

International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

G.9961

Amendment 1
(02/2020)

SERIES G: TRANSMISSION SYSTEMS AND MEDIA,
DIGITAL SYSTEMS AND NETWORKS

Access networks – In premises networks

Unified high-speed wireline-based home networking
transceivers – Data link layer specification

Amendment 1

Recommendation ITU-T G.9961 (2018) – Amendment 1

ITU-T



ITU-T G-SERIES RECOMMENDATIONS
TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS

INTERNATIONAL TELEPHONE CONNECTIONS AND CIRCUITS	G.100–G.199
GENERAL CHARACTERISTICS COMMON TO ALL ANALOGUE CARRIER-TRANSMISSION SYSTEMS	G.200–G.299
INDIVIDUAL CHARACTERISTICS OF INTERNATIONAL CARRIER TELEPHONE SYSTEMS ON METALLIC LINES	G.300–G.399
GENERAL CHARACTERISTICS OF INTERNATIONAL CARRIER TELEPHONE SYSTEMS ON RADIO-RELAY OR SATELLITE LINKS AND INTERCONNECTION WITH METALLIC LINES	G.400–G.449
COORDINATION OF RADIOTELEPHONY AND LINE TELEPHONY	G.450–G.499
TRANSMISSION MEDIA AND OPTICAL SYSTEMS CHARACTERISTICS	G.600–G.699
DIGITAL TERMINAL EQUIPMENTS	G.700–G.799
DIGITAL NETWORKS	G.800–G.899
DIGITAL SECTIONS AND DIGITAL LINE SYSTEM	G.900–G.999
MULTIMEDIA QUALITY OF SERVICE AND PERFORMANCE – GENERIC AND USER-RELATED ASPECTS	G.1000–G.1999
TRANSMISSION MEDIA CHARACTERISTICS	G.6000–G.6999
DATA OVER TRANSPORT – GENERIC ASPECTS	G.7000–G.7999
PACKET OVER TRANSPORT ASPECTS	G.8000–G.8999
ACCESS NETWORKS	G.9000–G.9999
Metallic access networks	G.9700–G.9799
Optical line systems for local and access networks	G.9800–G.9899
In premises networks	G.9900–G.9999

For further details, please refer to the list of ITU-T Recommendations.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at <http://www.itu.int/ITU-T/ipr/>.

© ITU 2020

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

Table of Contents

	Page
1	Scope..... 1
2	References..... 1
3	Definitions 3
3.1	Terms defined elsewhere 3
3.2	Terms defined in this Recommendation 3
4	Abbreviations and acronyms 6
5	Home network architecture and reference models 11
6	Profiles..... 11
7	Physical layer specification 11
8	Data link layer specification 11
8.1	Functional model and frame formats..... 11
8.2	MAP controlled medium access 25
8.3	Transmission opportunities (TXOPs) and time slots (TSs)..... 27
8.4	Control parameters for APC, LLC, and MAC 56
8.5	Functions of the endpoint node 61
8.6	Domain master node functional capabilities 77
8.7	Addressing scheme 179
8.8	Medium access plan (MAP) frame..... 180
8.9	Retransmission and acknowledgement protocol 217
8.10	Management and control message format 230
8.11	Channel estimation protocol..... 242
8.12	Connection management 264
8.13	Message flooding..... 270
8.14	Operation in the presence of neighbouring domains – Distributed domain coordination (NDIM)..... 270
8.15	Coexistence with alien power-line networks..... 320
8.16	PHY multicast binding protocol 320
8.17	DLL multicast stream 330
8.18	Inter-OFB interoperability 342
8.19	Node information and capabilities exchange 344
8.20	Metrics acquisition 345
8.21	Operation in power saving modes 346
8.22	Layer 2 Configuration and Management Protocol (LCMP)..... 353
8.23	Run-length compression of fields of management messages..... 360
8.24	Payload compression in management messages 362
8.25	TLV pool 363

	Page
9 Security	367
9.1 Encryption	367
9.2 Authentication and key management procedures	373
Annex A – Application protocol convergence sublayer	395
A.1 Ethernet APC (EAPC)	395
A.2 Other types of APC	399
Annex B to Annex W	400
Annex X – Test vectors	401
X.1 CCM encryption	401
X.2 PAK test vectors	403
X.3 Huffman coding	406
Bibliography	412

Amendment 1 to Recommendation ITU-T G.9961

Unified high-speed wireline-based home networking transceivers – Data link layer specification

Amendment 1

1 Scope

This Recommendation specifies reference models and functionality for all components of the data link layer (DLL) of home network transceivers designed for the transmission of data over premises wiring including inside telephone wiring, coaxial cable, and power-line wiring, and combinations of these.

This includes support of:

- contention-free TDMA and contention-based CSMA medium access control;
- parameter-based and priority-based QoS;
- security and confidentiality for the home network, including authentication, encryption and key management procedures;
- hidden nodes and data relaying;
- internal and external management communications;
- unicast and multicast retransmission based on selective acknowledgement and frame-based acknowledgement protocols;
- power-saving mechanisms;
- bidirectional transmission;
- neighbouring domain interference mitigation (NDIM);
- network management procedures, such as:
 - network initialization procedures,
 - admission control,
 - node authentication and encryption key assignment,
 - connection management,
 - channel estimation,
 - bandwidth reservation and flow control,
 - topology maintenance and routing mechanisms, and
 - recovery procedures after domain master failure.
- Layer 2 configuration and management protocol (LCMP).

Corrigendum 1 includes clarifications on the terminology used for auxiliary information fields length and for LCMP messages. It also includes corrections on timeout values and clarifies the MAP addressing fields.

[Amendment 1 includes support for the extension of Recommendation ITU-T G.9960 to operate with Profile 2 OFBs \(K-OFB\) over coaxial and phonenumber mediums.](#)

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the

editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

- [[ITU-T G.9960](#)] Recommendation ITU-T G.9960 (2015⁸), *Unified high-speed wireline-based home networking transceivers – System architecture and physical layer specification*.
- [[ITU-T G.9962](#)] Recommendation ITU-T G.9962 (2018), *Unified high-speed wireline-based home networking transceivers – Management specification*.
- [[ITU-T G.9963](#)] Recommendation ITU-T G.9963 (2015), *Unified high-speed wireline-based home networking transceivers – Multiple input/multiple output specification*.
- [ITU-T G.9964] Recommendation ITU-T G.9964 (2011), *Unified high-speed wireline-based home networking transceivers – Power spectral density specification*.
- [[ITU-T G.9972](#)] Recommendation ITU-T G.9972 (2010), *Coexistence mechanism for wireline home networking transceivers*.
- [ITU-T G.9978] Recommendation ITU-T G.9978 (2018), *Secure admission in G.hn network*.
- [[ITU-T G.9991](#)] Recommendation ITU-T G.9991 (2019), *High-speed indoor visible light communication transceiver – System architecture, physical layer and data link layer specification*.
- [[ITU-T X.1035](#)] Recommendation ITU-T X.1035 (2007), *Password-authenticated key exchange (PAK) protocol*.
- [IEEE 802.1ad] IEEE 802.1ad-2005, *IEEE Standard for Local and metropolitan area networks: Provider Bridges*.
- [IEEE 802.1D] IEEE 802.1D-2004, *IEEE Standard for Local and metropolitan area networks Media Access Control (MAC) Bridges*.
<<http://standards.ieee.org>>
- [IEEE 802.1Q] IEEE 802.1Q-2005, *IEEE Standard for Local and Metropolitan Area Networks – Virtual Bridged Local Area Networks – Revision*.
<<http://standards.ieee.org>>
- [IEEE 802.3] IEEE 802.3-2008, *IEEE Standard for Information technology-Specific requirements – Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications*.
<<http://standards.ieee.org>>
- [NIST FIPS 197] FIPS-PUB-197-2002, *Specification for the Advanced Encryption Standard (AES)*, National Institute of Standards and Technology, November, 2001. <<http://csrc.nist.gov/publications/>>
- [NIST 800-38C] Special Publication 800-38C, *Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality*, National Institute of Standards and Technology, May 2004.
<http://csrc.nist.gov/publications/nistpubs/800-38C_updated-July20_2007.pdf>
- [NIST FIPS 180-3] FIPS PUB 180-3 (2008), *Secure Hash Standard (SHS)*, National Institute of Standards and Technology, October, 2008.
<<http://csrc.nist.gov/publications/>>

3 Definitions

3.1 Terms defined elsewhere

None.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 address association table (AAT): A table that associates MAC addresses of the application entities with the DEVICE_ID of the nodes through which these application entities can be reached.

3.2.2 automatic traffic classification: A service that enables a node to establish a data flow automatically. This service is required to support client-based applications that are not capable of generating and transmitting a TSpec prior to data communication. The TSpec in this case is preconfigured in the node.

3.2.3 bit error ratio: A ratio of the number of data bits received in error to the total number of received data bits. The bit error ratio can be used for the total stream of data bits and for any of its tributary bit streams.

3.2.4 broadcast: A type of communication where a node sends the same frame simultaneously to all other nodes in the home network or in the domain.

3.2.5 channel: A transmission path between nodes. One channel is considered to be one transmission path. Logically, a channel is an instance of a communications medium used for the purpose of passing data between two or more nodes.

3.2.6 client: An application entity distinguished in the network by its unique address (e.g., MAC address).

3.2.7 connection: A flow between a node and one or more other nodes uniquely identified by the following parameters carried in the PHY-frame header:

- Source ID (SID);
- Destination ID (DID);
- Multicast indication (MI); and
- Connection identifier (CONNECTION_ID).

and the following parameter carried in the LPH of the associated LPDUs:

- Management queue flag (MQF).

3.2.8 contention-based time slot (CBTS): A time slot for which contention-based access is allowed amongst a group of nodes.

3.2.9 contention-based transmission opportunity (CBTXOP): A shared transmission opportunity for which only contention-based access is defined.

3.2.10 contention-free time slot (CFTS): A time slot within a shared transmission opportunity assigned to a single node.

3.2.11 contention-free transmission opportunity (CFTXOP): A transmission opportunity allocated to a single node.

3.2.12 data: Bits or bytes transported over the medium or across a reference point that individually convey information. Data includes both user (application) data and any other auxiliary information (overhead, including control, management, etc.). Data does not include bits or bytes that, by themselves, do not convey any information, such as preamble.

3.2.13 data connection: A connection for delivering data LLC frame blocks.

3.2.14 data frame: An ordered group of bits or bytes, specific to the application layer (e.g., Ethernet frame), with start and stop delimiters.

3.2.15 data LPDU: LPDU with MQF set to zero.

3.2.16 data LLC frame: An LLC frame carrying an APDU.

3.2.17 data rate: The average number of bits communicated (transmitted) in a unit of time. The usual unit of time for data rate is 1 second.

3.2.18 DEVICE_ID: A unique identifier allocated to a node operating in the network by the domain master during registration.

3.2.19 domain: A part of an ITU-T G.9960/1 home network comprising the domain master and all those nodes that are registered with the same domain master. In the context of this Recommendation, use of the term 'domain' without a qualifier means 'G.9960/1 domain', and use of the term 'alien domain' means 'non-ITU-T G.9960/1 domain'. Additional qualifiers (e.g., 'power-line') may be added to either 'domain' or 'alien domain'.

3.2.20 domain master: A node that manages (coordinates) all other nodes of the same domain (i.e., assigns bandwidth resources and manages user priorities). A node with domain master capabilities has all the capabilities of an endpoint node and may act as a relay node.

3.2.21 domain name: A 32-byte domain identifier assigned by the user for admission of nodes to the particular domain.

3.2.22 endpoint node: This term is used in this Recommendation according to the context to differentiate between the domain master node functionalities and non-domain master node functionalities.

3.2.23 flow: A unidirectional stream of data between two nodes related to a specific application, or characterized by a set of QoS requirements, or both.

3.2.24 FLOW_ID: A unique identifier allocated to a service flow by the node originating the flow.

3.2.25 frame: An ordered group of bits or bytes with start and stop delimiters.

3.2.26 hidden node: A node that cannot communicate directly with some other nodes within a domain.

NOTE – A hidden node may be able to communicate with another node or with a domain master using a relay node. A node that is hidden from a domain master uses a relay node as a proxy to communicate with the domain master.

3.2.27 in-band management message: A management message (see Figure 8-54) exchanged between the application entity (AE) and the data link layer (DLL) management entity via A-interface. It is represented as a standard application data primitive (ADP) set at the A-interface.

3.2.28 inter-domain bridge: A bridging function above the physical layer to interconnect nodes of two different domains.

3.2.29 inter-frame gap: The time measured from the last sample of the last symbol of a PHY frame to the first sample of the first symbol of the preamble of the subsequent PHY frame.

3.2.30 jitter: A measure of the latency variation above and below the mean latency value. The maximum jitter is defined as the maximum latency variation above and below the mean latency value.

3.2.31 latency: A measure of the delay from the instant when the last bit of a frame has been transmitted through the assigned reference point of the transmitter protocol stack to the instant when a whole frame reaches the assigned reference point of receiver protocol stack. Mean and maximum latency estimations are assumed to be calculated on the 99th percentile of all latency measurements. If retransmission is enabled for a specific flow, latency also includes retransmission time.

- 3.2.32 leaf node:** A node within a spanning tree that is linked to a single node.
- 3.2.33 management connection:** A connection for delivering management logical link control (LLC) frame blocks.
- 3.2.34 management LLC frame:** A logical link control (LLC) frame carrying a logical link control data unit (LCDU).
- 3.2.35 management LPDU:** A logical link control protocol data unit (LPDU) with the management queue flag (MQF) set to one.
- 3.2.36 medium:** A wireline facility, of a single wire class, allowing physical connection between nodes. Nodes connected to the same medium may communicate on the physical layer, and may interfere with each other unless they use orthogonal signals (e.g., different frequency bands, different time periods).
- 3.2.37 multicast:** A type of communication where a node sends the same frame simultaneously to one or more other nodes in the home network.
- 3.2.38 multicast client:** The application generating the request to receive a multicast stream above the A-interface of the receiving node.
- 3.2.39 multicast group:** Subset of the receivers of a multicast stream that were assigned by the transmitter to use the same bit allocation tables (BATs) and assigned Mc-ACK frame slots and identified by a multicast group ID.
- 3.2.40 multicast group ID:** A combination of the multicast destination ID (DID) assigned at the time of multicast group creation and the Device ID of the transmitter. It uniquely identifies the multicast group at the receivers.
- 3.2.41 multicast source:** The application generating the multicast stream above the A-interface of the transmitting node.
- 3.2.42 node (network node):** Any network device that contains an ITU-T G.9960/1 transceiver. In the context of this Recommendation, the use of the term 'node' without a qualifier means 'ITU-T G.9960/1 node', and use of the term 'alien node' means 'non-ITU-T G.9960/1 node'. Additional qualifiers (e.g., 'relay') may be added to either 'node' or 'alien node'. See related definitions: domain master node, endpoint node and relay node.
- NOTE – The entities: endpoint node, relay node, and domain master node refer to certain functionalities of a node according to the context. A certain node may act as more than one entity. For example, a domain master may act as a relay node or may act as an endpoint node.
- 3.2.43 non-leaf node:** A node within a spanning tree that is linked to more than one node.
- 3.2.44 primitives:** Variables and functions used to define logical interfaces and reference points.
- 3.2.45 quality of service:** A set of quality requirements on the communications in the home network. Support of quality of service refers to mechanisms that can provide different priority to different flows, or can guarantee a measurable level of performance to a flow based on a set of quality of service parameters.
- 3.2.46 reference point:** A location in a signal flow, either logical or physical, that provides a common point for observation and/or measurement of the signal flow.
- 3.2.47 registration:** The process used by a node to join the domain.
- 3.2.48 registration contention-based time slot (RCBTS):** A type of contention-based time slot used exclusively for registration.
- 3.2.49 relay node:** A node that acts as a relay unit in the domain to relay link control data units (LCDUs) and APC protocol data units (APDUs) between hidden nodes, in addition to its main role

(as domain master node or endpoint node). The domain master node and any endpoint node may act as a relay node in the domain.

3.2.50 service flow: A flow for which parameterized QoS is used for traffic delivery.

3.2.51 shared transmission opportunity (STXOP): A transmission opportunity allocated to a group of nodes.

3.2.52 spanning tree: A graph that represents the domain topology in the form of a tree that connects all nodes in the domain so that no loops (or cycles) are formed.

3.2.53 sub-carrier (OFDM sub-carrier): The centre frequency of each OFDM sub-channel onto which bits may be modulated for transmission over the sub-channel.

3.2.54 sub-channel (OFDM sub-channel): A fundamental element of OFDM modulation technology. The OFDM modulator partitions the channel bandwidth into a set of parallel sub-channels.

3.2.55 symbol (OFDM symbol): A fixed time-unit of an OFDM signal carrying one or more bits of data. An OFDM symbol consists of multiple sine-wave signals or sub-carriers, each modulated by a number of data bits and transmitted during the fixed time called symbol period.

3.2.56 time slot (TS): A time interval within a shared transmission opportunity representing an opportunity for one or more nodes to start transmitting.

3.2.57 traffic contract: An agreement between a node and the domain master that stipulates a certain guaranteed amount of bandwidth and QoS parameters, such as latency and jitter. The TSpec is provided by the node to the domain master to establish the parameters of the traffic contract during establishment of a data flow. If the contract cannot be established given the parameters contained in the TSpec, the domain master may refuse to establish the flow. This is called denial of service.

3.2.58 transmission opportunity (TXOP): An interval of time during which a node or a group of nodes has the right to initiate transmission.

3.2.59 unicast: A type of communication where a node sends a frame to another single node.

3.2.60 user priority: A value, denoted PRI, assigned by the classifier to the specific frame that determines the relative importance of the frame compared to other frames.

3.2.61 wire class: One of the classes of wire, having the same general characteristics: coaxial cable, home electrical power wire, phone line wire or Category 5 cable.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

AAT	Address Association Table
ACE	Additional Channel Estimation
ACK	ACKnowledgement
ACKI	ACKnowledgement Information
ADP	Application Data Primitive
AE	Application Entity
AES	Advanced Encryption Standard
AIFG	ACK Interframe Gap
AKM	Authentication and Key Management
APC	Application Protocol Convergence

APDU APC Protocol Data Unit
ARQ Automatic Repeat Request
BACK Bidirectional ACKnowledgement
BAT Bit Allocation Table
BC Back-off Counter
BEF Burst End Flag
BIFG Burst Inter-Frame Gap
B-LCDU Broadcast Link Control Data Unit
BMSG Bidirectional Message
BRT Broadcast Routing Table
BRURQ Bandwidth Reservation Update Request
BSC Back-off Stage Counter
BTXEF Bidirectional Transmission End Flag
BTXGL Bidirectional Transmission Grant Length
CBR Constant Bit Rate
CBTS Contention-Based Time Slot
CBTXOP Contention-Based Transmission Opportunity
CCM Counter with Cipher block chaining Message authentication code
CCMP CCM Protocol
CCMPI CCMP header present Indication
CE Channel Estimation
CFTXOP Contention-Free Transmission Opportunity
CFTS Contention-Free Time Slot
CID Clear to send proxy Identification
CIR Committed information rate
CMH Control Message Header
CMPL Control Message Parameter List
CRC Cyclic Redundancy Check
CRS Carrier Sense
CRTM Centralized Routing and Topology Management
CSLT Current Schedule Life Time
CSMA Carrier Sense Multiple Access
CSMA/CA Carrier Sense Multiple Access with Collision Avoidance
CSP Cluster Synchronization Point
CTMG Control and Management frame
CTS Clear To Send
CURRTS Current Time Slot

CYCSTART MAC Cycle Start
CW Contention Window
DA Destination Address
DC Defer Counter
DID Destination ID
DLL Data Link Layer
DM Domain Master
DNI Domain Name Identifier
DOD Domain ID
DRTM Distributed Routing and Topology Management
DSL Digital Subscriber Line
EACK Extended ACK
EFD Enhanced Frame Detection
FACK Frame ACKnowledgement
FCS Frame Check Sequence
FDX Full Duplex
FLEN Frame Length
FN Frame Number
FSB Force Sequence Bit
FSLT Future Schedule Life Time
FT Frame Type
HCS Header Check Sequence
HDX Half Duplex
HOIP Handover In Progress
IDB Inter-Domain Bridge
IDCC Inter-Domain Communication Channel
IDM Inter-Domain Message
IDPS Inter-Domain Presence Signal
IDSW Inter-Domain Signalling Window
IFG Inter-Frame Gap
IGMP Internet Group Management Protocol
Imm-ACK Immediate ACKnowledgement
ITS Idle Time Slot
LAAT Local Address Association Table
LCDU Link Control Data Unit
LCMP Layer 2 Configuration Management Protocol
LFBO Logical link control Frame Boundary Offset

LFH Logical link control Frame Header
 LLC Logical Link Control
 LLCFT Logical Link Control Frame Type
 LPCS Logical link control Protocol data unit Check Sequence
 LPDU Logical link control Protocol Data Unit
 LPH Logical link control Protocol data unit Header
 LPRI LLC Frame Priority
 LSB Least Significant Bit
 LSSN Lowest Segment Sequence Number
 MAC Medium Access Control
 MAP Medium Access Plan
 MAP-A Active Medium Access Plan
 MAP-D Default Medium Access Plan
 Mc-ACK Multi-cast ACKnowledgement
 MCCD Medium access control Cycle Countdown
 MI Multicast Indication
 MIC Message Integrity Code
 MIR Maximum information rate
 MLD Multicast Listener Discovery
 MMH Management Message Header
 MMPL Management Message Parameter List
 MPDU Medium access control Protocol Data Unit
 MPR Multipoint Relay
 MQF Management Queue Flag
 MSC Message Sequence Chart
 MSG Message
 MSID Multicast Stream Identifier
 NACK Negative ACKnowledgement
 NDID Neighbouring Domain Identifier
 NDIM Neighbouring Domain Interference Mitigation
 NMK Network Membership Key
 NSC Node to Security Controller
 NTR Network Time Reference
OFB Operational Frequency Band
 OFDM Orthogonal Frequency Division Multiplexing
 OPSF Oldest Pending Segment Flag
 PAK Password Authentication Key

PBU Partial Bit allocation table Update
PDU Protocol Data Unit
PFH PHY-Frame Header
PMI Physical Medium Independent
PON Passive Optical Network
PR Priority Resolution
PRI User Priority
PRS Priority Resolution Slot
PSD Power Spectral Density
PSM Power spectral density Shaping Mask
QoS Quality of Service
RAAT Remote Address Association Table
RCBTS Registration Contention-Based Time Slot
[RCM Robust Communication Mode](#)
REGID Registration Identifier
RMAP Relayed Medium Access Plan
RMAP-A Relayed Medium Access Plan – Active
RMAP-D Relayed Medium Access Plan – Default
RPRQ Reply Required
RTS Request To Send
SC Security Controller
SID Source ID
SM Sub-carrier Mask
SSN Segment Sequence Number
STXOP Shared Transmission Opportunity
TCP Transmission Control Protocol
TDMA Time Division Multiple Access
TLV Type Length Value
TS Time Slot
TSMP Time Stamp
TSMPI Time Stamp Present Indication
TTL Time To Live
TXOP Transmission Opportunity
VBR Variable Bit Rate
VSF Valid Segment Flag

5 Home network architecture and reference models

See clause 5 of [ITU-T G.9960].

6 Profiles

See clause 6 of [ITU-T G.9960].

7 Physical layer specification

See clause 7 of [ITU-T G.9960].

8 Data link layer specification

8.1 Functional model and frame formats

8.1.1 Functional model of the data link layer (DLL)

The functional model of the DLL is presented in Figure 8-1. The A-interface is the demarcation point between the application entity (AE) and the data link layer (DLL); the physical medium independent (PMI) interface is the demarcation point between the DLL and the physical (PHY) layer. Internal reference points x1 and x2 show logical separation between the APC and LLC and between the LLC and MAC, respectively.

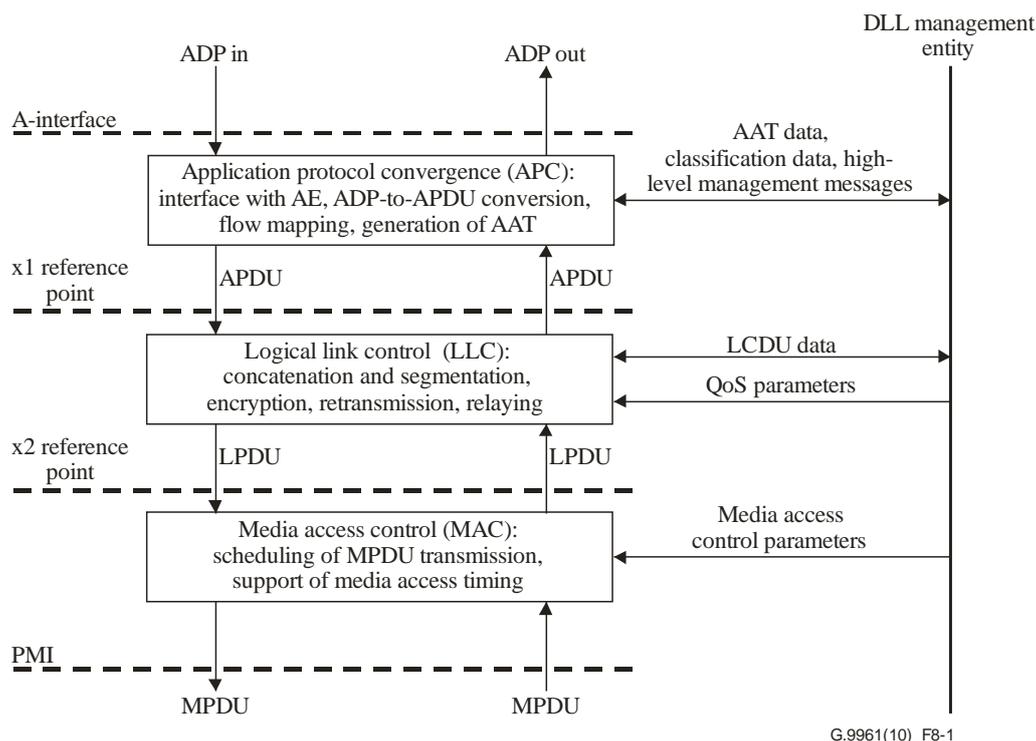


Figure 8-1 – Functional model of the DLL

In the transmit direction, application data primitive (ADP) sets enter the DLL from the AE across the A-interface. Every incoming ADP set meets the format defined by the particular application protocol; for an Ethernet type AE, the ADP set has one of the standard Ethernet formats, as presented in Annex A (Ethernet APC). Each incoming ADP set is converted by the APC into APC protocol data units (APDUs), which include all parts of the ADP set intended for communication to the destination node(s). The APC also identifies ADP classification primitives (e.g., priority tags), which can be used by the LLC to support QoS requirements assigned to the service delivered by the ADP. Further, the

APC is responsible for establishing flows of APDUs between peer APCs and assigning one or more queues for these flows according to the classification information associated with each APDU. The number of queues may depend on the profile of the device; for the Ethernet APC, mapping of user priorities to the same destination into priority queues (traffic classes) shall follow Table III.1 of [ITU-T G.9960].

The APDUs are transferred to the LLC across the x1 reference point, which is both application independent and medium independent. In addition, LLC receives management data primitives from the DLL management entity intended for LLC control frames, which are mapped into link control data units (LCDUs). The LLC is responsible for establishing flows of LCDU (control frames) between peer LLCs.

In the LLC, the incoming APDU and LCDU are converted into LLC frames and may be encrypted using assigned encryption keys (see clause 9.2). LLC frames are subject to concatenation and segmentation, as described in clause 8.1.3.2. Segments are transformed into LLC protocol data units (LPDUs) by adding an LPDU header (LPH) and CRC. LPDUs are then passed to the MAC across the x2 reference point. The LLC is also responsible for retransmission and relay operations.

The MAC is responsible for concatenating LPDUs into MAC protocol data units (MPDUs) and then conveying these MPDUs to the PHY in the order determined by the LLC (scheduling, using number of transmission queues) and applying medium access rules established in the domain.

In the receive direction, MPDUs from the PHY enter the MAC across the PMI together with associated PHY frame error information. The MAC disassembles the received MPDU into LPDUs, which are passed over the x2 reference point to the LLC. The LLC recovers the original APDUs and LCDUs from the LPDUs, performs decryption if required, and conveys them to the APC and LLC management entity, respectively. In the APC, ADPs are generated from the received APDUs and conveyed to the AE.

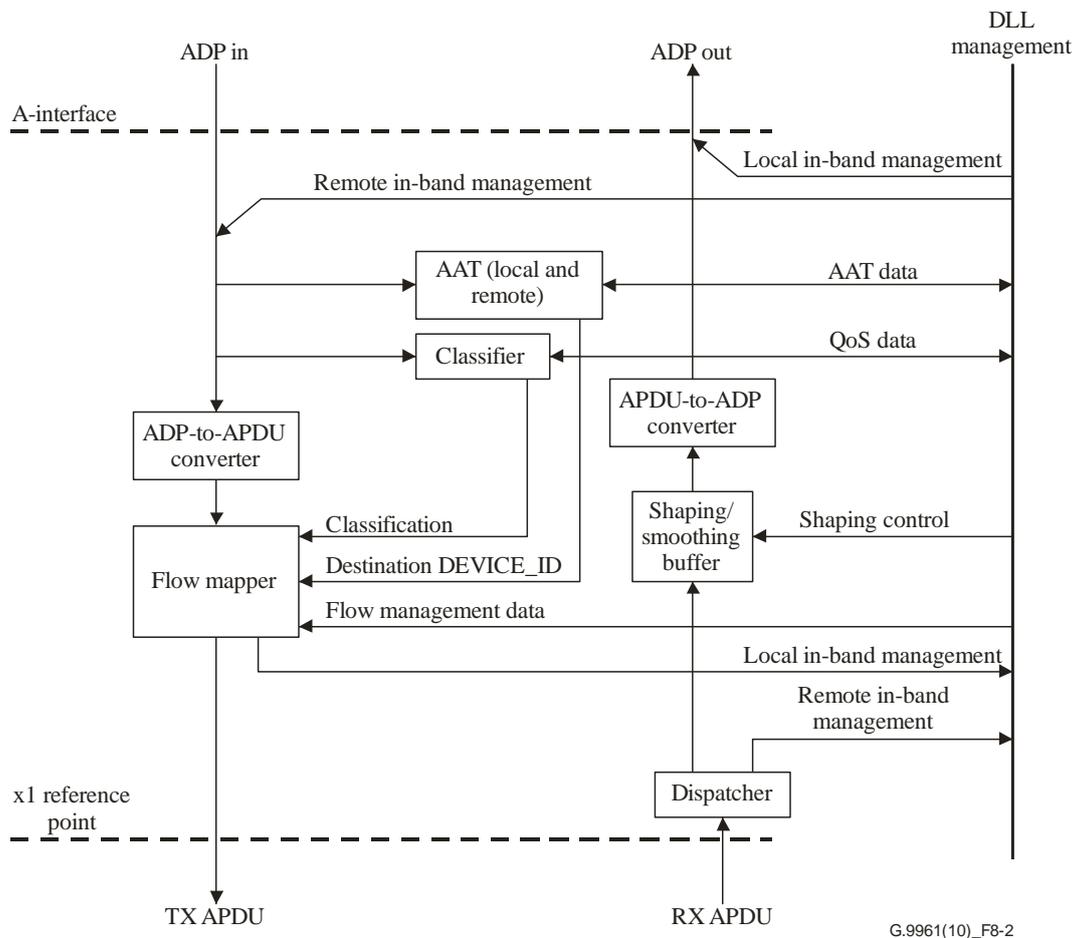
The LLC is responsible for the decision regarding errored LPDUs. It decides whether to request retransmission of errored LPDUs (and generates the ACK response to assist retransmission), or to discard the errored LPDUs.

The functionality of the APC, LLC, and MAC is the same for all types of medium, although some of their functions and control parameters may be adjusted for efficient operation of the transceiver over particular medium. Specific control parameters for APC, LLC, and MAC are described in clause 8.4.

NOTE – No assumptions should be made on partitioning of APC, LLC, and MAC in particular implementations; x1 and x2 are reference points and serve for convenience of system definition.

8.1.2 Application protocol convergence (APC)

The functional model of the APC is presented in Figure 8-2. It is intended to describe in more detail the APC functional block presented in Figure 8-1.



G.9961(10)_F8-2

Figure 8-2 – Functional model of APC

In the transmit direction, the incoming ADP is converted into an APDU as defined in Annex A. The flow mapper maps APDUs into flows, depending on their destination `DEVICE_ID`, class of service, and QoS support capabilities of the communicating nodes. Flows are established in the APC by DLL management after receiving relevant data units from the AE, or during admission to the network, or by high-level management requests coming across the A-interface, or upon request from another node (by means of flow establishment protocol messages coming across the x1 reference point). After mapping, each APDU, tagged with its `FLOW_ID` or `PRI-Q`, is sent across the x1 reference point to the LLC. The order of outgoing APDUs at the x1-reference point associated with a particular DID and a particular user priority, and within a particular service flow shall be the same as the order of arrival of the ADPs sourcing these APDUs.

The data units of the in-band management messages arriving across the A-interface and addressed to the local node are directed to the DLL management entity ("Local in-band management", at the bottom of Figure 8-2). The in-band management messages generated by DLL management entity for the remote AE are converted to APDUs, and sent across the x1 reference point ("Remote in-band management" at the top of Figure 8-2).

In the receive direction, the incoming APDUs crossing the x1 reference point are converted back into the ADP data unit primitives of the relevant application protocol. A shaping (smoothing) buffer, controlled by DLL management entity, may be included for traffic shaping of the outgoing (i.e., in the direction of the AE) ADP data units.

If addressed to the node, APDUs carrying in-band management messages from the remote AE are dispatched to the DLL management entity ("Remote in-band management" at the bottom of Figure 8-2). If addressed to the local AE, APDUs carrying in-band management messages from the

remote AE are converted to a standard ADP and passed to A-interface. The in-band management messages (e.g., responses) generated by DLL management entity for the local AE are sent to the AE across the A-interface as standard sets of data unit primitives ("Local in-band management" at the top of Figure 8-2).

The classification information embedded in the ADP is extracted from the incoming data units and may be used to set an appropriate type of traffic (flow) or to assign a user priority, or both, to convey the corresponding APDU through the network. Classification parameters are presented in Annex A.

The local address association table (LAAT) contains in its first entry the MAC address of the node itself (i.e., REGID) and, in the rest of the table, the MAC addresses of the clients associated with the node. This data is collected from the incoming ADP data units; LAAT data is passed to the DLL management entity for network management purposes (see clause 8.5.3).

The remote address association table (RAAT) stores MAC addresses of other nodes in the domain and their associated clients that were advertised on the network.

The address association table (AAT) is formed by the aggregation of LAAT and RAAT.

8.1.3 Logical link control (LLC)

The functional model of the LLC is presented in Figure 8-3. It is intended to describe in more detail the LLC functional block presented in Figure 8-1.

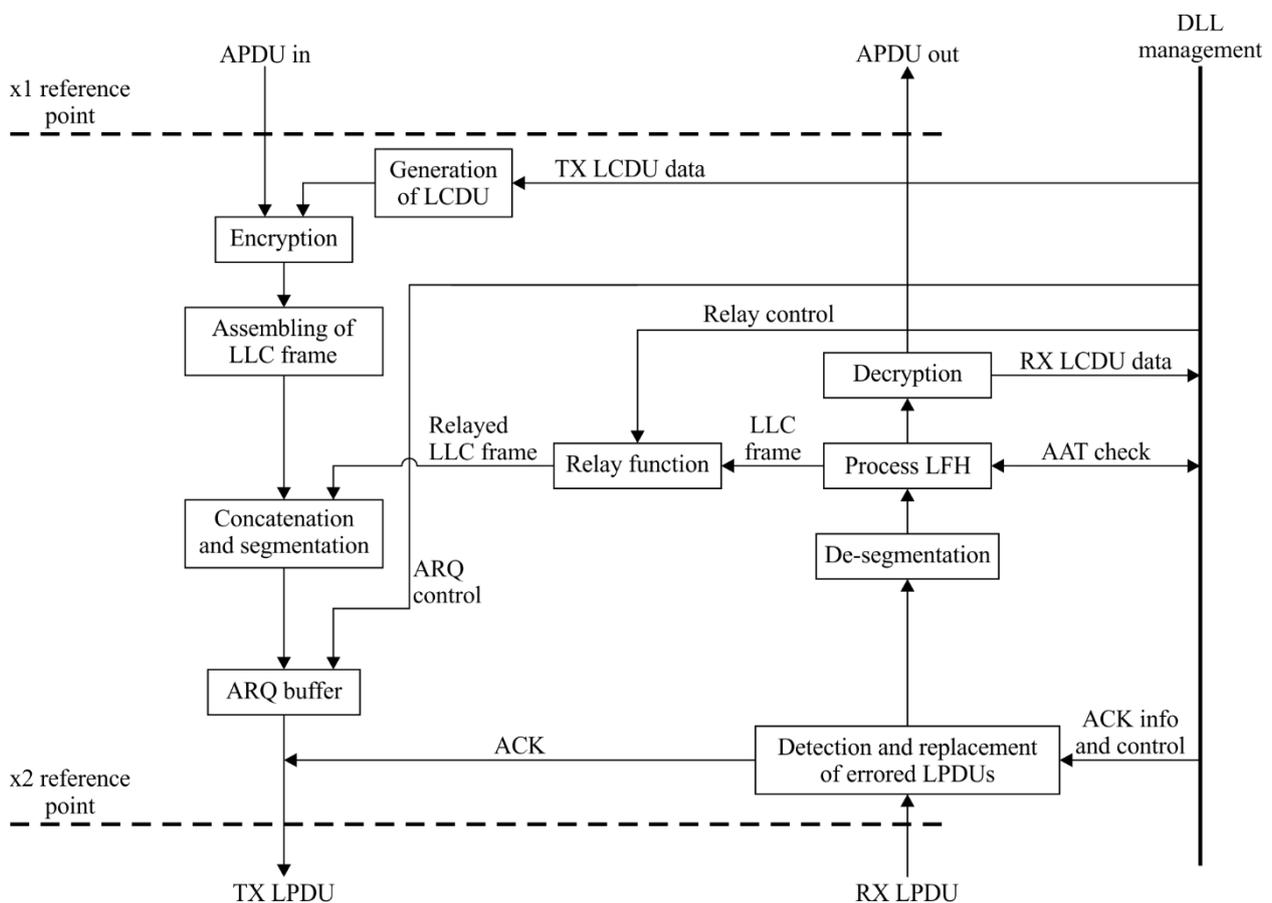


Figure 8-3 – Functional model of LLC

In the transmit direction, an LLC frame is formed from each incoming APDU crossing the x1 reference point, which may be encrypted using encryption rules defined in clause 9.1. One or more LLC frames are concatenated and further divided into segments of equal size. Each segment is pre-pended by an LPDU header and appended with an LPDU CRC, forming an LPDU.

8.1.3.1.1 LLC frame header fields

The LLC frame header (LFH) is composed of the fields described in Table 8-1. The LFH is 6 octets long if no time stamp is present and 10 octets long if time stamp is present. Octet 0 shall be passed to the MAC first.

Table 8-1 – LLC header fields format

Field	Octet	Bits	Description	Reference
LLCFT	0	[2:0]	LLC frame type	Clause 8.1.3.1.1.1
TSMPI		[3]	Time stamp present indication	Clause 8.1.3.1.1.2
CCMPI		[4]	CCMP header present indication	Clause 8.1.3.1.1.3
LPRI		[7:5]	User priority of the LLC frame	Clause 8.1.3.1.1.4
FLEN	1 and 2	[13:0]	LLC frame body length in bytes	Clause 8.1.3.1.1.5
MCSTI		[14]	Multicast stream indicator	Clause 8.1.3.1.1.11
Reserved		[15]	Reserved by ITU-T (Note)	
OriginatingNode	3	[7:0]	DEVICE_ID of the node that created the LLC frame	Clause 8.1.3.1.1.6
DestinationNode	4	[7:0]	Destination identifier that indicates the node(s) whither the LLC frame is finally destined.	Clause 8.1.3.1.1.10
BRCTI	5	[0]	Broadcast indicator	Clause 8.1.3.1.1.7
Reserved		[1]	Reserved by ITU-T (Note)	
TTL		[7:2]	Time to live	Clause 8.1.3.1.1.8
TSMP	6 to 9	[31:0]	Time stamp. This field is included in the header only when TSMPI is set to one	Clause 8.1.3.1.1.9
NOTE – Fields or bits reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.				

8.1.3.1.1.1 LLC frame type (LLCFT)

The LLCFT field indicates the type of frame that makes up the LLC frame. It is formatted as a 3-bit unsigned integer.

Table 8-2 lists the valid LLC frame types:

Table 8-2 – LLC frame types

LLC frame type	Value
Padding last segment (see clause 8.1.3.2)	0
Management frame (LCDU)	1
Data frame (APDU)	2
NULL frame	3
Reserved by ITU-T	4 to 7

8.1.3.1.1.2 Timestamp present indication (TSMPI)

TSMPI is a 1-bit field that is used to indicate whether or not the timestamp (TSMP) field is included in the LLC frame header. If set to one, the LFH shall include the TSMP field. If set to zero, no TSMP field shall be present (in this case the LFH length is 6 octets).

8.1.3.1.1.10 DestinationNode

This field is populated by the originating node of the LLC frame and its contents shall not be changed when an LLC frame is relayed by another node. This field shall be used by the relay nodes for routing LLC frames.

The DestinationNode field indicates one or more nodes whither the LLC frame is finally destined. It shall be set to the DEVICE_ID of the final destination node in case of unicast (BRCTI=0, MCSTI=0). It shall be set to the multicast stream identifier (MSID) assigned by the multicast source in case of multicast (BRCTI=0, MCSTI=1). It shall be set to BROADCAST_ID by the originating node in case of broadcast (BRCTI=1, MCSTI=0). It shall be set to 0 for the LLC frames in which the DA is set to reserved MAC address 01-19-A7-52-76-96.

The content of this field shall not be changed when an LLC frame is relayed by another node. This field shall be used by the relay nodes for routing LLC frames.

8.1.3.1.1.11 MCSTI

The MCSTI field shall be set to one for the DestinationNode field to represent a DLL multicast stream identifier.

NOTE – The MCSTI field is different from the MI field of the PFH in that (OriginatingNode, DestinationNode, MCSTI) tuple defines a DLL multicast stream defined within a domain whereas (SID, DID, MI) tuple defines a PHY-level multicast traffic.

8.1.3.2 Generation of LPDUs

The process of generating LPDUs from LLC frames is presented in Figure 8-5.

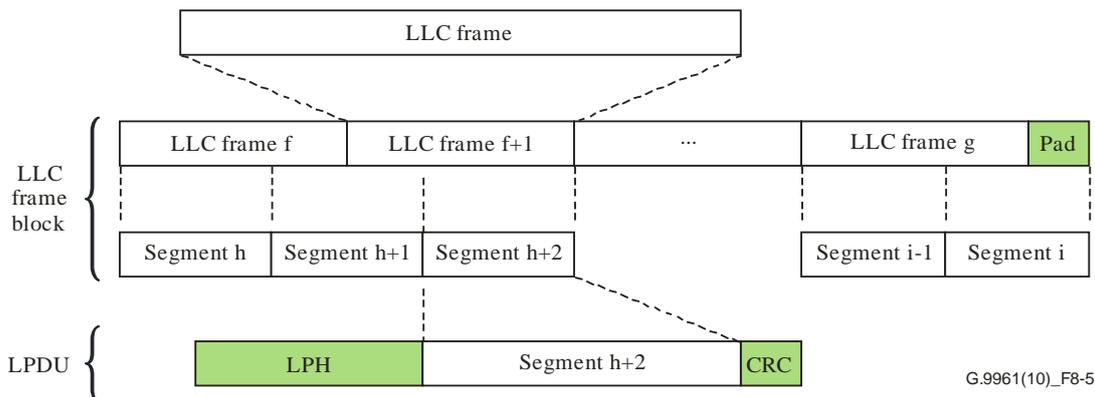


Figure 8-5 – Generation of LPDUs from LLC frames

An LLC frame shall be formed by pre-pending an LLC frame header (LFH) to an APDU or to an LCDU, and can either be unencrypted or encrypted as described in clause 8.1.3.1. When encrypted, a CCMP header and MIC are added, in addition to the LFH, as described in Figure 8-4. The format of LFH is defined in clause 8.1.3.1.1.

Multiple LLC frames carrying APDUs (data LLC frames) associated with the same data connection can be concatenated to form a data LLC frame block. LLC frames containing LCDUs (management LLC frames) that belong to the same management connection can be concatenated to form a management LLC frame block. The LLC frame block may also include LLC frames intended to be relayed that are associated with the same connection. The number of concatenated LLC frames for an LLC frame block is determined by DLL management entity and is vendor discretionary. The order of LLC frames of the same user priority in the LLC frame block (see Figure 8-5) containing APDUs shall be the same as the order of arrival of the APDUs sourcing these frames. The order of LLC frames in the LLC frame block containing LCDUs shall be the same as the order that these LCDUs are generated by the DLL management entity. The order of bytes in the LLC frame payload shall be

the same as in sourcing APDUs or LCDUs, and in the same relative order, bytes shall be passed to the MAC as LPDUs that the MAC maps into MPDU. Mixing data LLC frames and management LLC frames into the same LLC frame block (mixed LLC frame block) is allowed only if the lowest user priority associated with the corresponding prioritized data connection is equal to or greater than six and the highest user priority associated with the corresponding prioritized data connection is equal to seven (i.e., in case three or more priority queues are supported as described in Table III.1 of [ITU-T G.9960]). In this case LCDUs shall be mapped to the same prioritized data connection where APDUs with user priority 7 for the same destination are mapped. Mixed LLC frame blocks are not allowed for data connections associated with service flows. A mixed LLC frame block shall be treated as a data LLC frame block (i.e., The MQF flag in the LPH shall be set to zero; see clause 8.1.3.2.1.4).

NOTE 1 – Mixing data LLC frames and management LLC frames into the same LLC frame block can result in segments containing fragments of both a data LLC frame and a management LLC frame.

Each LLC frame block shall be segmented as presented in Figure 8-5. The first segment of an LLC frame block shall start from the first byte of the first LLC frame of that LLC frame block. The size of the segment shall be equal to the LPDU size of the FEC block (for Profile 2 OFBs) or the size of the FEC block (for Profile 1 OFBs) specified for the connection, minus the size of the LPDU header and minus the size of the LPDU check sequence, as described in clauses 8.1.3.2.1 and 8.1.3.2.2, respectively. If the last segment is incomplete, padding is required to fill up the last segment and provide an integer number of segments in the LLC frame block. Padding of the last segment shall be done by insertion of an all ZERO octet in the place of octet 0 of the LFH (see Table 8-2) followed by vendor discretionary octets as required to fill the segment.

NOTE 2 – Padding adds overhead and, therefore, should be avoided whenever possible.

If one LLC frame is divided into several segments, all those segments except the first one containing fragments of the LLC frame are called trailing segments.

For decoding purposes, the FEC block size is specified in the PHY-frame header (PFH) (see clause 7.1.2.3/10.1.2.3 of [ITU-T G.9960]).

Each segment shall be pre-pended with a LPDU header (LPH). The LPDU header contains information necessary to recover LLC frames from the segments at the receiver. The format of the LPDU header is defined in clause 8.1.3.2.1.

An LPDU that contains a trailing segment is called a trailing LPDU.

Each segment shall be appended with a CRC for error detection. A segment pre-pended with the LPDU header and appended with the LPDU CRC is referred to as an LPDU. The CRC shall be computed as defined in clause 8.1.3.2.2.

The segment size for all LPDUs of a connection shall be the same throughout the lifetime of that connection.

8.1.3.2.1 LPDU header format

Table 8-3 shows the format of the LPDU header (LPH). Octet 0 shall be passed to the PHY first.

Table 8-3 – LPDU header format

Field	Octet	Bits	Description	Reference
SSN	0 and 1	[15:0]	Segment sequence number	Clause 8.1.3.2.1.1
LFBO	2 and 3	[9:0]	LLC frame boundary offset	Clause 8.1.3.2.1.2
VSF		[10]	Valid segment flag	Clause 8.1.3.2.1.3
MQF		[11]	Management queue flag	Clause 8.1.3.2.1.4
OPSF		[12]	Oldest pending segment flag	Clause 8.1.3.2.1.5

be set to one to indicate that the LPDU is carrying a segment belonging to a management LLC frame block. It shall be set to zero to indicate that the LPDU is carrying a segment belonging to a data LLC frame block.

8.1.3.2.1.5 Oldest pending segment flag (OPSF)

For connections with acknowledgements, this field indicates whether the segment is the oldest pending segment in the transmitter queue associated with the connection. This enables the receiver to determine that all older segments are dropped, thus enabling it to process the oldest pending segment and subsequent segments without waiting for older segments. When OPSF is set to one, it indicates that the segment is the oldest segment present at the transmitters queue. When set to zero, it indicates that the segment is not the oldest pending segment in the transmitters queue.

For connections without acknowledgements and for payload not belonging to any connection (CNN_MNGMT field of PHY frame header equal to 1111₂), this field shall be set to one for the first segment in the MPDU and shall be set to zero for all other segments in that MPDU.

8.1.3.2.2 LPDU check sequence (LPCS)

The LPCS field is for LPDU verification. The LPCS is a 32-bit cyclic redundancy check (CRC) and shall be computed over all the fields of the LPDU in the order they are transmitted, starting with the LSB of the SSN field of the LPDU header (clause 8.1.3.2.1) and ending with the MSB of the last octet of the LPDU segment (clause 8.1.3.2).

The LPCS shall be computed using the following generator polynomial of degree 32:

$$G(x) = x^{32} + x^{28} + x^{27} + x^{26} + x^{25} + x^{23} + x^{22} + x^{20} + x^{19} + x^{18} + x^{14} + x^{13} + x^{11} + x^{10} + x^9 + x^8 + x^6 + 1$$

The LPCS shall be constructed as follows:

- 1) The n bits of the LPDU subject to LPCS are considered to be the coefficients of a polynomial of degree $n-1$. In particular, the LSB of the first octet of the LPDU header is the coefficient of the x^{n-1} term, and the MSB of the last octet of the LPDU segment is the coefficient of the x^0 term. This polynomial is referred to as $N(x)$.
- 2) Replace the 32 highest-order coefficients (i.e., the first 32 bits) of $N(x)$ with their one's complement.
- 3) Multiply the result of step 2 by x^{32} . This result is referred to as $N_1(x)$.

The LPCS is then the one's complement of the remainder of $N_1(x)$ divided by $G(x)$.

The bits of the LPCS shall be transmitted in sequential order, starting from the coefficient of the highest order term (x^{31}), referred as the MSB, and continuing to the x^0 term.

8.1.3.3 Generation of LPDUs for retransmission

LPDUs assigned for retransmission (see clause 8.9) shall be assembled into an outgoing MPDU with no changes in the LLC frame block segment and in all the fields of the LPH excluding the OPSF field. The OPSF field may change as described in clause 8.9.5.3.1. If the OPSF field changes, the LPCS shall be recalculated as described in clause 8.1.3.2.2. The location of the retransmitted LPDU in the MPDU shall be as described in clause 8.1.4.1.

8.1.3.4 LCDU frame format

The LCDU format, including size of the fields, shall be as presented in Figure 8-6.

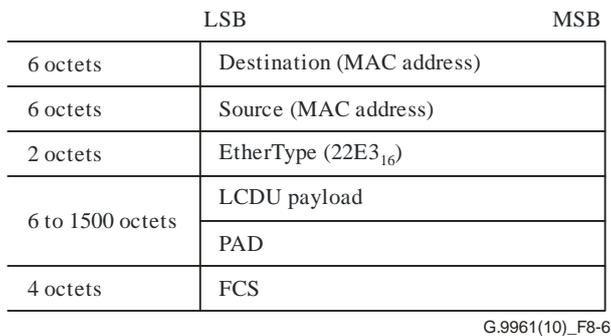


Figure 8-6 – LCDU format

The LCDU is identified at the destination node by its source/destination MAC address. The EtherType field is intended to identify the management message. The content of the EtherType field shall be 22E3₁₆.

The PAD field shall complete the total length of LCDU to its minimum value of 64 bytes. The PAD field, if present, shall be set to zero.

The FCS shall be computed over all LCDU fields, from the first bit (LSB) of the destination MAC address field to the last bit (MSB) of the PAD using the standard IEEE 802.3 Ethernet 32-bit FCS computation algorithm. The FCS field shall not be included when MIC is used in the LLC frame encapsulating the LCDU.

Bits of LCDU shall be transmitted starting from the first octet of the destination MAC address.

The encapsulation of encrypted and unencrypted LCDUs into LLC frames is described in Figure 8-7.

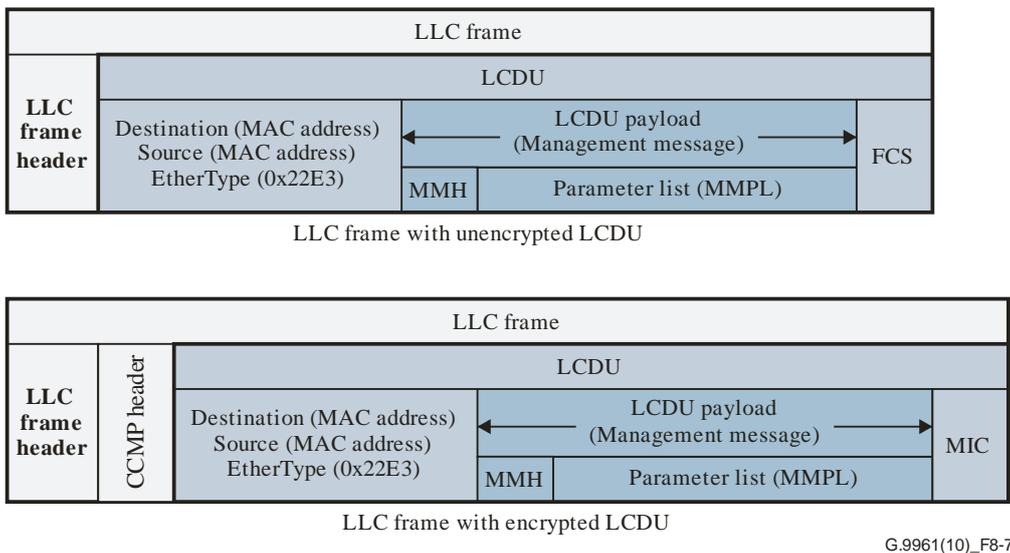


Figure 8-7 – Encapsulation of an LCDU into an LLC frame

8.1.4 Medium access control (MAC)

The functional model of the MAC is presented in Figure 8-8. It is intended to describe in more detail the MAC functional block presented in Figure 8-1.

TXOP associated with the first TXOP descriptor in the MAP is implicitly defined as the start of the MAC cycle.

The explicit specification of the start-time of a particular TXOP is relative to the start time of the MAC cycle.

The duration of each TXOP shall include the time required for transmission and an idle time at the end of TXOP (T_{IDLE}) needed to separate the last frame sent in a TXOP from the first frame sent in the following TXOP.

A node shall not transmit within a TXOP after the time instant computed as:

$$TXOP_{LatestTime} = TXOP_{StartTime} + TXOP_{Length} - T_{IFG_MIN}$$

In case of transmission of a frame sequence (e.g., MSG/ACK, RTS/CTS/MSG/ACK), the transmission of the last entire frame sequence shall complete no later than $TXOP_{LatestTime}$.

The transmission time line in a MAC cycle is divided into time units of duration $TIME_UNIT$ where the duration of a $TIME_UNIT$ shall be $TICK$ duration (see clause 8.4) times a constant factor defined in the MAP (see $TICK_Factor$, clause 8.8.3). All TXOPs shall start on a $TIME_UNIT$ boundary. The duration of any TXOP shall be equal to an integral number of $TIME_UNITs$.

The start time of a particular TXOP relative to the start time of the MAC cycle is represented by the count of $TIME_UNITs$, where the $TIME_UNIT$ count at the start of the MAC cycle is zero. Nodes in the domain shall synchronize their transmission timing to the start of a TXOP by counting $TIME_UNITs$ from the start of the MAC cycle.

8.3 Transmission opportunities (TXOPs) and time slots (TSs)

The MAC cycle includes one or more transmission opportunities (TXOPs) of different types. The following types of TXOPs are defined:

- Contention-free TXOP (CFTXOP);
- Shared TXOP (STXOP).

An STXOP is divided into one or more time slots (TSs) where each TS represents an opportunity to start transmitting for the node or nodes assigned to this TS. A node assigned to the TS may either use the opportunity to start transmitting during the TS, or pass on the opportunity to transmit.

Transmission rules within a TS depend on the type of the TS. If the node passes on the opportunity, it shall wait until the next opportunity to transmit in a subsequent TS assigned for this node. The duration of a TS ($TS_DURATION$) is medium-dependent and is defined in clause 8.4.

An STXOP can contain the following types of TSs:

- Contention-free TS (CFTS);
- Contention-based TS (CBTS).

An STXOP can be composed of only CFTSs, only CBTSs, or both CFTSs and CBTSs. An STXOP that is composed of CBTSs only is denoted as CBTXOP.

An example of a MAC cycle composed of TXOPs of different types is illustrated in Figure 8-12.

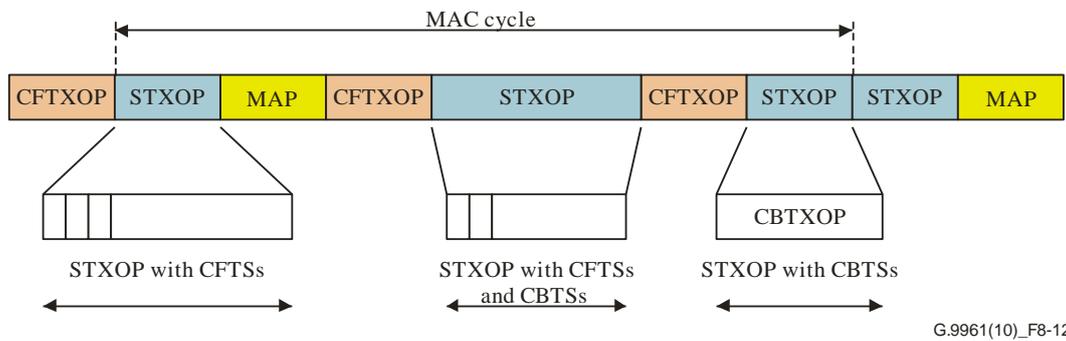


Figure 8-12 – Example of a MAC cycle structure

At least one MAP shall be sent each MAC cycle in a dedicated CFTXOP assigned by the domain master. This MAP shall be transmitted via a MAP-A frame (see clause 8.8.1).

The domain master shall plan medium access during a MAC cycle by dividing the available medium access time within the MAC cycle time into TXOPs. The domain master shall partition the MAC cycle into CFTXOPs and STXOPs.

NOTE – The above partitioning should be done in accordance with service requirements of network nodes and domain scheduling decisions.

Medium access within STXOPs shall be performed using CFTSs and CBTSs. Each STXOP may contain zero, one or more CFTSs, each assigned to a given node. Similarly, each STXOP may contain zero, one, or more CBTSs, and each CBTS is assigned to several nodes potentially contending for this CBTS.

The type, placement and duration of TXOPs within a MAC cycle and the order of the TSs inside a STXOP is assigned by the domain master according to internal scheduling decisions, which are beyond the scope of this Recommendation.

The format for describing the TXOPs and TS assignments in the MAP is described in clause 8.8.

8.3.1 Assignment of nodes and connections to TXOPs and TSs

Nodes and connections may be assigned to a particular TXOP or TS by the domain master. The assignment is performed based on domain master internal scheduling decisions, which are beyond the scope of this Recommendation. When a node signals an establishment of a new flow, the domain master may assign the corresponding connection to a TXOP or TS.

A node shall only transmit within a TXOP and TS to which it has been assigned or in which it is allowed to transmit.

8.3.1.1 Persistent and non-persistent TXOPs

MAP schedule persistence allows the domain master to inform nodes about TXOP allocations valid for a number of consecutive MAC cycles (see clause 8.8.6).

A TXOP can be assigned as persistent or non-persistent:

- A non-persistent TXOP is only valid for the next MAC cycle;
- A persistent TXOP is valid for the next and a number of subsequent MAC cycles.

All TXOP types described in clause 8.3 can be either persistent or non-persistent (default). The duration of a persistent TXOP is fixed during the time it is persistent.

The MAP message shall indicate if the allocated TXOP is persistent or not, and for how many MAC cycles it is persistent. The duration of all MAC cycles during the time of persistency shall be constant.

8.3.3.1 TS size and timing

The TS start times shall be calculated relative to a single time base T_{base} . The time base T_{base} shall initially be set to the start time of the TXOP, which is also the start time of the first TS of the TXOP, as presented in Figure 8-13. The time base shall be adjusted at the end of each transmitted frame sequence (P) on the medium. The adjusted time base T'_{base} shall be set to the time $T_{end} + T_{IFG_MIN}$, where T_{end} is the time when the last frame of the transmitted frame sequence is completed, and T_{IFG_MIN} is the duration of the idle time serving to form an inter-frame gap between two subsequent frame sequences.

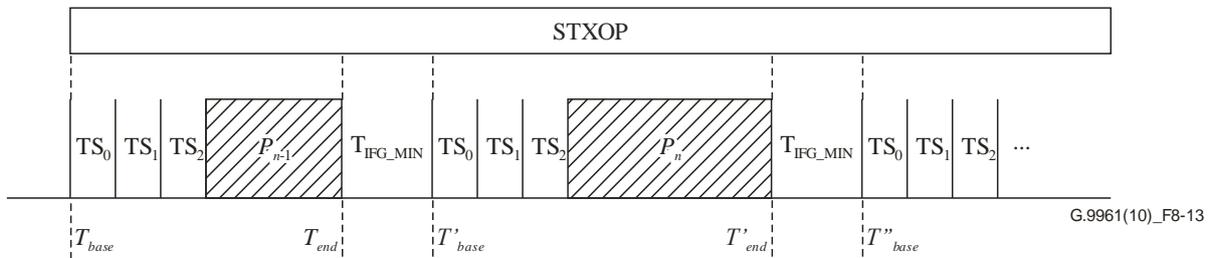


Figure 8-13 – Time slot timing

The duration of an unused time slot (e.g., TS₀ in Figure 8-13) is specified by parameter TS_DURATION, which is medium dependent and defined in clause 8.4.

8.3.3.2 TS assignment rules

The TSs scheduled in the MAP within an STXOP form a grid of transmission opportunity start times that starts at the beginning of the STXOP as described in clause 8.3.3.1. Each TS in the grid serves as a placeholder, reserving an opportunity for the nodes associated with this TS to transmit.

Nodes associated with a TS may either utilize the opportunity to transmit or to pass on the TS. If a TS is utilized, the next TS shall start T_{IFG_MIN} after the end of the frame sequence transmission in the TS. If a TS is not utilized, the next TS shall start TS_DURATION after the start of this unutilized TS. In either case, the nodes associated with the next TS in the grid sequence shall be given the opportunity to transmit in that TS.

Nodes sharing an STXOP shall follow the grid of TSs according to the TS assignment rules advertised in the MAP for that STXOP and actual TS usage in order to determine which TS is the next TS on the line as described in clauses 8.3.3.2.1 and 8.3.3.2.2.

Since operation in STXOPs is based on carrier sensing, the domain master should avoid assigning mutually hidden nodes to the same STXOP. Identification of mutually hidden nodes is based on the topology information communicated by nodes to the domain master, as defined in clause 8.6.4.

8.3.3.2.1 Sequential TS assignment rule

The sequential TS assignment rule is the default rule for TSs in an STXOP.

If the current TS is associated with the sequential TS assignment rule, the next TS of that STXOP shall be the next TS described in the MAP for that STXOP, regardless of whether this TS was used or not.

If the current TS is the last TS of the STXOP described in the MAP, the next TS of that STXOP shall be the first TS described in the MAP for that STXOP.

The example in Figure 8-14 describes the grid of TSs within a STXOP. The numbers shown in the TSs represent DEVICE_IDs. The figure shows eight TSs, seven CFTSs and one CBTS (identified by DEVICE_ID = 0), when the sequential TS assignment rule is used for all the TSs of the STXOP. The

order of the TSs in the STXOP in the MAP starts with a CFTS for node with DEVICE_ID "7" and ends with the CBTS.

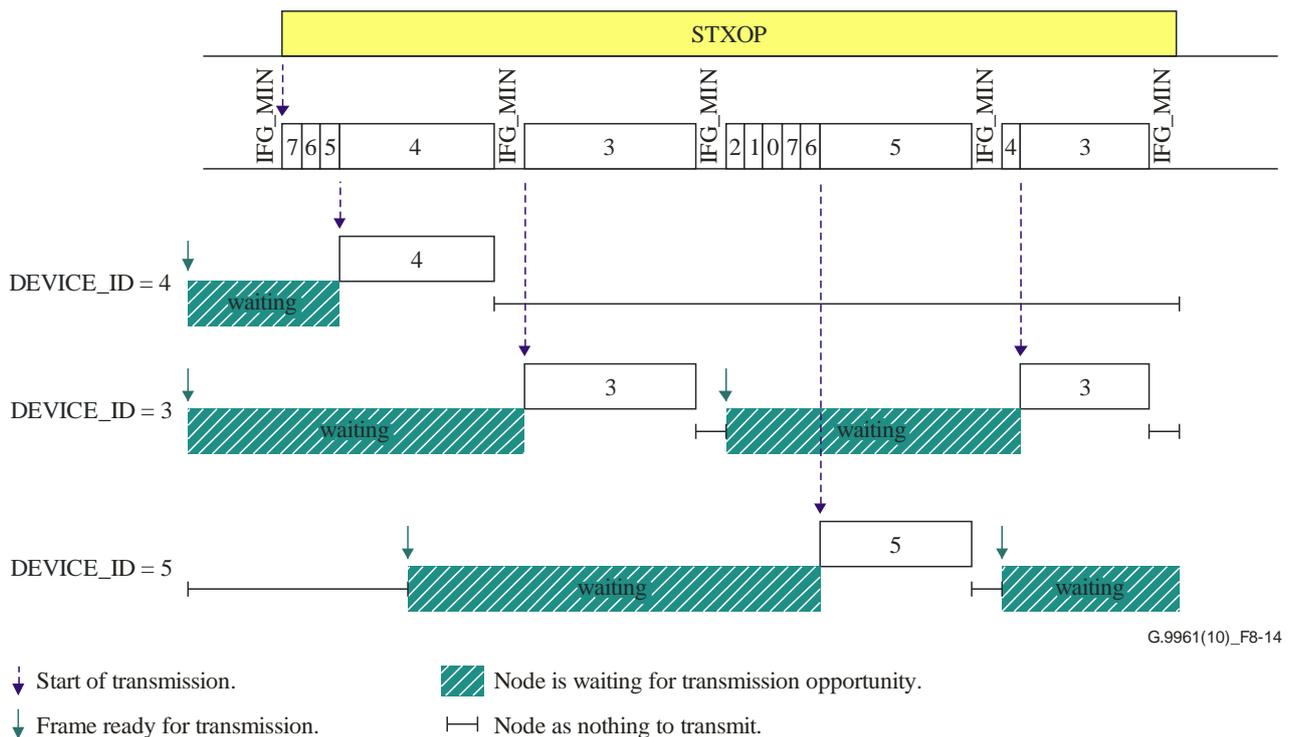


Figure 8-14 – Sequential TS assignment rule example

In Figure 8-14, each TS is marked with a number that is associated with the node whether an opportunity to transmit has been assigned. The first opportunity to transmit in Figure 8-14 is reserved for the node associated with TS marked "7". The node associated with TS "7" passes this opportunity to transmit, and so do nodes associated with TS "6" and TS "5". The node associated with TS "4" utilizes the TS for transmission.

The grid of TSs continues sequentially in the order the TSs were described in the MAP regardless if a TS was used or not. Once the current TS on the line is the last TS that was described for the STXOP in the MAP (the CBTS in Figure 8-14), the next TS on the line is the first TS of the STXOP as was described in the MAP (the CFTS of node with DEVICE_ID "7").

Figure 8-14 also shows the time that each node waits before its assigned TS starts.

8.3.3.2.2 Line activity dependent TS assignment rule

Several TSs within the same STXOP can be grouped together (TS grouping and numbering of the groups is described in clause 8.8.4). Each of these groups can have common attributes via a group information extension (see clause 8.8.4.1.3). The maximum number of groups within a STXOP is 127 groups due to the limitation on the number of TXOP descriptors in the MAP, describing TSs, within a single STXOP (see clause 8.8.4). Line activity dependent TS assignment rules are specified for these groups of TSs to allow passing the media access opportunity from one group of TSs to another, depending on the usage of those TSs.

A TS that provides an opportunity to access the medium at the current time is referred to as the current TS, and the group that this TS belongs to is referred to as the current group.

The next TS of a group is the TS that follows the current TS in that group as described in the MAP. The next TS of that STXOP shall be the next TS of the group of TSs whether control is passed following the current TS. The group to which control is passed following the current TS is defined by the sequential and line-activity dependent TS assignment rules.

field and other relevant fields (e.g., RPRQ, BEF) of the PHY-frame header of the transmitted frames in the frame sequence.

Figure 8-19 describes a duration-based CBTS closure. The CBTS closes after the entire frame sequence duration.

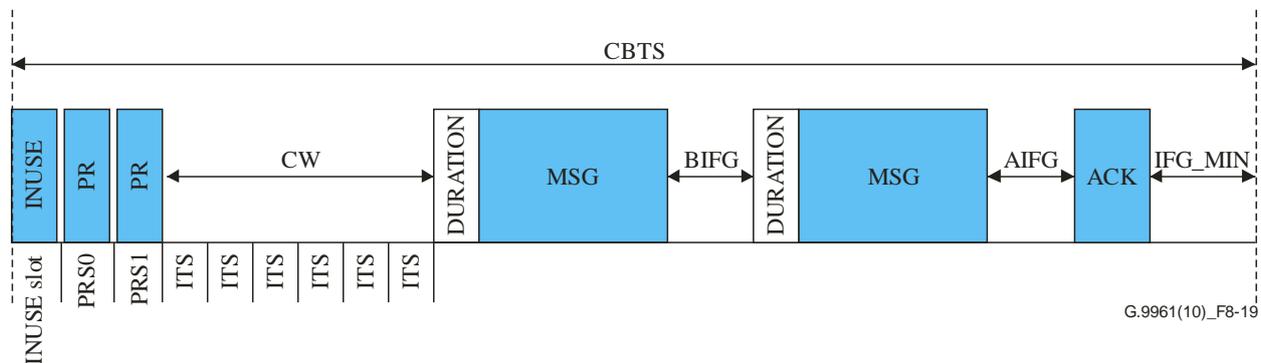


Figure 8-19 – Example of duration-based CBTS closure

If a node is unable to determine the exact closure point of the CBTS, due to a PHY-frame header error, inferring a collision or due to misdetection of a frame in the transmitted frame sequence, the node shall infer loss of synchronization with the TS grid of the STXOP and shall refrain from transmission in the STXOP until it resynchronizes with the TS grid as described in clause 8.3.3.6.

8.3.3.4.5.1.2 Timeout-based from frame sequence start CBTS closure

When the CBTS closure mode is timeout-based from frame sequence start, all nodes sharing a STXOP shall close a CBTS that was used for transmission at the minimum of (Max_TS_Length, the remaining time until the end of the STXOP) after the start of the frame sequence transmission, where Max_TS_Length is defined for the CBTS in the maximum transmission limitation extension in the MAP (see clause 8.8.4.1.4).

When the description of the CBTS in the MAP does not include maximum transmission limitation extension, DEFAULT_TBFSS_TIMEOUT (see clause 8.4) shall be used instead of Max_TS_Length.

Figure 8-20 describes a timeout-based from frame sequence start CBTS closure.

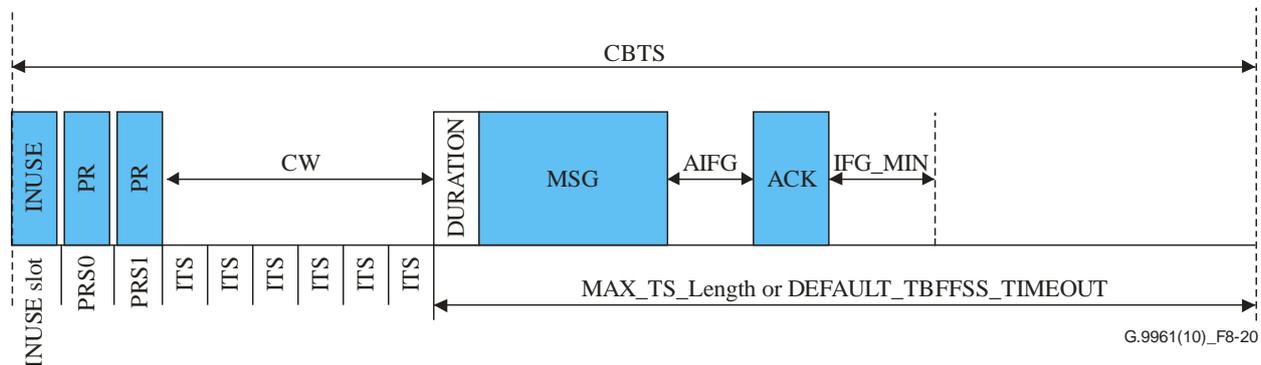


Figure 8-20 – Example of timeout-based from frame sequence start CBTS closure

If a node is unable to determine the exact closure point of the CBTS, due to misdetection of the first frame in the transmitted frame sequence, the node shall infer loss of synchronization with the TS grid of the STXOP and shall refrain from transmission in the STXOP until it resynchronizes with the TS grid as described in clause 8.3.3.6.

If INUSE signal is not required and RTS/CTS protocol is used, the node shall follow the rules described in clause 8.3.3.4.4. The CBTS shall be closed using the rules defined in clause 8.3.3.4.5.

8.3.3.4.7 Use of INUSE signal in CBTS

The domain master shall assign INUSE signal for all CBTS in a STXOP that contains one or more CFTS and for all CBTS in a CBTXOP that contains CBTS assigned in the MAP with different user priorities, or with different set of contending nodes, or with different use of PR signals.

The domain master shall not assign INUSE signal for CBTS of a CBTXOP containing only CBTS with same attributes describing user priorities, set of contending nodes, and use of PR signals.

8.3.3.4.8 Use of CBTS for node registration

An RCBTS, which is a type of CBTS that is identified using the special TXOP descriptor in the MAP as defined in clause 8.8.4.2, shall be used for registration purposes only.

Any CBTS can be used for registration if it is allowed by the domain master, as specified in the MAP (see clause 8.8.4.1.5). The registering node shall send its registration messages (see clause 8.6.1.1.4) using MA priority MA1.

NOTE – In a domain where hidden nodes are expected, the domain master should allocate CFTXOPs or CBTXOPs for registration of new nodes. Usage of RCBTS, or usage of CBTS in a STXOP that contains one or more CFTSs or contains CBTSs with different attributes, for registration of new nodes, is not recommended.

8.3.3.4.9 Collision inference

A node that has transmitted a frame sequence in a CBTS shall infer a collision if it has indicated that acknowledgement is required for the transmitted frame and any of the following cases occurs:

- 1) None of the expected acknowledgement frames (ACK or Mc-ACK) were received.
- 2) All of the expected acknowledgement frames (ACK or Mc-ACK) were received but they all indicate that all segments in all frames of the frame sequence were received in error (indicated by the BAD_BURST field in clause 7.1.2.3.2.3.7/[10.1.2.3.2.3.7](#) of [\[ITU-T G.9960\]](#)).
- 3) Only some of the acknowledgement frames (Mc-ACK) were received, but they all indicate that all segments in all frames of the frame sequence were received in error.

8.3.3.5 Enhanced frame detection (EFD) STXOP

An STXOP may be assigned by the domain master as an EFD STXOP only if this STXOP requires tracking of the grid of TS (i.e., contains CFTSs, a mixture of CFTSs and CBTSs or a mixture of CBTSs of different attributes – see clause 8.3.3). The size of a TS within an EFD STXOP shall be $2 \times \text{TS_DURATION}$ (see clause 8.4).

A node that intends to transmit in a CFTS within an EFD STXOP shall transmit an INUSE signal within a time window of TX_ON from the beginning of the TS before any other transmission in the TS. The frame following the INUSE signal shall be transmitted within a time window of TX_ON starting TS_DURATION microseconds from the beginning of the TS.

A node that intends to contend for transmission in a CBTS within an EFD STXOP shall transmit an additional INUSE signal within a time window of TX_ON from the beginning of the TS, before any other transmission in the TS, which already starts with an INUSE signal (i.e., two INUSE signals are transmitted). The node shall transmit the second INUSE signal within a time window of TX_ON starting TS_DURATION microseconds after the start of the TS.

Nodes sharing an EFD STXOP shall consider a CFTS as used when either the INUSE signal or a frame transmission (frame preamble) is detected.

Nodes sharing an EFD STXOP shall consider a CBTS as used when at least one of the two INUSE signals at the beginning of the TS is detected. It shall then follow the rules in clause 8.3.3.4.5 to close the CBTS.

A node that had detected an INUSE signal in a CFTS within an EFD STXOP but has not detected the frame transmission after, shall infer that it has lost synchronization with the TS grid and shall act as described in clause 8.3.3.6.

8.3.3.6 TS grid synchronization loss and recovery

8.3.3.6.1 TS grid synchronization loss detection

A node shall infer it has lost synchronization with the TS grid of a STXOP in the following cases:

Table 8-9 – Error conditions used to infer TS grid synchronization loss

Error condition	Applicable TS
Invalid PHY-frame header	CFTSs and duration-based CBTSs
Inferring a collision (see clause 8.3.3.4.9)	Duration-based CBTSs
No frame detection after INUSE was detected	1) Duration-Based CBTS with INUSE 2) Timeout-Based CBTS with INUSE counted from frame sequence start 3) CFTS in an enhanced frame detection (EFD) STXOP
No frame detection after PRS was detected	Duration-based CBTS without INUSE
CURRTS (see clause 7.1.2.3.2.2.21/10.1.2.3.2.2.21 of [ITU-T G.9960]) in the received PHY-frame header differs from the node's view of the current TS identity	All types of TSs

A node that has inferred loss of synchronization with the TS grid within a CBTXOP that does not require INUSE signal shall follow the CBTS closure rules described in clause 8.3.3.4.5.3. In all other cases, a node that has inferred loss of synchronization with the TS grid shall refrain from transmission until it resynchronizes with the TS grid as described in clause 8.3.3.6.2 if resynchronization with the TS grid is required or until the end of the STXOP. The domain master shall indicate whether to attempt resynchronization with the TS grid via the TXOP attributes extension data (see clause 8.8.4.1.1).

8.3.3.6.2 TS grid synchronization recovery

Upon reception of a valid PHY-frame header a node shall first resynchronize with the TS grid timing by setting T_{base} as described in clause 8.3.3.1 and then:

- if a sequential TS assignment rule is used, the node shall resynchronize with the TS grid identity using the CURRTS field of the received PHY-frame header;
- If a line activity dependent TS assignment rule is used, the node should try to resynchronize with the TS grid identity using the CURRTS field of the received PHY-frame header. If synchronization cannot be recovered, the node shall refrain from transmission for the remainder of the STXOP.

8.3.3.7 Silent TXOP or TS

A TXOP or TS of this type prohibits transmission by all nodes within the domain. A silent TXOP or TS shall be identified in the MAP message as described in clause 8.8.4.2.

NOTE – Example uses of this type of TXOP or TS might be coexistence with legacy devices, coordination with neighbouring networks, and interference or noise measurement.

8.3.4 Medium access in CFTXOPs

Each CFTXOP may be assigned to:

- A single source node and a minimum user priority identified by the tuple (SID, PRI); or
- a data connection that a single source node originates, identified by the tuple (SID, FLOW_ID); or
- a single source node, a single destination node and a minimum user priority identified by the tuple (SID, DID, PRI).

If the CFTXOP is assigned to a certain data connection identified by the tuple (SID, FLOW_ID) only traffic of that data connection and management connection corresponding to the same destination node may be sent using this CFTXOP.

If the CFTXOP is assigned to a source node with a certain user priority identified by the tuple (SID, PRI), only MPDUs with equal or higher MPDU priority may be sent by that node using this CFTXOP.

If the CFTXOP is assigned to a source node, a destination node and a certain user priority identified by the tuple (SID, DID, PRI), only MPDUs with equal or higher MPDU priority addressed to that destination node may be sent using this CFTXOP.

Only the assigned node may start a frame sequence within that CFTXOP. Other nodes may transmit within the same CFTXOP if their transmission is part of the same frame sequence (e.g., bidirectional transmission). When Imm-ACK or Mc-ACK is requested, each receiver that is requested to acknowledge shall send its acknowledgement within that CFTXOP.

A node associated with a CFTXOP may transmit one or more frame sequences in its assigned CFTXOP. A frame sequence transmission may start at any time within the CFTXOP, but shall be complete, including Imm-ACK or Mc-ACK, if requested, before the end of that CFTXOP.

NOTE – Receivers should use carrier sensing to detect the start of the transmitted frames within a CFTXOP.

8.3.5 Transmission using PHY frame bursting

The PHY frame bursting is a type of transmission when several PHY frames that are part of the same burst are transmitted in succession without relinquishing the medium. A single ACK frame shall acknowledge the status of the LPDUs in all the frames of the burst, if required. Each of the PHY frames in the burst shall be separated from each other by a gap called the burst inter-frame gap (BIFG). The ACK frame shall be separated from the burst by a gap called the ACK inter-frame gap (AIFG). The duration of AIFG (T_{AIFG}) and BIFG (T_{BIFG}) are defined in clause 8.4.

If the transmitter has no knowledge of the 'receiver specific' AIFG (see clauses 8.6.1.1.4.1 and 8.6.4.3.1) or if the last frame of the PHY frame burst includes less than MIN_SYM_VAR_AIFG symbols, the gap between the frame and the following Imm-ACK shall be $T_{AIFG-DR}$, specified by the receiver upon registration, or T_{AIFG-D} , if the transmitter has no knowledge of the $T_{AIFG-DR}$ for the receiver (see clause 8.4). ~~Otherwise the gap shall be T_{AIFG} .~~ ~~The~~

If the transmitter has no knowledge of the 'receiver specific' MIN_SYM_VAR_AIFG announced by the receiver upon registration, it shall use the parameter MIN_SYM_VAR_AIFG_DEFAULT is defined in clause 8.4, for each media.

The transmitter indicates usage of ~~either T_{AIFG} , or T_{AIFG-D}~~ or $T_{AIFG-DR}$ by using the AIFG_IND bit and AIFG_TYPE bits in the PHY-frame header (see clause 7.1.2.3.2.2.16/10.1.2.3.2.2.16 of [ITU-T G.9960]).

The source node shall include at least MIN_SYM_VAR_AIFG symbols in each PHY frame within a PHY frame burst except for the last frame in the burst. In case the source node does not have enough symbols to fulfil this condition, it shall terminate the burst by setting the BEF field to one.

Figure 8-23 shows an example of PHY frame bursting with three PHY frames in a burst.

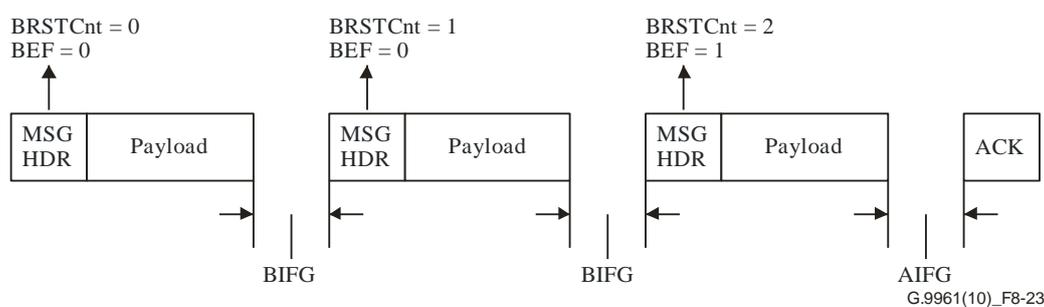


Figure 8-23 – Example of PHY frame transmission with bursting and Imm-ACK

The BRSTCnt is set to zero on the first PHY frame in the burst and is incremented by one on each subsequent PHY frame in the burst. The third PHY frame in the burst has the burst end flag (BEF) field of the PHY-frame header set to one to indicate that it is the last PHY frame in the burst. The ACK frame contains acknowledgement information for all three PHY frames.

Rules for setting BEF and BRSTCnt are described in clauses 7.1.2.3.2.2.15/[10.1.2.3.2.2.15](#) and 7.1.2.3.2.2.14/[10.1.2.3.2.2.14](#) of [ITU-T G.9960], respectively.

PHY frame bursting shall be limited to four PHY frames in a burst.

When using PHY frame bursting mechanism, the PHY frame burst shall be considered as a single PHY frame for the purpose of filling the RPRQ field of the PHY frames composing the burst (see Table 7-15/[10-15](#)). The RPRQ field of all the individual PHY frames of the burst shall be set to the same value.

When the destination node receives a PHY frame with BEF set to zero, it shall not transmit an ACK and shall store the corresponding ACK information. This process continues until a PHY frame with BEF = 1 is received, which is the last frame of the burst. Upon receiving the last PHY frame in the burst, the receiver shall transmit an ACK frame, acknowledging all PHY frames belonging to that burst.

The PHY frame bursting can be used both in CCTXOP and STXOP. When PHY frame bursting is used in a STXOP, all nodes that are sharing the STXOP shall determine the end of the frame sequence, including frames of the burst and the ACK, and further track the TS grid in the same way as in the case of a non-bursting PHY frame transmission. The maximum duration of the PHY frame burst shall not exceed the maximum PHY frame duration allowed for a transmission of a non-bursting frame sequence in a TXOP or TS, determined by the parameter MAX_TS_Length and indicated in the MAP. The TS used for burst transmission shall be closed using the same rules as for non-bursting transmission, using the duration of the whole burst (from the start of the first frame to the end of the last frame) as if it were a duration of a single MSG frame in non-bursting transmission. The end of the last frame in the burst is indicated in the header of the last frame (with BEF=1).

If the node misses a PHY frame in the burst, it shall wait for the last frame in the burst (with BEF=1). If the last frame is not detected prior to the expiration of the maximum allowed PHY frame duration of a non-bursting transmission, the node shall consider the duration of the burst unknown and the receiving node shall not transmit the ACK frame. If the last frame (BEF=1) is detected prior to the expiration of the maximum allowed PHY frame duration of a non-bursting transmission, the receiving node shall send the ACK frame. In both cases the TS shall be closed as described above. The loss of TS grid synchronization, if detected, shall be recovered as defined in clause 8.3.3.6.

NOTE 1 – The transmitter of a PHY frame burst needs to ensure that the total number of LPDUs transmitted in all PHY frames of the burst is consistent with the flow control information provided by the receiver (see clause 8.12.4).

All PHY frames in a burst shall only contain LPDUs belonging to a single data connection and a single management connection.

NOTE 2 – PHY frame bursting improves the medium access efficiency by combining multiple PHY frame transmissions into a single PHY frame sequence with smaller inter-frame gaps and a single acknowledgment. For example, a transmitter can send up to four frames in a single PHY frame burst, with each frame sent in a different BAT region as shown in Figure 7-11.1, instead of sending separate PHY frames with larger inter-frame gap.

8.3.6 Scheduled inactivity

A node is said to be in inactive state if it is not ready to receive any PHY frames, and is not engaged in serving traffic. Otherwise, a node is said to be in active state. The continuous time that a node remains in active state is denoted as active period. The continuous time that a node remains in inactive state is denoted as inactive period.

A node in inactive state does not serve any traffic; hence the domain master should not assign any dedicated resource (for example, CFTSs or CFTXOPs) to this node. However, a node in inactive state may still transmit in a TXOP or TS in which it is allowed to transmit as specified in the MAP (see clause 8.8.4). If a node in inactive state transmits requesting acknowledgement, it shall be ready to receive possible ACK frames from the destination node. The destination node is allowed to acknowledge transmissions from a node in inactive state. Other than this case nodes shall not transmit to a node in inactive state. A node that has traffic to send to an inactive node may request the DM the activation of that node if the inactive node is in L3 mode (see clause 8.21.4).

Using scheduled inactivity, the domain master can schedule for any node one or more inactive periods. Active and inactive periods for all nodes are ultimately determined by the domain master and broadcast to all nodes of the domain via the MAP. A node can use this feature to implement power saving strategies.

8.3.6.1 Scheduled inactivity over multiple MAC cycles

Using this mechanism, a node can stay in inactive state over multiple MAC cycles. Each inactive period starts at the beginning of the MAC cycle, and stops at the end of the same or one of the subsequent MAC cycles. During an inactive period, a node is not required to receive MAP frames and decode MAPs. This feature may be used for low-power mode (L2 – see clause 8.21.3), or standby mode (L3 – see clause 8.21.4).

8.3.6.1.1 Long inactivity scheduling

A node may request the domain master for inactivity scheduling for multiple MAC cycles by sending an IAS_LongInactivity.req message. The node may request two types of long inactivity scheduling: If the node wants this schedule to be effective only once, it indicates the requested duration of the inactive period. If the node wants this schedule to be effective more than once, it indicates the requested duration of the inactive period and the requested duration of the active period that follows the inactive period. The sum of both periods is called a sleep cycle. In this case the specified inactive period followed by the active period will repeat until it is cancelled or changed by the domain master.

A default sleep cycle and an indication of the beginning of the next default sleep cycle shall be published by the DM through the power saving update auxiliary information field of the MAP. Nodes sending a Long Inactivity request should align their requests to this default sleep cycle.

The domain master, if the request is accepted, shall announce the inactivity schedule as proposed by the receiver. The start time and duration of the inactive period and the duration of the following active period (if applicable) for a long inactivity schedule shall be transmitted in the auxiliary information field of the MAP message. The domain master may use the validity counter-based update (AUX_VALID = 3-7 and ModificationFlag = 1) or the immediate update (ModificationFlag = 0) for long inactivity scheduling announcement (see clause 8.8.5).

The domain master may directly force any node in the domain to follow a long inactivity scheduling, defined in the auxiliary information field of the MAP message (see clause 8.8.5.3). The scheduling shall coincide with the low power mode capability of the EP, exchanged in the registration request

Table 8-10 – Format of the MMPL of the IAS_LongInactivity.req message

Field	Octet	Bits	Description
LIS_ACT_DUR	3 and 4	[15:0]	Requested duration of the active period that immediately follows the inactivity period specified by LIS_INACT_DUR, expressed in units of 1/16 th of the MAC CYCLE duration, represented as a 16-bit unsigned integer. This value shall be larger than or equal to the length of one MAC cycle or set to zero.

NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.

Table 8-11 – Format of the MMPL of the IAS_LongInactivity.cnf message

Field	Octet	Bits	Description
ReasonCode	0	[3:0]	Reason why inactivity is denied (Note 1) 0000 = no reason specified 0001 = proposed inactivity period is too long 0010 = proposed inactivity period is too short 0011 = proposed inactivity rejected due to unsolvable scheduling issues 0100 = proposed inactivity rejected due to scheduling issues being solved. Retry is allowed 0101 = proposed inactivity rejected due to unsolvable topology issues 0110 = proposed inactivity rejected due to topology issues being solved. Retry is allowed.
Reserved		[7:4]	Reserved by ITU-T (Note 2)

NOTE 1 – Definition of other ReasonCodes is for further study.
 NOTE 2 – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.

Table 8-11.1 – Format of the MMPL of the IAS_LongInactivity.ind message

Field	Octet	Bits	Description
Message confirmation	0	[0]	This bit shall be set to one.
Reserved		[7:1]	Reserved by ITU-T (Note 1)

NOTE 1 – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.

A node that needs to transmit to another node of the domain that is in standby mode L3 (Long Inactivity Scheduling) may request the DM to instruct the node in L3 to leave L3 state. For this, the requesting node shall send the message LP_RequestWakeUp.req to the DM (see clause 8.21.4).

The format of the MMPL of the LP_RequestWakeUp.req message shall be as shown in Table 8-11.2.

Table 8-11.2 – Format of the MMPL of the LP_RequestWakeUp.req message

Field	Octet	Bits	Description
NumNodesWakeUp	0	[7:0]	Indicates the number of nodes (k) that are being requested to leave L3 power saving mode
WakeUpRequest[0]	1	[7:0]	DeviceID of the first node that is requested to leave L3 power saving mode.
...
WakeUpRequest[k-1]	k	[7:0]	DeviceID of the last node that is requested to leave L3 power saving mode.

8.3.6.2 Scheduled inactivity in a single MAC cycle

Using this mechanism, a node may switch between active and inactive states during periods of time shorter than a MAC cycle. Regardless of the start and stop times of the inactivity periods, a node shall listen to the MAP message. This feature may be used for reduced-power mode (L1).

8.3.6.2.1 Short inactivity scheduling

A node may request the domain master for an inactivity scheduling for a fraction of a MAC cycle by sending an IAS_ShortInactivity.req message. This message defines the inactive periods within a MAC cycle. The node may request two types of short inactivity scheduling: valid once or valid until cancelled or changed. A domain master that received an IAS_ShortInactivity.req message shall respond with an IAS_ShortInactivity.cnf message.

The domain master, if the request is accepted, shall announce the inactivity scheduling as proposed by the receiver or a scheduling adapted to the needs of the whole domain. The inactive and active portions of the MAC cycle for a short inactivity schedule shall be transmitted in the auxiliary information field of the MAP message (see clause 8.8.5.4). The domain master may use the validity counter-based update (AUX_VALID = 3-7 and ModificationFlag = 1) or the immediate update (ModificationFlag = 0) for short inactivity scheduling announcement (see clause 8.8.5).

The domain master may directly force any node in the domain to follow a short inactivity scheduling, defined in the auxiliary information field of the MAP message (see clause 8.8.5.4). The scheduling shall coincide with the low power mode capability of the EP, exchanged in the registration request message (ADM_NodeRegistrRequest.req). The EP, receiving the short inactivity schedule, shall confirm the successful message reception by sending the IAS_ShortInactivity.ind message to the domain master. The EP shall conduct the inactivity scheduling in the next MAC cycle after sending out the IAS_ShortInactivity.ind message.

The node that requested inactivity may transition into the inactive state as instructed in the MAP. If the MAP does not include the inactivity schedule within 100 ms after the request was sent, the node may repeat the request 200 ms after it transmitted the last request.

The domain master, if the request is rejected, shall indicate why the request for inactivity is denied with a reason by sending the IAS_ShortInactivity.cnf message. The node that received an inactivity denial shall act based on the ReasonCode.

Nodes in short inactivity scheduling shall be able to decode the MAP at every MAC cycle. The DM shall schedule MAP transmission(s) and inactivity schedules in a way that ensures that each node can receive MAP(s) during its active portion of the MAC cycle.

A node can change the current inactivity schedule by sending another IAS_ShortInactivity.req while it is in active state. The domain master can terminate or change the current inactivity scheduling any time by sending different inactivity schedules.

The format of the MMPL of the IAS_ShortInactivity.req, IAS_ShortInactivity.cnf and IAS_ShortInactivity.ind messages shall be as shown in Table 8-12, Table 8-13 and Table 8-13.1, respectively.

Table 8-12 – Format of the MMPL of the IAS_ShortInactivity.req message

Field	Octet	Bits	Description
SIS_TYPE	0	[2:0]	Proposed type of short inactivity scheduling 0: inactivity schedule is valid only once. 1: inactivity schedule repeats itself. In this case the inactivity schedule is valid until it is cancelled or changed. 2: inactivity schedule is cancelled by the node. In this case SIS_IND shall be set to zero. Other values are reserved by ITU-T.
Reserved		[7:3]	Reserved by ITU-T (Note)
SIS_IND	1 and 2	[15:0]	Requested indication of one or more inactive periods within a MAC cycle represented, as a 16-bit unsigned integer. 16-bit map is used to represent inactive periods. The bit0 (LSB) and bit15 (MSB) correspond to the first and last 1/16-th portions of a MAC cycle, respectively. A bit corresponding to each portion shall be set to one if the node is active during that time, and set to zero otherwise.

NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.

Table 8-13 – Format of the MMPL of the IAS_ShortInactivity.cnf message

Field	Octet	Bits	Description
ReasonCode	0	[3:0]	Confirmation code 0000 ₂ = Rejected - no reason specified 0001 ₂ = Rejected - proposed inactivity period is too long 0010 ₂ = Rejected - proposed inactivity period is too short 0011 ₂ = Rejected - proposed inactivity rejected due to unsolvable scheduling issues 0100 ₂ = Rejected - proposed inactivity rejected due to scheduling issues being solved. Retry is allowed. 0101 ₂ = Accepted –scheduling granted as requested 0110 ₂ = Accepted –scheduling granted with modifications provided in the MAP 0111 ₂ = Accepted – default scheduling to be used. Other values are reserved by ITU-T.
Reserved		[7:4]	Reserved by ITU-T (Note 1)

NOTE 1 – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.

Table 8-13.1 – Format of the MMPL of the IAS_ShortInactivity.ind message

Field	Octet	Bits	Description
Message confirmation	0	[0]	This bit shall be set to one.
Reserved		[7:1]	Reserved by ITU-T (Note 1)

NOTE 1– Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.

8.3.7 Bidirectional transmissions

Bidirectional transmissions between two nodes may be used to improve throughput and minimize latency of a traffic that is bidirectional in nature, such as TCP traffic with acknowledgements. The defined bidirectional mechanism is only applicable to nodes communicating directly (i.e., not via a relay node).

In case of bidirectional transmission, a node originating (sourcing) the bidirectional traffic and the destination node exchange special frames: a bidirectional message (BMSG) frame and a bidirectional acknowledgement (BACK) frame. Both BMSG and BACK carry data, and in the case of acknowledged transmissions, also an acknowledgement on the recently received frame.

If using acknowledged bidirectional transmission, the BMSG PHY frames shall use the format described in Tables 7-47/10-47 and 7-53/10-53 of [ITU-T G.9960], and the BACK PHY frames shall use the format described in Tables 7-48/10-48 and 7-54/10-54 of [ITU-T G.9960], in which the PHY frame header contains:

- $2 \times \text{PHY}_H$ information bits (EHI bit, in the PHY frame header, is set to one, see clause 7.1.2.3.1.7 of [ITU-T G.9960]) for Profile 1 OFBs;
- $2 \times \text{PHY}_{HCH}$ information bits for Profile 2 OFBs.

–If using unacknowledged bidirectional transmission and Profile 1 OFBs, the BMSG and BACK PHY frames shall use the format described in Tables 7-47~~3~~ and 7-48~~5~~ of [ITU-T G.9960], respectively, in which the PHY frame header contains PHY_H information bits (EHI bit in the PHY frame header ~~is shall be~~ set to zero in Profile 1 OFBs).

An exchange of BMSG and BACK frames forms a bidirectional frame sequence that shall last strictly inside the boundaries of the particular TXOP or TS assigned in the MAP for the node sourcing the bidirectional transmission, see Figure 8-24. With an acknowledged bidirectional transmission, a destination node may request for delayed acknowledgement (RPRQ field set to 10) when answering with a BACK frame to a BMSG frame indicating the closure of the bidirectional transmission (i.e., case 4-b below). In all other cases, only immediate acknowledgement is allowed (the valid values of RPRQ field are 00 and 01 only).

NOTE – Using delayed acknowledgement for the last BACK frame allows termination of the bi-directional transmission with a BACK frame, improving TCP efficiency.

A bidirectional transmission may be initiated by either a source node or a destination node using one of the following methods:

- A destination node, in case of acknowledged transmission, transmits to the source node, in response to a MSG frame requesting immediate acknowledgement, an ACK frame with the BTXRQ bit set to one;
- A destination node, in case of un-acknowledged transmission, transmits to the source node a MSG frame with BTXRQ bit set to one;
- A source node transmits to the destination node a BMSG frame with the BTXGL field set to a non-zero value.

- c) By sending an Imm-ACK frame, in case of acknowledged transmission, instead of BMSG frame. Previous BMSG frames in the frame sequence shall all carry $BTXEF = 0$.
- 5) The destination node may indicate that bidirectional transmission is not further needed (advice for termination of bidirectional transmission) by setting the $BTXRL = 0$ in the BACK frame. In response, the source node may terminate bidirectional transmission using any of three methods described above.

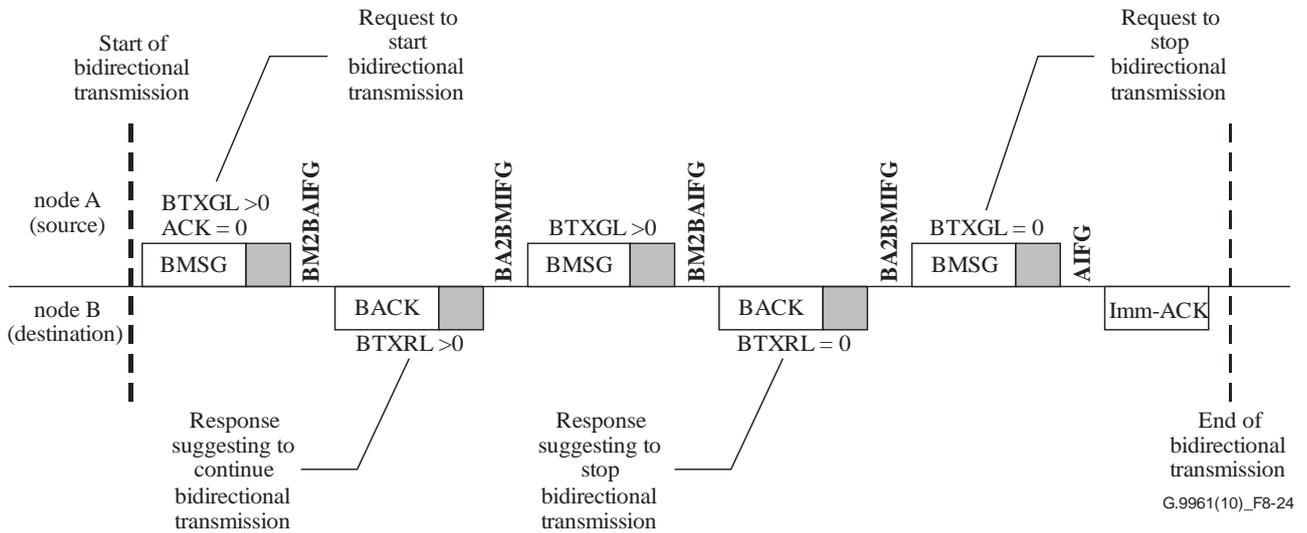


Figure 8-24 – Example of bidirectional transmission (invited by the originating node)

NOTE – Figure 8-24 presents a case when the destination node suggests to terminate bidirectional transmission and the source node requests that termination shall be done by the destination node (the destination node sends Imm-ACK). The source node may also terminate the bidirectional transmission itself by sending Imm-ACK instead on the last BMSG frame with the $BTXGL$ field set to zero.

The maximum duration of a BACK frame is determined by the source node in the $BTXGL$ field of the PHY-frame header. The destination node only indicates the desired duration of BACK frame in the $BTXRL$ field of the PHY-frame header of the previous BACK frame, but the final decision on the BACK frame duration limit (including the following IFG) is done by the source node. If a destination node indicates in the $RPRQ$ field that Imm-ACK is requested, the source node shall set the maximum granted length for BACK transmission so that there is sufficient time for the source node to transmit an Imm-ACK frame at the end of the transmission sequence (in response to the last BACK frame).

A responding BACK frame shall be transmitted $T_{BM2BAIFG}$ after the BMSG frame, and the responding BMSG frame shall be transmitted $T_{BA2BMIFG}$ after the BACK frame. The Imm-ACK frame shall be transmitted T_{AIFG} after the BMSG frame or after the BACK frame, respectively. In all of the following frame sequences:

- BMSG followed by a BACK;
- BACK followed by a BMSG;
- BMSG followed by an Imm-ACK;
- BACK followed by an Imm-ACK.

If the transmitter of the first frame has no knowledge of the 'receiver specific' AIFG (see clause 8.6.1.1.4.1 and clause 8.6.4.3.1) or if the first frame in any of the above frame sequences includes less than $MIN_SYM_VAR_AIFG$ symbols, the gap between this frame and the following frame shall be $T_{AIFG-DR}$, specified by the receiver upon registration, or T_{AIFG-D} if the transmitter has no knowledge of the $T_{AIFG-DR}$ for the receiver (see clause 8.4).~~5~~ Otherwise the gap shall be T_{AIFG} .

Table 8-14 – Parameters for APC, LLC and MAC

Parameter	Description	Medium			
		Power-line baseband (Note 2)	Coax BB	Coax RF	Phoneline
	<u>MIN_SYM_VAR_AIFG for the receiver</u>				
T _{RCIFG}	Inter frame gap between RTS and CTS	110 μs	29 μs	29 μs	74 μs
T _{CCIFG}	Inter frame gap between CTS and MSG frame	110 μs	29 μs	29 μs	74 μs
T _{BM2BAIFG}	Inter frame gap between BMSG and BACK frame	T _{AIFG}	T _{AIFG}	T _{AIFG}	T _{AIFG}
T _{BA2BMIFG}	Inter frame gap between BACK and BMSG frame	T _{AIFG}	T _{AIFG}	T _{AIFG}	T _{AIFG}
T _{BIFG}	Inter frame gap between MSG frames in PHY frame bursting	20.48 μs	Note 3	Note 3	20.48 μs
T _{McAIFG}	Inter frame gap between multicast ACK frames	20.48 μs	5.12 μs	5.12 μs	20.48 μs
TICK	The basic MAC resolution (at TXOP level)	10 ns	10 ns	10 ns	10 ns
MAP_TX_SETUP_TIME	The minimum time between the MAP and the MAC cycle it describes	2 ms	2 ms	2 ms	2 ms
MAX_ARQ_SLOTS	Maximum number of Mc-ACK slots in multicast acknowledgment	7	7	7	7
DEFAULT_TBFFSS_TIMEOUT	Default timeout used for closing the CBTS in "Timeout-based from frame sequence start" mode	2.5 ms	2.5 ms	2.5 ms	2.5 ms
DEFAULT_TBFCS_TIMEOUT	Default timeout used for closing the CBTS in "Timeout-based from CBTS start" mode	3.5 ms	2.5 ms	2.5 ms	2.5 ms
DEFAULT_ERR_CWOL_TIMEOUT	Default timeout for error conditions in "CBTS without	4.9 ms	3.61 ms	3.61 ms	4.04 ms

Table 8-14.2 – Handling of broadcast LLC frames (LCDU case)

Case	Type of broadcast	Leaf/ Non-leaf	LCDU DA	Broadcasting actions	Example
12			node's MAC address	Pass the frame to DLL management; Optional: forward through BRT	
13			Broadcast address	Pass the frame to DLL management and forward through BRT	
14			Reserved address	Pass the frame to DLL management and forward through BRT	Management message intended to all nodes
15			Destination MAC address not covered by cases 10-14	Forward through BRT	
<p>NOTE 1 – LAAT does not need to be consulted. NOTE 2 – Following BRT rules leads to a drop. NOTE 3 – RAAT does not need to be consulted. NOTE 4 – Although this table addresses how to handle the reception of any broadcast LLC frames, not all of these cases are currently allowed in the specification. For example, the DestinationNode is set to 0 by the transmitter for LLC frames in which the DA is set to the reserved MAC address 01-19-A7-52-76-96 (see clause 8.1.3.1.1.10).</p>					

Table 8-14.3 – Handling of broadcast LLC frames (APDU case)

Case	Type of broadcast	Leaf/ Non-leaf	APDU DA	Broadcasting actions	Example
16	Broadcast frame intended for another node in the network	Leaf	–	Drop the frame	Unicast frame not found by a previous relay node in its internal unicast routing tables and relayed in broadcast
17	(BRCTI = 1; MCSTI = 0; DestinationNode = DeviceID _{OtherNode})	Non-Leaf	–	Follow BRT rules.	Unicast frame not found by a previous relay node in its internal unicast routing tables and relayed in broadcast

Table 8-14.3 – Handling of broadcast LLC frames (APDU case)

Case	Type of broadcast	Leaf/ Non-leaf	APDU DA	Broadcasting actions	Example	
18	Broadcast frame intended for this node (BRCTI = 1; MCSTI = 0;	Leaf	–	Consider the frame as non-broadcast (unicast) and follow the corresponding rules (cases 1-6 of Table 8-14.5). Stop the broadcasting	Endpoint of a unicast frame not found by a previous relay node in its internal unicast routing tables and relayed in broadcast	
19	DestinationNode = DeviceID _{Node})	Non-Leaf		Consider the frame as non-broadcast (unicast) and follow the corresponding rules (cases 1-6 of Table 8-14.5). Stop the broadcast	Endpoint of a unicast frame not found by a previous relay node in its internal unicast routing tables and relayed in broadcast	
20	Broadcast frame intended to all the nodes	Leaf	In LAAT except node's MAC address	Pass to A interface; Do not relay (Note 1)		
21	(BRCTI = 1; MCSTI = 0; DestinationNode = BroadcastID)		In RAAT	Do not relay (Note 2) Optional: Pass to A interface.		
22			Node's MAC address	Pass to DLL management. Stop the broadcast through BRT		
23			Broadcast address	Pass to A interface; Pass to DLL management. Stop the broadcast through BRT	Standard broadcast	
24			Reserved address	Pass to DLL management. Stop the broadcast through BRT		
25			Destination MAC address not covered by cases 20-24	Pass to A-interface. Do not relay	Unknown Destination frames broadcast	
26			Non-Leaf	In LAAT except node's MAC address	Pass to A interface Optional: relay following BRT	
27				In RAAT	Relay following BRT Optional: Pass to A-interface	

A relayed MAP frame contains the time stamp of its transmission start time, which is an estimate of the domain master's NTR (see clause 7.1.2.3.2.1.2/[10.1.2.3.2.1.2](#) of [ITU-T G.9960]), and a time stamp marking the start time of the next MAC cycle (CYCSTART), so that each node that receives and decodes an RMAP will be able to determine the exact location of the next MAC cycle and the TXOPs and TSs described in that RMAP. All relayed MAP frames shall contain the same MAC cycle duration and MAP sequence number of the MAP frame that is relayed.

The relayed RMAP frame shall indicate the number of hops the RMAP relay node is from the domain master (see NUM_HOPS in clause 7.1.2.3.2.1.12/[10.1.2.3.2.1.12](#) of [ITU-T G.9960]).

8.5.6.1 MAP relaying for registration of hidden nodes

A node that intends to join the domain may not detect the MAP-D frames (see clause 8.8.1) transmitted by the domain master (i.e., the node is hidden from the domain master). In order for such a node to register with the domain master, another endpoint node (that is not hidden from the registering node) shall transmit MAP-D frames at the domain master's request. The domain master shall specify in the transmitted MAP message a TXOP descriptor to schedule the transmission of the RMAP-D and to specify the relay node (see Table 8-70).

A node that is assigned via the MAP to transmit an RMAP-D frame shall generate an RMAP-D frame according to the most updated information it currently has which is needed to build an RMAP-D frame. The RMAP-D shall contain all the auxiliary information that is needed by a registering node to synchronize with the MAC cycle and to transmit the registration request message frame (see clause 8.8.1).

8.5.6.2 MAP relaying for operation of registered hidden nodes

The domain master shall ensure that every node admitted to the domain can receive either a MAP-A frame or an RMAP-A frame in every MAC cycle.

When the domain master learns that at least one of the nodes in the domain is hidden from the domain master, it shall designate one or more nodes to relay the MAP-A frame in every MAC cycle. A node is designated if the domain master allocates a TXOP or TS to it to transmit an RMAP-A frame (see clause 8.8.4.2). The set of relays shall be selected using the topology information collected as described in clause 8.6.4.

NOTE – Selection of MAP relays (used for relaying MAP-A or MAP-D frames) can be done according to the following procedure:

- Step I: Build a topological representation of the domain using topology information described in clause 8.6.4;
- Step II: Build a logical spanning tree that includes all nodes in the domain and has the domain master as the root;
- Step III: Designate all non-DM non-leaf nodes as MAP relays.

In addition to the nodes selected with the procedure above, the domain master may designate other nodes to relay the MAP-A and MAP-D.

It is assumed that implementers choose a "shortest-path tree" when choosing a spanning tree, in order to minimize the number of MAP relays between the domain master and any given node.

8.5.7 Relaying messages

In some media types some of the nodes are hidden from others, and may be hidden from the domain master. In order to allow communication between hidden nodes, other nodes act as relays. The determination of the relays to be used to deliver a given LLC frame to its destination is explained in clause 8.6.4.

8.5.7.1 Relaying of LCDU

Any LLC frame received from the medium that contains an LCDU shall be relayed according to the following rules:

If the frame has been received with BRCTI = 0, MCSTI = 0 and the DestinationNode is the DEVICE_ID of the receiving node, the node shall extract the LCDU and pass it to the DLL management (cases 1,2,3,4,5 and 6 of Table 8-14.4).

If the frame has been received with BRCTI = 0, MCSTI = 0 and the DestinationNode is zero, the node shall:

- a) If the DA is the same as the address of the node or the standard broadcast address or the reserved MAC address 01-19-A7-52-76-96 (cases 9, 10 and 11 of Table 8-14.4), it shall extract the LCDU and pass it to the DLL management. The action taken by the DLL management entity depends on the contents of the LCDU and the role of the node in the domain;
- b) In all other cases (cases 7, 8 and 12 of Table 8-14.4), the node shall drop the LLC frame.

If the frame has been received with BRCTI = 0, MCSTI = 0 and the DestinationNode is present in the unicast routing tables (case 13 of Table 8-14.4), the node shall relay it to the appropriate node or nodes as indicated in the routing table.

If the frame has been received with BRCTI = 0, MCSTI = 0 and the DestinationNode is not present in the unicast routing tables:

- If the receiving node is a leaf node (case 14 of Table 8-14.4), the frame is dropped;
- If the receiving node is a non-leaf node (case 15 of Table 8-14.4), the node shall set the BRCTI to 1 and broadcast the received LLC frame to the nodes that are specified in the BRT while keeping the DestinationNode and OriginatingNode fields.

If the frame has been received with BRCTI = 0, MCSTI = 1 and with a known MSID (cases 16 and 18 of Table 8-14.4), the frame shall be relayed as specified in clause 8.17. In addition, if the node is a member of the specified MSID DLL multicast group (case 18 of Table 8-14.4), the node shall extract the LCDU and pass it to the DLL management. If the frame has been received with an unknown MSID (case 17 of Table 8-14.4), the frame shall be dropped and the transmitter shall be informed as specified in clause 8.17.

If the frame has been received with BRCTI = 1, the frame is processed as specified in clause 8.5.4.

Table 8-14.4 – Relaying of LLC frames (LCDU case)

Case	Type of relaying	Leaf/ Non- leaf	LCDU DA	Relaying actions	Example
1	Unicast frame intended to the node (BRCTI = 0; MCSTI = 0; Destination Node = DeviceID _{Node})	Leaf/ Non- leaf	In LAAT except node's MAC address	Send to DLL management	
2			In RAAT	Send to DLL management	Management message with proxy (ADM_NodeRegisterRequest.req)
3			Node's MAC address	Send to DLL management	Management message (or remote in-band message)

bidirectional service flow establishment, the domain master shall allocate bandwidth for the aggregated bandwidth requests of the forward flow and the reverse flow (see clause 8.3.7 for bidirectional transmission, and clause 8.6.2.3.8 for flow admission request). The DM may allocate the aggregated bandwidth in the same TXOP, or it may assign separate TXOPs for the forward flow and the reverse flow.

8.5.9.2 Bidirectional prioritized data connections

Bidirectional transmission can also be used in the context of prioritized data connections. In this case, the MPDU priority of the BACK frame shall be greater than or equal to the minimum allowed user priority in the TXOP/TS where the bidirectional transmission takes place.

8.6 Domain master node functional capabilities

A domain master-capable node is a node that, in addition to supporting all of the required capabilities of an endpoint node, is also able to assume the role of a domain master.

A domain master-capable node shall support all of the functions specified in the following clauses.

At any given time, only one node is allowed to act as a domain master for a domain. All other nodes within the domain are managed (coordinated) by this domain master. If a domain master fails, another node of the same domain, capable of operating as a domain master, should pick up the function of the domain master.

The domain master shall perform medium access using the same medium access rules as for endpoint (non-domain master) nodes and using the same MAP distributed to the endpoint nodes.

NOTE – It is not a requirement that every node be domain master capable.

The DM is responsible for communicating the latest node information via NodeInformationTLVs (see Table 8-16.1) supported by the nodes of the domain to all the nodes of the domain. The details of how this is done are described in clause 8.19.

8.6.1 Network admission

To join the network, all nodes shall first "register" with the domain master using the network admission protocol described in clause 8.6.1.1.

Normally, non registered nodes are able to receive successfully the MAP frames only if the MAP is transmitted in the default MAP format (MAP-D). Therefore, the domain master shall periodically transmit MAP-D messages in addition to MAP-A transmission to enable registration.

If a node does not have direct communication with the domain master (i.e., is hidden from the domain master), this node can still register and become part of the network using relayed admission as described in clause 8.6.1.2.

For registration, a unique registration identifier (REGID) is assigned to every node prior to its installation. REGID is intended exclusively for registration and may be communicated unencrypted. The value of the REGID shall be equal to the MAC address of the node.

The registering node shall identify the domain it wishes to join by comparing the domain name information in the received MAP-D frames as described in clause 8.8.3, with the target domain name(s) provided to the node by the user (to distinguish his/her network from neighbouring networks) or obtained during the first registration, if a device has no user interface.

The registering node shall first search for a MAP frame bearing a DNI field whose value coincides with the value of a target DNI in its information database. When a MAP frame meeting the target DNI is detected, the node shall verify the full value of the domain name in the domain information sub-field of the MAP (see clause 8.8.5.2) and use the DOD value of this MAP frame as the DOD for its registration messages described in clause 8.6.1.1.4 to indicate the particular domain it intends to join.

If the domain operates in non-secure mode, a node which successfully registered with the domain master can communicate with other nodes in the domain. If the domain operates in a secure mode, a registered node shall also authenticate itself, as described in clause 9.2. After authentication, the node becomes a member of the secure network and is in a position to establish communication with any other node in the domain/network.

In case a node has no interface for configuring a target domain(s), the manufacturer shall provide the node with a 6-byte registration code, which is also supplied explicitly to the user for configuring the domain master. If so configured, the domain master shall include this registration code in the auxiliary information field of MAP-Ds to allow the node to join the domain. Such a node provided with a registration code shall search for a MAP bearing this registration code in the auxiliary information field (see clause 8.8.5.9). After registration, the node shall memorize (optimally to a non-volatile memory) the domain name communicated in the MAP and the value of DNI, and use it as a target DNI for future registrations, and the domain master may remove the registration code from the auxiliary information field of MAP-Ds.

In case a node has an interface for configuring a target domain(s), it shall be configured with the target domain(s). The list may include more than one entry. If a node fails to register to one domain on the list, it shall try to register to another one on the list.

If no MAP frame meeting the target DNI is found, a node that is not capable of acting as a domain master may continue searching for the target DNI. A node that is capable of acting as a domain master shall establish a new domain, as described in clause 8.6.6.

The DEVICE_ID (and OriginatingNode) of the registering node shall be set to zero. After registration (for an unsecure domain) or authentication (for a secure domain) is complete, the DEVICE_ID shall be set to the value assigned by the domain master, as described in clause 8.7.1.1. A node shall not establish connections until it has been assigned a DEVICE_ID. From the first transmitted frame, the node shall comply with the transmission schedule posted in the MAP and shall meet all spectral compatibility requirements described in the PSD-related domain info field of the MAP (SM, PSD mask, Transmission power limit, etc. – See clause 8.8.5.5).

8.6.1.1 Network admission protocol

8.6.1.1.1 Registration into the domain

The protocol diagram of node registration into the domain is presented in Figure 8-25.

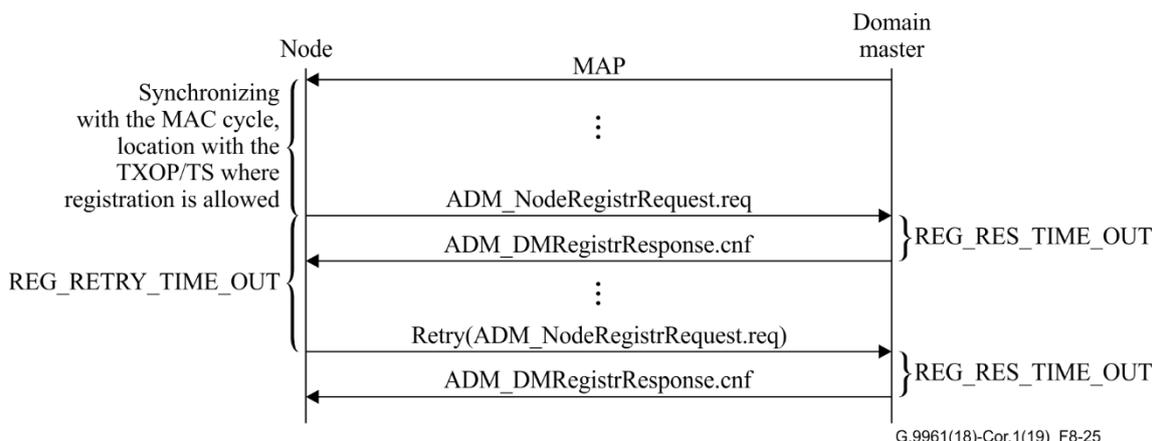


Figure 8-25 – Protocol diagram describing node registration

Prior to registration, the node shall synchronize with the network so that it can identify the MAC cycle, detect the MAP of the domain it intends to register, and locate the registration TS (for RCBTS see clause 8.3, for CBTS that permits registration, see clauses 8.8.4.1.5 and 8.8.4.2).

Table 8-16 – Format of the MMPL of the ADM_NodeRegistrRequest.req message

Field	Octet	Bits	Description
Attempt	0	[1:0]	00 ₂ for initial attempt, 01 ₂ , 10 ₂ , 11 ₂ for the second, third and fourth attempts
ProxyReg		[2]	Proxy registration flag; shall be set to one for registration through proxy (see clause 8.6.1.2) and zero otherwise.
PairingReq		[3]	Registration request under pairing procedure
Reserved		[7:4]	Reserved by ITU-T (Note 1)
ProxyDevID	1	[7:0]	Device ID of the Registration proxy (Note 2).
Parameters	2	[0]	Set to one if node is capable of operating as a domain master, zero otherwise
		[1]	Set to one if relaying is supported, zero otherwise
		[4:2]	Indicates the bandplan OFB that the node shall use after registration represented as described in clause 7.1.2.3.2.2.9/10.1.2.3.2.2.9 of [ITU-T G.9960] (BNDPL OFB ID/GRP_ID field).
		[5]	Set to one if the device is registering using registration code, zero otherwise.
		[7:6]	Reserved by ITU-T (Note 1)
T_AIFG	3	[7:0]	The value of T _{AIFG} supported by the node, represented as $n \times 1.28 \mu\text{s}$; the value of n is an unsigned integer in the range between 4 and 96. Valid values for each medium are specified in Table 8-14.
NumNodeInformationTLVs	4	[7:0]	Number of NodeInformationTLVs (N) included in this message.
Parameters	5	[0]	Set to one if node is capable of calculating routing tables, zero otherwise.
		[7:1]	Reserved by ITU-T (Note 1)
NodeInformationTLVs[0]	Variable	Variable See Table 8-16.1	Information related to the features implemented and capabilities of the registering node. It shall be coded as described in Table 8-16.1
...
NodeInformationTLVs[N-1]	Variable	See Table 8-16.1 Variable	Information related to the features implemented and capabilities of the registering node. It shall be coded as described in Table 8-16.1

NOTE 1 – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.

NOTE 2 – This field shall be set to zero by the transmitter and ignored by the receiver when the ProxyReg field is set to zero.

Table 8-16.1 – Format of NodeInformationTLVs[i] and DmInformationTLVs[i]

Field	Octet	Bits	Description
InformationType	Variable0	[7:0]	Type of information field 0 – ITU-T information (see Table 8-16.2). 1 – Reserved for HGF information All other values of this field are reserved by ITU-T.
InformationLength	Variable1	[7:0]	Length in bytes of information value field.
InformationValue	Variable	Variable	Value of information field.

Table 8-16.2 – Format of the InformationValue field when the InformationType field is set to ITU-T information in Table 8-16.1

Field	Octet	Bits	Description
ITUFIELDSContents	0	[7:0]	Bits [7:0] represent the information related to the presence of the different components of the ITU-T information field. Bit 0: This bit shall be set to ZERO. Bit 1: If set to one, Capabilities field is present. If set to zero, Capabilities field is not present. Other bits are reserved by ITU-T and shall be set to 0 (Note)
Capabilities	Variable	See Table 8-16.4	If present, this field contains the information on specific capabilities that the node implements. The format of this field is described in Table 8-16.4
NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

Table 8-16.3 – Placeholder table

(This table has been intentionally left blank.)

Table 8-16.4 – Format of the Capabilities field

Field	Octet	Bits	Description
NumCapabilities	0	[7:0]	Number of Capabilities (M) indicated in Capabilities field
CapabilityType[0]	1	[7:0]	Capability Type. The format of this field is described in Table 8-16.5
CapabilityLength[0]	2	[7:0]	Length of the capability type indicated, represented as an 8-bit unsigned integer (see Table 8-16.5).
CapabilityValue[0]	Variable	Variable	Value of the capability type indicated.
...			
CapabilityType[M-1]	Variable	[7:0]	Capability Type. The format of this field is described in Table 8-16.5

Table 8-16.5 – List of Capabilities

Capability Type	Capability name	Capability Length value	Capability Value field
07 ₁₆	Power Saving support	1	Bit map indicating the capability of the node to enter each of the power saving modes (see clause 8.21). Bit 0: Support for L0 mode. This bit shall always be set to one. Bit 1: If one, it indicates support for L1 mode. Bit 2: If one, it indicates support for L2 mode. Bit 3: If one, it indicates support for L3 mode. Bit 4: If one, it indicates support for L4 mode. Bits 5 to 7: Reserved by ITU-T (Note 1)
08 ₁₆	MMPL Compression support	1	See Table 8-16.10
09 ₁₆	IfExtendBits for NUM_VALID_DUR	1	If set to 01 ₁₆ , it indicates the capability of the node to use extended bits for NUM_VALID_DUR (see clause 8.11.7.3 of [ITU-T G.9961]), 00 ₁₆ otherwise. All other values are Reserved by ITU-T
0A ₁₆ -FF ₁₆	Reserved by ITU-T		Reserved by ITU-T
<u>A0₁₆</u>	<u>Profile 2 OFBs Info</u>	<u>4</u>	<u>See Table 8-16.6.1</u>
<u>A1₁₆</u>	<u>TXOP timing Info</u>	<u>1</u>	<u>See Table 8-16.6.2</u>
<u>A2₁₆</u>	<u>Receiver Timing Capabilities</u>	<u>1</u>	<u>See Table 8-16.6.3</u>
<u>A3₁₆</u>	<u>Payload encoding for Profile 2 OFBs Capabilities</u>	<u>1</u>	<u>See Table 8-16.11</u>
<u>A4₁₆-FF₁₆</u>	<u>Reserved by ITU-T</u>		<u>Reserved by ITU-T</u>

NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.

Table 8-16.6.3 – Receiver timings Info Capability Value field

Field	Octet	Bits	Description
<u>T_{AIFG-DR}</u>	<u>0 and 1</u>	<u>[13:0]</u>	Minimum value accepted by the node as default <u>T_{AIFG-DR}</u> , represented as $n \times 1.28 \mu\text{s}$; the value of <u>n</u> is an unsigned integer. Valid values of <u>T_{AIFG-DR}</u> for each medium are the same as the <u>T_{AIFG}</u> values specified in Table 8-14.
<u>Reserved</u>		<u>[15:14]</u>	<u>Reserved by ITU-T (Note)</u>
<u>MIN_SYM_VAR_AIFG</u>	<u>2</u>	<u>[2:0]</u>	The default minimum number of symbols following the header required in a frame to use receiver specific <u>T_{AIFG}</u> , instead of a default <u>T_{AIFG}</u> , as the AIFG gap between the frame and the following immediate acknowledgment for this node as receiver. Valid values for this field are between 0 and <u>MIN_SYM_VAR_AIFG_DEFAULT</u> (valid values of for each medium are specified in Table 8-14).
<u>Reserved</u>		<u>[7:3]</u>	<u>Reserved by ITU-T (Note)</u>

NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver

Table 8-16.7 – PSD Information Capability Value field

Field	Octet	Bits	Description
APSD_MAX support	0	[0]	If set to one, it indicates that the node is reporting APSD_MAX. Zero otherwise
APSD_MAX extension support		[1]	If set to one, the APSD_MAX extension bit is used by the node to convey the APSD_MAX with a granularity of 1 dB. Zero otherwise.
APSD_MAX		[7:2]	If APSD_MAX support is set to one, this field indicates the highest value of the PSD that the node is capable of using. The valid range of values is from -50 dBm/Hz (value 000000 ₂) to -100 dBm/Hz (value 110010 ₂) in steps of 1 dB. This field shall be set to 0 if APSD_MAX support field is set to zero.

Table 8-16.8 – Run-length compression support

Field	Octet	Bits	Description
BAT compression	0	[0]	If set to one, it indicates the capability of the node to compress and decompress BAT fields of CE_ParamUpdate.req messages making use of the Run-Length compression mechanism described in clause 8-23, zero otherwise.
BMAT compression	0	[1]	Reserved for [ITU-T G.9963]
Reserved	0	[7:2]	Reserved by ITU-T (Note)

NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.

Table 8-16.9 – Secure admission methods

Field	Octet	Bits	Description
GeneralMethods	0	[7:0]	Bitmap representing the secure admission methods that are supported by the node. Bit 0 represents the LSB. When the bit is set to one, the corresponding methods are accepted as a valid method. When the bit is set to zero, the methods is not accepted as a valid method. Bit 0: Admission through passphrase (see clause 6.3 of [ITU-T G.9978]) Bit 1: Admission through auto-pairing (see clause 6.4 of [ITU-T G.9978]) Bits 2-7: Reserved by ITU-T (Note)
PairingMethods	1	[7:0]	Bitmap representing the secure admission methods that are supported by the node. Bit 0 represents the LSB. When the bit is set to one, the corresponding methods are accepted as a valid method. When the bit is set to zero, the methods is not accepted as a valid method Bit 0: Admission through generic pairing – Single-node pairing mode (see clause 6.2.2.1 of [ITU-T G.9978]) Bit 1: Admission through generic pairing – Multi-node pairing mode (see clause 6.2.2.2 of [ITU-T G.9978]) Bit 2-7: Reserved by ITU-T (Note)
MACMethods	2	[7:0]	Bitmap representing the secure admission methods that are supported by the node. Bit 0 represents the LSB. When the bit is set to one, the corresponding methods are accepted as a valid method. When the bit is set to zero, the methods is not accepted as a valid method Bit 0: Admission through MAC authorization (see clause 6.1 of [ITU-T G.9978]) Bit 1-7: Reserved by ITU-T (Note)
NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

Table 8-16.10 – MMPL compression support capability value field

Field	Octet	Bits	Description
MMPL Compression Coding	0	[7:0]	This bitmap indicates the compression coding supported by the node Bit 0, if set to one, indicates that the node support Huffman code compression for MMPL Bits 1 to 7 are reserved by ITU-T (Note)
NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

Table 8-16.11 – Profile 2 OFBs capability value field

<u>Field</u>	<u>Octet</u>	<u>Bits</u>	<u>Description</u>
<u>Payload encoding</u>	<u>0</u>	<u>[7:0]</u>	<p><u>Bitmap representing the capability of a node's receiver to support different payload encodings (see clause 7.1.3.3/10.1.3.3 of [ITU-T G.9960]). A bit set to one represents that the node's receiver supports that payload encoding. A bit set to zero represents that the node's receiver does not support that payload encoding.</u></p> <p><u>Bit 0: Full mode payload encoding</u></p> <p><u>Bit 1: MLC mode payload encoding</u></p> <p><u>Bits 2 to 7: Reserved by ITU-T</u></p>

8.6.1.1.4.2 Registration response message (ADM_DmRegistrResponse.cnf)

The ADM_DmRegistrResponse.cnf message is a unicast management message sent by the domain master to the registering node (directly or via a proxy), and is intended to be used for registration response only. The format of the MMPL of the ADM_DmRegistrResponse.cnf message shall be as shown in Table 8-17.

Table 8-17 – Format of the MMPL of the ADM_DmRegistrResponse.cnf message

Field	Octet	Bits	Description
REGID	0 to 5	[47:0]	REGID of the node that requested the admission in standard format of a MAC address
DEVICE_ID	6	[7:0]	An ID assigned to the node by the domain master; shall be set to 00 ₁₆ in case registration is denied
Registration flag	7	[0]	Set to one for successful registration, set to zero for registration denied
<u>BandplanOFB_ID</u>		[3:1]	<u>Bandplan-OFB</u> used by new registering node represented as described in clause 7.1.2.3.2.2.9/10.1.2.3.2.2.9 of [ITU-T G.9960] (<u>BNDPLOFB_ID</u> /GRP_ID field)
Rejection code		[6:4]	As described in Table 8-15
Security mode		[7]	Set to zero for an unsecure domain, set to one for a secure domain
Attempt	8	[1:0]	00 ₂ for response on the initial attempt, 01 ₂ , 10 ₂ , 11 ₂ for the response on the second, third and fourth attempts, respectively
Extended rejection code		[4:2]	As described in Table 8-15.1
SC configuration		[5]	Set to one if SC is assigned to a specific node and can only be changed by user/operator control, zero otherwise
Reserved		[7:6]	Reserved by ITU-T (Note)
Security	9 to 15	[55:0]	<p>If Security mode is set to zero, this field shall be set to zero.</p> <p>If Security mode is set to one, the eight LSBs of this field represent the DEVICE_ID of the security controller, and the 48 MSBs</p>

Table 8-17 – Format of the MMPL of the ADM_DmRegistrResponse.cnf message

Field	Octet	Bits	Description
			represent the REGID of the security controller
NumTLVs	16	[7:0]	Number of TLVs (<i>n</i>)
TLV[<i>n</i>]	Variable	[7:0] See Table 8-17.1	A list of <i>n</i> TLVs. Each TLV shall be formatted as described in Table 8-17.1

NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.

Table 8-17.1 – Format of the TLV field

Field	Octet	Bits	Description
TLV type	0	[7:0]	Indicates the TLV type (See Table 8-17.2)
TLV length	1	[7:0]	Indicates the length of the following TLV value field in octets (<i>L</i>). The valid range is from 0 to 255
TLV value	2 to <i>L</i> +1	[7:0]	This field shall be formatted as indicated in Table 8-17.2, for the TLV type. This field exists if and only if <i>L</i> > 0

Table 8-17.2 – TLV Type assignments

TLV Type Value	Description	Clause	TLV value format
1	Non-default CW parameters	8.8.5.11	Table 8-85.3
2	Backup SC information	8.6.1.1.4.2	Table 8-17.3
3	Pairing response information	6.2.2.1.2 of [ITU-T G.9978]	Table 8-17.4
0 and 4-255	Reserved by ITU-T for future assignments		

Table 8-17.3 – Backup SC information

Field	Octet	Bits	Description
BackupSC_DeviceID	1	[7:0]	The 8-bit DEVICE_ID of a node that is assigned as the backup SC. If the backup SC node has not registered to the domain, this field shall be set to 0.
BackupSC_RegID	2 to 7	[47:0]	The 48-bit REGID of a node that is assigned as the backup SC. If the backup SC node has not registered to the domain, this field shall be set to 0.

Table 8-17.4 – Pairing response information

Field	Octet	Bits	Description
Password	1 to 12	[95:0]	This field contains the password the domain master delivers to a pairing registering node in case of positive response.

8.6.1.1.4.3 Resignation request message (ADM_NodeResignRequest.req)

The ADM_NodeResignRequest.req message is a unicast management message sent by a node to the domain master, and is intended to be used for resignation request only. The format of the MMPL of the ADM_NodeResignRequest.req message shall be as shown in Table 8-18.

Table 8-18 – Format of the MMPL of the ADM_NodeResignRequest.req message

Field	Octet	Bits	Description
Attempt	0	[1:0]	00 ₂ for initial attempt, 01 ₂ , 10 ₂ , 11 ₂ for the second, third and fourth attempts
PowerModeResignation		[2]	One if the node is requesting to resign because it enters into idle power saving mode (L4); zero otherwise.
Reserved		[7:3]	Reserved by ITU-T (Note)

NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.

8.6.1.1.4.4 Resignation confirmation message (ADM_DmResign.cnf)

The ADM_DmResign.cnf message is a unicast management message sent by the domain master to a node that requested resignation, and is intended to confirm or reject the node's resignation request. The format of the MMPL of the ADM_DmResign.cnf message shall be as shown in Table 8-19.

Table 8-19 – Format of the MMPL of the ADM_DmResign.cnf message

Field	Octet	Bits	Description
Node ID	0	[7:0]	DEVICE_ID of the resigned node
RegID	1 to 6	[47:0]	MAC address of the resigned node
Reason	7	[1:0]	00 ₂ if the request was accepted 01 ₂ if the request was rejected due to topology issues 10 ₂ , 11 ₂ – Reserved by ITU-T
Reserved		[7:2]	Reserved by ITU-T (Note)

NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.

8.6.1.1.4.5 Forced resignation request message (ADM_DmForcedResign.req)

The ADM_DmForcedResign.req message is a unicast management message sent by the domain master to a node, and is intended to force the node to resign from the domain. The format of the MMPL of the ADM_DmForcedResign.req message shall be as shown in Table 8-20.

Table 8-20 – Format of the MMPL of the ADM_DmForcedResign.req message

Field	Octet	Bits	Description
ReasonCode	0	[1:0]	Reason why the node was resigned • 00 ₂ =Reserved • 01 ₂ = SC in the domain has failed • 10 ₂ -11 ₂ =Reserved
Reserved		[7:2]	Reserved by ITU-T (Note)

NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.

8.6.1.1.4.6 Re-registration request message (ADM_NodeReRegistrRequest.req)

The ADM_NodeReRegistrRequest.req message is a unicast management message sent by a node to the domain master, and is intended to be used for periodic re-registration requests.

The MMPL of the ADM_NodeReRegistrRequest.req message shall be empty.

8.6.1.1.4.7 Re-registration response message (ADM_DmReRegistrResponse.cnf)

The ADM_DmReRegistrResponse.cnf message is a unicast management message sent by the domain master to the node, and is intended to be used for periodic re-registration response. The MMPL of the ADM_DmReRegistrResponse.cnf message shall be empty.

8.6.1.1.4.8 Re-registration initiation message (ADM_DmReRegistrInitiate.ind)

The ADM_DmReRegistrInitiate.ind message is a unicast management message sent by the domain master to a node, and is intended to force the node to initiate re-registration process immediately. The format of the MMPL of the ADM_DmReRegistrInitiate.ind message shall be as shown in Table 8-20.1.

Table 8-20.1 – Format of the MMPL of the ADM_DmReRegistrInitiate.ind message

Field	Octet	Bits	Description
Attempt	0	[1:0]	00 ₂ for the initial attempt 01 ₂ for the second attempt 10 ₂ , 11 ₂ – Reserved by ITU-T
Reserved		[7:2]	Reserved by ITU-T (Note)
NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

8.6.1.1.4.9 New node registered indication to SC (ADM_NewNodeRegistered.ind)

The ADM_NewNodeRegistered.ind message is a unicast management message sent by the DM to the SC, and is intended to convey the identity of the new node that has been registered by the DM. The format of the MMPL of the ADM_NewNodeRegistered.ind message shall be as shown in Table 8-20.2.

Table 8-20.2 – Format of the MMPL of the ADM_NewNodeRegistered.ind message

Field	Octet	Bits	Description
Device ID	0	[7:0]	DEVICE_ID of the newly registered node

8.6.1.2 Admission via proxy

Provided that proxy nodes are available, nodes that are hidden from the domain master shall register into the domain via one of the proxy nodes using the procedure described in this clause.

The protocol diagram of node registration into the domain via a proxy node is presented in Figure 8-27.

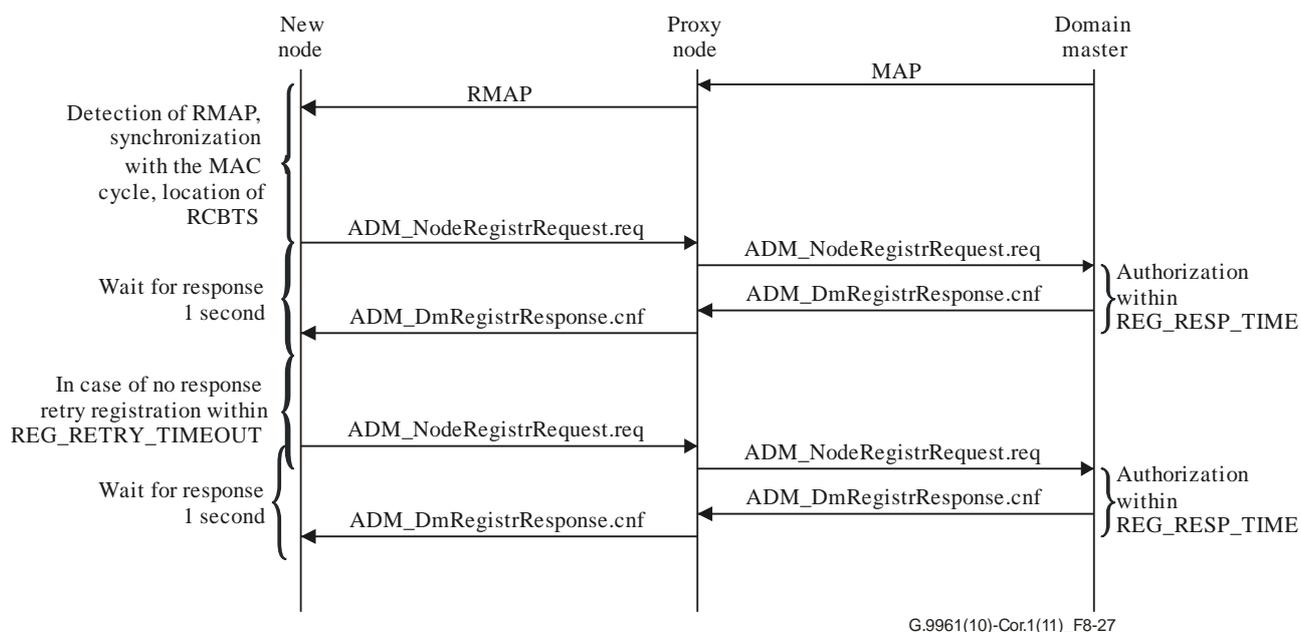


Figure 8-27 – Protocol diagram describing node registration via proxy

Prior to registration, the node shall detect the RMAP-D of the domain it intends to register to. If it also detects RMAP-A, it should use the information from RMAP-D to decode RMAP-A, and use the node that transmits the RMAP-A as the registration proxy. The relayed RMAP frame shall indicate the number of hops the RMAP relay node is from the domain master (see NUM_HOPS in clause 7.1.2.3.2.1.12/10.1.2.3.2.1.12 of [ITU-T G.9960]). If RMAP-A is not detected, the node shall use the node transmitting RMAP-D as the registration proxy. The registering node shall synchronize with the MAC cycle, and locate the registration TS (RCBTS or CBTS that permits registration, see clauses 3 and 8.8.4.1.5, respectively) described in the MAP. To start the registration via proxy, the node shall send a registration request message (ADM_NodeRegistrRequest.req, see clause 8.6.1.1.4.1) to the domain master via the registration proxy.

After receiving the ADM_NodeRegistrRequest.req from the node, the proxy shall relay it towards the domain master. To facilitate relaying of ADM_NodeRegistrRequest.req, the registering node shall use the addressing scheme shown in Table 8-20.3.

Table 8-20.3 – Addressing fields of the ADM_NodeRegistrRequest.req from registering node to proxy node

Field	Value
DA of the LCDU carrying the message	REGID of the domain master
SA of the LCDU carrying the message	REGID of the registering node
OriginatingNode of the LLC frame carrying the LCDU	0
DestinationNode of the LLC frame carrying the LCDU	DEVICE_ID of the proxy node
SID of the PHY frame carrying the message	0
DID of the PHY frame carrying the message	DEVICE_ID of the proxy node

In addition, the ProxyReg flag in ADM_NodeRegistrRequest.req shall be set to one, and the field ProxyDevID shall contain the DEVICE_ID of the proxy node, obtained from the SID field of the PFH for the PHY frame carrying the RMAP. The registering node obtains the REGID of the domain master from the LCDU that conveys the RMAP message, as described in clause 8.8.

The proxy shall relay the received ADM_NodeRegistrRequest.req message by recreating a new ADM_NodeRegistrRequest.req with the same contents as the received one and shall send it to the DM using the addressing scheme shown in the Table 8-20.4.

Table 8-20.4 – Addressing fields of the ADM_NodeRegistrRequest.req from proxy to the DM

Field	Value
DA of the LCDU carrying the message	REGID of the domain master
SA of the LCDU carrying the message	REGID of the registering node
OriginatingNode of the LLC frame carrying the LCDU	0
DestinationNode of the LLC frame carrying the LCDU	DEVICE_ID of the DM
SID of the PHY frame carrying the message	DEVICE_ID of the proxy node
DID of the PHY frame carrying the message	DEVICE_ID of the DM or the next relay node towards the DM in case where the proxy has not a direct link with the DM

The registering hidden node shall send ADM_NodeRegistrRequest.req only during the CBTS for which registration is allowed, using medium access rules for registration described in clause 8.3.3.4.8. The domain master shall process the ADM_NodeRegistrRequest.req message and reply to the registration proxy with a registration response message (ADM_DmRegistrResponse.cnf, see clause 8.6.1.1.4.2) using the addressing scheme shown in Table 8-20.5. If the node is admitted and the domain is a secure domain, the DM shall also send a ADM_NewNodeRegistered.ind message (see clause 8.6.1.1.4.9) to the SC.

Table 8-20.5 – Addressing fields of the ADM_DmRegistrResponse.cnf from DM to the proxy node

Field	Value
DA of the LCDU carrying the message	REGID of the proxy node
SA of the LCDU carrying the message	REGID of the DM
OriginatingNode of the LLC frame carrying the LCDU	DEVICE_ID of the DM
DestinationNode of the LLC frame carrying the LCDU	DEVICE_ID of the proxy node
SID of the PHY frame carrying the message	DEVICE_ID of the DM
DID of the PHY frame carrying the message	DEVICE_ID of the proxy node or the next relay node towards the proxy in case where the DM has not direct link with the proxy

The registration proxy node shall then unicast the received ADM_DmRegistrResponse.cnf message to the new node using the addressing scheme shown in Table 8-20.6.

**Table 8-20.6 – Addressing fields of the ADM_DmRegistrResponse.cnf
from the proxy node to the registering node**

Field	Value
DA of the LCDU carrying the message	REGID of the registering node
SA of the LCDU carrying the message	REGID of the proxy node
OriginatingNode of the LLC frame carrying the LCDU	DEVICE_ID of the proxy node
DestinationNode of the LLC frame carrying the LCDU	0
SID of the PHY frame carrying the message	DEVICE_ID of the proxy node
DID of the PHY frame carrying the message	0

The behaviour specified in clause 8.6.1.1.1 regarding retransmission of registration messages and rejection codes shall also apply to nodes registering via proxy.

After sending the ADM_DmRegistrResponse.cnf message with the registration flag set to one (successful registration), the domain master shall designate the registration proxy node as an RMAP-A relay. After getting the topology update messages from the newly registered node, the domain master may change the assignment of the RMAP-A relay for this node as described in clause 8.5.6.2.

For re-registration and resignation, the node shall follow the procedures described in clauses 8.6.1.1.2 and 8.6.1.1.3, respectively.

8.6.1.3 Domain maintenance after node admission

When a new node is admitted into the domain, the DM shall update the related domain wide information in order to take into account backward compatibility issues.

If any of the information included in the additional domain information auxiliary information sub-field of the MAP needs to be updated based on the node-information of the registering node (see Table 8-16.1), the DM shall send a MAP-A with an updated additional domain information auxiliary information sub-field. The domain master shall use the validity counter-based update (AUX_VALID = 3 and ModificationFlag = 1):

- Whenever a registering node indicates in its node-information (see Table 8-16.1) during registration that it does not support a DM-generated Domain-specific constellation scrambler initialization (see clause 7.1.4.5.3.2.1/[10.1.4.5.3.2.1](#) of [\[ITU-T G.9960\]](#)) the DM shall update the DM_Defined_Seed field of the additional domain information auxiliary information sub-field to the first allowed seed for that DOD as indicated in Table 7-75/[10-75](#) of [\[ITU-T G.9960\]](#) in the case of powerline and to the default value of the seeds for the rest of the mediums.
- Whenever a registering node indicates in its node-information during registration that it does not support a DM-defined initial seed (see clause 7.1.4.2.6/[10.1.4.2.6](#) of [\[ITU-T G.9960\]](#)) the DM shall update the UnloadedSubcarrierInitialSeed field of the additional domain information auxiliary information sub-field to the default value.
- Whenever a new node registers into the domain with OFB information (OFB_ID) that has an impact on MAP-A transmission (e.g., the new node only supports Profile 1 OFBs in a context where the MAP-A is sent in Profile 2 OFBs), the DM shall send MAP-A transmissions using an OFB that all the nodes of the domain are able to decode.
- Whenever a new node registers into the domain with OFB information (OFB_ID) that has an impact on scheduling (e.g., the new node only supports Profile 1 OFBs in a context where

the TXOPs are only assigned to Profile 2 OFBs), the DM should adapt the scheduling to the new situation.

8.6.2 Bandwidth management

The domain master shall be capable of allocating (scheduling) TXOPs and TSs to different nodes, user priorities and service flows. These allocations should be such that nodes transmitting within the assigned TXOPs and TSs should meet priority constraints for priority traffic and QoS bandwidth, latency and jitter constraints specified in the TSpec for the established service flows, even in the presence of neighbouring domains operating in the same medium.

The domain master shall be responsible for managing available bandwidth. It should try to satisfy bandwidth requests from the different nodes, balancing the demands for bandwidth defined in the traffic specifications of the established flows with the total amount of available bandwidth.

The domain master may reserve periods of time for use by other domains by scheduling silent TXOPs in its own domain.

The way in which the domain master manages the available bandwidth and the particular schedules it generates are beyond the scope of this Recommendation.

The output of the scheduling process is the MAP (see clause 8.8). The MAP is transmitted each MAC cycle and defines the TXOPs and TSs allocated to node(s), user priorities and service flows in the next MAC cycle(s).

The domain master should maintain state information concerning the allocation of medium resources in the domain and shall control the admission of new service flows and the allocation of medium resources.

Admission control of new service flows should guarantee that the minimum QoS requirements for existing services are not violated.

The domain master shall service requests to add and remove service flows and requests to change service flow characteristics as described in the following sub-clauses.

If a request is made to add a new service flow and the QoS requirements specified in the TSpec cannot be met, the domain master shall deny admission of the new service flow and a denial of service status shall be returned to the requestor.

Note that denial of service flow establishment means that no QoS guarantees can be given to a particular service flow. In this case, medium access may still be performed on a priority-basis within STXOPs.

Changes in line conditions may be detected by channel estimation. If the line conditions change and the transmitting node is forced to use a lower bit loading for an admitted service flow, it shall notify the domain master. The domain master should then reallocate medium bandwidth reservations to account for the change.

If there are not enough bandwidth resources to support the low bit loading, the domain master may decide to reduce the allocations of one of the current active flows by updating its TSpec attributes or the domain master may end one flow or more in order to release the needed bandwidth resources.

NOTE – The decision on which flow TSpec to change, or which flow to end is a domain master scheduling decision and is out of scope of this Recommendation.

If the service flow data rate is changed at the A-interface, the originating node shall initiate a new admission procedure with the domain master to update the TSpec attributes of the service flow according to the new service flow characteristics.

8.6.2.1 Description of TSpec parameters

Terms related to traffic specifications and quality of service are described in this clause.

Traffic specification (TSpec) – Describes the set of parameters, characteristics, and expected quality of service related to a particular data flow. The TSpec may be provided to the node by its associated client before the data flow is established. The TSpec may include any of the following QoS attributes: traffic priority, maximum information rate, maximum traffic burst, committed information rate, tolerated jitter and maximum latency, unsolicited grant interval, unsolicited polling interval, APDU size.

Traffic priority – Given two flows with identical TSpec parameters except for priority, the higher priority flow should be given lower delay. For otherwise non-identical flows, the priority parameter should not take precedence over any conflicting flow QoS parameter. The domain master shall use this parameter when determining precedence in CFTXOP and CFTS allocations. When available bandwidth is not sufficient for all the service flows, service flows with higher priority shall be assigned resources at the expense of service flows with lower priority.

Maximum information rate (MIR) – Defines maximum information rate of the flow. The rate is expressed in bits per second and pertains to the APDU at the input to the APC. Hence, this parameter does not include ITU-T G.9960/1 overhead. If this parameter is omitted or set to zero, then there is no explicitly mandated maximum rate. This field specifies only a bound, not a guarantee that the rate is available. APDUs deemed to exceed the maximum information rate may be, for instance, delayed or dropped.

Maximum traffic burst – Describes the maximum continuous burst in kbytes that the node should accommodate for the flow, assuming the flow is not currently using any of its available resources. This parameter is needed because the physical speed of the A-interface might be greater than the maximum information rate parameter for a flow. Maximum traffic burst set to zero shall mean no maximum traffic burst reservation requirement.

Committed information rate (CIR) – Specifies the minimum rate reserved for this flow. The rate is expressed in bits per second and specifies the minimum amount of data to be transported on behalf of the flow. The domain master and the relevant nodes shall be able to transport traffic at its committed information rate. If the actual information rate is less than the committed information rate for a flow, the domain master may reallocate the excess reserved bandwidth for other purposes. The data for this parameter are measured at the input of the APC. If this parameter is omitted, then no bandwidth need be reserved for the flow.

Tolerated jitter – This parameter defines the maximum delay variation (jitter) in ms for the flow after the domain master has allocated a specific TXOP therefor. The jitter is computed as the difference between the maximum real measured delays to minimum real measured delays.

Maximum latency – The value of this parameter specifies the maximum interval in ms between the arrival time of an ADP at the A-interface of the originating node and the departure time of the ADP at the A-interface of the endpoint node. If defined, this parameter represents a flow commitment (or admission criteria) at the domain master and the involved nodes and shall be guaranteed by the domain master and the nodes. The domain master and the involved nodes do not have to meet this flow commitment for flows that exceed their committed information rate.

Grant interval – The value of this parameter specifies the nominal interval in ms between successive CFTXOP allocations for this flow. The target schedule for enforcing this parameter is defined by a reference time t_0 , with the desired transmission time $t_i = t_0 + i \times \text{interval}$. The actual CFTXOP time, t_i' shall be in the range $t_i - \text{jitter}/2 \leq t_i' \leq t_i + \text{jitter}/2$, where interval is the grant interval value specified, and jitter is the tolerated jitter. The size of the CFTXOP is specified by the parameter APDU size. When grant interval is specified then APDU size must be specified as well.

Polling interval – This parameter is used when the flow traffic characterized by a fixed packet size in a fixed interval when there is not always a packet to transmit, for example, VOIP with silence suppression. The value of this parameter, in ms, specifies the maximum nominal interval between

successive transmissions opportunities for this flow. The ideal schedule for enforcing this parameter is defined by a reference time t_0 , with the desired polling time $t_i = t_{(i-1)} + \text{interval}$.

APDU size – The value of this parameter specifies the length of the APDU in bytes. This parameter is used only if the flow consists of fixed-length APDUs.

8.6.2.2 Lifecycle of a data flow

A data flow is created, exists for some time, and is then terminated. Figure 8-28 shows the lifecycle of a flow.

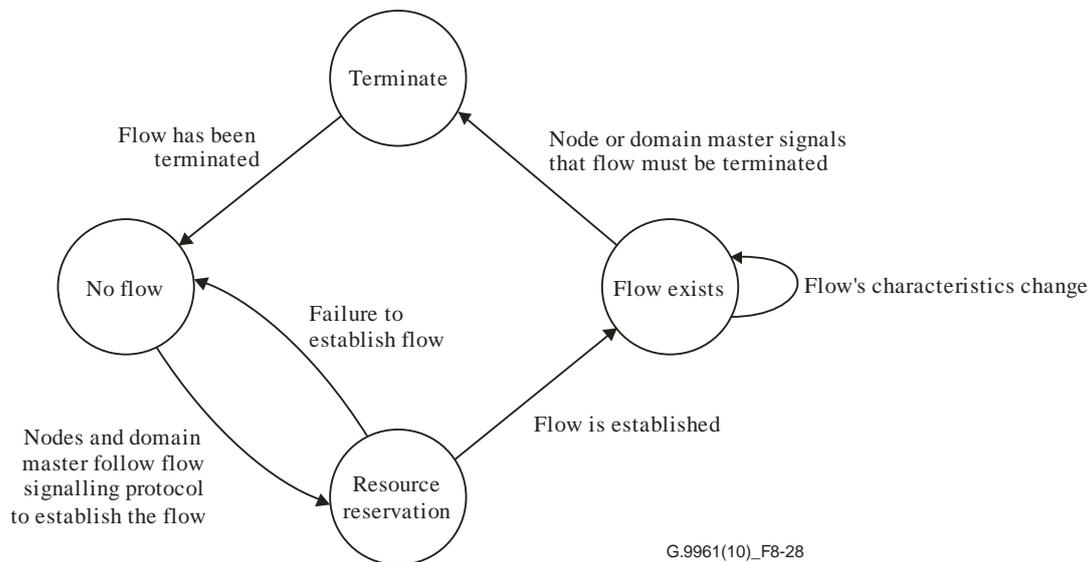


Figure 8-28 – Lifecycle of a flow

8.6.2.2.1 Flow establishment

This clause defines the procedures used to establish data flow. This protocol is supported by management messages and by fields contained in PHY-frame headers. Nodes follow this protocol when establishing QoS parameters for a flow, and this protocol is also followed between nodes and the domain master for flow management purposes.

The protocol defines a series of messages used for communication between the application entity, nodes, and the domain master.

A flow shall be established following these steps:

- When an application entity residing on the client associated with a node needs to establish a flow with a peer application entity, it shall signal that it needs to create a flow between the originating node and a peer application entity residing on a designated application entity specified MAC Address. This signal can be explicit (i.e., conveyed as an in-band management message, CL_EstablishFlow.req, across the A-interface from the AE) or implicit (the application simply starts sending data);
- If the signal is explicit, the application entity shall send the CL_EstablishFlow.req message that shall contain the required traffic specification (TSpec). The specified TSpec shall include at least one of the TSpec parameters;
- If the signal is implicit, the node may use the automatic traffic classification service. This service allows the node to generate a traffic specification that will describe the flow's characteristics;
- If the signal is explicit and the originating node does not have enough resources, it shall reply to the AE with the CL_EstablishFlow.cnf message with a failure code;

- If the originating node can support the flow, it shall determine the destination DEVICE_ID of the endpoint that the specified application entity is above its A-interface, then it shall allocate a FLOW_ID that uniquely represents the associated connection using the tuple (SID, FLOW_ID) where SID is the DEVICE_ID assigned to the originating node and FLOW_ID is defined as described in clause 8.7.2 (the FLOW_ID uniquely defines the DID of the connection);
- If the endpoint node is hidden from the originating node, the flow establishment shall be as specified in clause 8.6.2.2.2. Otherwise, it shall continue with the next bullet;
- The originating node shall send the FL_AdmitFlow.req message to the domain master to establish a traffic contract with the domain master. The domain master shall then assess whether the flow can be established given its TSpec (in case of bidirectional flow, two TSpecs, one for the forward direction and one for the reverse direction) and the available bandwidth;
- If the domain master decides to reject the flow admission request, it shall notify the originating node by replying with the FL_AdmitFlow.cnf message with a failure code, and the originating node shall release the allocated FLOW_ID and its allocated resources. If the flow establishment request was received explicitly from the AE, the originating node shall reply to the AE with the CL_EstablishFlow.cnf message indicating that the flow establishment request was rejected. By this the flow establishment procedure is ended;
- If the domain master decides to confirm the flow admission request, it shall reply with an FL_AdmitFlow.cnf with a success code;
- Upon receiving the FL_AdmitFlow.cnf message with a success code, the originating node shall send the FL_OriginateFlow.req message to the designated endpoint node to establish the flow;
- If the endpoint node is unable to support the new flow, it shall notify the originating node by sending the FL_OriginateFlow.cnf message with a failure code. The originating node shall notify the domain master about the flow establishment failure by sending the FL_AdmitFlow.ind message to it and release the allocated FLOW_ID. The domain master shall send the FL_AdmitFlow.rsp message to the originating node and then release the reserved bandwidth for the allocated FLOW_ID:
 - Otherwise, if the endpoint node is able to support the new flow, it shall notify the originating node by sending the FL_OriginateFlow.cnf message with the success code. In case of bidirectional flow, the FL_AdmitFlow.ind message shall contain the FLOW_ID for reverse flow. The originating node shall notify the domain master that the flow was established successfully by sending the FL_AdmitFlow.ind message with a success code. In case of bidirectional flow the FL_AdmitFlow.ind message shall contain the FLOW_ID for reverse flow in addition to the FLOW_ID for the forward flow. The Domain Master shall acknowledge the FL_AdmitFlow.ind by sending the FL_AdmitFlow.rsp message to the originating node.
- If the request from the AE was explicit, the originating node shall send a CL_EstablishFlow.cnf message to the AE indicating whether the flow was established successfully or not. In case that the flow was established successfully, the CL_EstablishFlow.cnf shall contain the established FLOW_ID. In case of bidirectional flow it shall include the FLOW_ID for reverse flow as well;
- Once the flow has been established, the corresponding data connection shall be established following the procedure described in clause 8.12.2 or clause 8.12.1.

The originating node and the endpoint node (in case of bidirectional flow) may begin transmitting via the established connection using contention-based mechanisms before the domain master allocates resources in the MAP for this flow.

Figure 8-29 describes a successful flow establishment (explicit signal from the AE).

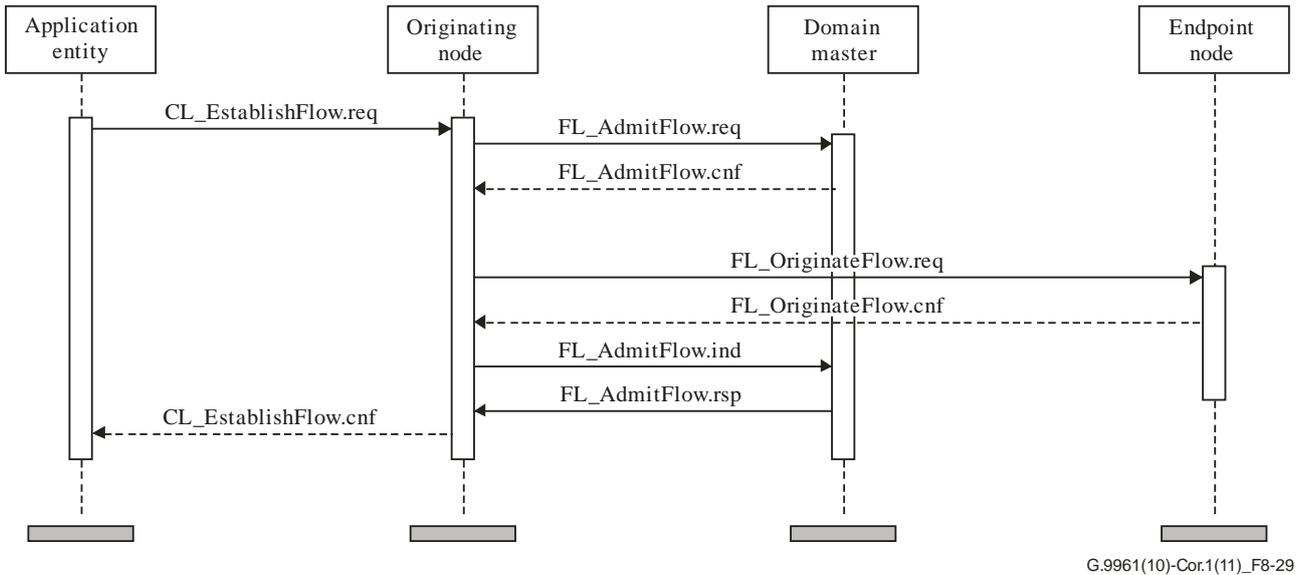


Figure 8-29 – Successful flow establishment

Figure 8-30 describes a failure in a flow establishment request by explicit signal from the AE that is rejected due to rejection by the originating node rejection (explicit signal from AE).

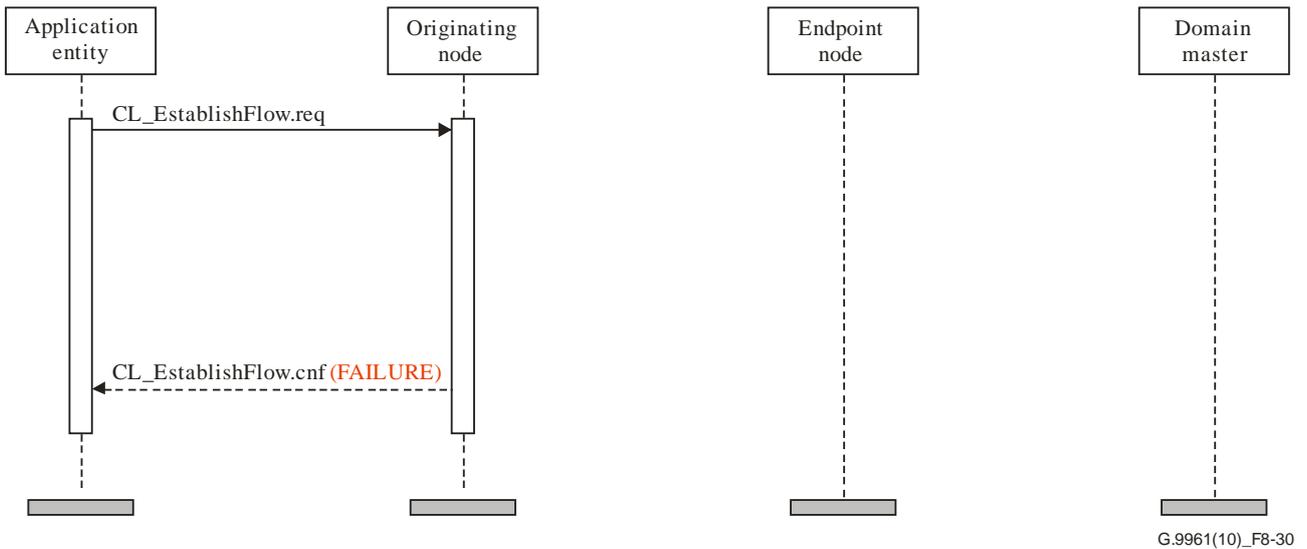


Figure 8-30 – Failure in flow establishment due to rejection by the originating node rejection

Figure 8-31 describes a flow establishment request by explicit signal from AE that is rejected by the endpoint node.

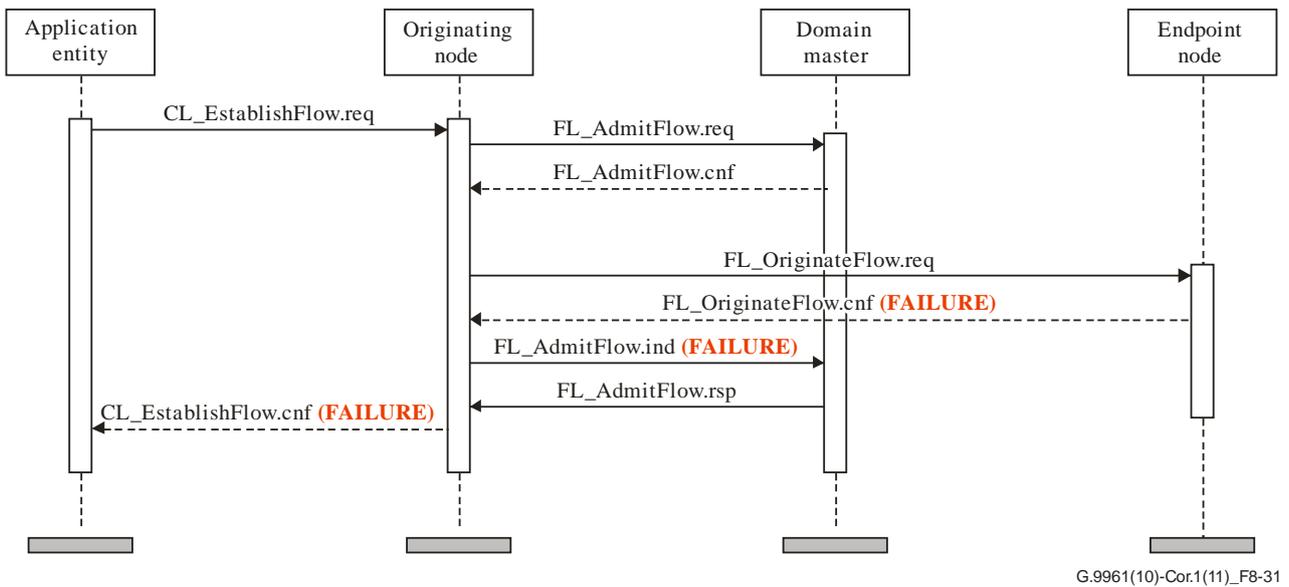


Figure 8-31 – Failure in flow establishment due to rejection by endpoint node

Figure 8-32 describes a failure in flow establishment due to rejection by the domain master (explicit signal from AE).

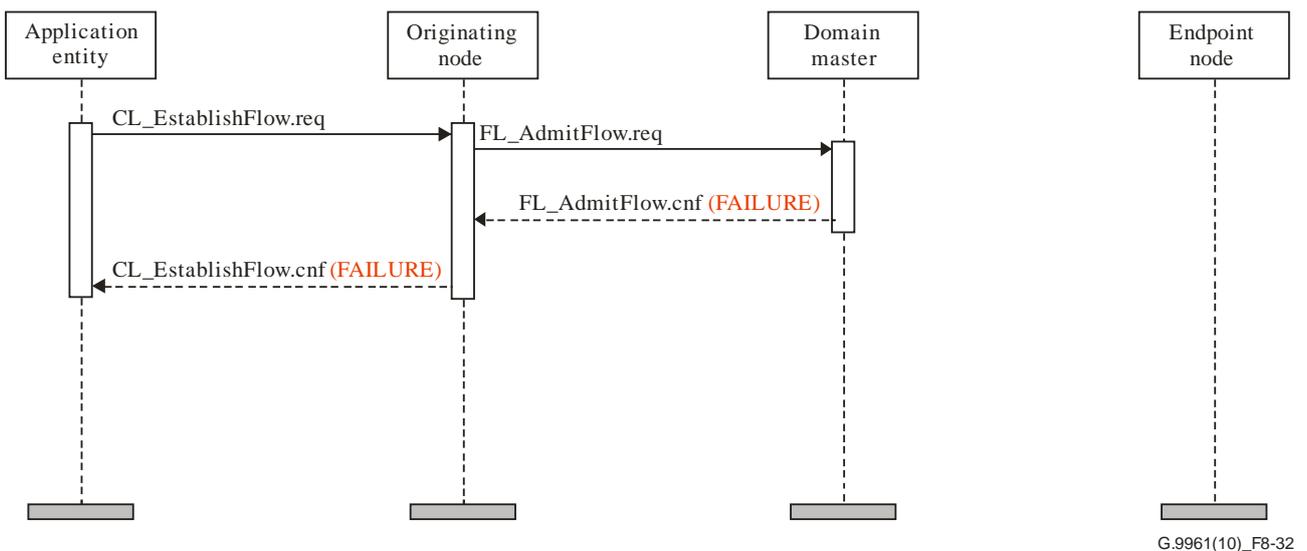


Figure 8-32 – Failure in flow establishment due to rejection by the domain master

8.6.2.2.2 Flow establishment via relay nodes

Flow establishment via relay includes the same first steps as specified in the previous clause. If the originating node determines that the designated endpoint node is hidden, the originating node shall initiate the tunnel establishment procedure that includes the following steps:

- a) The originating node shall allocate a FLOW_ID that shall uniquely identify the flow and the tunnel by using the tuple (DEVICE_ID, FLOW_ID), where DEVICE_ID is the originating node DEVICE_ID, the FLOW_ID is defined by the originating node.
 - If the originating node cannot support the flow establishment, it shall abort the tunnel establishment procedure. The abort procedure includes: notifying the higher layer by sending the message CL_EstablishFlow.cnf with failure code and releasing the allocated resources (allocated FLOW_ID, etc.).

- b) The originating node shall send an FL_AdmitFlow.req message to the domain master to reserve bandwidth resources for the flow. The message includes the tunnel identification, the designated endpoint and the TSpec. The domain master shall determine the route from the originating node to the designated endpoint node, determining the relay nodes along the route. The domain master shall assess whether there is enough available bandwidth resources to support all the flows that compose the tunnel, given the traffic specification and the line data rate for each flow of each hop in the tunnel.
- If the domain master can support the tunnel:
 - i) the domain master shall confirm the established tunnel flow by sending the FL_AdmitFlow.cnf message with success code. The FL_AdmitFlow.cnf message shall include a list with the relay nodes in route toward the designated endpoint;
 - ii) the domain master shall reserve the needed bandwidth resources for the identified tunnel: [SID, FLOW_ID, TUNNEL]. SID is the originating node ID. The FLOW_ID is as defined by the originating node and TUNNEL is indication that this flow is served via tunnel;
 - iii) the domain master shall reserve the resources for a period of time sufficient to complete the tunnel establishment. If the tunnel establishment is not completed during this period of time, the domain master shall release all the reserved resources.
 - Otherwise, if the domain master cannot support the tunnel bandwidth resources requirements:
 - i) the domain master shall notify the originating node by sending the FL_AdmitFlow.cnf message with the failure code;
 - ii) the originating node shall abort the flow tunnel establishment procedure.
- c) The originating node shall get from the received FL_AdmitFlow.cnf message the next relay toward the designated endpoint and shall send the FL_OriginateFlow.req message to the next relay node to verify whether the flow can be supported by the next relay node. The FL_OriginateFlow.req message shall include the list of the relay nodes toward the designated endpoint node.
- d) If the relay node that receives the FL_OriginateFlow.req message can support the requested flow establishment:
- it shall allocate a FLOW_ID that shall uniquely identify the flow toward the next node and the tunnel using the tuple (Originating node DEVICE_ID, FLOW_ID). It shall bind the previous FLOW_ID with the next flow FLOW_ID that it has just allocated. It shall get from the FL_OriginateFlow.req message the next relay node and set to the FL_OriginateFlow.req message the new allocated FLOW_ID and send it to the next relay node;
 - otherwise, if the relay node cannot support the new flow establishment, then it shall reply to the node that sent him the FL_OriginateFlow.req message by sending FL_OriginateFlow.cnf message with failure code, and abort the flow tunnel establishment procedure.
- e) The next relay node that receives FL_OriginateFlow.req shall execute step (d). Step (d) shall be executed as long as there are still relay nodes along the path toward the designated endpoint. If the relay node in step (d) has sent the FL_OriginateFlow.req message to the designated endpoint, then the procedure continues at step (f).
- f) The endpoint node that receives FL_OriginateFlow.req message shall evaluate its current actual capabilities to support the requested flow establishment:

- If the endpoint node cannot support the new flow establishment, then it shall reply by sending an FL_OriginateFlow.cnf message with failure code, and abort the tunnel establishment process;
 - Otherwise, if the endpoint node can support the new flow establishment, then it shall reply to the relay node that sent it the FL_OriginateFlow.req message by sending the FL_OriginateFlow.cnf message with success code.
- g) The relay node that receives the FL_OriginateFlow.cnf with success code shall update the FL_OriginateFlow.cnf message by adding to the route flows list the flow ID that it has established and shall send the FL_OriginateFlow.cnf message as a reply to the node that sent it originally the FL_OriginateFlow.req.
- h) Step (g) shall be executed until the FL_OriginateFlow.cnf is received by the originating node.
- i) When the originating node receives the FL_OriginateFlow.cnf message with the success code it shall send the FL_AdmitFlow.ind message to the domain master to notify that the tunnel establishment has completed successfully. The FL_AdmitFlow.ind shall include all the established flows that compose the tunnel. If the request to set up the flow was explicitly sent by the application entity, the originating node shall send to the application entity the CL_EstablishFlow.cnf message with success code to the application entity.
- j) After the domain master receives the FL_AdmitFlow.ind it shall acknowledge it by sending the FL_AdmitFlow.rsp message and allocate the actual bandwidth resources required to serve the flows that compose the tunnel.

In each one of the steps, in case of failure, the node that receives the failure indication shall abort the establishment process by forwarding the message with the failure code toward the originating node that is responsible for the whole tunnel establishment procedure, and release any allocated resource including the allocated FLOW_ID, etc. The originating node shall inform the domain master about the tunnel establishment failure and the domain master shall release the reserved bandwidth resources.

Figure 8-33 illustrates the message sequence chart (MSC) example of a successful tunnel establishment that includes three hops with two relay nodes and three flows.

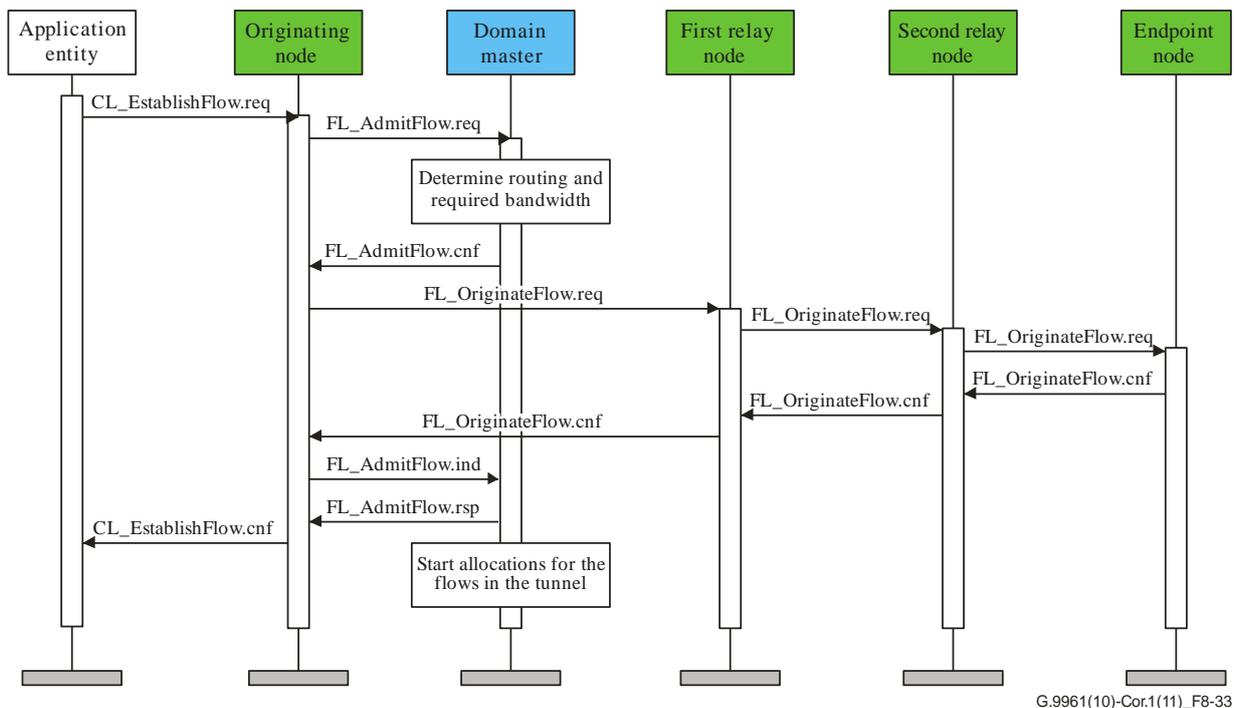


Figure 8-33 – Example of a successful tunnel establishment with two relay nodes

In the example shown in Figure 8-33, the originating node sends an admission request to establish the tunnel with the given traffic specifications and endpoint node. The domain master determines the routing from the originating node towards the endpoint node and determines the relay nodes along the route. The domain master shall assess whether there is enough available bandwidth resources to support all the flows that compose the tunnel, given the traffic specification and the line data rate for each service flow in each hop in the tunnel. The domain master reserves the needed bandwidth resources and confirms to the originating node the tunnel admission. The admission confirmation message includes the relay route toward the endpoint. The originating node starts the tunnel establishment by sending an originate request to the next relay node, which includes the routing path to the endpoint and the required traffic specifications. Each relay node that confirms the service flow establishment forwards the request to the next relay node until the endpoint. The endpoint replies with confirmation (positive or negative), and each relay node that receives the confirmation propagates the confirmation back to the originating node. After the originating node receives the confirmation, it indicates to the domain master to start the tunnel allocation take effect.

Figure 8-34 illustrates a successful tunnel establishment.

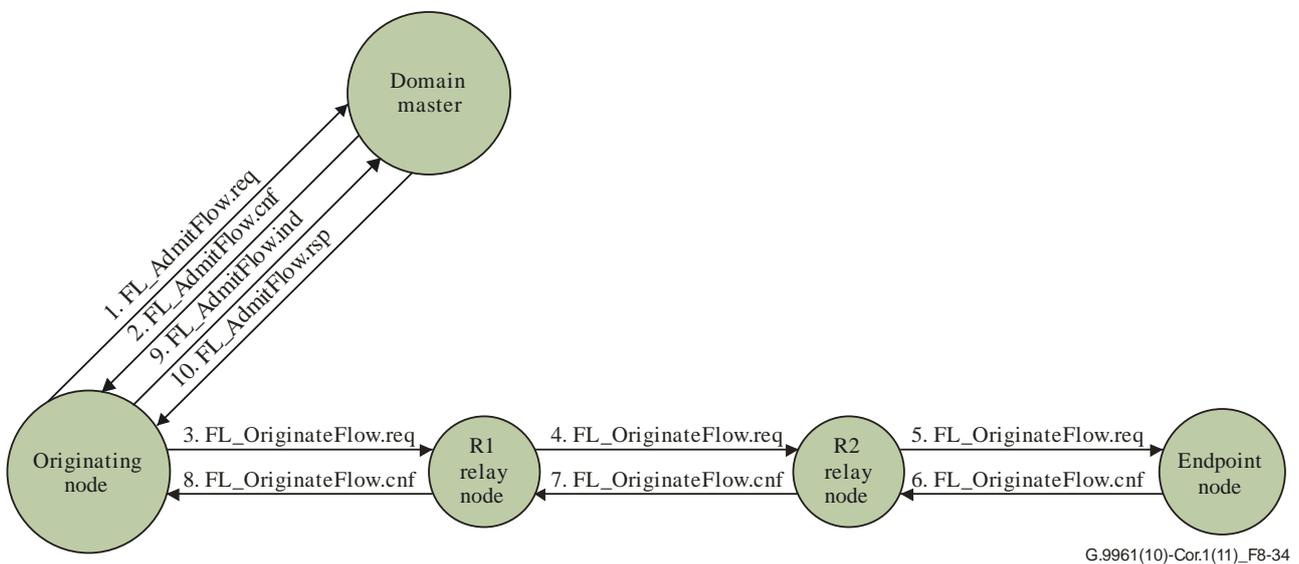


Figure 8-34 – Example of a successful tunnel establishment via two relay nodes

Figure 8-35 illustrates a tunnel establishment failure scenario due to the domain master rejecting the admission request due to insufficient bandwidth resources. In such a case the originating node should not start any transaction with the relay node to establish the flow and should abort the tunnel establishment.

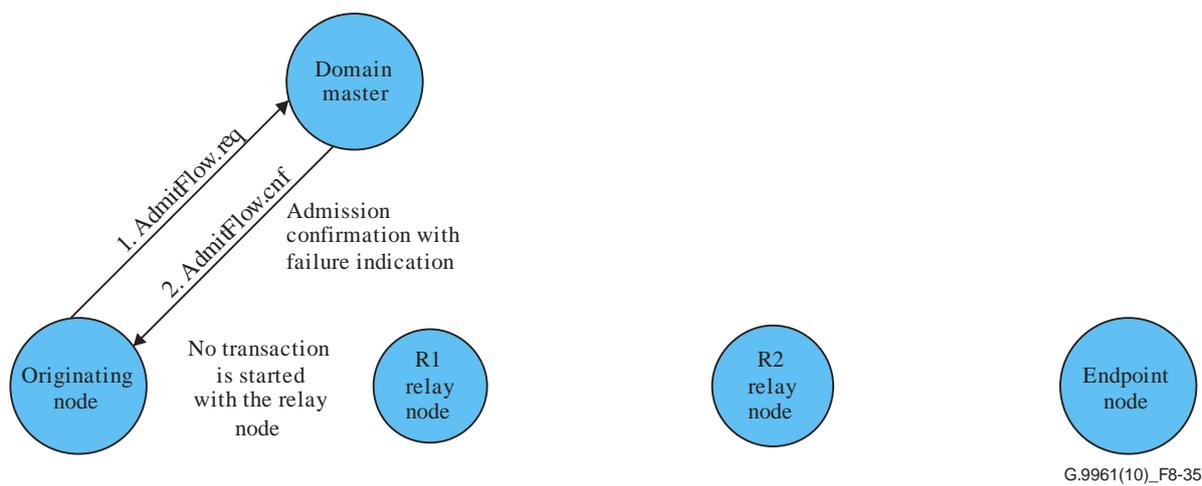


Figure 8-35 – Example of a tunnel establishment failure scenario due to the domain master rejecting the admission request

Figure 8-36 illustrates an example of a tunnel establishment failure in the case where the second relay node R2 rejects the flow establishment request and sends to the previous node, R1, a confirmation with failure code. Thus, relay node R1 sends a rejection message to the originating node. The originating node sends a failure indication to the domain master to release the reserved bandwidth resources and aborts the tunnel establishment process. The domain master shall acknowledge the abort indication for the tunnel establishment.

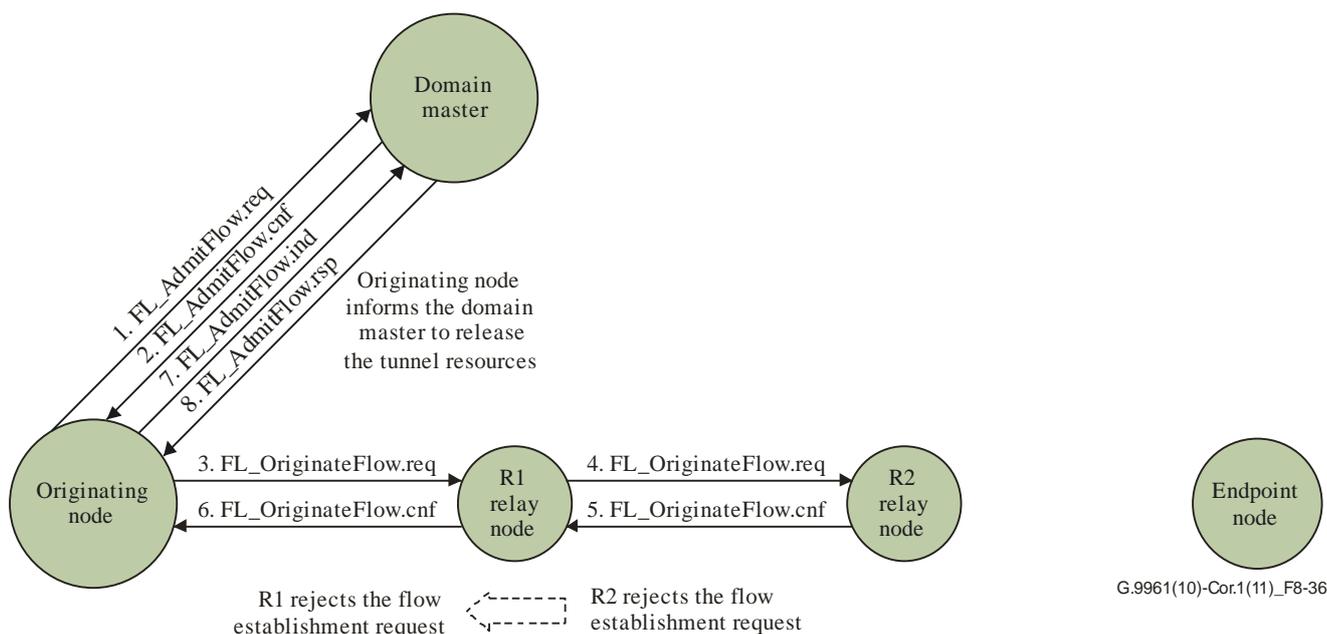


Figure 8-36 – Example of a tunnel establishment failure in the case where the second relay node, R2, rejects the flow establishment request

8.6.2.2.3 Flow maintenance

The flow maintenance is supported by management messages and by fields contained in PHY-frame headers. The protocol defines a series of messages used for communication between the application entity, nodes, and the domain master.

In addition to the management messages defined in the protocol, the bandwidth reservation update request (BRURQ) field that is carried in MSG-type PHY-frame headers enable a node to specify

bandwidth increases or decreases in number of bytes in a specified connection queue and the current used rate (bytes per symbol) (see clause 7.1.2.3.2.2.19/10.1.2.3.2.2.19 of [ITU-T G.9960]).

Once a service flow has been established, it shall be maintained by the originating node and by the domain master to fulfil the TSPEC contract using the following rules:

- When the bit loading employed between the two endpoints is reduced to such an extent that the bandwidth to support the agreed-upon traffic contract between the originating node and the domain master is insufficient, the originating node shall inform the domain master that it is being provided with insufficient bandwidth in its current CFTXOP by sending the FL_ModifyFlowParameters.ind message and by setting the number of bytes that shall be transmitted in the BRURQ field (see 7.1.2.3.2.2.19/10.1.2.3.2.2.19 of [ITU-T G.9960]) conveyed in the transmitted message PFH;
- When the bit loading employed between the two endpoints is increased, such that the flow begins to consume only a small fraction of the bandwidth allocated in the CFTXOP, the originating node shall inform the domain master of the situation by sending the FL_ModifyFlowParameters.ind message. If the domain master infers from inspection of a BRURQ field conveyed in MSG frame PHY-frame header or by receiving FL_ModifyFlowParameters.ind message with indication that the duration of the CFTXOP may be reduced while still complying with the terms of the traffic contract, it shall decrease the CFTXOP allocations accordingly;
- When there are user data traffic flows that are characterized by fixed packet size and fixed intervals between packets arriving via the A-interface, the node may adjust the allocations of the CFTXOP in a MAC cycle for this type of traffic using the FL_ModifyFlowAllocations.req message.

A node that has CFTXOP allocations for one of its flows shall update the number of bytes that shall be transmitted in this flow by appropriate settings in BRURQ field in the PHY frame header (see clause 7.1.2.3.2.2.19/10.1.2.3.2.2.19 of [ITU-T G.9960]) of the frames of this flow.

A node that is determined according to the routing table as hidden from the domain master shall send a FL_ModifyFlowParameters.ind message to the domain master (via a relay node) to inform the domain master on changes required in the CFTXOP allocations of its flow.

Once it has been informed that the node's bandwidth requirements for the specified flow have changed, the domain master may choose to expand or to contract the allocation made for the flow. This change will be reflected in a MAP message sent in the current or in one of the following MAC cycles.

If the domain master changes the allocation for a persistent flow, the new allocation for the flow conveyed in the schedule will become effective once the domain master has counted down the upcoming change in the MAP frame.

The internal rules used by the domain master to decide whether an allocation should be expanded or contracted due to ongoing flow maintenance done by the bandwidth management function are out of the scope of this Recommendation.

If the domain master is unable to expand the allocation of the flow the domain master may choose to offer a change in the flow parameters by sending a FL_ModifyFlowParameters.req message to the originating node to inform the node that its traffic contract must be changed in order to support the required allocation.

The node shall transmit a FL_ModifyFlowParameters.cnf message to the domain master indicating whether the offered flow parameters can be accepted or not.

If the node has not accepted the offered flow parameters, it shall end the flow.

Figure 8-37 describes an example of a change in flow parameters offered by the domain master as a result of a request made by the node to change the flow allocation

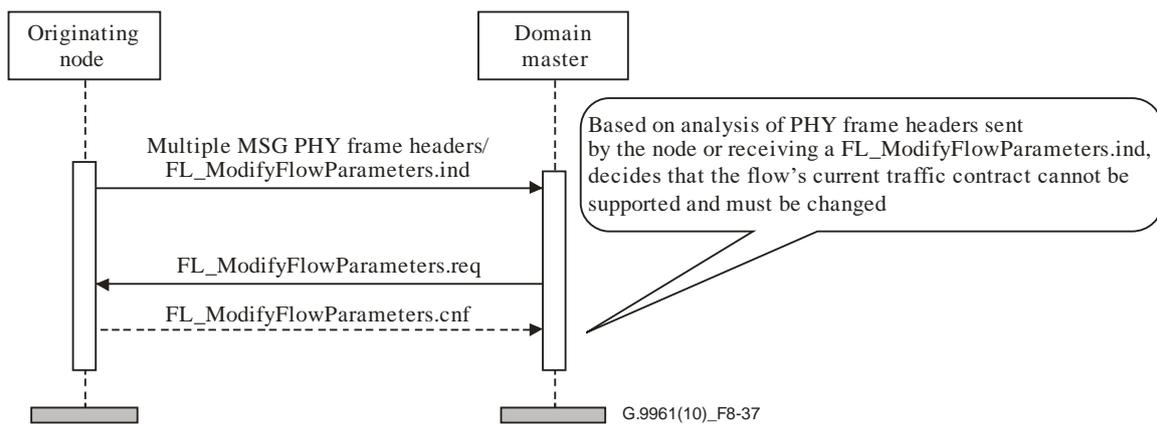


Figure 8-37 – Flow parameters modification example

In the above example, the domain master determined, either from the BRURQ field in the PHY-frame header of the received data messages or from the received FL_ModifyFlowParameters.ind message, that the bandwidth allocation previously made for this flow cannot be increased. If the domain master has to inform the node that its traffic contract must be changed, it shall send the FL_ModifyFlowParameters.req message to the node that originated the flow. The node shall respond to this message, either by changing accordingly the flow's characteristics or by ending the flow. In either case, it shall send a result code in the FL_ModifyFlowParameters.cnf message to the domain master.

NOTE – If the bandwidth allocation for the flow can be changed, this will be reflected in the MAP describing the following MAC cycles.

8.6.2.2.3.1 Timing adjustment of CFTXOP

When an active flow is served by the domain master by a periodic CFTXOP allocation, the node that originated the flow may send the FL_ModifyFlowAllocations.req message to the domain master to request an adjustment of the timing of the CFTXOP allocation by specifying an offset time to postpone or advance the timing of the CFTXOP allocation relative to the last allocated CFTXOP in the MAC cycle. The domain master shall acknowledge reception of FL_ModifyFlowAllocations.req message by sending FL_ModifyFlowAllocations.cnf message back to the node. If a node does not receive the FL_ModifyFlowAllocations.cnf message within 40 ms, then it may repeat the request. If a node got the acknowledgement from the domain master, it shall not request the adjustment during at least the next two MAC cycles.

NOTE – The restriction of two MAC cycles is due to the built-in delay between receiving any update request until the change is reflected in the MAP, and the MAP relevancy is to the next MAC cycle. In case the CFTXOP was advertised in the MAP in a persistent way, then the change can be effective only after the persistence can be expired.

After sending the FL_ModifyFlowAllocations.cnf message, the domain master may modify the timing allocation of the CFTXOP according to the value specified in the FL_ModifyFlowAllocations.req message sent by the requesting node.

8.6.2.2.4 Flow termination

In most home networks, a flow has a limited lifetime. For example, several hours for a video stream or several minutes for an audio stream.

There are several situations that require a flow to be terminated:

- After the application entity on the originating node no longer has data to send using a particular flow, it may signal the originating node that the flow has ended, or the originating node may infer that the application entity has finished sending data;
- In the case of a flow that was established following automatic traffic classification, the originating node shall determine that the application entity has finished sending data associated with the flow;
- The domain master may end selected flows as channel conditions change. For example, a decrease in the possible bit load between two or more nodes may result in over-subscription of the channel. This means that there is now insufficient bandwidth to support one or more existing flows, and that ending one or more flows may free up sufficient bandwidth so that some other flows can continue. The internal rules used by the domain master to decide when a flow must be terminated are out of the scope of this Recommendation;
- The domain master may determine that one or more of the nodes associated with a flow has left the domain without notification. For example, a node may be turned off or could fail while receiving data. In this case the domain master would eventually infer that the node has left and would then end the flow.

The originating nodes and the domain master shall follow the flow signalling protocol when ending a flow.

8.6.2.2.4.1 Message sequence chart for flow termination

The MSC in Figure 8-38 shows an example of how a flow can be terminated.

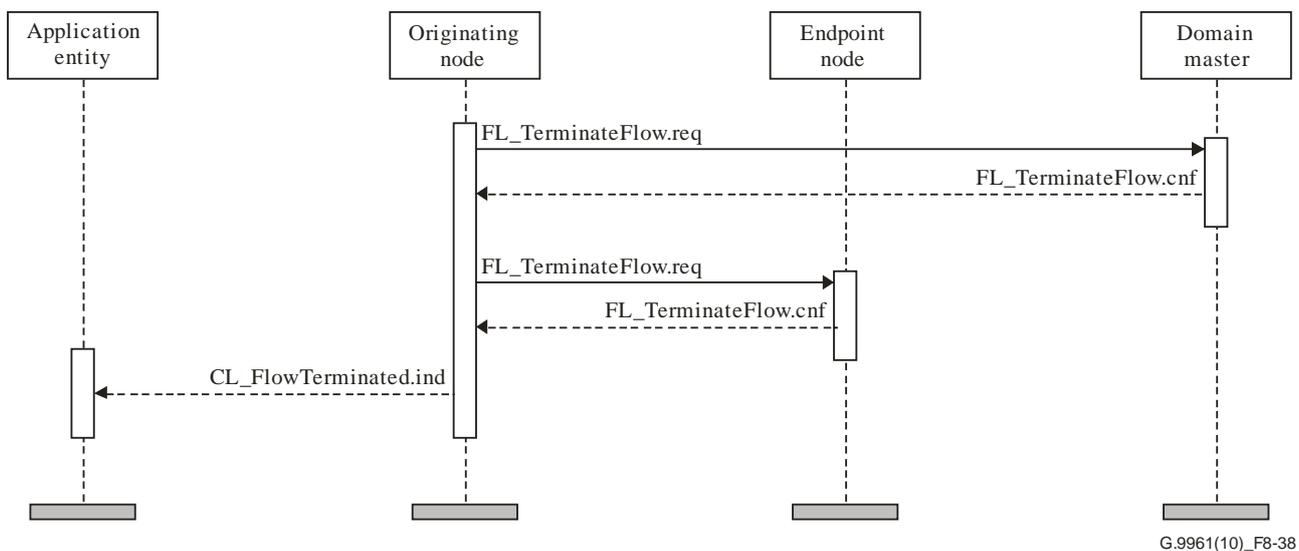


Figure 8-38 – MSC showing an example of how a flow can be terminated

In this example, the originating node has determined that the transmission of data associated with the flow has ended, through inference and not by the reception of message `CL_TerminateFlow.req` transmitted from the application entity residing on the client. The node frees all the resources allocated to support the flow and sends `FL_TerminateFlow.req` to the domain master, telling it that resources previously committed to the flow may be freed. The domain master then sends `FL_TerminateFlow.cnf` to the originating node.

If the domain master has sent the `FL_TerminateFlow.req` message, then the originating node frees all the resources allocated to support the flow and sends `FL_TerminateFlow.cnf` to the domain master.

Next, the originating node sends FL_TerminateFlow.req to the endpoint node. That node frees its own resources that it allocated to support the flow, and sends FL_TerminateFlow.cnf to the originating node.

Finally, the originating node sends CL_FlowTerminated.ind to the application entity running on its associated client.

8.6.2.2.5 Tunnel reconstruction or termination

The following situations require a tunnel to be terminated:

- After the application entity on the originating node no longer has data to send using a particular flow, it may signal to the originating node that the flow has ended or the originating node may infer that the application entity has finished sending data;
- When a tunnel has been established following automatic traffic classification and the originating node determines that the application entity has finished sending data associated with the tunnel;
- When the domain master determines that there is insufficient bandwidth to support that tunnel;
- When the domain master determines that the originating node or the endpoint node has left the domain;
- Any node in the tunnel that infers a broken link and informs the domain master about the broken link, the domain master may request the originating node to restore the tunnel via another alternative route or request the originating node to terminate the tunnel.

The originating node, the endpoint node, the relay nodes and the domain master shall follow the flow termination protocol when terminating a tunnel.

8.6.2.2.5.1 Normal tunnel termination

If a tunnel was established due to an application entity request via a CL_EstablishFlow.req message the tunnel termination shall be started only after the application entity sends a CL_TerminateFlow.req message. If the tunnel was established due to a request from an application entity and the originating node determines that the application entity has finished sending data associated with the flow, it shall send a CL_TerminateFlow.ind message to the application entity. The application entity upon receiving a CL_TerminateFlow.ind message may send to the originating node a CL_TerminateFlow.req message to terminate the flow.

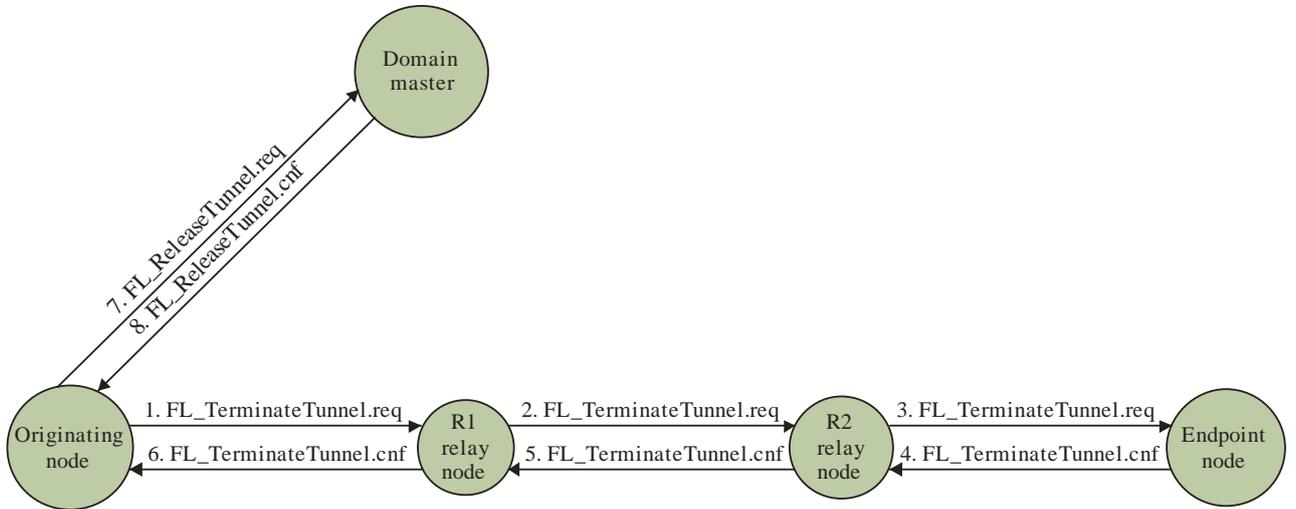
If a tunnel was established following automatic traffic classification and the originating node determines that the application entity has finished sending data associated with the tunnel, it shall terminate the tunnel.

When the originating node has to terminate a tunnel, it shall send to the adjacent relay node a FL_TerminateTunnel.req message to terminate the tunnel. The relay node shall continue to forward the FL_TerminateTunnel.req message to the next node in the tunnel towards the endpoint node. When the endpoint node receives the FL_TerminateTunnel.req message, it shall reply with a FL_TerminateTunnel.cnf message and free resources allocated for the tunnel.

Upon receiving the FL_TerminateTunnel.cnf message, the relay node shall send the FL_TerminateTunnel.cnf message to the node that sent it the FL_TerminateTunnel.req message and release the resources allocated for the tunnel. When the originating node receives the FL_TerminateTunnel.cnf message, it shall release the resources allocated for the tunnel and send an FL_ReleaseTunnel.req message to the domain master. Upon receiving the FL_ReleaseTunnel.req message the domain master shall reply with the FL_ReleaseTunnel.cnf message and release the resources allocated for the tunnel.

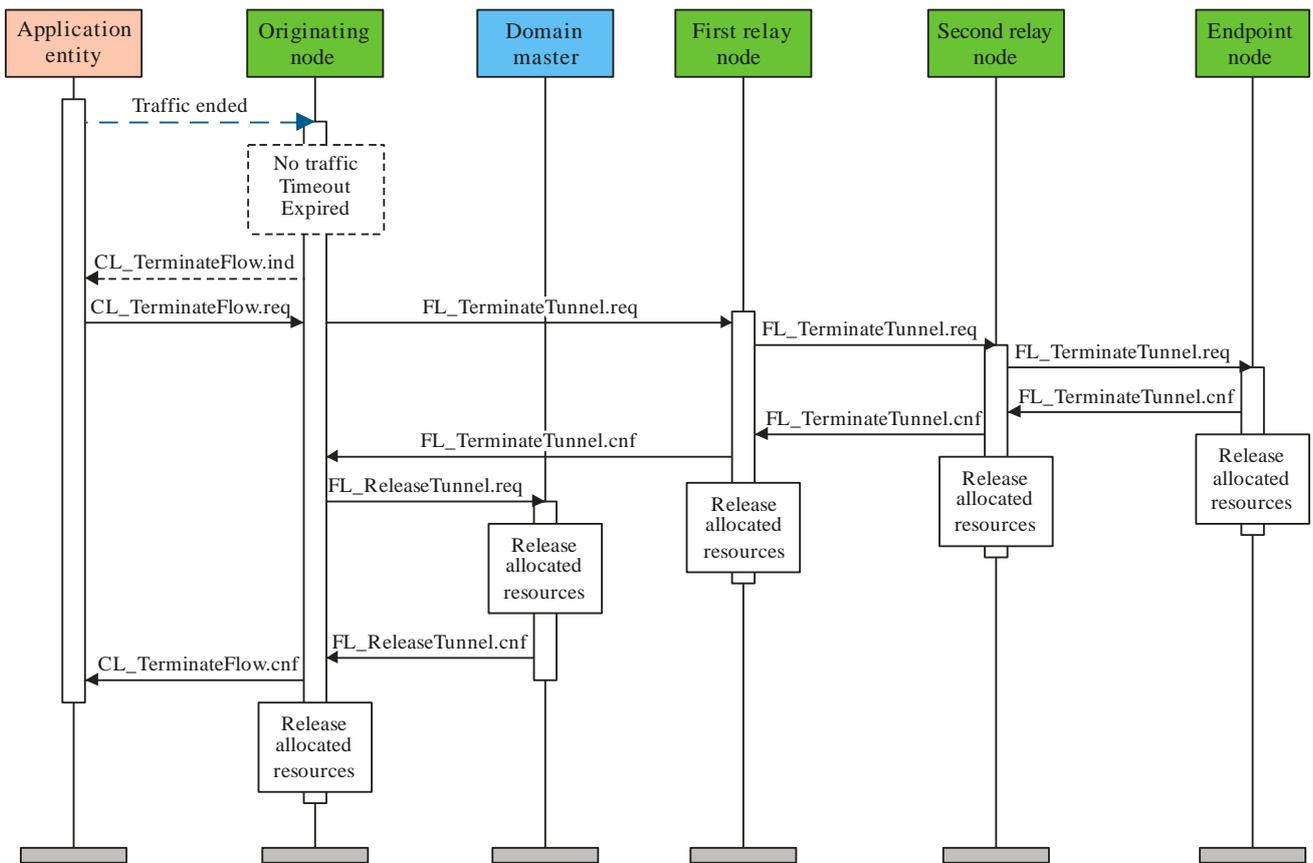
Finally, the originating node shall send CL_TerminateFlow.cnf to the application entity, if the tunnel termination was based on a CL_TerminateFlow.req message sent by the application entity.

For all the terminations procedures, the node that transmits request or indication messages, should consider the number of hops to the endpoint node while setting up timeouts to infer loss of confirmation or response messages.



G.9961(10)-Cor.1(11)_F8-38.1a

Figure 8-38.1a – Normal tunnel termination



G.9961(10)-Cor.1(11)_F8-38.1b

Figure 8-38.1b – MSC of normal termination of a tunnel

8.6.2.2.5.2 Tunnel reconstruction or termination due to a broken link

The protocol for tunnel reconstruction or termination due to a broken link is defined in the following steps:

- a) When a node that is participating in a tunnel determines that a link to the next node in a tunnel is broken, it shall send an FL_BrokenTunnel.ind message to the domain master to inform about the broken link. The DM shall reply with an FL_BrokenTunnel.rsp message to confirm receiving the FL_BrokenTunnel.ind message;
- b) The domain master shall calculate an alternative new path from the originating node towards the endpoint node:
 - If the domain master succeeds in calculating an alternative path, it shall reserve the needed resources to support the new alternative path for the tunnel and shall build an FL_DM_RenewTunnel.req message that contains the new alternative path for the tunnel and send the FL_DM_RenewTunnel.req message to the originating node.

NOTE – The new path may include some flows that are still valid in the current tunnel.

- Otherwise, if the domain master did not succeed in calculating or allocating an alternative path to reconstruct the broken tunnel, it shall send an FL_TerminateFlow.req message to the originating node, to terminate the tunnel as specified in clause 8.6.2.2.5.3.
- c) Upon receiving the FL_DM_RenewTunnel.req message, the originating node shall build an FL_RenewTunnel.req message based on the received FL_DM_RenewTunnel.req message and send it to the next relay node according to the path as specified in the received FL_DM_TerminateFlow.req message. The FLOW_ID allocated by the originating node to the renewed tunnel shall be the same as the FLOW_ID allocated to the original tunnel with a broken link.
- d) If the broken link is between the originating node and the first relay node, as shown in Figure 8-38.2, then the originating node shall replace the old broken flow ('a') with the new established flow 'd' with the new alternative relay node (AR1). The originating node shall establish the flow with AR1 by sending the FL_RenewTunnel.req message to a new alternative relay node (AR1). The originating node shall delete the flow ('b') between the relay node (R1) and the next relay node (R2) by indirectly sending via relay nodes (because the link to R1 is broken) an FL_DeleteFlow.req message to the old relay node (R1). The relay node (R1) shall send an FL_DeleteFlow.cnf message to the originating node (via its available route to the originating node).

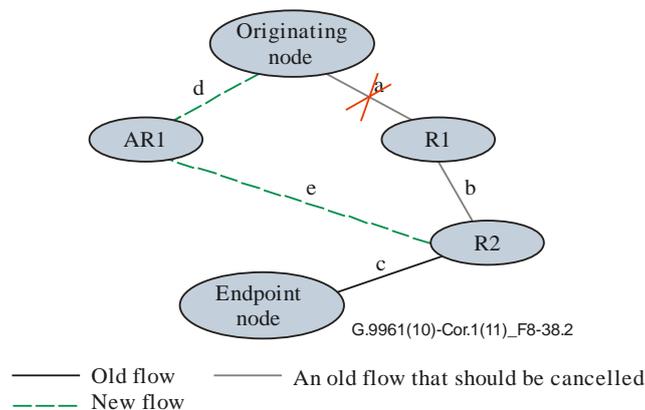


Figure 8-38.2 – Tunnel renewal due to a broken link: case 1

- e) A relay node participating in this tunnel that receives the FL_RenewTunnel.req message and has a valid flow for this tunnel towards the next node in the path as specified in the received FL_RenewTunnel.req message (as relay node R1 in Figure 8-38.3), shall update the FL_RenewTunnel.req message with the actual RelFLOW_ID ('b') that it has with the next node (R2), and then it shall relay the FL_RenewTunnel.req message to the next node (R2).
- f) A relay node (as for example relay node R2 in Figure 8-38.3) participating in a tunnel, that receives an FL_RenewTunnel.req message that specifies that it has to reconstruct the tunnel by establishing a flow to a new relay node (AR3) shall do the following actions:
- It shall allocate a FLOW_ID for the new established flow, update the received FL_RenewTunnel.req message with the allocated FLOW_ID (RelFLOW_ID) and send the updated FL_RenewTunnel.req message to the next new relay node (AR3) to establish the flow;
 - It shall bind the actual flow ('b' in Figure 8-38.3) that it has with the previous node in the tunnel with the new established flow ('f' in Figure 8-38.3).

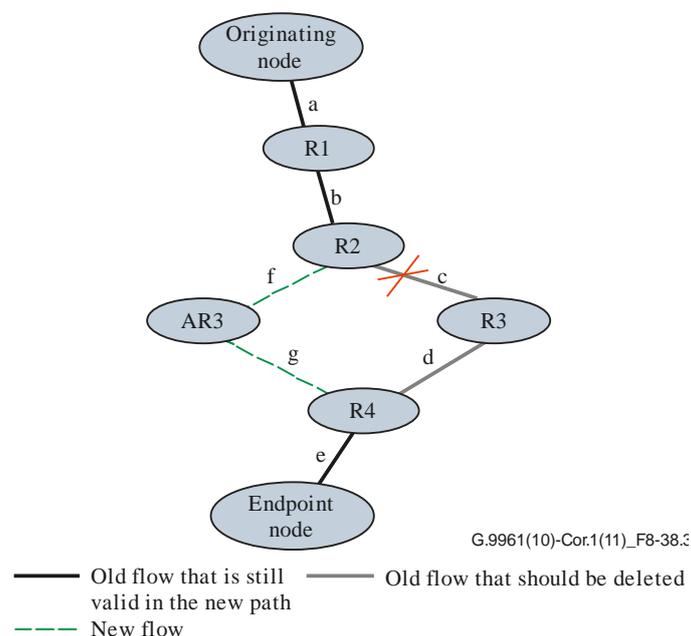


Figure 8-38.3 – Tunnel renewal due to a broken link

- g) A node (for example AR3 in Figure 8-38.3) that receives an FL_RenewTunnel.req message that requests that the node has to reconstruct a tunnel shall do the following:
- It shall save the FLOW_ID (flow f in Figure 8-38.3) that is specified in the FL_RenewTunnel.req message that was established by the node that sent it the FL_RenewTunnel.req message (R2 Figure 8-38.3);
 - It shall allocate a FLOW_ID (flow g in Figure 8-38.3) for the flow that it has to establish with the next relay node (R4 in Figure 8-38.3);
 - It shall update the received FL_RenewTunnel.req message with the allocated FLOW_ID (RelFLOW_ID) and send the updated FL_RenewTunnel.req message to the next new relay node (R4) to establish the flow;
 - It shall bind the flow ('f' in Figure 8-38.3) that it has with the previous node (R2) with the flow ('g' in Figure 8-38.3) that it establishes with the next node (R4).
- h) If a relay node that receives an FL_RenewTunnel.req message which specifies that it has to reconstruct the tunnel by establishing a flow to a new relay node cannot establish the new requested flow (lack of internal resources), it shall reply to the node that sent it the FL_RenewTunnel.req message with an FL_RenewTunnel.cnf message with a failure code

and abort the tunnel reconstruction procedure. The FL_RenewTunnel.cnf with the failure code shall be relayed by the relay nodes in the tunnel toward the originating node.

- i) when a relay node (for example R4 Figure 8-38.3) participating in a tunnel receives an FL_RenewTunnel.req message from a new relay node (AR3 Figure 8-38.3) in the tunnel and it has a valid flow towards the next node in the path specified in FL_RenewTunnel.req message, it shall release the old flow ('d') that it has with the old relay node and bind the new flow specified in the FL_RenewTunnel.req message ('g') with the next flow ('e') that it has with the next node. It shall update the field RelFLOW_ID in the FL_RenewTunnel.req message with the actual FLOW_ID ('e') that it has with the next node (endpoint in Figure 8-38.3) and relay the updated FL_RenewTunnel.req message to the next node (endpoint in Figure 8-38.3).
- j) Each relay node along the tunnel path that receives an FL_RenewTunnel.req message shall execute one of the previous steps (d, e, f, g and i) according to its topology state and conditions. It means that the originating node shall do the operations defined in step d and a relay node participating in this tunnel that receives the FL_RenewTunnel.req message and has a valid flow for this tunnel towards the next node in the path as specified in the received FL_RenewTunnel.req message shall execute step e and so on.
- k) When the endpoint node receives the FL_RenewTunnel.req message, it shall reply to the relay node that sent it the FL_RenewTunnel.req message with an FL_RenewTunnel.cnf message. If it has an old flow with another relay node for that tunnel, it shall replace the old flow with the new flow (RelFLOW_ID) that is specified in the received FL_RenewTunnel.req message. If the flow is a bidirectional flow, the endpoint shall update the FL_RenewTunnel.cnf message with the BFLOW_ID of the flow in the reverse direction.
- l) Each relay node in the tunnel that receives an FL_RenewTunnel.cnf with a success code shall update the FL_RenewTunnel.cnf message by adding to the flows list that compose the tunnel its own established RelFLOW_ID that it has established. If the flow is a bidirectional flow, the relay node shall update the FL_RenewTunnel.cnf message with the assigned BFLOW_ID of the flow in the reverse direction. Then the relay node shall send the FL_RenewTunnel.cnf message as a reply to the node that has sent it the FL_RenewTunnel.req message. This step shall be executed by all intermediate nodes until the FL_RenewTunnel.cnf is sent to the originating node.
- m) When the originating node receives the FL_RenewTunnel.cnf message with a success code it shall build and send an FL_DM_RenewTunnel.cnf message to the domain master to notify that the tunnel reconstruction has been completed successfully. The FL_DM_RenewTunnel.cnf message shall include all the flows that implement the reconstructed tunnel.
- n) The originating node shall send an FL_DeleteFlow.req message to each one of the nodes that were part of the original tunnel and not part of the restored tunnel. Upon receiving the FL_DeleteFlow.req message, the node shall release the resources allocated to this flow and reply with an FL_DeleteFlow.cnf message.
- o) After the domain master receives the FL_DM_RenewTunnel.cnf message it shall allocate the bandwidth resources required to serve the flows in the reconstructed tunnel and release allocated resources of the deleted flows.

In each one of the steps above, in case of failure, the node that receives a message with a failure indication shall abort the establishment process and forward the message with the failure code towards the originating node that is responsible for the entire tunnel reconstruction procedure and release any allocated resources (allocated RelFLOW_ID, etc.). The originating node shall inform the domain master about the tunnel renewal failure and the domain master shall release the reserved bandwidth resources. If the tunnel establishment was triggered by an application entity, the originating node shall inform the application entity that the tunnel has been terminated.

The following clauses contain some examples of the protocol according to different cases.

8.6.2.5.2.1 MSC of tunnel reconstruction for case 1

In case 1 the link between the originating node and the first relay node (R1) is broken as shown in Figure 8-38.2. The new calculated path goes from the originating node to relay node AR1. The originating node has to send an FL_DeleteFlow.req message to node R1 because relay node R1 is not involved in the new restored tunnel. The following figure shows the MSC of tunnel reconstruction for case 1.

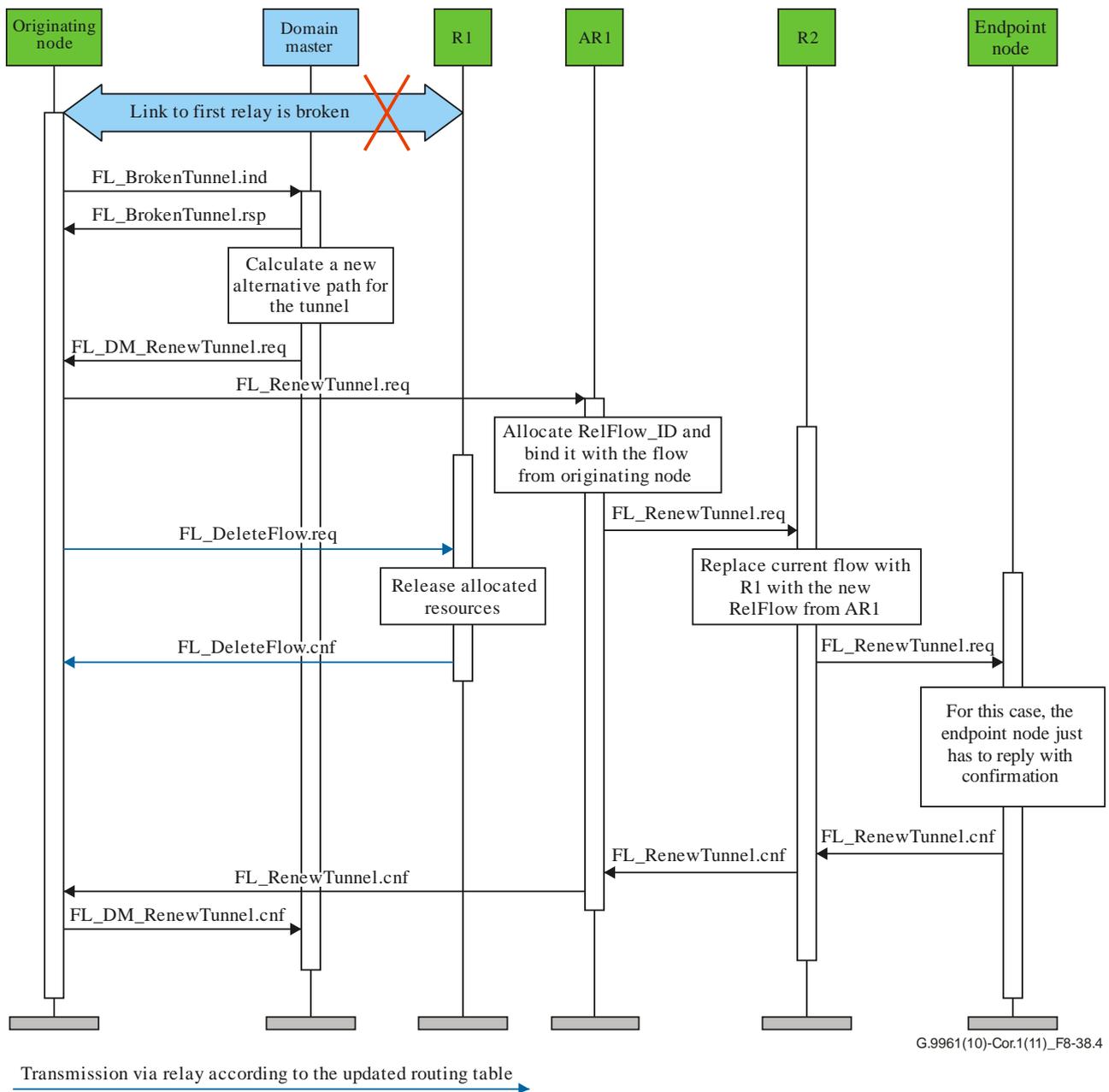


Figure 8-38.4 – MSC of tunnel renewal due to a broken link case 1

8.6.2.5.2.2 Tunnel reconstruction for case 2

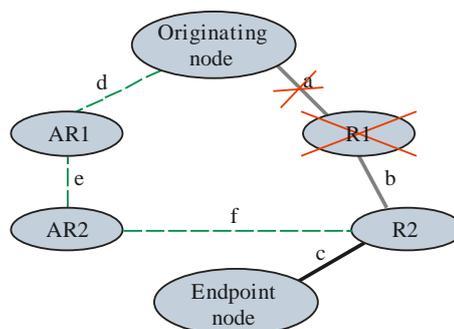
In case 2, flow 'a' is broken because node R1 is disconnected. In this case the originating node should not send an FL_DeleteFlow.req to node R1 because node R1 is disconnected.

Originating node sends RenewTunnel to AR1 to renew a tunnel to EP via AR1, AR2, R2

AR1 sends RenewTunnel to AR2 to establish a tunnel to EP via R2

AR2 sends a RenewTunnel to R2 to replace flow R1 ↔ R2 with the flow AR2 ↔ R2

R2 replace flow R1 ↔ R2 with the flow AR2 ↔ R2



G.9961(10)-Cor.1(11)_F8-38.5

Figure 8-38.5 – Tunnel renewal due to a broken link case 2

8.6.2.2.5.2.3 Tunnel reconstruction for case 3

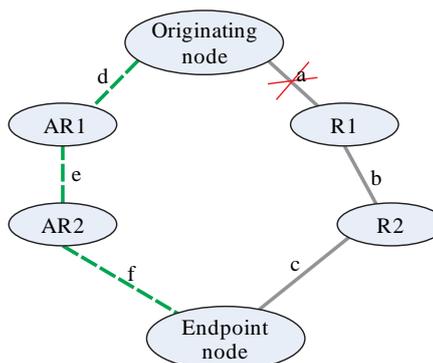
The following case is an example to explain how the nodes shall act in a scenario where all the old relay nodes are no longer in the renewed tunnel, and all the relay nodes in the reconstructed tunnel are new. In this case the alternative path includes only new relay nodes: AR1 AR2. The endpoint node completes the tunnel restoring procedure by replacing the old flow 'c' with the new flow 'f'.

Originating node sends RenewTunnel to AR1 to establish flow to EP via AR2

AR1 allocates flow e and sends RenewTunnel to AR2 to establish a tunnel towards node EP

AR2 sends RenewTunnel to EP to replace flow c with flow f (R2 ↔ EP with AR2 ↔ EP)

Originating node sends FL_DeleteFlow.req to R1 and R2 to delete flows b and c



— Old flow
 - - - New flow
 — An old flow that should be cancelled

Figure 8-38.6 – Tunnel renewal due to a broken link case 3

8.6.2.2.5.2.4 Tunnel reconstruction for case 4

In case 4 the link between two relay nodes R1 ↔ R2 is broken and the alternative new path consists of some of the old flows together with some new flows. Relay node R1 establishes flow 'd' with AR2 and replaces flow 'b' with flow 'd'. Relay node AR2 establishes flow 'e' with R2 and relay node R2 replaces flow 'b' with flow 'e'. The originating node shall not send FL_DeleteFlow.req message to relay node R1 and relay node R2 to delete flow 'b', because it was replaced by relay node R1 and relay node R2.

Link b is broken

Originating node sends FL_RenewTunnel.req to R1 to renew the tunnel to EP via AR2, R2 (to replace flow b (R1 ↔ R2) with flow d (R1 ↔ AR2) and flow e (AR2 ↔ R2))

R1 sends FL_RenewTunnel.req to AR2 to establish flow d (R1 ↔ AR2)

AR2 sends FL_RenewTunnel.req to R2 to establish flow e and to replace flow b (R1 ↔ R2) with flow e (AR2 ↔ R2)
R2 ends the renewal process by replying with FL_RenewTunnel.cnf

Flow c and flow a from the original tunnel continue as before

- Old flow
- - - New flow
- An old flow that should be cancelled

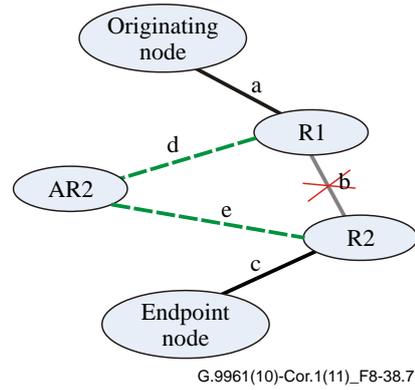


Figure 8-38.7 – Tunnel renewal due to a broken link case 4

The following MSC describes the protocol behaviour for case 4.

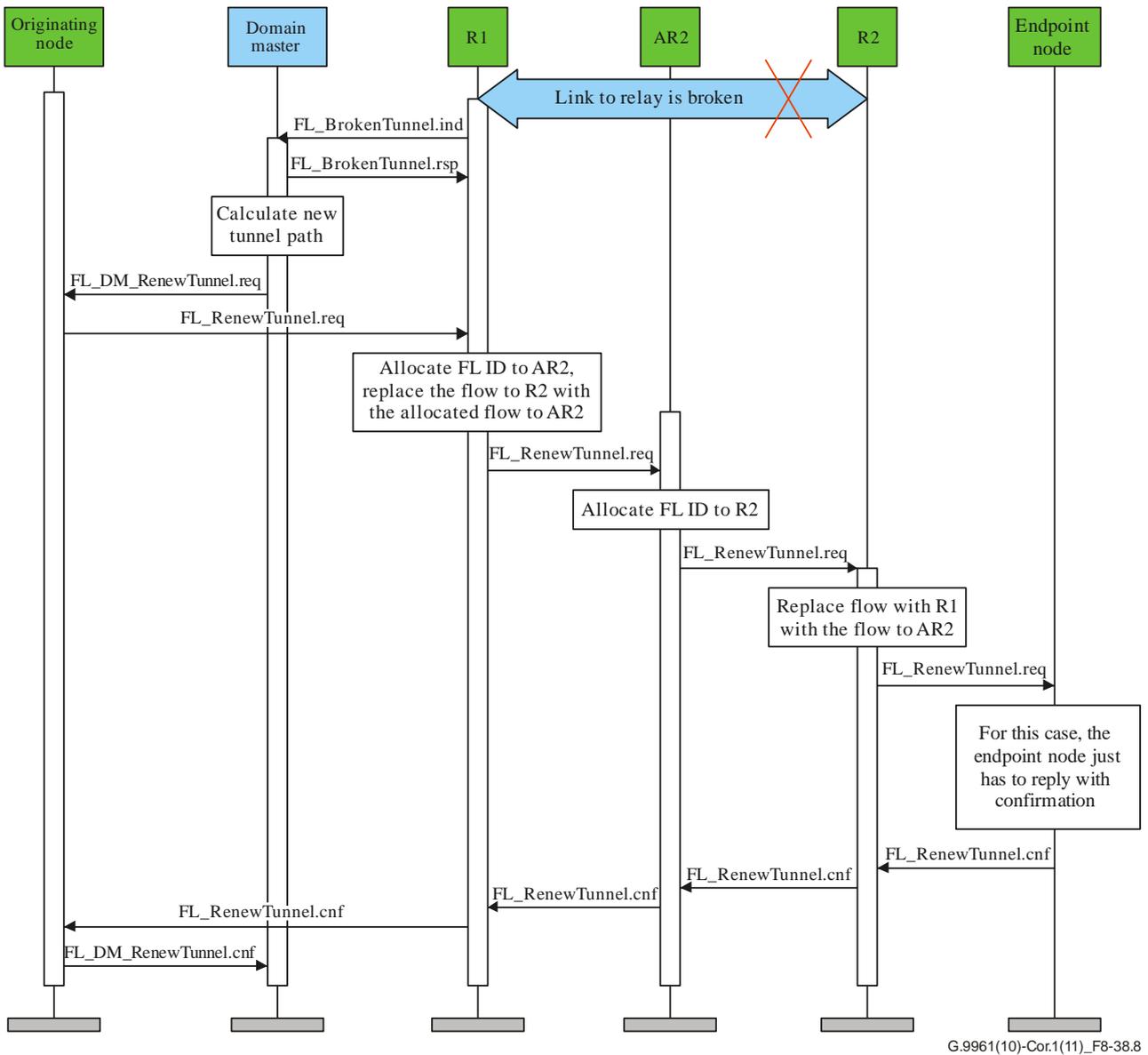
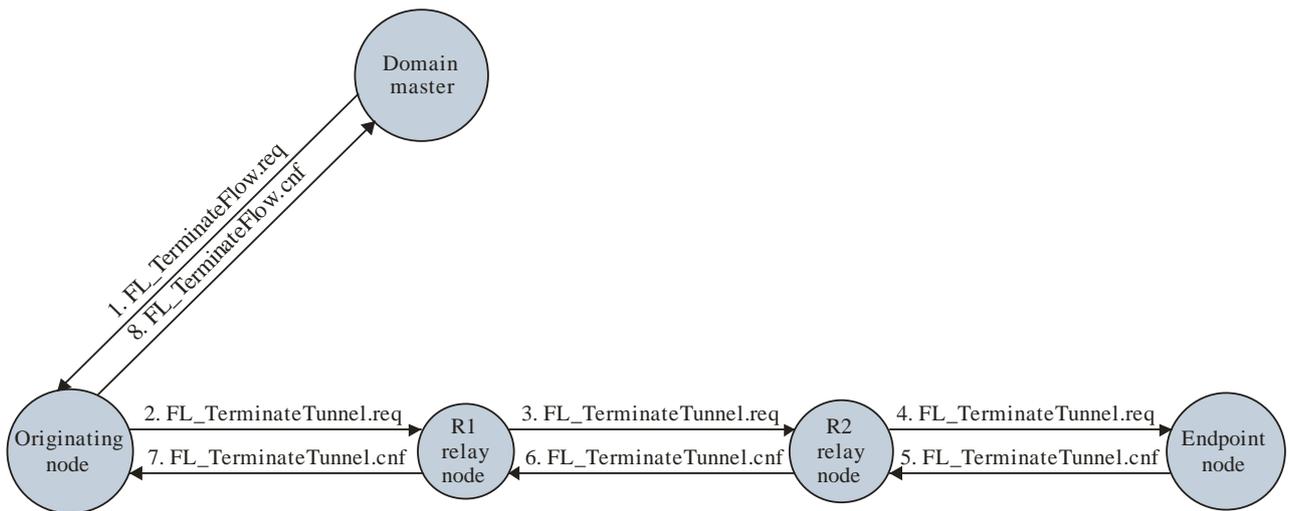


Figure 8-38.8 – MSC of tunnel renewal due to a broken link case 4

8.6.2.2.5.3 Domain master triggered tunnel termination

If the domain master decides to terminate a tunnel due to lack of resources to support the tunnel, it shall send an FL_TerminateFlow.req to the originating node, with the ReasonCode set to 01₁₆ (termination by domain master due to lack of resources). The originating node upon receiving the FL_TerminateFlow.req message shall start the tunnel termination protocol as specified in clause 8.6.2.2.5.1 but with one difference: the originating node shall send the FL_TerminateFlow.cnf message to the domain master instead of sending the FL_ReleaseFlow.req message.

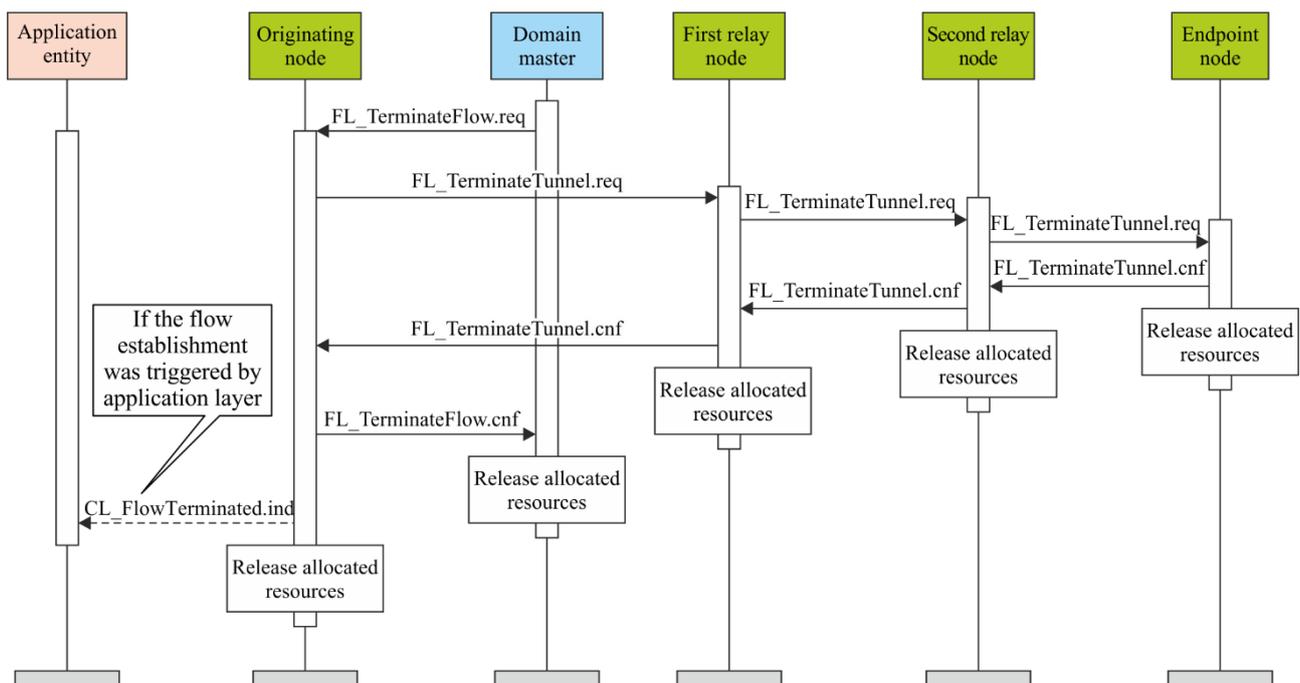
Figure 8-38.9a shows the protocol for the case that the domain master triggers tunnel termination due to lack of resources to support the tunnel.



G.9961(10)-Cor.1(11)_F8-38.9a

Figure 8-38.9a – Domain master triggers tunnel termination

Figure 8-38.9b shows the MSC for tunnel termination protocol when the tunnel termination is triggered by the domain master.



G.9961(15)-Cor.2(16)_F8-38.9b

Figure 8-38.9b – MSC for tunnel termination triggered by the domain master

8.6.2.2.5.4 Tunnel termination due to disconnection of the originating node

If the domain master determines that the originating node is disconnected, the domain master shall release the resource allocations for the tunnel and it shall send an FL_TerminateTunnel.req message to the first relay node with the ReasonCode field indicating that the tunnel has been terminated because the originating node has been disconnected. The relay node shall forward the message to the next node in the tunnel path as done in tunnel termination that is initiated by the originating node. When the first relay node receives the FL_TerminateFlow.cnf reply message, it shall send an FL_TerminateTunnel.cnf message to the domain master.

Figure 8-38.10 shows the MSC protocol for the case where the domain master terminates a tunnel when the originating node is disconnected.

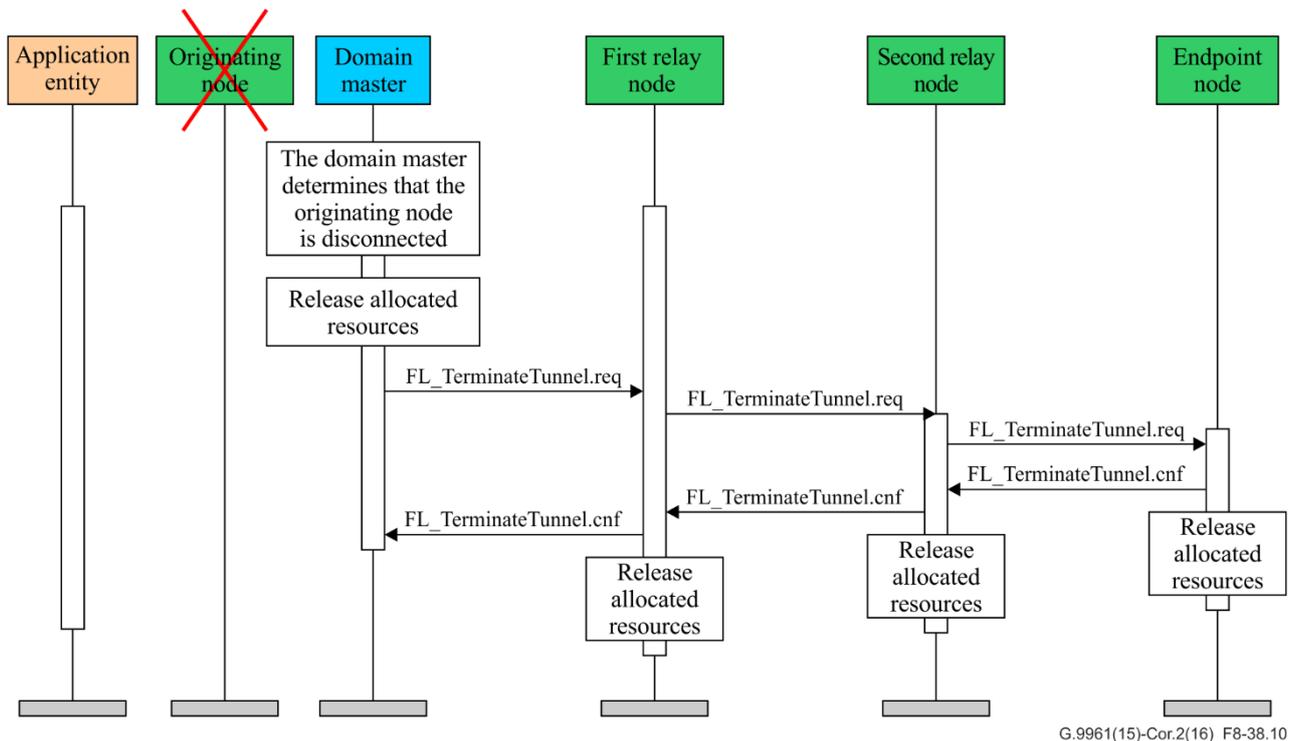


Figure 8-38.10 – MSC for tunnel termination when the originating node is disconnected

8.6.2.3 Flow signalling protocol messages

The following clauses specify the messages to support the flow signalling protocol.

8.6.2.3.1 Format of CL_EstablishFlow.req

This message is sent by the application entity residing on the client associated with a node. This message contains the following parameters: flow destination MAC address, the flow classifiers, the flow TSpec and the bidirectional indication. In case the bidirectional indication is set, the following fields for the flow in the reverse direction shall be included in the message as well: the destination address, the Tspec and classifiers for the reverse direction. The format of the MMPL of the CL_EstablishFlow.req message shall be as shown in Table 8-21.

Table 8-21 – Format of the MMPL of the CL_EstablishFlow.req message

Field	Octet	Bits	Description
DA_F	0 to 5	[47:0]	Flow Destination MAC address. APDUs whose destination MAC address is specified in this field should be transmitted via this flow.
Classifiers	6 to (7+j)	See Table 8-22	This field shall contain traffic classifiers. APDUs whose destination MAC address is the specified MAC address and header conforms to the specified classifiers should be transmitted via this flow.
TSpec	✚Variable	See Table 8-24	Traffic specification for this flow may include the following fields: Traffic Priority, Maximum information Rate, Maximum Traffic Burst, committed information rate, Tolerated Jitter, Maximum Latency, Unsolicited Grant Interval, Unsolicited Polling Interval and APDU Size. N – The length of this field is variable according to the actual number of included traffic specification fields. The TSpec format is as specified in Table 8-24
Bidirectional	✚Variable	[7:0]	When set to 01 ₁₆ this field indicates that the flow is a bidirectional flow. When set to 00 ₁₆ this field indicates that the flow is a unidirectional flow.
DA_B	✚Variable	[47:0]	Destination MAC address for the established flow in the reverse direction (Note).
TSpec_B	✚Variable	See Table 8-24	Contains the TSpec of the flow in the reverse direction (Note).
Classifiers_B	✚Variable	See Table 8-22	Contains the traffic classifiers used to classify APDUs to be transmitted in the reverse direction (Note).
NOTE – These fields shall only exist in the message if bidirectional field is set to 01 ₁₆ .			

Table 8-22 – Format of classifiers structure

Field	Octet	Bits	Description
Length	0	[7:0]	Length of the list of classifiers (j) in bytes
Num	1	[7:0]	Number of classifiers (k) in the classifiers list
Classifier[0]	2 to (m+3)	See Table 8-23	First classifier in the list. Format of the classifiers is specified in Table 8-23. m+2 is the classifier length (Note).
...
Classifier[k-1]	✚Variable	See Table 8-23	Last classifier in the list (Note).
NOTE – More than one classifier can carry the same classifier type with different values. For example, Classifier[0] = "(", Classifier[1] = Address X, Classifier[2] = "AND", Classifier[3] = Destination Port 0, Classifier[4] =), Classifier[5] = "OR", Classifier[6] = "(", Classifier[7] = Address Y, Classifier[8] = "AND", Classifier[9] = Destination Port 1, Classifier[10] = ")" implies that any packet with (IP Address X and Destination Port 0) or (IP Address Y and Destination Port 1) belongs to the same flow.			

Table 8-23 – Format of classifier structure

Field	Octet	Bits	Description
Length	0	[7:0]	Length of the classifier parameter (m) in bytes
Classifier_typ	1	[7:0]	Type of classifier: 0: IP v4 Address (m = 4) 1: TOS (m = 1) 2: VLAN priority (m = 1, only the 3 LSBs are meaningful) 3: VLAN TAG (m = 4) 4: Destination Port (m = 2) 5: Source port (m = 2) 6: IP v6 Address (m = 16) 7: Generic Classifier: offset, length, value, where m = offset (2 bytes) + length (1 byte) + value (≤ 252 bytes) is a variable (Note 1) 8: EtherType (m=2) 9-254: Reserved by ITU-T (Note 2) 255: Operator (m = 1) (Note 3)
Classifier parameter	2 to (1+m)	[(m*8)-1:0]	Contains the classifier value, for example 32 bits of IP address. m is the length of the field in bytes and is a function of the Classifier_typ.

NOTE 1 – The offset is the number of bits from the beginning of the APDU where the classifier looks for a match within the APDU, the length is the classifier field size in bits to be matched, the value contains the value of the classifier field to be matched.

NOTE 2 – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.

NOTE 3 –The coding of the operator classifier parameter is as shown in Table 8-23.1.

Table 8-23.1 – Coding of operator classifier parameter

Parameter	Operator	Classifier description
0	(Open parenthesis
1	AND	Logical AND operator
2	OR	Logical OR operator
3)	Close parenthesis
4-255		Reserved by ITU-T

Table 8-24 – Format of TSpec field

Field	Octet	Bits	Description																						
Length	0	[7:0]	The length of the TSpec sub-fields following this field expressed in number of octets in the range between 2 and 255.																						
TSpecBitMask	1 & 2	[15:0]	<p>Traffic specifications bit mask. Each bit represents one traffic specification attribute field. When a represented bit value is set to one, the associated traffic specification attribute field shall be present in the TSpec field following this mask. When a represented bit value is set to zero, the associated traffic specification attribute field shall not be present. See clause 8.6.2.1 for the definition of these parameters. Traffic specification attribute fields that are present shall appear in the TSpec field in the following order:</p> <table border="1" data-bbox="742 701 1347 1205"> <thead> <tr> <th>Bit</th> <th>TSpec attribute</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Traffic Priority</td> </tr> <tr> <td>1</td> <td>Maximum Information Rate</td> </tr> <tr> <td>2</td> <td>Maximum Traffic Burst</td> </tr> <tr> <td>3</td> <td>Committed Information Rate</td> </tr> <tr> <td>4</td> <td>Tolerated Jitter</td> </tr> <tr> <td>5</td> <td>Maximum Latency</td> </tr> <tr> <td>6</td> <td>Grant Interval</td> </tr> <tr> <td>7</td> <td>Polling Interval</td> </tr> <tr> <td>8</td> <td>APDU Size</td> </tr> <tr> <td>9 to 15</td> <td>Reserved by ITU-T</td> </tr> </tbody> </table> <p>If bit 3 is set (CIR field is present), then bit 0 shall also be set (TrafficPriority is present).</p>	Bit	TSpec attribute	0	Traffic Priority	1	Maximum Information Rate	2	Maximum Traffic Burst	3	Committed Information Rate	4	Tolerated Jitter	5	Maximum Latency	6	Grant Interval	7	Polling Interval	8	APDU Size	9 to 15	Reserved by ITU-T
Bit	TSpec attribute																								
0	Traffic Priority																								
1	Maximum Information Rate																								
2	Maximum Traffic Burst																								
3	Committed Information Rate																								
4	Tolerated Jitter																								
5	Maximum Latency																								
6	Grant Interval																								
7	Polling Interval																								
8	APDU Size																								
9 to 15	Reserved by ITU-T																								
TrafficPriority	Variable	[7:0]	Specifies the traffic priority, represented as an 8-bit unsigned integer in the range from 0 to 7. The value 7 represents the highest priority. This field shall be present if CIR field is present. This field shall only be present if TSpecBitMask bit 0 is set to one.																						
MIR	Variable	[31:0]	Specifies the Maximum Information Rate in bit/s, represented as a 32-bit unsigned integer. This field shall only be present if TSpecBitMask bit 1 is set to one.																						
MaxTBurst	Variable	[15:0]	Specifies the Maximum Traffic Burst (see clause 8.6.2.1) in kbytes, represented as a 16-bit unsigned integer. This field shall only be present if TSpecBitMask bit 2 is set to one.																						
CIR	Variable	[31:0]	Specifies the Committed Information Rate (see clause 8.6.2.1) in bit/s, represented as a 32-bit unsigned integer. This field shall only be present if TSpecBitMask bit 3 is set to one.																						
ToleratedJitter	Variable	[7:0]	Specifies the Tolerated Jitter in ms, represented as an 8-bit unsigned integer. This field shall only be present if TSpecBitMask bit 4 is set to one.																						

Table 8-24 – Format of TSpec field

Field	Octet	Bits	Description
MaxLatency	Variable	[7:0]	Specifies the Maximum Latency in ms, represented as an 8-bit unsigned integer. This field shall only be present if TSpecBitMask bit 5 is set to one.
GrantInterval	Variable	[7:0]	Specifies Grant Interval in ms, represented as an 8-bit unsigned integer. This field shall only be present if TSpecBitMask bit 6 is set to one.
PollingInterval	Variable	[7:0]	Specifies the Polling Interval in ms, represented as an 8-bit unsigned integer. This field shall only be present if TSpecBitMask bit 7 is set to one.
APDU Size	Variable	[15:0]	APDU Size in bytes, represented as a 16-bit unsigned integer This field shall only be present if TSpecBitMask bit 6 (GrantInterval) and bit 8 are both set to one.

8.6.2.3.2 Format of CL_EstablishFlow.cnf

This message is sent by the node associated with a client to the application entity residing on the client, in response to a CL_EstablishFlow.req message. This message contains the status of the attempt to establish a flow. If successful, this message also contains the tuple (DeviceID, FlowID) that uniquely identifies the flow in the domain. If the status is a failure due to inability to meet the TSpec requirements in the CL_EstablishFlow.req message, then the rejected or wrong TSpec attributes shall be indicated by TSpecReject. In case the established flow is a bidirectional flow and the status is successful, this message shall also contain additional tuple (DeviceID, FlowID) with DeviceID corresponding to the endpoint node's DEVICE_ID, uniquely identifying the reverse flow in the domain. In case the request is for establishing a bidirectional flow and the status is failure due to the inability to establish the reverse flow, the StatusCode shall show the corresponding failure in establishing that flow.

The format of the MMPL of the CL_EstablishFlow.cnf message is shown in Table 8-25.

Table 8-25 – Format of the MMPL of the CL_EstablishFlow.cnf message

Field	Octet	Bits	Description
DeviceID	0	[7:0]	DEVICE_ID of the originating node.
FlowID	1	[7:0]	FLOW_ID assigned by the originating node.
StatusCode	2	[7:0]	Status of the request to establish a flow: 00 ₁₆ = Success. 01 ₁₆ = Failure – maximum number of flows already started by the node. 02 ₁₆ = Failure – error in TSpec passed in CL_EstablishFlow.req. 03 ₁₆ = Failure – insufficient capacity to admit the flow. 04 ₁₆ = Failure – failed to establish flow in reverse direction because maximum number of flows already started by the endpoint node.

Table 8-25 – Format of the MMPL of the CL_EstablishFlow.cnf message

Field	Octet	Bits	Description
			<p>05₁₆ = Failure – error in TSpec passed in CL_EstablishFlow.req for the flow in the reverse direction</p> <p>06₁₆ = Failure – insufficient capacity to start the flow in the reverse direction</p> <p>07₁₆ = Failure – classifier rule is not supported.</p> <p>08₁₆ – FF₁₆ = Reserved (Note 1).</p>
TSpecReject	3 & 4	[15:0]	<p>This field contains TSpec failure bit mask. In case StatusCode indicates failure, this field specifies which TSpec attributes are wrong or were rejected. Each bit represents one traffic specification attribute. When a represented bit value is set to one, the associated traffic specification field is wrong or could not be delivered.</p> <p>0: if bit 0 is set to one then Traffic Priority was rejected</p> <p>1: if bit 1 is set to one then Maximum Information Rate was rejected.</p> <p>2: if bit 2 is set to one then Maximum Traffic Burst was rejected.</p> <p>3: if bit 3 is set to one then Committed Information Rate was rejected.</p> <p>4: if bit 4 is set to one then Tolerated Jitter was rejected.</p> <p>5: if bit 5 is set to one then Maximum Latency was rejected.</p> <p>6: if bit 6 is set to one then Grant Interval was rejected.</p> <p>7: if bit 7 is set to one then Polling Interval was rejected.</p> <p>8: if bit 8 is set to one then APDU Size was rejected.</p> <p>9-15: reserved by ITU-T.</p> <p>(Note 1)</p>
Bidirectional	5	[7:0]	Set to 01 ₁₆ if bidirectional flow establishment was requested in CL_EstablishFlow.req
DeviceID_B	6	[7:0]	DEVICE_ID of the Endpoint node (Note 2)
FlowID_B	7	[7:0]	FLOW_ID assigned by the endpoint node in case of a bidirectional flow. In case it is a unidirectional flow this field shall contain zero (Note 2)
<p>NOTE 1 – If StatusCode is lower than 02₁₆ then the TSpecReject field shall be ignored.</p> <p>NOTE 2 – If Bidirectional field is set to zero these fields shall not appear in the message.</p>			

8.6.2.3.3 Format of CL_TerminateFlow.req

This message is sent by the application entity residing on the client to the originating node to signal that the specified flow shall be terminated. This message contains the tuple (DeviceID, FlowID) that uniquely identifies the flow in the domain. When CL_TerminateFlow.req message specifies that a bidirectional service flow is to be terminated, the endpoint node shall respond to the request to terminate the forward flow and also terminate the reverse flow.

The format of the MMPL of the CL_TerminateFlow.req message shall be as shown in Table 8-26.

Table 8-26 – Format of the MMPL of the CL_TerminateFlow.req message

Field	Octet	Bits	Description
DeviceID	0	[7:0]	DEVICE_ID of the originating node
FlowID	1	[7:0]	FLOW_ID assigned by the originating node

8.6.2.3.4 Format of CL_TerminateFlow.cnf

This message is sent by the originating node to the application entity residing on the client after the specified flow has been terminated. This message contains the tuple (DeviceID, FlowID) that uniquely identifies the flow in the domain and includes a ReasonCode explaining why the flow was terminated.

The format of the MMPL of the CL_TerminateFlow.cnf message shall be as shown in Table 8-27.

Table 8-27 – Format of the MMPL of the CL_TerminateFlow.cnf message

Field	Octet	Bits	Description
DeviceID	0	[7:0]	DEVICE_ID of the originating node
FlowID	1	[7:0]	FLOW_ID assigned by the originating node
ReasonCode	2	[7:0]	Reason why the flow was terminated: <ul style="list-style-type: none"> • 00₁₆ = Normal termination in response to CL_TerminateFlow.req • 01₁₆-FF₁₆ = Reserved

8.6.2.3.5 Format of CL_FlowTerminated.ind

This unsolicited message is sent by the originating node to the application entity residing on the client after the specified flow has been terminated. This message contains the tuple (DeviceID, FlowID) that uniquely identifies the flow in the domain and includes a ReasonCode explaining why the flow was terminated.

The format of the MMPL of the CL_FlowTerminated.ind message shall be as shown in Table 8-28.

Table 8-28 – Format of the MMPL of the CL_FlowTerminated.ind message

Field	Octet	Bits	Description
DeviceID	0	[7:0]	DEVICE_ID of the originating node
FlowID	1	[7:0]	FLOW_ID assigned by the originating node
ReasonCode	2	[7:0]	Reason why the flow was terminated: <ul style="list-style-type: none"> • 00₁₆ = Originating node has inferred that the traffic flow has ended • 01₁₆ = Terminated at request of the domain master. • 02₁₆-FF₁₆ = Reserved

8.6.2.3.6 Format of FL_OriginateFlow.req

This message is sent by a node that needs to originate a flow to a selected endpoint. This message contains the TSpec that is used by the originating and endpoint nodes and the FLOW_ID allocated by the originating node. In case the flow is bidirectional, the message shall also contain the TSpec and classifiers for the flow in the reverse direction. In case the endpoint node is hidden, the tunnel field shall be set and the endpoint DEVICE_ID shall be included in the message together with the route list. The route list shall include the list of relay nodes toward the endpoint.

The format of the MMPL of the FL_OriginateFlow.req message shall be as shown in Table 8-29.

Table 8-29 – Format of the MMPL of the FL_OriginateFlow.req message

Field	Octet	Bits	Description
DeviceID	0	[7:0]	DEVICE_ID of the originating node.
FlowID	1	[7:0]	FLOW_ID assigned by the originating node.
TSpec	2 to (N+1)	[8*N-1:0]	See Table 8-24.
Bidirectional	Variable	[7:0]	When set to 01 ₁₆ it indicates that the flow is a bidirectional flow. When set to 00 ₁₆ it indicates that the flow is a unidirectional flow.
DA_B	Variable	[47:0]	Destination MAC address for the established flow in the reverse direction (Note).
TSpec_B	Variable	See Table 8-24	It contains the TSpec of the flow in the reverse direction (Note).
Classifiers_B	Variable	See Table 8-22	Classifiers to classify APDUs to be transmitted via the flow in the reverse direction (Note).
Tunnel	Variable	[7:0]	00 ₁₆ – Direct flow establishment 01 ₁₆ – Flow via relays establishment
EndPoint	Variable	[7:0]	DEVICE_ID of the endpoint hidden node
RouteList	Variable	See Table 8-30	Routing list toward the destination endpoint
NOTE – These fields shall exist in the message only if Bidirectional field is set to 01 ₁₆ .			

Table 8-30 – Format of RouteList

Field	Octet	Bits	Description
NumRelays	0	[7:0]	Number of relay nodes (n) in the RouteList
RelayNode	1	[7:0]	DEVICE_ID of the first relay node in the list
....
RelayNode	n	[7:0]	DEVICE_ID of the last relay node in the list

8.6.2.3.7 Format of FL_OriginateFlow.cnf

This message is sent by the endpoint node to the node that is attempting to originate a new flow. This message contains the status of the attempt to originate the flow and the FLOW_ID previously provided in message FL_OriginateFlow.req that allows the originator and the endpoint to coordinate flow set-up requests. In case the flow is a bidirectional flow, the message shall also contain the FLOW_ID of the reverse flow assigned by the endpoint node. In case tunnel flow is requested, then the confirmation contains the list of flow IDs from the endpoint until the originating node.

The format of the MMPL of the FL_OriginateFlow.cnf message shall be as shown in Table 8-31.

Table 8-31 – Format of the MMPL of the FL_OriginateFlow.cnf message

Field	Octet	Bits	Description
DeviceID	0	[7:0]	DEVICE_ID of the originating node.
FlowID	1	[7:0]	FLOW_ID assigned by the originating node.
StatusCode	2	[7:0]	Status of the request to establish a flow: <ul style="list-style-type: none"> • 00₁₆ = Success. • 01₁₆ = Failure – Maximum number of flows already started by the endpoint node. • 02₁₆ = Failure – Error in TSpec passed in FL_OriginateFlow.req. • 03₁₆ = Failure – Insufficient resources • 04₁₆ = Failure – Failed to establish flow in reverse direction because maximum number of flows already started by the endpoint node. • 05₁₆ = Failure – Error in TSpec passed in CL_EstablishFlow.req for the bidirectional flow other direction • 06₁₆ = Failure – Insufficient capacity to start the flow in the reverse direction • 07₁₆ – FF₁₆ = Reserved.
Bidirectional	3	[7:0]	When set to 01 ₁₆ it indicates that the flow is a bidirectional flow. When set to 00 ₁₆ it indicates that the flow is a unidirectional flow.
FlowID_B	4	[7:0]	FLOW_ID assigned by the endpoint node in case of bidirectional flow. This field shall be present in the message only if Bidirectional field is set to 01 ₁₆ .
Tunnel	5	[7:0]	00 ₁₆ – Direct flow establishment is confirmed. 01 ₁₆ – Tunnel flow establishment is confirmed.
NumFlowIDs	6	[7:0]	Number of flows IDs (n) in the list.
FlowID	7	[7:0]	First flow ID in the list. It is the flow ID of the last hop assigned by the last relay node toward the endpoint.
...
FlowID	7 + n-1	[7:0]	Last flow ID in the list. It is the flow ID of the first relay note in the route toward the endpoint.

8.6.2.3.8 Format of FL_AdmitFlow.req

This message is sent by the originating node to the domain master, for flow admission, to establish a traffic contract. This message contains the TSpec, the actual PHY data rate and a FLOW_ID. The actual PHY data rate enables the domain master to allocate the estimated number of symbols needed to serve the flow transmission according to number of bytes needed to be transmitted and the actual PHY data rate. In case of bidirectional flow, the message shall include the TSpec of the flow in the reverse direction and its actual PHY data rate. In case the tunnel field is set to 01₁₆ then the hidden endpoint shall be also included in the message.

The format of the MMPL of the FL_AdmitFlow.req message shall be as shown in Table 8-32.

Table 8-32 – Format of the MMPL of the FL_AdmitFlow.req message

Field	Octet	Bits	Description
DeviceID	0	[7:0]	DEVICE_ID of the originating node.
FlowID	1	[7:0]	FLOW_ID assigned by the originating node.
TSpec	Variable	See Table 8-24	See Table 8-24.
TX Rate	Variable	See Table 8-33	The actual PHY data rate used by the transmitter, specified in bits per second for each channel estimation window, based on the bit loading per symbol, the symbol time, the FEC rate, and the number of repetitions. The format of the TX rate field is described in Table 8-33. The offset of this field depends on the actual length of the previous (TSpec) field. Note that the TX Rate should be specified per each channel estimation window.
Bidirectional	Variable	[7:0]	Set to 01 ₁₆ in case the established flow to be admitted is a bidirectional flow.
DeviceID_B	Variable	[7:0]	DEVICE_ID of the endpoint node (Note).
TSpec_B	Variable	See Table 8-24	The TSpec of the flow in reverse direction (Note).
TX Rate_B	Variable	See Table 8-33	TX Rate for the reverse direction (Note).
Tunnel	Variable	[7:0]	00 ₁₆ – direct flow admission is requested 01 ₁₆ – tunnel flow admission is requested
EndPoint	Variable	[7:0]	DEVICE_ID of the endpoint node.
NOTE – These fields appear only if Bidirectional field is set to 01 ₁₆ .			

Table 8-33 – Format of the TX rate field

Field	Octet	Bits	Description
NumCEWindows	0	[4:0]	Number of items in the following list. Each item contains information for one channel estimation window. Each item includes three fields: CE_STime, CE_ETime and BitsPerSecond. The list shall not exceed n=32 items.
EstimOverhead		[7:5]	Estimated DLL overhead in percentage represented as an unsigned integer minus 1 (Note 1). A value of zero represents 1% overhead. A value of 7 represents ≥ 8 % overhead.
CE_STime	1	[7:0]	Start time as specified in Table 8-98 for first channel estimation window.
CE_ETime	2	[7:0]	End time as specified in Table 8-99 for first channel estimation window.
BitsPerSecond	3 and 4	[15:0]	The PHY data rate in bits per second for the first channel estimation window in steps of 32 kbit/s (Note 2).
CE_STime	4n-3	[7:0]	Start time as specified in Table 8-98 for last channel estimation window.

CE_ETime	4n-2	[7:0]	End time as specified in Table 8-99 for last channel estimation window.
BitsPerSecond	4n-1 to 4n	[15:0]	The PHY data rate in bits per second for the last channel estimation window in steps of 32 kbit/s (Note 2).
<u>XputMultiplierFactor</u> <u>or</u>	<u>4n+1</u>	<u>[2:0]</u>	<u>Multiplier factor (F) to be used to retrieve the throughput information for a channel estimation window.</u> <u>The final data rate for a channel estimation window is computed by multiplying the value of BitsPerSecond field by the factor (F+1).</u> <u>The field is coded as an unsigned integer.</u> <u>This field is only applicable to transmissions using a Profile 2 OFB.</u>
<u>OFB_ID</u>		<u>[5:3]</u>	<u>OFB used to estimate the BitsPerSecond field</u> <u>See clause 7.1.2.3.2.1.9 /10.1.2.3.2.1.9 of [ITU-T G.9960]</u>
<u>Reserved</u>		<u>[7:6]</u>	<u>Reserved by ITU-T</u>
<p>NOTE 1 – Defined as (Number of bytes crossing the PMI – number of bytes crossing the A-interface)/Number of bytes crossing the A-interface * 100% associated with a flow, including retransmission. The estimation of this parameter shall be vendor discretionary.</p> <p>NOTE 2 – $\text{BitsPerSecond} = (\text{floor}(k_P/N_{REP}) \cdot R \cdot F_{SC}) / (1 + N_{GI}/N)$ where k_P is the number of loaded bits (see clause 7.1.3.3.1/10.1.3.3.1 of [ITU-T G.9960]), N_{REP} is the number of repetitions (see clause 7.1.3.3.1), R is the code rate (see clause 7.1.3.2/10.1.3.2 of [ITU-T G.9960]), F_{SC} is the sub-carrier spacing, N_{GI} is the guard interval, and N is the number of sub-carriers (see clause 7.1.4.6/10.1.4.6 of [ITU-T G.9960]) for a payload OFDM symbol transmitted over a specified channel estimation window.</p>			

8.6.2.3.9 Format of FL_AdmitFlow.cnf

This message is sent by the domain master to the originating node after it has assessed whether a traffic contract can be provided for a new flow (i.e., whether the flow can be supported by allocating sufficient resources). This message contains the status of the attempt to admit the new flow and the FLOW_ID. In case of bidirectional flow, the status may include a failure code for the flow in the reverse direction. If the FL_AdmitFlow.req contains the Tunnel field set to 01₁₆, the FL_AdmitFlow.cnf shall contain this field set to 01₁₆ as well, and include a list with the relay nodes toward the endpoint hidden node.

The format of the MMPL of the FL_AdmitFlow.cnf message shall be as shown in Table 8-34.

Table 8-34 – Format of the MMPL of the FL_AdmitFlow.cnf message

Field	Octet	Bits	Description
DeviceID	0	[7:0]	DEVICE_ID of the originating node.
FlowID	1	[7:0]	FLOW_ID assigned by the originating node.
StatusCode	2	[7:0]	Status of the request to establish a flow: <ul style="list-style-type: none"> • 00₁₆ = Success. • 01₁₆ = Failure – maximum number of flows already started by originating node. • 02₁₆ = Failure – error in TSpec passed in FL_AdmitFlow.req. • 03₁₆ = Failure – insufficient capacity to admit the flow given the TSpec passed in FL_AdmitFlow.req.

Table 8-44.9 – Format of the MMPL of the FL_RenewTunnel.cnf message

Field	Octet	Bits	Description
			<ul style="list-style-type: none"> • 06₁₆ = Failure – insufficient capacity to start the flow in the reverse direction • 07₁₆ – FF₁₆ = Reserved.
NumRelays	3	[7:0]	Number of relays in the reconstructed tunnel (n). If StatusCode is not success this field shall contain zero.
RelDeviceID[0]	4	[7:0]	DEVICE_ID of the first relay node (Note 1).
RelFlowID[0]	5	[7:0]	FLOW_ID assigned by the first relay node (Note 1).
....			
RelDeviceID[n-1]	4+ $2 \times (n-1)$ Variable	[7:0]	DEVICE_ID of the nth relay node in the reconstructed tunnel (Note 1).
RelFlowID[n-1]	5+ $2 \times (n-1)$ Variable	[7:0]	FLOW_ID assigned by nth relay node in the reconstructed tunnel (Note 1).
Bidirectional	6+ $2 \times (n-1)$ Variable	[7:0]	When set to 01 ₁₆ it indicates that the flow is bidirectional flow. When set to 00 ₁₆ it indicates that the flow is unidirectional flow.
RelBFlowID[0]	1+ 6+ $2 \times (n-1)$ Variable	[7:0]	FLOW_ID of the inverse bidirectional flow assigned by the endpoint node (Notes 1 and 2).
RelBFlowID[1]	2+ 6+ $2 \times (n-1)$ Variable	[7:0]	FLOW_ID of the inverse bidirectional flow assigned by the last relay node (Notes 1 and 2).
...			
RelBFlowID[n]	n+6+ $2 \times (n-1)$ Variable	[7:0]	FLOW_ID of the inverse bidirectional flow assigned by the first relay node when the flow is a bidirectional flow (Notes 1 and 2).
DeviceID	7Variable	[7:0]	DEVICE_ID of the node that could not establish a flow (Note 3).
<p>NOTE 1 – This field should exist in the message only if StatusCode is success.</p> <p>NOTE 2 – This field should exist in the message only if the Bidirectional field contains 01₁₆.</p> <p>NOTE 3 – This field shall exist in the message only if StatusCode is not success. The failure status corresponding to this node is communicated in the StatusCode field.</p>			

8.6.2.3.27 Format of the FL_DeleteFlow.req message

This message is sent by originating node to delete a not valid flow of a relay node that is removed from a tunnel. The format of the MMPL of the FL_DeleteFlow.req message shall be as shown in Table 8-44.10.

Table 8-44.10 – Format of the MMPL of the FL_DeleteFlow.req message

Field	Octet	Bits	Description
DeviceID	0	[7:0]	DEVICE_ID of the originating node.
FlowID	1	[7:0]	FLOW_ID assigned by the originating node that identifies the reconstructed tunnel.
RelDeviceID	2	[7:0]	DEVICE_ID of the relay node.
RelFlowID	3	[7:0]	FLOW_ID assigned by the relay node that should be deleted.

8.6.2.3.28 Format of the FL_DeleteFlow.cnf message

This message is sent by a relay node to confirm receiving the FL_DeleteFlow.req message. The format of the MMPL of the FL_DeleteFlow.cnf message shall be as shown in Table 8-44.11.

Table 8-44.11 – Format of the MMPL of the FL_DeleteFlow.cnf message

Field	Octet	Bits	Description
DeviceID	0	[7:0]	DEVICE_ID of the originating node.
FlowID	1	[7:0]	FLOW_ID assigned by the originating node that identifies the reconstructed tunnel.
RelDeviceID	2	[7:0]	DEVICE_ID of the relay node that deleted the flow.
RelFlowID	3	[7:0]	FLOW_ID of the deleted flow.

8.6.2.3.29 Format of CL_TerminateFlow.ind

This message is sent by the originating node to the application entity residing on the client when it determines that the application entity has finished sending data associated with a flow.

The format of the MMPL of the CL_TerminateFlow.ind message shall be as shown in Table 8-44.12.

Table 8-44.12 – Format of the MMPL of the CL_TerminateFlow.ind message

Field	Octet	Bits	Description
DeviceID	0	[7:0]	DEVICE_ID of the originating node.
FlowID	1	[7:0]	FLOW_ID assigned by the originating node.

8.6.2.3.30 Format of FL_TerminateTunnel.req

The message is sent to request a tunnel termination.

The format of the MMPL of the FL_TerminateTunnel.req message shall be as shown in Table 8-44.13.

Table 8-44.13 – Format of the MMPL of the FL_TerminateTunnel.req message

Field	Octet	Bits	Description
TerminatorID	0	[7:0]	DEVICE_ID of the originating node or of the domain master that triggers the tunnel termination.
DeviceID	1	[7:0]	DEVICE_ID of the originating node.
FlowID	2	[7:0]	FLOW_ID of the tunnel to be terminated.
ReasonCode	3	[7:0]	Reason why the tunnel is being terminated: <ul style="list-style-type: none"> • 00₁₆ = Normal termination initiated by the originating node. • 01₁₆ = Termination by domain master due to lack of resources. • 02₁₆ = Termination by the domain master because the originating node is disconnected. • 03₁₆ – FF₁₆ = Reserved.

8.6.2.3.31 Format of FL_TerminateTunnel.cnf

This message shall be sent as a reply to a received FL_TerminateTunnel.req message. This message contains the flow's identity.

The format of the MMPL of the FL_TerminateTunnel.cnf message shall be as shown in Table 8-44.14.

Table 8-44.14 – Format of the MMPL of the FL_TerminateTunnel.cnf message

Field	Octet	Bits	Description
DeviceID	0	[7:0]	DEVICE_ID of the node originating the tunnel.
FlowID	1	[7:0]	FLOW_ID of the terminated tunnel.
Status	2	[7:0]	Status of termination. <ul style="list-style-type: none"> • 00₁₆ = Termination completed successfully. • 01₁₆ = Tunnel termination failed due to a broken link. • 02₁₆ = Flow unknown. • 03₁₆ – FF₁₆ = Reserved.

8.6.2.4 Bandwidth update protocol for prioritized connections

This clause defines the mechanism used by the nodes to inform the domain master on the actual status of its connection queues. This protocol is supported by management messages described in clause 8.6.2.4.1.

A node should follow the bandwidth update protocol when a node does not have direct visibility with the domain master (the BRURQ field in the PHY-frame header of transmitted PHY frames is not received by the domain master).

In addition, any node should use this protocol when:

- The domain master does not allocate in the MAP enough TXOPs allowing to transmit user priorities queued in the node;
- The node wants to inform the domain master on the status of a particular connection.

The domain master may take this information into account when assigning resources for a given connection. As prioritized connections are not QoS guaranteed, the domain master may change bandwidth allocations on its own discretion.

To inform the DM about the necessity to allocate bandwidth for prioritized connections a node shall send a BU_BWUpdate.req message to its domain master including the information about the reported connection status (user priority in the connection queue, bandwidth request update).

Upon reception of BU_BWUpdate.req, the domain master shall send to the reporting node a BU_BWUpdate.cnf message acknowledging the reception of the information. If a node does not receive the BU_BWUpdate.cnf message within 200 ms, it may repeat the report.

If the request is acknowledged by the DM, the node shall refrain from reporting a new bandwidth update for at least the next 1 second.

NOTE – A node should report the status of its queues when traffic conditions change but taking care not to flood the DM with report messages. If a node has enough resources with the current allocation by the domain master, it should only send a report when its traffic requirements change.

8.6.2.4.1 Bandwidth update protocol messages

The following sub-clauses specify the messages that are needed to support the bandwidth update protocol.

8.6.2.4.1.1 Format of BU_BWUpdate.req

This message is sent by the reporting node to the Domain Master and contains the user priorities, the bandwidth request update for the given connection by means of BRURQ indication, and PHY data rate used by the transmitter.

The format of the MMPL of the BU_BWUpdate.req shall be as shown in Table 8-44.15.

Table 8-44.15 – Format of the MMPL of the BU_BWUpdate.req message

Field	Octet	Bits	Description
DeviceID	0	[7:0]	DEVICE_ID of the originating node.
NumDID	1	[7:0]	Number of the destination nodes (N) of connections originated by the originating node and reported in this message.
DID[0]	2	[7:0]	DEVICE_ID of the first destination node of connections originated by the originating node.
TxRate[0]	Variable	See Table 8-33	The actual PHY data rate used by the originating node transmitter to communicate with the first destination node, specified in bits per second for each channel estimation window, based on the bit loading per symbol, the symbol time, the FEC rate and the number of repetitions. The format of the TX rate field is described in Table 8-33. Note that the TX Rate should be specified per each channel estimation window.
PRIQUEUE_info[0]	Variable	See Table 8-44.16	Information of the priority queue related to the bandwidth update request for the connection between the originating node and the first destination node.
...

Table 8-44.15 – Format of the MMPL of the BU_BWUpdate.req message

Field	Octet	Bits	Description
DID[N-1]	Variable	[7:0]	DEVICE_ID of the last (Nth) destination node of connections originated by the originating node.
TxRate[N-1]	Variable	See Table 8-33	The actual PHY data rate used by the originating node transmitter to communicate with the last (Nth) destination node, specified in bits per second for each channel estimation window, based on the bit loading per symbol, the symbol time, the FEC rate and the number of repetitions. The format of the TX rate field is described in Table 8-33. Note that the TX Rate should be specified per each channel estimation window.
PRIQUEUE_info[N-1]	Variable	See Table 8-44.16	Information of the priority queue related to the bandwidth update request for the connection between the originating node and the last (Nth) destination node.
AdditionalTxRateInfo[0]	Variable	[7:0]	Additional information to calculate the actual PHY data rate for the first destination node. This field is only valid for Profile 2 OFBs. This field shall be set to zero for Profile 1 OFBs. See Table 8-44.16.1
...
AdditionalTxRateInfo[N-1]		[7:0]	Additional information to calculate the actual PHY data rate for the last (Nth) destination node. This field is only valid for Profile 2 OFBs. This field shall be set to zero for Profile 1 OFBs. See Table 8-44.16.1

Table 8-44.16 – Format of the PRIQUEUE_info field of the BU_BWUpdate.req message

Field	Octet	Bits	Description
NumQueue	0	[7:0]	Number of priority queues reported for destination node i.
PRIORITY_QUEUE[0]	1	[7:0]	Priority queue with the lowest priority corresponding to the BRURQ we are reporting.
Priorities[0]	2	[7:0]	User priority bit mask. Each bit represents one user priority. Bit 0 represents user priority 0 and bit 7 user priority 7. Each bit signals the presence of at least one APDU with this user priority in the selected connection
BRURQ[0]	3 and 4	[15:0]	See in clause 7.1.2.3.2.2.19/ 10.1.2.3.2.2.19 of [ITU-T G.9960] (Note).
...
PRIORITY_QUEUE[k-1]	Variable	[7:0]	Priority queue with the highest priority corresponding to the BRURQ we are reporting.
Priorities[k-1]	Variable	[7:0]	User priority bit mask. Each bit represents one user priority. Bit 0 represents user priority 0 and bit 7 user priority 7. Each bit signals the presence of at least one APDU with this user priority in the selected connection

Table 8-44.15 – Format of the MMPL of the BU_BWUpdate.req message

Field	Octet	Bits	Description
BRURQ[k-1]	Variable	[15:0]	See in clause 7.1.2.3.2.2.19/ 10.1.2.3.2.2.19 of [ITU-T G.9960] (Note).
NOTE – The reporting node may set this field to zero in order to indicate that the DM may release the resources allocated for the connection that is being reported.			

Table 8-44.16.1 – Format of the Additional TxRate information field of the BU_BWUpdate.req message

Field	Octet	Bits	Description
XputMultiplierFactor	0	[2:0]	Multiplier factor (F) to be used to retrieve the throughput information. The final TxRate value is computed by multiplying the value of TxRate field by (F+1). The field is coded as an unsigned integer.
OFB_ID		[5:3]	OFB used to estimate the TxRate See clause 7.1.2.3.2.1.9/10.1.2.3.2.1.9 of [ITU-T G.9960]
Reserved		[7:6]	Reserved by ITU-T (Note)
NOTE – Fields or bits reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

8.6.2.4.1.2 Format of BU_BWUpdate.cnf

This message is sent by the Domain Master to the reporting node after it has assessed whether the bandwidth allocation for the node can be provided.

The format of the MMPL of the BU_BWUpdate.cnf shall be as shown in Table 8-44.17.

Table 8-44.17 – Format of the MMPL of the BU_BWUpdate.cnf message

Field	Octet	Bits	Description
DeviceID	0	[7:0]	DEVICE_ID of the reporting node.
NumDID	1	[7:0]	Number of the destination nodes (N) of connections originated by the originating node and reported in this message.
DID[0]	2	[7:0]	DEVICE_ID of the first destination node of connections originated by the originating node.
...
DID[0]	2	[7:0]	DEVICE_ID of the first destination node of connections originated by the originating node.
Status_info[0]	Variable	See Table 8.44.18	Status of the request for bandwidth update for connections associated with DID[0].
...

Table 8-44.17 – Format of the MMPL of the BU_BWUpdate.cnf message

Field	Octet	Bits	Description
DID[N-1]	Variable	[7:0]	DEVICE_ID of the last destination node of connections originated by the originating node.
Status_info[N-1]	Variable	See Table 8.44.18	Status of the request for bandwidth update for connections associated with DID[N-1].

Table 8-44.18 – Format of the Status_info field of the BU_BWUpdate.cnf message

Field	Octet	Bits	Description
NumQueue	0	[7:0]	Number of connection queues (k) associated with destination node i.
PRIORITY_QUEUE[0]	1	[7:0]	The first priority queue associated with destination node i specified by the reporting node in the BU_BWUpdate.req message
StatusCode[0]	2	[7:0]	Status of the request for bandwidth allocation: <ul style="list-style-type: none"> • 00₁₆ = Success. • 01₁₆ = Failure – Insufficient resources. • 02₁₆ = Failure – Insufficient resources for the requested priority • 03₁₆ – FF₁₆ = Reserved.
...
PRIORITY_QUEUE[k-1]	Variable	[7:0]	The last priority queue associated with destination node i specified by the reporting node in the BU_BWUpdate.req message
StatusCode[k-1]	Variable	[7:0]	Status of the request for bandwidth allocation: <ul style="list-style-type: none"> • 00₁₆ = Success. • 01₁₆ = Failure – Insufficient resources. • 02₁₆ = Failure – Insufficient resources for the requested priority • 03₁₆ – FF₁₆ = Reserved.

8.6.3 Synchronization to an external source

Synchronization of the MAC cycle to an external source by the domain master is optional when operating over phone line or coax media. Over power-line medium connected to mains supply, the AC cycle shall be considered as the external source.

When operating over phone line or coax media and the domain master is synchronized with a period of an external source, the starting point of the MAC cycle shall be aligned with the period of the external source with a maximum tolerance of \pm EXT_SYNC_ACCURACY and the period of the MAC cycle shall be NUM_SYNC_PERIODS times the period of the external source. Specific values shall be chosen for EXT_SYNC_ACCURACY and NUM_SYNC_PERIODS depending on the type of the external source.

In the example in Figure 8-39, the start of the MAC cycle is shown to be within the boundaries of \pm EXT_SYNC_ACCURACY around the arrival of the external synchronization signal. The parameter NUM_SYNC_PERIODS is set to three so MAC cycle starts at every third appearance of the external synchronizing source.

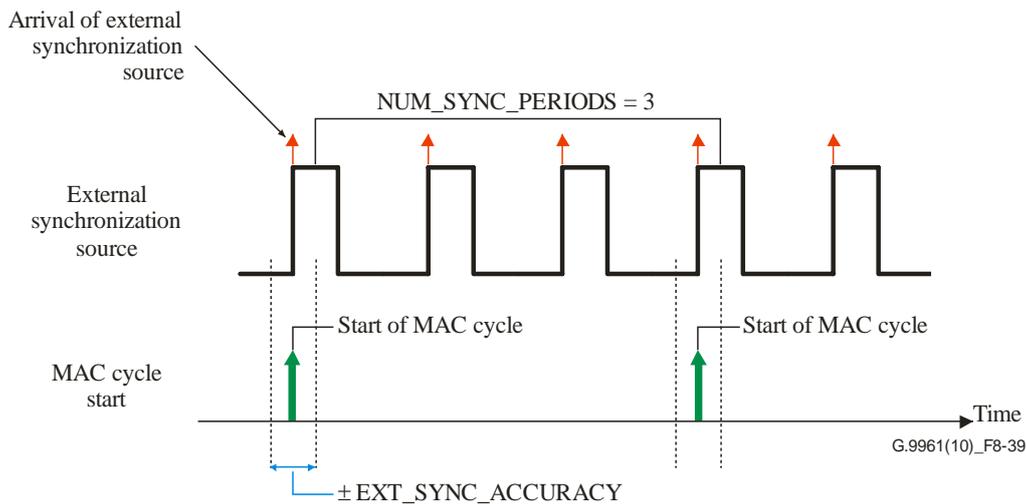


Figure 8-39 – Example of synchronization of MAC cycle to a periodic external source

8.6.3.1 AC line cycle synchronization

When operating over a public utility supplied AC power-line medium with a nominal cycle frequency of 50 hertz or 60 hertz, the domain master shall synchronize the MAC cycle to the power-line cycle and the NUM_SYNC_PERIODS shall be equal to two AC cycles. The start of the MAC cycle shall be at a constant angular offset Δ (see clause 8.14.3) from the AC cycle zero-crossing point with a maximum tolerance of $\text{EXT_SYNC_ACCURACY} = 100 \mu\text{s}$.

Variances at power generators and through the power distribution system and the presence of noise on the line cause the actual power-line frequency supplied to a node to jitter, compared to the nominal 50 or 60 Hz line frequency. In order to cope with the jitter in the zero-crossing point, the zero-crossing detector should include a smoothing function.

NOTE – Synchronization to the AC line cycle can be achieved by having the domain master track a particular point (shown as Δ in Figure 8-40).

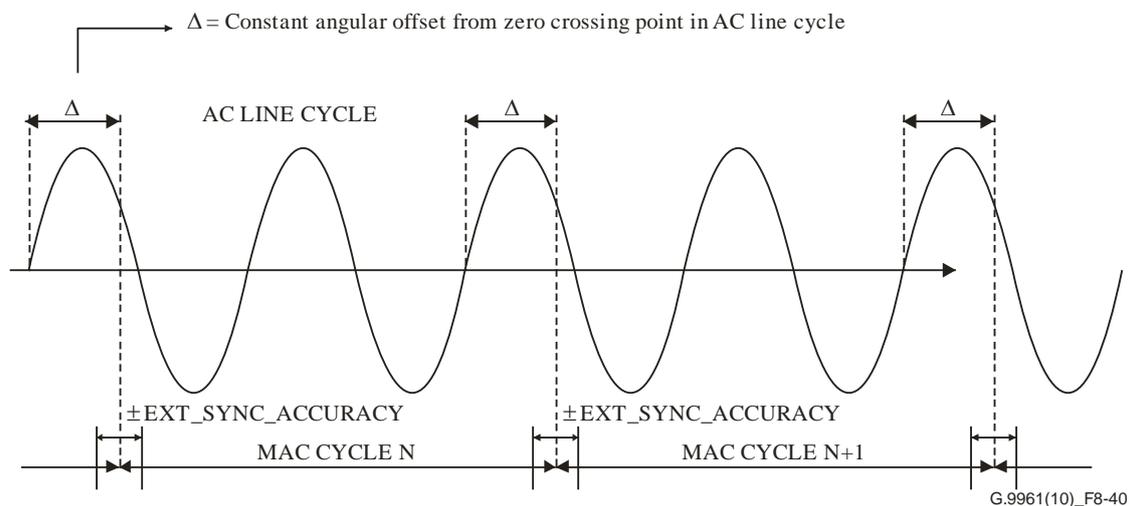


Figure 8-40 – Relationship between the AC line cycle and the MAC cycle

8.6.4 Routing and topology management

This Recommendation describes two modes for routing and topology management:

- 1) Centralized routing and topology management (CRTM) mode – All the nodes in the domain transmit their topology information to the domain master as described in clause 8.6.4.2.1.1. The domain master calculates the routing table for all nodes of the domain and sends the

calculated routing table and topology information to all nodes in the domain as described in clause 8.6.4.1.1.2. The nodes in the domain use the routing table received from the domain master.

- 2) Distributed routing and topology management (DRTM) mode – Each node calculates its own routing table using the standard algorithm indicated in the MAP (see Table 8-62) by the domain master. The mechanism used by each node to distribute its topology information to all other nodes in the domain is for further study.

A domain may operate in CRTM or in the DRTM mode. The domain master shall advertise the particular mode in the MAP (see clause 8.8.3). All nodes admitted to the domain shall use the routing and topology management mode indicated by the domain master.

A node shall indicate its capability to support the DRTM mode during the registration process (see clause 8.6.1.1.4.1) and in the topology update message (see Table 8-47). The domain master shall use CRTM mode if not all nodes in the domain are capable of calculating routing tables.

The domain master and the backup domain master shall be capable of gathering and holding information describing the network topology. This information is collected from TM_NodeTopologyChange.ind and TM_NodeTopologyChange.cnf messages, sent by nodes in the domain including nodes that are hidden from the domain master or the backup domain master. AAT information from other network domains is collected from nodes of the domain that are associated with IDBs to the corresponding domains.

Nodes shall transmit topology messages in accordance with the topology update interval (see clause 8.6.4.2) and upon particular events and management procedures in the domain that change or may potentially change the topology.

The domain master and the backup domain master may also request topology information from one or more nodes using a topology information request message TM_NodeTopologyChange.req. The domain master and the backup domain master shall specify which fields of the topology information are requested. A node receiving the request shall report the topology information to the requesting node according to the specified report request field (ReqRep) using message TM_NodeTopologyChange.cnf.

In DRTM mode the backup domain master shall maintain full topological map of the domain using the information collected from the TM_NodeTopologyChange.ind messages, sent by the nodes in the domain.

In CRTM mode, the domain master shall transmit to the backup domain master all the information from the received TM_NodeTopologyChange.ind messages associated with any newly added nodes or lost nodes from the visibility list of any endpoint node via TM_DMBBackup.ind messages. The backup domain master shall then update its routing tables, to be used in case it becomes the domain master.

8.6.4.1 Domain master operation for routing and topology management

8.6.4.1.1 Domain master operation in CRTM mode

The domain master shall update its topology information whenever any of the following events that change the domain's topology occurs:

- it receives a message TM_NodeTopologyChange.ind from one of the nodes;
- when a node joins the domain, or leaves the domain (i.e., a node resigns from the domain, or the domain master expels a node from the domain);
- the domain master detects that a node has not re-registered (i.e., may be possibly turned off or has failed) or a node failed the re-authentication.

NOTE – Inactivity schedule of nodes is advertised in the MAP and is not a part of topology information.

The domain master shall broadcast updates of its topology information by transmitting a message TM_DomainRoutingChange.ind within T_{DM_UPDATE} ms after it receives any update.

Multiple changes to the domain's topology may be included within a single TM_DomainRoutingChange.ind message.

The domain master shall periodically update the topology information communicated to all the nodes by sending a TM_DomainRoutingChange.ind message with full topology information. The periodicity of these updates is at the discretion of the domain master but shall be between 0.5 and 30 seconds.

In case a new registered node sends its first TM_NodeTopologyChange.ind to the DM, the DM shall send a full topology report through the TM_DomainRoutingChange.ind acknowledging the TM_NodeTopologyChange.ind from the joining node. In case of an admission procedure via proxy, this TM_DomainRoutingChange.ind shall also assign the proxy used for registration as MPR.

In case the update of the topology received by the domain master requires update of the routing table, the domain master shall calculate a new routing table (see clause 8.6.4.1.1.1) and a new BRT (see clause 8.6.4.1.1.3), and send those to all nodes in the domain using TM_DomainRoutingChange.ind message (see clause 8.6.4.1.1.2). The algorithm used by the domain master to compute the table is vendor discretionary. The domain master shall indicate the sequence number of the last transmitted TM_DomainRoutingChange.ind message in each transmitted MAP.

Nodes that conclude, according to the RoutingSequenceNumber received in a MAP, that they did not receive the last update of the routing table shall request it by sending TM_ReturnDomainRouting.req message.

NOTE – Nodes should take into account that, due to delivery delays, the TM_DomainRoutingChange.ind message with an updated sequence number may be received after the MAP with the same updated sequence number.

If the RoutingSequenceNumber received in a MAP has an older value than the latest received value in the TM_DomainRoutingChange.ind message, the node should assume that it is synchronized with the domain master and that the MAP will be updated in further transmissions.

The domain master may receive TM_NodeTopologyChange.ind messages via the mechanism described in clause 8.6.4.2.1.1. After the domain master receives the topology change messages and updates its routing tables, it shall generate and distribute a new TM_DomainRoutingChange.ind message. This message shall include the updated routing table, and shall also indicate the nodes that generated the TM_NodeTopologyChange.ind messages that requested an acknowledgement (i.e., with AckType set to 00₂ or 10₂) via the TM_DomainRoutingChange.ind (see Table 8-46), and the sequence numbers of those messages (see Table 8-50). This is an acknowledgement to the nodes from which the domain master received the topology change messages and updated the routing table.

A node may also request the domain master for the latest routing tables by sending a TM_ReturnDomainRouting.req message. The domain master shall reply with a TM_ReturnDomainRouting.cnf message. The message sequence number of the TM_ReturnDomainRouting.cnf message header and the message sequence number specified in the last transmitted TM_DomainRoutingChange.ind message header are unrelated.

8.6.4.1.1.1 Generation of the unicast routing table

The domain master maintains a routing table that contains routing information from each node to all other destination nodes in the domain. This table contains NxN elements, where N is the number of nodes in the domain. Each element RI[i,j] in the table is the DEVICE_ID of the next node on the route from source node i toward final destination node j. In case that node i has a direct link to node j, the element RI[i,j] in the table is set to zero.

Figure 8-41 shows an example of the optimal routing paths for a network of 12 nodes, where each ellipse contains the nodes that are visible to each other (i.e., nodes that do not share the same ellipse are hidden from each other). Table 8-45 shows the routing table for this example, where each row of the routing table is referred to as a record.

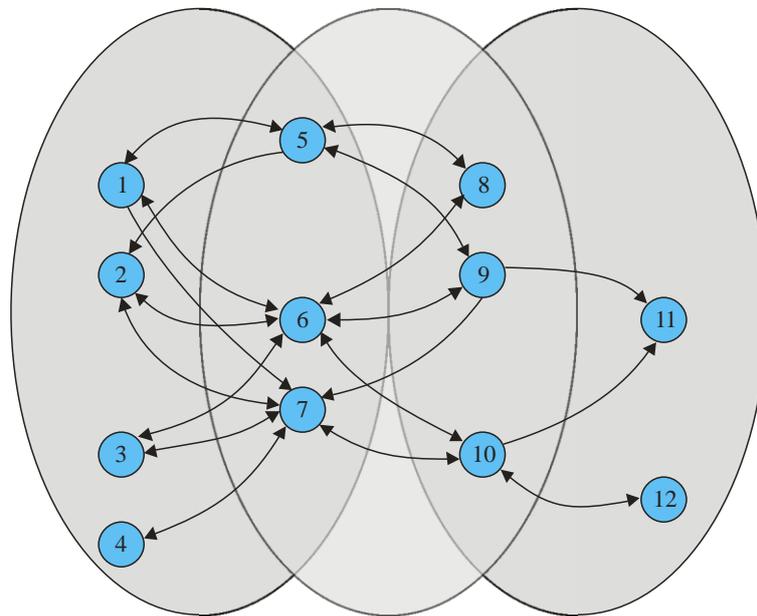


Figure 8-41 – Example 12-node network showing topology coverage ellipses and optimal paths

Table 8-45 – Routing table for example network in Figure 8-41

To (j) From (i)	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	0	0	0	0	0	5	6	7	6	7
2	0	0	0	0	0	0	0	5	6	7	6	7
3	0	0	0	0	0	0	0	7	6	7	6	7
4	0	0	0	0	0	0	0	7	6	7	6	7
5	0	0	0	0	0	0	0	0	0	0	9	10
6	0	0	0	0	0	0	0	0	0	0	9	10
7	0	0	0	0	0	0	0	0	0	0	10	10
8	5	5	6	6	0	0	0	0	0	0	0	0
9	6	6	6	7	0	0	0	0	0	0	0	0
10	6	6	7	7	0	0	0	0	0	0	0	0
11	9	9	9	9	9	9	9	0	0	0	0	0
12	10	10	10	10	10	10	10	0	0	0	0	0

The domain master, as a part of domain topology maintenance, broadcasts routing information in TM_DomainRoutingChange.ind messages as described in clause 8.6.4.1.1.2. The format of these messages shall be as defined in clause 8.6.4.3.5.

8.6.4.1.1.2 Distribution of routing tables

A multipoint relay (MPR) is a node designated by the domain master for relaying the topology and routing change messages to other nodes in the domain. The TM_DomainRoutingChange.ind message contains the routing tables and also contains, for each node, the HopCount field indicating the number of hops between the node and domain master and a field for each node indicating if the node is designated as an MPR (see Table 8-50). The TM_DomainRoutingChange.ind message shall use the reserved MAC address 01-19-A7-52-76-96 as the DA and zero as DestinationNode of the LLC frame. The domain master shall send TM_DomainRoutingChange.ind to all nodes in its visibility list (i.e., those with hop count = 0). Each MPR that receives this message, shall update its visibility list and the hop counts of the associated nodes, as indicated in the message, and then further relay the message to all the nodes in this updated visibility list that have a hop count greater than its own hop count. The TM_DomainRoutingChange.ind message shall include indication on the TM_NodeTopologyChange.ind that triggered the routing change as specified in Table 8-50. This is intended to be an acknowledgement for those messages. All the TM_NodeTopologyChange.ind messages should be sent using connections with acknowledgements. TM_DomainRoutingChange.ind messages may be sent using connections with acknowledgements. If a node that has sent one or more TM_NodeTopologyChange.ind messages with the same sequence number does not receive a TM_DomainRoutingChange.ind message that includes the sequence number of one of these TM_NodeTopologyChange.ind messages, it shall retry the procedure described in clause 8.6.4.2.1.1, after incrementing the repetition number of the TM_NodeTopologyChange.ind message. After three attempts, if the node does not get the indication that the TM_NodeTopologyChange.ind message it has sent was received by the domain master, it shall stop the procedure and initiate the self-resignation protocol (see clause 8.6.1.1.3.1). The time interval between attempts shall be at least 100 milliseconds.

A node relaying a TM_DomainRoutingChange.ind message as described in this section, shall first check if it has already relayed that message with a particular repetition number to all its destinations, using the sequence number and the repetition number of the message. If it has already relayed the message, it shall drop the message instead of relaying it further.

NOTE – A domain master should choose some of MPRs as MAP relays.

8.6.4.1.1.3 Generation of the BRT

The domain master shall maintain a broadcast routing table (BRT) to be used to broadcast LLC frames in the domain. The domain master shall ensure that the BRT does not contain loops and that broadcast LLC frames originated by one node are delivered to all nodes in the domain either following a direct link or through relay nodes.

The BRT contains $N \times N$ elements, where N is the number of nodes in the domain. Each element $DI[i,j]$ in the table contains routing information for node i including:

- DEVICE_ID of the node that shall deliver to node i LLC frames with OriginatingNode equal to j when following the BRT. The node with that DEVICE_ID is known as the root path of node i for LLC frames originated by node j ;
- A list of DEVICE_IDs of the nodes to which node i shall transmit LLC frames with OriginatingNode equal to j when following the BRT. That list of DEVICE_IDs is known as the branch path of node i for LLC frames originated by node j .

The domain master shall communicate the BRT to all nodes in the domain using the TM_DomainRoutingChange.ind message in a compressed format. The domain master shall generate the BRT entries for each node in the domain. An entry for node i is defined as:

- List of DEVICE_IDs corresponding to originating nodes (i.e., list of originating nodes), including the node itself;
- Root path of node i for LLC frames originated by all nodes in the list of originating nodes;

- Branch path of node *i* for LLC frames originated by all nodes in the list of originating nodes.

The domain master shall not include an entry for node *i* in the `TM_DomainRoutingChange.ind` if:

- the list of originating nodes contains only one `DEVICE_ID`; and
- the root path is equal to that `DEVICE_ID`; and
- the branch path is an empty list.

NOTE – These conditions describe the case of a leaf node that receives broadcast LLC frames originated by node *k* directly from it (without a relay in the middle).

The format used to describe the BRT is described in clause 8.6.4.3.5.

8.6.4.1.2 Domain master operation in DRTM mode

The domain master shall play the same role as any other endpoint node in topology maintenance as described in clause 8.6.4.2.2.

8.6.4.2 Endpoint node topology maintenance

Each node in the domain shall participate in the topology maintenance by sending a message `TM_NodeTopologyChange.ind` with the `AckType` field set to `002` (see Table 8-46) whenever any of the following events occurs:

- after the node successfully joined the domain (in an unsecure domain, after node successfully registered by the domain master, in a secure domain, after node successfully authenticated by the SC);
- when the node's list of topology-related parameters (e.g., the list of other nodes that it can detect) has changed; or
- when the contents of the LAAT associated with the node has changed.

NOTE 1 – If a node provides IDB to another domain, changes in MAC addresses associated with this domain are considered as changes in the AE of the node. Reporting on changes in data rates of incoming and outgoing streams is for further study.

NOTE 2 – The values of data rates of the incoming and outgoing streams that are part of the report may change frequently, causing congestion of the domain with topology update messages. Relevant criterion on reporting of data rate variations is needed to resolve the issue.

In addition, the domain master may request the nodes in the domain to periodically report their topology information, by including in the transmitted MAP frame the timer-related domain info auxiliary information field that contains the topology update interval sub-field with a non-zero value in the `TopologyPeriodicInterval` field (Table 8-83). The domain master shall specify the scope of the request topology report by setting the required value to the `RequestReport` field. The domain master may stop the requested periodic topology report, by setting the `TopologyPeriodicInterval` field to zero.

The criteria for determining whether the list of other nodes that a node can detect has changed is vendor discretionary. All nodes in the domain should attempt to receive the headers of all PHY-frames communicated in the domain to collect a more comprehensive set of topology information.

8.6.4.2.1 Endpoint node topology maintenance in CRTM mode

A node shall reply with a `TM_NodeTopologyChange.cnf` message upon receiving a `TM_NodeTopologyChange.req` message sent by the domain master or backup domain master requesting topology information. The message shall be sent within T_{N_RSP} ms (see clause 8.4) after reception of the `TM_NodeTopologyChange.req` message from the domain master.

In case one of the events occurs (see clause 8.6.4.2), `TM_NodeTopologyChange.ind` message shall be sent to the domain master as soon as possible.

If the DM requests a periodic topology report through the MAP (see clauses 8.6.4.2 and 8.8.5.8.1), every node in the domain shall transmit a `TM_NodeTopologyChange.ind` message randomly within each interval according to the requested report specified in the `RequestReport` field included in the topology update interval sub-field. Each node shall start counting the interval from the first received MAP that contains a request for a periodic topology report. Each new interval starts after the end of the previous interval.

At its own discretion, a node may request an acknowledgement via a `TM_DomainRoutingChange.ind` message, that the domain master received the periodically sent `TM_NodeTopologyChange.ind` message, by setting the `AckType` field to 10₂ (see Table 8-46). If a node requests an acknowledgement, then the domain master shall include the `DEVICE_ID` of this node and the sequence number of the topology change message (see Table 8-50) in the next `TM_DomainRoutingChange.ind` message.

NOTE – For routine `TM_NodeTopologyChange.ind` messages sent by a node as "keep-alive messages" it should not request an explicit acknowledgement in order to reduce unnecessary control traffic. If a node suspects that a `TM_NodeTopologyChange.ind` message that it sent was not received by the domain master, then it should request an explicit acknowledgement via `TM_DomainRoutingChange.ind` for the `TM_NodeTopologyChange.ind` message.

Messages `TM_NodeTopologyChange.ind` and `TM_NodeTopologyChange.cnf` shall contain the following updated information:

- the `DEVICE_ID` of each node in its own domain that it can detect. This is called the node's visibility list and it shall be included in the message if there was a change in the detected nodes;
- MAC addresses associated with the AE of the node; namely, the local AAT. (MAC addresses associated with another domain are considered as associated with a node that provides IDB to this domain). This component shall be included in the message if there was a change in the AAT list. Incremental information on added and deleted MAC addresses can also be sent using this message;
- data rates associated with each node in the domain that the reporting node could connect or detect. This component shall be included in the message if there was a change in the data rate;
- main node capabilities (`bandplanOFB`, capability to serve as a domain master, as a security controller, or as a relay);
- sequence number of the message (for monitoring purposes).

An endpoint node is not required to maintain complete topological information of the domain, as a domain master does (see clause 8.6.4.1) but only the information it needs for topology reporting and communication with the domain master and other nodes. A node that has been appointed as the domain master's backup shall maintain complete topological information of the domain (see clause 8.6.5). Nodes shall update their topological information using the received `TM_DomainRoutingChange.ind` message.

A node may also request the domain master for an update of domain topology by sending a `TM_ReturnDomainRouting.req` message to the domain master. If a node has received a `TM_DomainRoutingChange.ind` message from the domain master in the past `T_UPDATE_MIN` ms (see clause 8.4), it is not allowed to request the topology update from the domain master (i.e., to send a `TM_ReturnDomainRouting.req` message to the domain master).

A node that receives a `TM_DomainRoutingChange.ind` message shall update its local topology tables and routing table accordingly.

If a node that has a link to a destination node detects that the route to the destination node is broken, it may select an alternative route (if allowed by the `Routing Authorization` field) towards the

destination node based on the current routing table, until a new TM_DomainRoutingChange.ind message is received from the domain master.

A node whose current routing table differs from the last routing table indicated in the MAP shall request the domain master to update the routing information by sending a TM_ReturnDomainRouting.req message to the domain master.

8.6.4.2.1.1 Communication of endpoint topology change in CRTM mode

Each endpoint node shall maintain a visibility list. A node shall transmit a TM_NodeTopologyChange.ind message to the domain master if it can no longer communicate directly with one or more nodes from its existing visibility list. If the node has no direct link to the domain master, it shall transmit the message to all MPRs with a hop count lower than its own hop count that are in its visibility list using the reserved MAC address 01-19-A7-52-76-96 as the DA and zero as DestinationNode of the LLC frame. In secure mode, this message shall be encrypted using the DB encryption key. If there are no such MPRs in the visibility list of a node, it shall transmit the message to all nodes in its visibility list, except for the node from which it received the message. A node that receives the message shall follow the procedure described in this section to relay the message further, as if the message was generated by itself.

If the MPR that receives this message has a direct link to the domain master (i.e., hop count 0), it shall stop the relaying procedure, address the message to the domain master using the domain master's DA. In secure mode, this message shall be encrypted using the applicable encryption key (either NN or NMK). Otherwise, it shall relay it to all the MPRs that are in its visibility list and have hop count lower than its own hop count. If a MPR that receives this message, does not have a direct link to the domain master or any other MPR with a hop count lower than its own hop count, it shall transmit the message to all nodes in its visibility list (except for the node from which it received the message) and the node that receives the message shall follow the procedure described in this clause to relay the message, as if the message were generated by itself.

If the visibility information of a node does not change, but other topology-related parameters change, a TM_NodeTopologyChange.ind addressed to the domain master shall be sent using the routing tables described in clause 8.6.4.1.1.1, instead of using the procedure in this clause.

All nodes should establish connections for transmission and reception of management messages (i.e., management connection or data connection with mixed management messages) with acknowledgement enabled, with all the MPRs in their visibility list, to reliably transmit topology change messages and receive routing change messages.

A node relaying a TM_NodeTopologyChange.ind message as described in this clause, shall first check if it has already relayed that message with a particular repetition number to all its destinations, using the sequence number and the repetition number of the message and DEVICE_ID of the node whose information is conveyed in this message. If it has already relayed the message to a destination, it shall drop the message instead of relaying it further.

8.6.4.2.2 Endpoint node topology maintenance in DRTM mode

The mechanism for endpoint node topology maintenance in DRTM mode is for further study.

8.6.4.2.3 Flooding of topology information

The mechanism for flooding of topology messages is for further study.

8.6.4.3 Message formats

The following management messages shall be used to support topological discovery and maintenance. For a secure domain, all messages defined in this clause shall be sent encrypted. If the length of the message does not fit the maximum length of LCDU, the message shall be segmented using the segmentation rules for management messages described in clause 8.10.

8.6.4.3.1 Format of TM_NodeTopologyChange.ind

Message TM_NodeTopologyChange.ind is a management message that is transmitted by nodes as a part of domain topology maintenance. The format of the MMPL of a TM_NodeTopologyChange.ind message shall be as presented in Table 8-46.

The message is of variable length. It allows communication of a complete or partial topology report, which includes only AAT, or only visibility information, or only neighbouring domain DODs, or a combination of those. In case a node communicates its report as a sequence of partial reports, the total number of segments and the sequence number of the reported segment shall be indicated in the number of segments and sequence number fields of the MMH of the LCDU (see Table 8-87). The number of parts may be up to 16. All parts of the same report shall keep the same sequence number.

The TM_NodeTopologyChange.ind message may include an incremental set of topology parameters (e.g., only those that have changed since the last report). This is indicated by setting the type field to 02₁₆.

The sequence number of the first TM_NodeTopologyChange.ind message that a node sends after registering to a domain shall be zero.

The most recent update shall supersede all previous messages (to know how it is determined if a received message is older, equal or newer than the last correctly received message, see clause 8.10.1.2).

Table 8-46 – Format of MMPL of the TM_NodeTopologyChange.ind message

Field	Octet	Bits	Description
DEVICE_ID	0	[7:0]	DEVICE_ID of the node whose topology information is conveyed in this message.
Type	1	[3:0]	Shall be set to: <ul style="list-style-type: none"> – 0₁₆ if report includes all the topology parameters available to the node (full report) – 1₁₆ if the report includes any fraction of topology parameters available to the node – 2₁₆ if the report includes only parameters that changed relatively to the last report Other values are for further study.
AckType		[5:4]	This field shall be set to: <ul style="list-style-type: none"> • 00₂: Event driven update, ACK requested • 01₂: Periodic update, no ACK requested • 10₂: Periodic update, ACK is requested • 11₂: Reserved by ITU-T ACK in this case means acknowledgement via TM_DomainRoutingChange.ind is requested (see Table 8-50).
Reserved		[7:6]	Reserved by ITU-T (Note 1)
InterferenceInfo	2	[1:0]	Node's estimation of level of interference that it suffers from neighbouring domains. <ul style="list-style-type: none"> 00₂ – No interference 01₂ – Low interference 10₂ – High interference
Reserved		[7:2]	Reserved by ITU-T (Note 1)
Reserved	3 and 4	[15:0]	Reserved by ITU-T (Note 1)

Table 8-46 – Format of MMPL of the TM_NodeTopologyChange.ind message

Field	Octet	Bits	Description
NodeRec_Size	5 to 7	[23:0]	Size of the node record in bytes (S0) represented as a 24-bit unsigned integer.
NodeRec_Info	8 to (S0+7)	See Table 8-47	Node record information field, S0-byte long, with a format as defined in Table 8-47. $S0 = 8+(4*M)+(6*L)$
NumDomRecs (Note 2)	(S0+8)	[7:0]	Number of records (n) on neighbouring domains represented as an unsigned integer in the range from 0 to 255.
NeighbDom_ID [0]	(S0+9)	[7:0]	The DOD of the first neighbouring domain.
NeighbDom_Size[0]	(S0+10) to (S0+11)	[15:0]	The value of this field shall be set to zero.
...
NeighbDom_ID [n-1]	(S0+6+3*n)	[7:0]	The DOD of the last neighbouring domain.
NeighbDom_Size[n-1]	(S0+7+3*n) to (S0+8+3*n)	[15:0]	The value of this field shall be set to zero.

NOTE 1 – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.
NOTE 2 – The value of zero indicates that no information on neighbouring domain is available. The value of 255 indicates that no record on neighbouring domains is attached (while information on neighbouring domains is available).

Table 8-47 – Format of a NodeRec_Info field of the TM_NodeTopologyChange.ind message

Field	Octet	Bits	Description
NodeParam	0 to 2	[23:0]	A 24-bit field describing parameters and capabilities of the reporting node, formatted as described in Table 8-47.1.
NodeAIFG	3	[7:0]	The value of T_{AIFG} supported by the node, represented as $n \times 1.28 \mu s$; the value of n is an unsigned integer in the range between 4 and 96 (Note 1).
NumNodeInformationTLVs	4	[7:0]	Number of NodeInformationTLVs (N) included in this message. Set to 0 if no NodeInformationTLVs are included.
AAT_Size	5 and 6	[15:0]	Number (k) of local AAT entries associated with the reporting node (Note 2).
AAT [0]	7 to 13	[47:0]	The first entry in the AAT. It contains the first local MAC address (Note 3).
...
AAT [k-1]	$7+(6*K-1)+1$ to $(7+6*k)$	[47:0]	The last entry in the AAT (for $k>1$). It contains the last local MAC address.
RemAAT_Size	Variable	[15:0]	Number (p) of AAT entries that are removed from the node AAT (Note 4).
RemAAT [0]	Variable	[47:0]	First entry in the RemAAT. It contains the first MAC address that has been removed from the node AAT.

Table 8-47 – Format of a NodeRec_Info field of the TM_NodeTopologyChange.ind message

Field	Octet	Bits	Description
...
RemAAT [p-1]	Variable	[47:0]	Last entry in the RemAAT. It contains the last MAC address that has been removed from the node AAT.
NewAAT_Size	Variable	[15:0]	Number of AAT entries (q) that were added to the node AAT (Note 5).
NewAAT [0]	Variable	[47:0]	First entry in the NewAAT. It contains the first MAC address that has been added to the node AAT.
...
NewAAT [q-1]	Variable	[47:0]	Last entry in the NewAAT. It contains the last MAC address that has been added to the node AAT.
Visibility_Size (Note 6)	Variable	[7:0]	Number of nodes M in the domain which were detected by the reporting node, represented as an unsigned integer in the range between 1 and 249.
Visibility_List	Variable	[39:0]	List of M fields, 5 octets each, describing a single detected node, formatted as described in Table 8-48
NodeInformation TLVs[0]	Variable	Variable See Table 8-16.1	Information related to the features implemented and capabilities of the registering node. It shall be coded as described in Table 8-16.1
...
NodeInformation TLVs[N-1]	Variable	See Table 8-16.1 Variable	Information related to the features implemented and capabilities of the registering node. It shall be coded as described in Table 8-16.1
NodeVisibilityTLV[0]	Variable	See Table 8-140.1	Visibility information TLV for the first node whose visibility is being reported. It shall be formatted as shown in Table 8-140.1.
...
NodeVisibilityTLV[M-1]	Variable	See Table 8-140.1	Visibility information TLV for the last node whose visibility is being reported (Mth). It shall be formatted as shown in Table 8-140.1.

NOTE 1 – Once registered or upon re-registration in accordance with the topology update interval (see Table 8-82), a node shall not change the value of this field. Valid values for each medium are specified in Table 8-14.

NOTE 2 – If this field is zero, no AAT fields shall be included in the message. Otherwise, it contains the number of entries in the full local AAT that are specified in the message. The first time the node reports this message to the domain master, it shall include in the message its full local AAT.

NOTE 3 – The first MAC address shall be the REGID of the reporting node.

NOTE 4 – If this field is zero, no entries have been removed from the local AAT since the previous transmitted report for that node, and no RemAAT fields shall be included. Otherwise, it contains the number of removed entries from the local AAT. This field shall be set to zero if AAT_Size field is non-zero.

NOTE 5 – If this field is zero, no entries have been added to the local AAT since the previous transmitted report for that node and no NewAAT fields shall be included. Otherwise, it contains the number of added

Table 8-47 – Format of a NodeRec_Info field of the TM_NodeTopologyChange.ind message

Field	Octet	Bits	Description
new entries to the local AAT since last transmitted report for that node. This field shall be set to zero if AAT_Size field is non-zero.			
NOTE 6 – Value 255 indicates that no record on visibility is attached (while a node possesses information on visibility). Value 0, and values 251-254 are reserved by ITU-T.			

Table 8-47.1 – Format of a NodeParam field of a NodeRec_Info field of the TM_NodeTopologyChange.ind message

Field	Octet	Bits	Description
NodeParam	0	[0]	Set to one if node is capable of serving as a relay and 0 otherwise.
		[1]	Set to one if node is capable of serving as a MAP relay and 0 otherwise.
		[2]	Reserved by ITU-T (Note)
		[3]	Set to one if node is intended to use power saving mode and 0 otherwise.
		[4]	Reserved by ITU-T (Note)
		[7:5]	Indicates the bandplan-OFB used by the node as described in Table 7-11/10-11 of [ITU-T G.9960] (BNDPL_OFB_ID /GRP_ID field).
	1	[0]	Set to one if a node is capable of serving as a SC and zero otherwise.
		[1]	Reserved by ITU-T (Note)
		[2]	Set to one if node is configured by the user to operate as a domain master.
		[3]	Set to one if node is the assigned backup of the domain master and zero otherwise.
		[4]	Set to one if node is capable of serving as a domain master and zero otherwise.
		[6:5]	Indicates the profile of the node; value 00 ₂ is the default and indicates full compliance to the main body of this Recommendation.
		[7]	Set to one if node is connected to the IDB (gateway to another ITU-T G.9960/1 domain) and zero otherwise.
	2	[0]	Set to one if node is capable of calculating routing tables and zero otherwise.
		[2:1]	Indicates the standard routing algorithm (see Table 8-62) supported by the node. These bits shall be ignored if a node is not capable of calculating routing tables.
		[7:3]	Reserved by ITU-T (Note).

NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.

Table 8-48 – Format of a Visibility_List field

Field	Octet	Bits	Description
DeviceID	0	[7:0]	DEVICE_ID of a node that the reporting node detected.
BitsPerSecond	1 to 4	[31:0]	Bits [15:0] indicate the PHY data rate in bits per second from the reporting node to the detected node; Bits [31:16] indicate the PHY data rate from the detected node to the reporting node. (Notes 1, 2)
<p>NOTE 1 – Both data rates shall follow the formula $\sum_{i=1}^N \text{BitsPerSecond}_i * \frac{T_i}{T_{Cycle}}$ where N is the number of channel estimation windows in the MAC cycle, BitsPerSecond_i is the result of applying the same formula as Note 2 in Table 8-33 to the <i>i</i>-th channel estimation window, T_i is the duration of the <i>i</i>-th channel estimation window and T_{Cycle} is the duration of the MAC cycle. If no channel estimation is available for a particular window, robust communication mode (RCM) parameters shall be used for that window. If the data rate with the particular detected node is not available, the value of this parameter shall be set to FFFF₁₆.</p> <p>NOTE 2 – In case the DM receives conflicting information in the BitsPerSecond field from the Visibility_List of two different nodes for a given link, the DM shall take into account the minimum value between both.</p>			

8.6.4.3.2 Format of the TM_NodeTopologyChange.req

Message TM_NodeTopologyChange.req is a management message that shall be sent by a domain master to a particular node or broadcast to refresh its copy of the domain topology. By appropriate settings in the ReqRep field, the domain master may request only AAT information, or only visibility information, or only information on neighbouring domains, or a combination of them.

The MMPL of the TM_NodeTopologyChange.req message shall be as shown in Table 8-49.

Table 8-49 – Format of the MMPL of the TM_NodeTopologyChange.req message

Field	Octet	Bits	Description
ReqRep	0	[7:0]	<p>Bit 2 – Set to one if visibility information is required</p> <p>Bit 3 – Set to one if AAT information is required, set to zero if AAT information is not required (Note).</p> <p>Bit 4 – Set to one if the complete AAT information is requested, set to zero if the AAT information requested is relative to the last report sent by the node.</p> <p>Other bits are reserved by ITU-T and shall be set to zero.</p>
NOTE – Setting of both bits 2 and 3 to zero is not allowed.			

8.6.4.3.3 Format of TM_NodeTopologyChange.cnf

TM_NodeTopologyChange.cnf is a management message that shall be sent by a node in response to a TM_NodeTopologyChange.req message from the domain master or backup domain master.

If the requested topology information has not changed since sending the last TM_NodeTopologyChange.ind message, the MMPL of the TM_NodeTopologyChange.cnf message shall contain the sequence number of the last transmitted TM_NodeTopologyChange.ind message

and the rest of the MMPL shall contain the requested components from the MMPL of the last transmitted TM_NodeTopologyChange.ind message (Table 8-46).

If the requested topology information has changed since sending the last TM_NodeTopologyChange.ind message, the MMPL of the TM_NodeTopologyChange.cnf message shall contain the updated sequence number (i.e., greater than that of the last transmitted TM_NodeTopologyChange.ind message) and the rest of the MMPL shall contain the complete, updated topology information (even if relative information was requested) for the requested components.

The format of the TM_NodeTopologyChange.cnf message shall be as shown in Table 8-49.1.

Table 8-49.1 – Format of the MMPL of the TM_NodeTopologyChange.cnf message

Field	Octet	Bits	Description
ReqRep	0	[7:0]	This field contains the original received field as specified in the TM_NodeTopologyChange.req message.
OrigSequence	1 and 2	[15:0]	Sequence number of the TM_NodeTopologyChange.ind message whose MMPL information is included in the Info field
Info	Variable	VariableSee Table 8-46	This field contains the TM_NodeTopologyChange.ind message components as specified in Table 8-46 according to the specified ReqRep field.

8.6.4.3.4 Format of TM_DMBBackup.ind

Message TM_DMBBackup.ind is a management message that shall be sent by the domain master to the backup domain master to inform about changes in a node's topology information. The MMPL of the TM_DMBBackup.ind message shall contain the sequence number of the received TM_NodeTopologyChange.ind message whose MMPL information is contained in the Info field. The format of the TM_DMBBackup.ind message shall be as shown in Table 8-49.2.

Table 8-49.2 – Format of the MMPL of the TM_DMBBackup.ind message

Field	Octet	Bits	Description
OrigSequence	0 and 1	[15:0]	Sequence number of the received TM_NodeTopologyChange.ind message whose MMPL information is contained in the Info field.
Info	Variable	VariableSee Table 8-46	This field contains the TM_NodeTopologyChange.ind message components as specified in Table 8-46.

8.6.4.3.5 Format of TM_DomainRoutingChange.ind

Message TM_DomainRoutingChange.ind is a management message that shall be sent by the domain master as part of the domain topology maintenance. The message has a variable length, depending on the number of items included in the message such as the number of nodes in the domain and the size of the AAT of each node.

TM_DomainRoutingChange.ind may be segmented using the number of segments, segment number and sequence number fields of the MMH of the LCDU (see clauses 8.10.1 and 8.10.1.2). The total number of nodes in all segments shall be equal to the total number of reported nodes.

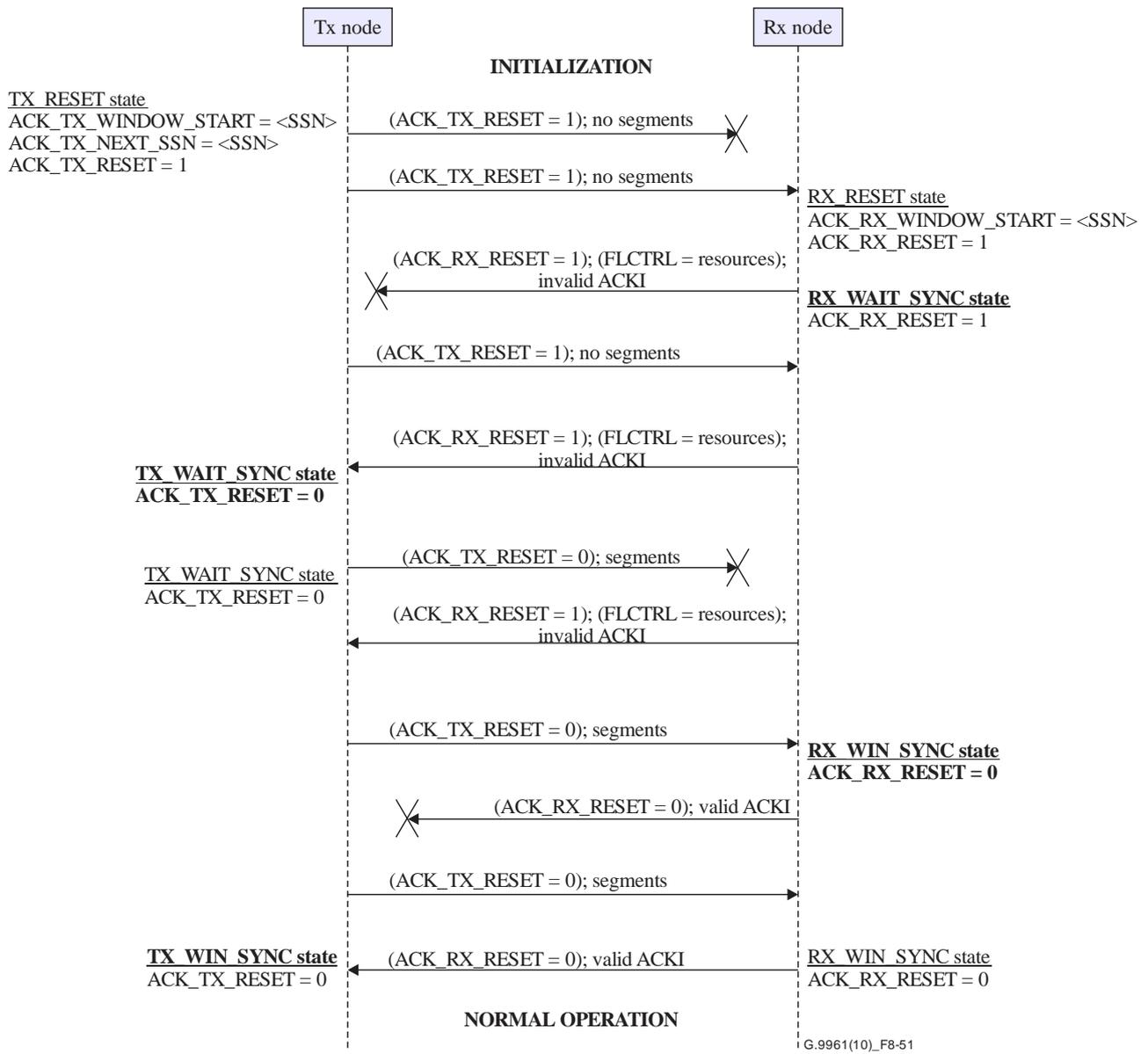


Figure 8-51 – Detailed initialization of the Acknowledgement protocol

If in `TX_WIN_SYNC` state the transmitting node receives `ACK_RX_RESET = 1`, the receiving node is in `RX_RESET` state. In this case, if the flow control information conveys a status report or a valid hold time, the transmitting node may transition into the `TX_RESET` state, send `ACK_TX_RESET = 1` and follow again the initialization procedure described in this clause; or it may discard the segments of the established connection and terminate that connection.

If in any state the transmitter receives flow control information indicating the unavailability of resources (see clause 8.12), the transmitter shall discard the segments of the established connection, terminate that connection and transition into the `TX_RESET` state.

8.9.5.3.1 Transmission window operation

Comparisons of SSNs that appear in this clause assume a previous normalization as described in clause 8.9.5.1. The term LSSN is used in this clause to refer to the value conveyed in the ACK frame fields LSSN (see clause 7.1.2.3.2.3.9.1.6/[10.1.2.3.2.3.9.1.6](#) of [\[ITU-T G.9960\]](#)) and MNMT_LSSN (clause 7.1.2.3.2.3.9.1.3.1/[10.1.2.3.2.3.9.1.3.1](#) of [\[ITU-T G.9960\]](#)).

The transmitter shall maintain an `ACK_TX_WINDOW` per connection established with the receiver.

In TX_WAIT_SYNC or TX_WIN_SYNC state, when an acknowledgement with ACK_RX_RESET = 0 is received, the transmitter shall process the conveyed acknowledgement data. The transmitter shall discard the acknowledgement data if the LSSN does not satisfy any of the following conditions:

- $ACK_TX_WINDOW_START \leq LSSN < ACK_TX_NEXT_SSN$;
- LSSN is equal to the N LSB bits of $ACK_TX_NEXT_SSN$ and there is no valid selective acknowledgement information (the ACKI field is set according to clause 7.1.2.3.2.3.9.1.7/10.1.2.3.2.3.9.1.7 of [ITU-T G.9960] to indicate that all data units have been received with errors).

NOTE – The previous conditions assure that either the LSSN is contained in ACK_TX_WINDOW or that the receiver is acknowledging all the contents of it. Then, $ACK_RX_WINDOW_START$ is equal to $ACK_TX_NEXT_SSN$.

Otherwise, the transmitter shall continue processing the received acknowledgement information.

If an acknowledgement message is not discarded, the transmitter shall interpret the contents (see clause 7.1.2.3.2.3/10.1.2.3.2.3 of [ITU-T G.9960]) and update the ACK_TX_WINDOW as described below.

The transmitter shall change to done state all the segments with SSNs that satisfy the condition $ACK_TX_WINDOW_START \leq SSN < LSSN$ and shall then update $ACK_TX_WINDOW_START$ to the SSN whose N LSB bits are equal to the received LSSN. After that, the transmitter shall interpret the contents of the selective acknowledgement information (ACKI) and shall change to done state the indicated segments whose SSNs fulfil the condition $ACK_TX_WINDOW_START \leq SSN < ACK_TX_NEXT_SSN$.

$ACK_TX_WINDOW_START$ and $ACK_RX_WINDOW_START$ shall be kept synchronized so that the receiver never awaits the reception of a segment that has been removed from the transmission window (passed to discarded state) and has never been received correctly in the receiver side. Therefore, the oldest pending segment flag (OPSF) is used to avoid this. The transmitter shall always set the OPSF of the oldest segment pending acknowledgement (not in done or discarded state) to one to inform the receiver. The OPSF of an LPDU shall not be modified between the transmission of a PHY-frame and the reception of the Imm-ACK in case it was requested.

When a segment is discarded after $ACK_BLOCK_LIFETIME$ (see clause 8.9.4.2) the transmitting node shall proceed to the next segment that is not in the done or discarded state and shall set its OPSF to one.

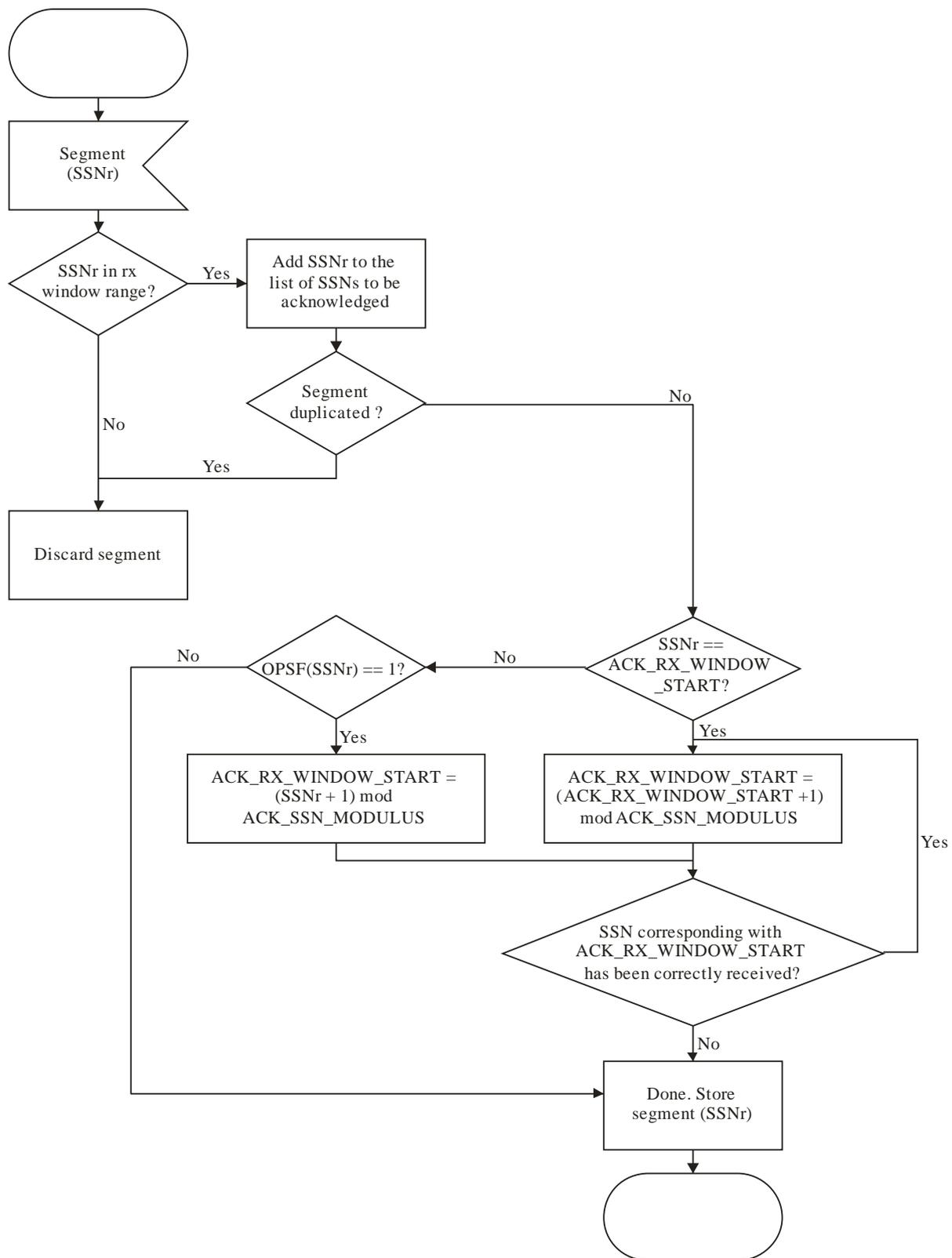
When ACK_TX_WINDOW is equal to $ACK_TX_CONF_WINDOW$ and all the segments in ACK_TX_WINDOW are in done or discarded state and the LPDU corresponding to $ACK_TX_WINDOW_START$ is in the discarded state, the transmitting node shall transition into the TX_RESET state, send $ACK_TX_RESET = 1$ and reset the connection (see clause 8.12.7).

8.9.5.3.2 Reception window operation

Comparisons of SSNs that appear in this clause assume a previous normalization as described in clause 8.9.5.1.

The receiver shall maintain an $ACK_RX_CONF_WINDOW$ per connection established with the transmitter. When a segment with a SSN that falls in $ACK_RX_CONF_WINDOW$ is received, the receiver shall accept it. Segments with SSNs outside $ACK_RX_CONF_WINDOW$ shall be rejected as out of order. The receiver shall discard duplicate segments (segments that where already received correctly) within the window.

In RX_WAIT_SYNC or RX_WIN_SYNC state, when a segment is received in a frame with $ACK_TX_RESET = 0$, the receiver shall follow the actions shown in Figure 8-52 to manage the receiver window variables.



G.9961(10)_F8-52

Figure 8-52 – Receiver SSN processing

The sliding window is maintained such that the `ACK_RX_WINDOW_START` variable always points to the lowest numbered segment that has not been received or has been received with errors. `OPSF` is used to synchronize `ACK_TX_WINDOW_START` and `ACK_RX_WINDOW_START` by moving the `ACK_RX_WINDOW_START` to the segment for which the `OPSF` is set.

8.9.5.4 Acknowledgement protocol state machine for multicast transmission

The protocol to be used between nodes to facilitate multicast transmission with acknowledgements is initialized as presented in Figure 8-53. Before initializing the acknowledgement protocol, the multicast binding protocol (clause 8.16) shall be completed to assign resources for the transmission and reception windows. The initialization is based on the use of the OPSF.

ACK_TX_RESET is sent in the PHY-frame header of the MSG frame (see clause 7.1.2.3.2.2.18/[10.1.2.3.2.2.18](#) of [\[ITU-T G.9960\]](#)). ACK_RX_RESET is sent in the PHY-frame header of the ACK frame (see clauses 7.1.2.3.2.3.5/[10.1.2.3.2.3.5](#) and 7.1.2.3.2.3.6/[10.1.2.3.2.3.6](#) of [\[ITU-T G.9960\]](#)) according to clause 8.9.2.

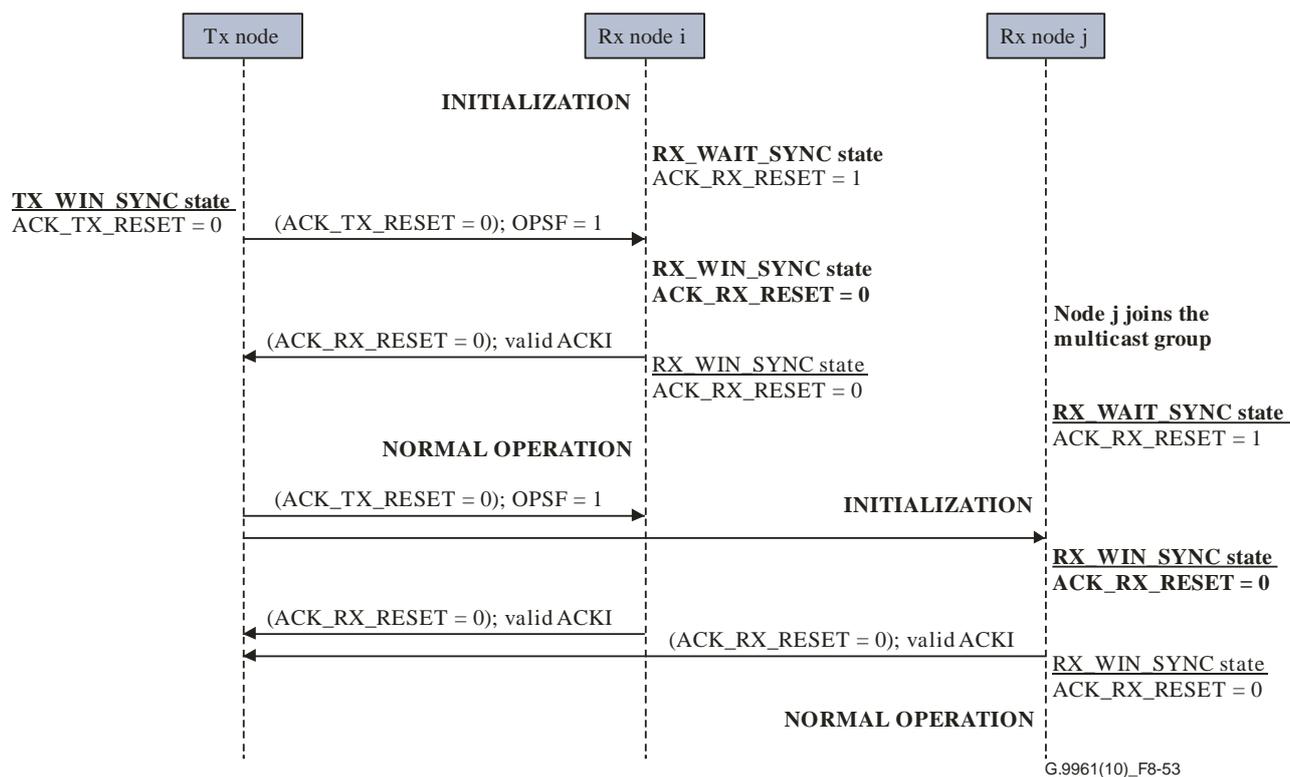


Figure 8-53 – Initialization of the acknowledgement protocol for multicast transmission

First, the receiving node state machine shall be in RX_WAIT_SYNC state with ACK_RX_RESET = 1. If the receiving node receives ACK_TX_RESET = 0 while being in RX_WAIT_SYNC state, it shall search for the segment with OPSF = 1 and set ACK_RX_WINDOW_START to the SSN of that segment. Then, it shall set the flag ACK_RX_RESET to zero and transition into RX_WIN_SYNC state.

Before sending the first segment of the connection, the transmitting node shall be in TX_WIN_SYNC state with ACK_TX_WINDOW_START equal to the SSN of the first segment of the connection. Then, it shall set the OPSF of that segment equal to one. Segments of the established connection may then be sent in this state as described in clause 8.9.5.2.

When transmitting and receiving nodes are in TX_WIN_SYNC state and RX_WIN_SYNC state, the initialization of the acknowledgement protocol is completed. After the initialization, the protocol enters its normal operation.

In TX_WIN_SYNC state the transmitting node shall ignore the reception of ACK_RX_RESET = 1.

8.9.5.4.1 Transmission window operation

In TX_WIN_SYNC state, when multicast acknowledgements with ACK_RX_RESET = 0 are received, the transmitter shall process the conveyed acknowledgement data. After receiving all the acknowledgements, as described in clause 8.9.2, the transmitter shall build a worst case Ack and operate the ACK_TX_WINDOW as described in clause 8.9.5.3.1.

The worst case Ack shall be built so that a segment is considered unacknowledged if it is indicated as unacknowledged by any of the multicast ACKs. If a NACK signal is detected, the worst case Ack shall consider all segments as unacknowledged.

8.9.5.4.2 Reset of a multicast connection with acknowledgements

To reset a multicast connection with acknowledgements a transmitter shall send a PHY frame with FT=MSG, no payload, RPRQ=01 or 11 (depending if the NACK signalling slot is used or not), NUM_MCAK_SLOTS equal to the number of Mc-ACK slots, START_SSN=ACK_TX_WINDOW_START and CNN_MNGMT=0111.

For a multicast group with Mc-ACK slots assigned for each group member (RPRQ = 01) the transmitter shall send a request to reset the connection and wait for an acknowledgement from each of the group members. If a positive acknowledgement is not received from all group members, the transmitter may continue to send additional reset requests until it receives a positive acknowledgement from each multicast group member or until an N_a number of attempts to reset the multicast connection have been performed. If a multicast group member ceases to respond in its Mc-ACK slot for N_a consecutive reset requests, the transmitter shall assume that the multicast group member is no longer active and the transmitter shall exclude the member from the multicast group. In this case, if the transmitter does not reassign the Mc-ACK slot for the excluded group member, the transmitter shall ignore any Mc-ACK frames received in that slot.

For a multicast group where the NACK signalling slot is used (RPRQ = 11) the transmitter may choose to reset the connection or to release the connection and re-establish it again. If the transmitter chooses to reset the connection, it shall follow the same procedure as described above (where only Mc-ACK slots are used) but in addition it shall also require that the NACK signalling slot be empty during at least one of the reset requests.

If N_a attempts to reset the multicast connection fails the transmitter shall release the multicast connection and may re-establish it again.

The value of N_a is vendor discretionary. The members of the multicast group shall set ACK_RX_WINDOW_START to the value conveyed in the START_SSN field.

8.9.5.4.3 Reception window operation

The reception window shall be operated as described in clause 8.9.5.3.2.

8.10 Management and control message format

8.10.1 Management message format

Internal management messages, intended for communication between nodes of the same domain, shall be mapped into an LCDU payload field (see Figure 8-6). In-band management messages intended for communication with entities that reside locally above the A-interface of a node or above the A-interface of another node in the domain (see clause 8.1.1) shall be mapped into an APDU payload field (see Figure A-1). All management messages shall be formatted as shown in Figure 8-54, including a management message header (MMH) and a management message parameter list (MMPL or CMMPL) or a segment of MMPL or CMMPL (if segmentation is needed, see clause 8.10.1.2). The first byte (octet 0) of the MMH shall be the first byte of the LCDU/APDU payload, as described in clause 8.1.3.3. Encapsulation of the management message into an LLC frame is shown in Figure 8-7.

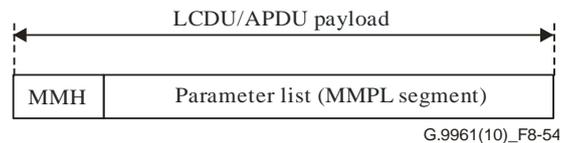


Figure 8-54 – Format of a management message

The MMH defines the length, the type, and other parameters of the message. The type of the message is identified by an OPCODE associated with a particular management function, as presented in Table 8-88. The MMPL (or CCMPL, see clause 8.24) includes a list of management message parameters, depending on the management function. The format of any management message shall be as shown in Table 8-87. An LCDU that contains a MAP message shall be carried only in the MAP or RMAP frame.

The format of MMPLs may be revised in future versions of this Recommendation by appending additional fields. Furthermore, fields may be defined using bits that are currently indicated as reserved for ITU-T. Nodes indicate the node information via NodeInformationTLVs during registration (see Table 8-16) and topology updates (see Table 8-47). Nodes shall be able to parse the MMPL (the length of the MMPL is specified in the MMH) but shall ignore the content of fields that they do not understand, i.e., those associated with later versions of the Recommendation.

Table 8-87 – Format of management messages

	Content	Octet	Bits	Description
MMH	Length	0 to 2	[11:0]	Length (LG) of the MMPL segment in octets, encoded as a 12-bit unsigned integer. The value of LG shall not exceed 1492.
	OPCODE		[23:12]	12-bit OPCODE, indicates message type (Note 1).
	MMPL Compression	3	[1:0]	Indicates whether MMPL data is compressed or not and which type of compression is used. – 00 ₂ if no compression is used – 01 ₂ if Huffman Coding is used Other values are reserved by ITU-T
			[7:2]	Reserved by ITU-T (Note 4).
	Number of segments	4	[3:0]	Number of segments minus 1, represented as an unsigned integer between 0 and F ₁₆ . It shall be set to 0 ₁₆ if the message is not segmented (Note 2).
	Segment number		[7:4]	Segment number, represented as an unsigned integer between 0 ₁₆ and F ₁₆ ; set to 0 ₁₆ for the first segment and if message is not segmented (Note 2).
	Sequence number	5 and 6	[15:0]	Sequence number of the segmented message in a format of 16-bit unsigned integer (Notes 2 and 3).
	Repetition number	7	[3:0]	Repetition number of the message formatted as a 4-bit unsigned integer whose initial value is 0. Each time a message is retransmitted (See clause 8.10.1.2) by the originating node this field shall be incremented.
	FSB		[4]	Force Sequence Bit. See clause 8.10.1.2
	Reserved		[7:5]	Reserved by ITU-T (Note 4).
MMPL	Message Parameters	8 to (LG+7)	[(8×LG–1):0]	Depends on the OPCODE, see Table 8-88.
NOTE 1 – The OPCODES are defined in Table 8-88.				

Table 8-87 – Format of management messages

	Content	Octet	Bits	Description
NOTE 2 – This field is not applicable for a MAP message, and shall be set to zero.				
NOTE 3 – The meaning of the sequence number depends on the OPCODE. See clause 8.10.1.2.				
NOTE 4 – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.				

8.10.1.1 Management message OPCODEs

Management message OPCODEs are formatted as 12-bit unsigned integers. Valid values of OPCODEs are presented in Table 8-88. OPCODEs are categorized (typically by their associated protocol or procedure) according to the value of their eight MSBs.

Table 8-88 – OPCODEs of management messages

Category	Message name	OPCODE (hex)	Description	MMPL Reference
Admission (01X)	ADM_NodeRegistrRequest.req	010	Registration request	Clause 8.6.1.1.4.1
	ADM_DmRegistrResponse.cnf	011	Registration response	Clause 8.6.1.1.4.2
	ADM_NodeResignRequest.req	012	Resignation request	Clause 8.6.1.1.4.3
	ADM_DmResign.cnf	013	Registration announcement	Clause 8.6.1.1.4.4
	ADM_DmForcedResign.req	014	Forced resignation request	Clause 8.6.1.1.4.5
	ADM_NodeReRegistrRequest.req	015	Periodic re-registration request	Clause 8.6.1.1.4.6
	ADM_DmReRegistrResponse.cnf	016	Periodic re-registration response	Clause 8.6.1.1.4.7
	ADM_DmReRegistrInitiate.ind	017	Re-registration initiation request	Clause 8.6.1.1.4.8
	ADM_NodeReportMAPD.ind	018	Report the reception of a MAP-D with matching domain name	Clause 8.6.6.1.4.1
	ADM_NodeReportMAPA.ind	019	Report the reception of a MAP-A with matching DNI	Clause 8.6.6.1.4.2
	Reserved	01A-01F	Reserved by ITU-T	
AKM (02X)	AUT_NodeAuthentication.req	020	Request for authentication	Clause 9.2.5.1.1
	AUT_Prompt.ind	021	Delivers authentication prompt	Clause 9.2.5.1.2
	AUT_Verification.rsp	022	Authentication prompt verification	Clause 9.2.5.1.3
	AUT_Confirmation.cnf	023	Authentication confirmation message	Clause 9.2.5.1.4
	AKM_KeyRequest.req	024	Request for secure communication with another node(s)	Clause 9.2.5.2.1
	AKM_NewKey.req	025	Message delivers the encryption key to the supplicant node	Clause 9.2.5.2.2
	AKM_KeyConfirmation.req	026	Message delivers the encryption key to the addressee node(s)	Clause 9.2.5.2.4

Table 8-88 – OPCODEs of management messages

Category	Message name	OPCODE (hex)	Description	MMPL Reference
	AKM_KeyUpdate.req	027	Request for re-authentication and update the keys	Clause 9.2.5.3.1
	AKM_NewKey.cnf	028	Addressee confirmation that encryption key was delivered	Clause 9.2.5.2.3
	SC_DMRes.req	029	Request to resign a node from the domain	Clause 9.2.5.2.5
	SC_DMRes.cnf	02A	Confirmation of resignation from the domain master	Clause 9.2.5.2.6
	AKM_AddClient.req	02B	Request to join a node to a multicast group	Clause 9.2.5.2.1.1
	Reserved	02C	Reserved by ITU-T	
	AKM_NewKey.ind	02D	Indication that the new encryption key is available for use	Clause 9.2.5.2.7
	AKM_DomainKeyUpdate.req	02E	Request to update the domain-wide encryption key, from SC to DM	Clause 9.2.5.3.2
	AKM_DomainKeyUpdate.cnf	02F	Confirmation for the request to update the domain-wide encryption key, from DM to SC	Clause 9.2.5.3.3
Topology maintenance (03X)	TM_NodeTopologyChange.ind	030	Topology report from a node	Clause 8.6.4.3.1
	TM_NodeTopologyChange.req	031	Request sent by the domain master to a particular node requesting its topology report	Clause 8.6.4.3.2
	TM_NodeTopologyChange.cnf	032	Topology report from a node in response to the message TM_NodeTopologyChange.req	Clause 8.6.4.3.3
	TM_DomainRoutingChange.ind	033	Optimal routing update from the domain master	Clause 8.6.4.3.5
	TM_ReturnDomainRouting.req	034	Request for routing update from the node to the domain master	Clause 8.6.4.3.6
	TM_ReturnDomainRouting.cnf	035	Reply on routing request by the Domain master	Clause 8.6.4.3.7
	TM_DMBBackup.ind	036	Topology report from a node sent by backup domain master to a node	Clause 8.6.4.3.4
	Reserved	037-03F	Reserved by ITU-T	

Table 8-88 – OPCODEs of management messages

Category	Message name	OPCODE (hex)	Description	MMPL Reference
Power-line coexistence with alien networks (04X)	Reserved for use by [ITU-T G.9972]	040-04F		
Multicast binding (05X)	MC_GrpInfoUpdate.ind	050	Multicast binding information update	Clause 8.16.5.1
	MC_GrpInfoUpdate.cnf	051	Multicast binding information update confirmation	Clause 8.16.5.2
	MC_GrpRemove.req	052	Multicast leave request from the transmitter	Clause 8.16.5.3
	MC_GrpRemove.cnf	053	Multicast leave confirmation from the receiver	Clause 8.16.5.4
	DMC_Path.req	054	DLL multicast path establishment request	Clause 8.17.6.1
	DMC_Path.cnf	055	DLL multicast path establishment confirmation	Clause 8.17.6.2
	DMC_PathReject.cnf	056	DLL multicast path establishment rejection	Clause 8.17.6.3
	DMC_EnforcePath.req	057	DLL multicast enforced path establishment request	Clause 8.17.6.4
	DMC_ReleasePath.req	058	A request to release a DLL multicast client node from its MSID	Clause 8.17.6.5
	DMC_ReleasePath.cnf	059	Confirmation of the release of a DLL multicast client node from its MSID	Clause 8.17.6.6
	DMC_PathAlive.ind	05A	DLL multicast path alive indication	Clause 8.17.6.7
	DMC_BrokenLink.ind	05B	DLL multicast broken link indication	Clause 8.17.6.8
	Reserved		05C-05F	Reserved by ITU-T
Domain master selection and backup domain master (06X)	DM_Handover.req	060	Domain master role handover request	Clause 8.6.6.5.1
	DM_Handover.cnf	061	Domain master role handover confirmation	Clause 8.6.6.5.2
	DM_Handover.ind	062	Domain state update	Clause 8.6.6.5.3
	DM_Handover.rsp	063	Domain state update confirmation	Clause 8.6.6.5.4
	DM_BackupAssign.req	064	Backup domain master assignment request	Clause 8.6.5.2
	DM_BackupAssign.cnf	065	Backup domain master assignment confirmation	Clause 8.6.5.2
	DM_BackupData.ind	066	Domain state update	Clause 8.6.5.2
	DM_BackupRelease.req	067	Release of a backup domain master	Clause 8.6.5.2

Table 8-88 – OPCODES of management messages

Category	Message name	OPCODE (hex)	Description	MMPL Reference
	DM_BackupRelease.cnf	068	Backup domain master release confirmation	Clause 8.6.5.2
	DM_HandoverRequest.ind	069	Endpoint node indication of need for domain master handover	Clause 8.6.6.5.5
	Reserved	06A-06F	Reserved by ITU-T	
Channel estimation (07X)	CE_ProbeSlotAssign.req	070	Channel estimation bandwidth assignment request	Clause 8.11.7.1
	CE_ProbeSlotRelease.req	071	Channel estimation bandwidth release request	Clause 8.11.7.2
	CE_ParamUpdate.req	072	Channel estimation parameters update request	Clause 8.11.7.3
	CE_ParamUpdateRequest.ind	073	Request for channel estimation parameter update	Clause 8.11.7.4
	CE_PartialBatUpdate.req	074	Partial BAT update request	Clause 8.11.7.5
	CE_ACESymbols.ind	075	Request for an ACE symbol attachment	Clause 8.11.7.6
	CE_ProbeSlotAssign.cnf	076	Channel estimation bandwidth assignment confirmation	Clause 8.11.7.7
	CE_ProbeSlotRelease.cnf	077	Channel estimation bandwidth release confirmation	Clause 8.11.7.8
	CE_ParamUpdate.cnf	078	Channel estimation parameters update confirmation	Clause 8.11.7.9
	CE_PartialBatUpdate.cnf	079	Partial BAT update confirmation	Clause 8.11.7.10
Reserved	07A-07F	Reserved by ITU-T		
Reserved (08X)	Reserved	080-08F	Reserved by ITU-T	
Inactivity scheduling (09X)	IAS_LongInactivity.req	090	Long inactivity scheduling request	Clause 8.3.6.1.1
	IAS_LongInactivity.cnf	091	Long inactivity scheduling confirmation	Clause 8.3.6.1.1
	IAS_ShortInactivity.req	092	Short inactivity scheduling request	Clause 8.3.6.2.1
	IAS_ShortInactivity.cnf	093	Short inactivity scheduling confirmation	Clause 8.3.6.2.1
	IAS_LongInactivity.ind	094	Long inactivity scheduling indication	Clause 8.3.6.1.1
	IAS_ShortInactivity.ind	095	Short inactivity scheduling indication	Clause 8.3.6.2.1
	Reserved	096-09F	Reserved by ITU-T	
	CL_EstablishFlow.req	0A0	Reserved by ITU-T	Clause 8.6.2.3.1

Table 8-88 – OPCODES of management messages

Category	Message name	OPCODE (hex)	Description	MMPL Reference
Flow establishment (0AX)	CL_EstablishFlow.cnf	0A1	Reserved by ITU-T	Clause 8.6.2.3.2
	FL_AdmitFlow.req	0A2	Flow admission request	Clause 8.6.2.3.8
	FL_AdmitFlow.cnf	0A3	Flow admission confirmation	Clause 8.6.2.3.9
	FL_AdmitFlow.ind	0A4	Flow admission indication	Clause 8.6.2.3.10
	FL_AdmitFlow.rsp	0A5	Flow admission acknowledgement	Clause 8.6.2.3.18
	FL_OriginateFlow.req	0A6	Flow origination request	Clause 8.6.2.3.6
	FL_OriginateFlow.cnf	0A7	Flow origination confirmation	Clause 8.6.2.3.7
	Reserved	0A8-0AF	Reserved by ITU-T	
Flow maintenance (0BX)	FL_ModifyFlowParameters.req	0B0	Modification of flow parameters and allocation	Clause 8.6.2.3.11
	FL_ModifyFlowParameters.cnf	0B1		Clause 8.6.2.3.12
	FL_ModifyFlowParameters.ind	0B2		Clause 8.6.2.3.15
	FL_ModifyFlowAllocations.req	0B3	Modification of flow allocation	Clause 8.6.2.3.16
	FL_ModifyFlowAllocations.cnf	0B4		Clause 8.6.2.3.17
	Reserved	0B5-0BF	Reserved by ITU-T	
Flow termination (0CX)	CL_TerminateFlow.req	0C0	Flow termination request and confirmation	Clause 8.6.2.3.3
	CL_TerminateFlow.cnf	0C1		Clause 8.6.2.3.4
	CL_FlowTerminated.ind	0C2		Clause 8.6.2.3.5
	FL_TerminateFlow.req	0C3	Request flow termination	Clause 8.6.2.3.13
	FL_TerminateFlow.cnf	0C4	Confirm flow termination	Clause 8.6.2.3.14
	FL_BrokenTunnel.ind	0C5	Indicate broken tunnel	Clause 8.6.2.3.19
	FL_BrokenTunnel.rsp	0C6	Response to indication	Clause 8.6.2.3.20
	FL_ReleaseTunnel.req	0C7	Request Release Tunnel	Clause 8.6.2.3.21
	FL_ReleaseTunnel.cnf	0C8	Confirm Release Tunnel	Clause 8.6.2.3.22
	FL_DM_RenewTunnel.req	0C9	DM renew tunnel request	Clause 8.6.2.3.23
	FL_DM_RenewTunnel.cnf	0CA	Confirm DM renew tunnel	Clause 8.6.2.3.24
	FL_RenewTunnel.req	0CB	Renew tunnel request	Clause 8.6.2.3.25
	FL_RenewTunnel.cnf	0CC	Confirm Renew tunnel	Clause 8.6.2.3.26
	FL_DeleteFlow.req	0CD	Delete Flow request	Clause 8.6.2.3.27
	FL_DeleteFlow.cnf	0CE	Confirm Delete Flow	Clause 8.6.2.3.28
	CL_TerminateFlow.ind	0CF	Indicate flow needs to be terminated	Clause 8.6.2.3.29
Media Access Plan (0DX)	MAP	0D0	MAP message	Clause 8.8
	Reserved	0D1-0DF	Reserved by ITU-T	
Channel Estimation 2 (0EX)	CE_Request.ind	0E0	Channel estimation trigger	Clause 8.11.7.11
	CE_Initiation.req	0E1	Channel estimation initiation request	Clause 8.11.7.12

Table 8-88 – OPCODES of management messages

Category	Message name	OPCODE (hex)	Description	MMPL Reference
	CE_Initiation.cnf	0E2	Channel estimation initiation confirmation	Clause 8.11.7.13
	CE_ProbeRequest.ind	0E3	Request for PROBE frame transmission	Clause 8.11.7.14
	CE_Cancellation.req	0E4	Channel estimation cancellation request	Clause 8.11.7.15
	CE_BatIdMaintain.ind	0E5	BAT ID maintenance	Clause 8.11.7.16
	CE_Cancellation.cnf	0E6	Channel estimation cancellation confirmation	Clause 8.11.7.17
	Reserved		0E7-0EF	Reserved by ITU-T
Secure Admission (OFX)	ADM_NodeReportPUSH_P.ind	0x0F0	Indication from a secure EP to its DM that the pairing window should open.	Clause 6.2.2.1.4.1 of [ITU-T G.9978]
	ADM_NodeReportPUSH_P.rsp	0x0F1	Response from a secure DM to the EP that requested to open the pairing window.	Clause 6.2.2.1.4.2 of [ITU-T G.9978]
	Reserved for use by HomeGrid Forum	0x0F2	Reserved for use by HomeGrid Forum	
	Reserved for use by HomeGrid Forum	0x0F3	Reserved for use by HomeGrid Forum	
	ADM_PairingWindow.ind	0x0F4	Broadcast message to indicate opening and closing of pairing window.	Clause 6.2.2.1.4.3 of [ITU-T G.9978]
	Reserved		0F5-0FF	Reserved by ITU-T
Neighbouring network coordination (10X to 13X)	NDIM_Start Alignment Procedure.ind	100	Request to start a MAC cycle alignment procedure (DM to proxy node)	Clause 8.14.9.1
	NDIM_IDCC Reserve.req	101	Slot reservation request	Clause 8.14.9.2
	NDIM_IDCC Reserve.cnf	102	Slot reservation confirmation	Clause 8.14.9.3
	NDIM_Report Alignment.req	103	Report on MAC cycle alignment	Clause 8.14.9.4
	NDIM_ReportAlignment.cnf	104	Confirm receiving NDIM_ReportAlignment.req	Clause 8.14.9.5
	NDIM_Remote Presence.req	105	Request to respond to ID_PresenceRequest	Clause 8.14.9.6
	NDIM_Remote Presence.cnf	106	Permission to respond to ID_PresenceRequest	Clause 8.14.9.7
	NDIM_Transmit.ind	107	DM to proxy node message to be transmitted to neighbouring domain	Clause 8.14.9.8
	NDIM_Receive.ind	108	Proxy node to DM message received from neighbouring domain	Clause 8.14.9.9

Table 8-88 – OPCODEs of management messages

Category	Message name	OPCODE (hex)	Description	MMPL Reference
	NDIM_InterferenceReport.ind	109	Indication of interference detected	Clause 8.14.9.12
	NDIM_IDCC_Release.req	10A	Release Slot reservation	Clause 8.14.9.10
	NDIM_IDCC_Release.cnf	10B	Confirm receiving NDIM_IDCC_Release.req	Clause 8.14.9.11
	Reserved	10C-10F	Reserved by ITU-T	
	IDM_ClusterAlignment.req	120	DM informs other DMs about new cluster alignment	Clause 8.14.10.1
	IDM_ClusterAlignment.cnf	121	DM confirm receiving IDM_ClusterAlignment.req	Clause 8.14.10.2
	IDM_InterfNodesInfo.ind	122	Proxy node to neighbouring domains indication of interfering nodes	Clause 8.14.10.3
	IDM_CoordDomainsInfo.ind	123	Proxy node to neighbouring domains indication of coordinating nodes	Clause 8.14.10.5
	IDM_ShareUnallocSlot.req	124	Request to share unallocated slots	Clause 8.14.10.7
	IDM_ShareUnallocSlot.cnf	125	Confirmation of request to share unallocated slots	Clause 8.14.10.8
	IDM_ShareUnallocSlot.ind	126	Indication of status of the request to share unallocated slots	Clause 8.14.10.9
	IDM_RequestUnallocSlot.req	127	Request assignment of unallocated slots	Clause 8.14.10.10
	IDM_RequestUnallocSlot.cnf	128	Confirmation of request for assignment of unallocated slots	Clause 8.14.10.11
	IDM_RequestUnallocSlot.ind	129	Indication of status of the request for assignment of unallocated slots	Clause 8.14.10.12
	IDM_SwapAllocSlot.req	12A	Request to swap allocated slots	Clause 8.14.10.13
	IDM_SwapAllocSlot.cnf	12B	Confirmation of the request to swap allocated slots	Clause 8.14.10.14
	IDM_SwapAllocSlot.ind	12C	Indication of status of the request to swap allocated slots	Clause 8.14.10.15
	IDM_CoordPref.ind	12D	Indication of preferred coordination method	Clause 8.14.10.16
	IDM_DmChange.ind	12E	Indication to neighbouring domain masters that the DM of the domain sending the message has changed	Clause 8.14.10.17

Table 8-88 – OPCODEs of management messages

Category	Message name	OPCODE (hex)	Description	MMPL Reference
	IDM_DniChange.ind	12F	Indication to neighbouring domain masters that the DNI of the domain sending the message has changed	Clause 8.14.10.18
	IDM_InterfNodesInfo.rsp	130	A message sent as a confirmation for a received IDM_ InterfNodesInfo.ind	Clause 8.14.10.4
	IDM_CoordDomainsInfo.rsp	131	A message sent as a confirmation for a received IDM_ CoordDomainsInfo.ind	Clause 8.14.10.6
	Reserved	132-13F	Reserved by ITU-T	
AKM 2 (14X)	AUT_NodeAuthenticated.req	140	Indication from the SC to DM that the node has been authentication	Clause 9.2.5.1.5
	AUT_NodeAuthenticated.cnf	141	Confirmation of the AUT_NodeAuthenticated.req message	Clause 9.2.5.1.6
	SC_NewSCDesignation.req	142	New SC designation request for SC re-establishment	Clause 9.2.6.3.1
	SC_NewSCDesignation.cnf	143	confirmation of SC_NewSCDesignation.req	Clause 9.2.6.3.2
	Reserved	144-14F	Reserved by ITU-T	
Power saving modes (15X)	LP_RequestWakeUp.req	150		Clause 8.3.6.1.1
	Reserved	151-15F	Reserved by ITU-T	
L2 configuration and management protocol (16X)	LCMP_Read.req	160	Read request via LCMP protocol	Clause 8.22.3.1
	LCMP_Read.cnf	161	Read confirmation via LCMP protocol	Clause 8.22.3.2
	LCMP_Write.req	162	Write request via LCMP protocol	Clause 8.22.3.3
	LCMP_Write.cnf	163	Write confirmation via LCMP protocol	Clause 8.22.3.4
	LCMP_Ctrl.req	164	Control operation request via LCMP protocol	Clause 8.22.3.5
	LCMP_Ctrl.cnf	165	Control operation confirmation via LCMP	Clause 8.22.3.6
	LCMP_Notify.ind	166	Notify operation request via LCMP protocol	Clause 8.22.3.7
	LCMP_Notify.rsp	167	Notify operation confirmation via LCMP	Clause 8.22.3.8
	LCMP.ind	168	For further study	Clause 8.22.3.9
	LCMP.rsp	169	For further study	Clause 8.22.3.10
	Reserved	16A-16F	Reserved by ITU-T	

Table 8-88 – OPCODEs of management messages

Category	Message name	OPCODE (hex)	Description	MMPL Reference
Flow termination (17X)	FL_TerminateTunnel.req	170	Terminate tunnel request	Clause 8.6.2.3.30
	FL_TerminateTunnel.cnf	171	Terminate tunnel confirmation	Clause 8.6.2.3.31
	Reserved	172-17F	Reserved by ITU-T	
Bandwidth update protocol (18X)	BU_BWUpdate.req	180	Request for bandwidth update for prioritized connections	Clause 8.6.2.4.1.1
	BU_BWUpdate.cnf	181	Confirmation for bandwidth update for prioritized connections	Clause 8.6.2.4.1.2
	Reserved	182-18F	Reserved by ITU-T	
Reserved	Reserved	190-7FF	Reserved by ITU-T	
MIMO (8XX – 9XX)	Reserved for use by [ITU-T G.9963]	800-9FF	Reserved by ITU-T	
HGF (AXX – A3X)	Reserved for use by HomeGrid Forum	A00-A3F	Reserved for use by HomeGrid Forum	
Reserved	Reserved	A40-AFF	Reserved by ITU-T	
Secure Admission (B0X)	Reserved for use by [ITU-T G.9978]	B00-B0F	Reserved by ITU-T	
Reserved	Reserved	B10-BFF	Reserved by ITU	
Transmission profile (C0X)	TP_TransmitPsd Change.req	C00	Transmit PSD mask change request	Clause 8.6.9.1
	TP_TransmitPsd Change.cnf	C01	Transmit PSD mask change confirmation	Clause 8.6.9.2
Reserved	Reserved	C02-FFF	Reserved by ITU-T	

8.10.1.2 Management of message sequence numbers and segmentation

The sequence number space shall be unique for each {OPCODE, OriginatingNode} tuple. The sequence number shall be incremented for each transmitted message except as follows:

- When the same message is retransmitted (e.g., when a message has been lost), the message sequence number shall be the same as the original transmitted message and the repetition number shall be incremented by 1;
- When a message is relayed, the sequence number and the repetition number fields shall not be modified.

NOTE 1 – The sequence number space used by an originating node for a given OPCODE is the same regardless of the destination (e.g., single counter per OPCODE).

When the field Force Sequence Bit (FSB) of the MMH is set to one, it indicates that the receiver shall process this message without performing any sequence filtering. The receiver shall also consider the sequence number of this message as the latest valid sequence number associated with the transmitter's DeviceID and OPCODE of the message.

NOTE 2 – The increment in the value of the message sequence number is independent of the value of the FSB field.

The following segmentation rules apply to any segmented management message:

- The segmentation shall be done in the ascending order of octets;
- All the segments shall have the same sequence number;
- The segmentation shall not be changed if the management message is retransmitted, unless a new sequence number is generated;
- The segmentation shall not be changed if the management message is relayed (the sequence number shall remain the same).

Segmentation shall only be done for management message with payload (MMPL or CMMPL) that lead to LCDUs greater than 1500 bytes.

Some management protocols may require knowing if the sequence number of a received LCDU is older, equal or newer than the last correctly received LCDU. The sequence number is a 16-bit unsigned integer used for that purpose and shall be in the range 0 to (SequenceModulus –1), where SequenceModulus is equal to 2^{16} . When it is equal to 2^{16} , it wraps-around to zero. If the FSB field of the MMH is set to one, the received LCDU shall be considered as the newest. If the FSB field of the MMH is set to zero, sequence numbers of LCDUs with the same OPCODE shall be compared according to the following rules:

- The first LCDU received from a node shall be considered as a new message containing new information. The node shall perform the operations required by the protocol that defines that OPCODE;
- If the sequence number of the new received LCDU is the same as the sequence number of the LCDU already kept by the node, the new received LCDU shall be considered to be equal to the LCDU kept by the node;
- If the sequence number of the new received LCDU is higher than the sequence number of the LCDU already kept by the node and the difference between the numbers is, in absolute value, less than half of SequenceModulus, the new received LCDU shall be considered to be newer. Otherwise it shall be considered to be older;
- If the sequence number of the new received LCDU is lower than the sequence number of the LCDU already kept by the node and the difference between the numbers is, in absolute value, lower than half of SequenceModulus, the new received LCDU shall be considered to be older. Otherwise it shall be considered to be newer.

In any of the above cases, the actions to perform by the node that receives the LCDU depend on the protocol that defines that OPCODE.

NOTE 3 – A transmitter may use the FSB bit to force synchronization with the receiver. Once the transmitter gets confirmation that the receiver is synchronized, it should set FSB to zero.

8.10.2 Control message format

This clause describes the format of short control messages, intended for communication between nodes of the same domain. All control messages carried over CTMG frames (clause 7.1.2.3.2.6/10.1.2.3.2.6 of [ITU-T G.9960]) shall be formatted as shown in Figure 8-55, including a control message header (CMH) and a control message parameter list (CMPL). A control message is carried in the PHY-frame header of CTMG frame, hence protected by the HCS and E_HCS (clauses 7.1.2.3.1.9/10.1.2.3.1.9 and 7.1.2.3.3.2/10.1.2.3.3.2 of [ITU-T G.9960]). The control messages carried over CTMG frames are not subject to relay. The first byte (octet 0) of the CMH shall be the first byte passed to the PHY layer. A CTMG frame transmitted in CBTS shall be considered as having an MPDU priority equal to 7.

CMH	Control Message Parameter List (CMPL)
-----	---------------------------------------

G.9961(10)_F8-55

Figure 8-55 – Format of a control message

The CMH defines the length and other parameters of the message. The type of the message is identified by an OPCODE associated with a particular control function as presented in Table 8-90. The CMPL includes a list of control message parameters depending on the control function. The format of control message shall be as shown in Table 8-89.

Table 8-89 – Format of control messages

	Content	Octet	Bits	Description
CMH	Length	0 and 1	[5:0]	Length of the CMPL in octets (V), encoded as a 6-bit unsigned integer. The valid range of V is 1 to 31.
	OPCODE		[15:6]	10-bit OPCODE, indicates control message type (Note 1).
	Reserved	2	[7:0]	Reserved by ITU-T (Note 2).
CMPL	Message parameters	3 to ($V+2$)	$[(8V-1):0]$	Depends on the OPCODE.
NOTE 1 – The OPCODEs are defined in Table 8-90.				
NOTE 2 – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.				

The format of CMPLs may be revised in future versions of this Recommendation by appending additional fields. Furthermore, fields may be defined using bits that are currently indicated as reserved for ITU-T. Nodes indicate the node information via NodeInformationTLVs during registration (see Table 8-16) and topology updates (see Table 8-47). Nodes shall be able to parse the CMPL (the length of the CMPL is specified in the CMH) but shall ignore the content of fields that they do not understand, i.e., those associated with later versions of the Recommendation.

8.10.2.1 Control message OPCODEs

Control message OPCODEs are formatted as 10-bit unsigned integers. Valid values of OPCODEs are for further study.

Table 8-90 – Placeholder table

(This table has been intentionally left blank)

8.11 Channel estimation protocol

The channel estimation protocol describes the procedure of measuring the characteristics of the channel between the transmitter (source) and the receiver (destination) nodes. The procedure involves initiation of channel estimation, transmissions of PROBE frames, and selection of parameters.

Channel estimation can be done in two phases:

- Channel discovery – Initial channel estimation;
- Channel adaptation – Subsequent channel estimation to adapt changing channel.

The protocols used for channel discovery and channel adaptation can be started either by the transmitter or by the receiver. The core part of the channel estimation protocol is identical in these two cases, and is always initiated by the receiver (receiver-initiated channel estimation). The transmitter can request the receiver to initiate channel estimation (transmitter-requested channel estimation).

During the initiation process, the transmitter and receiver jointly determine input parameters for channel estimation such as channel estimation window (a fraction of a MAC cycle over which channel estimation should be executed), the minimum value of G (G_{\min} , see clause 7.1.4.2.4/10.1.4.2.4 of [ITU-T G.9960]), and parameters for the PROBE frame. The receiver selects the BAT_ID associated with the BAT to be updated. This BAT_ID is used for an identifier for a particular channel estimation process throughout the rest of the process. The receiver shall consider its own bandplan-frequency information (namely the StartSubCarrier & StopSubCarrier) and that of the transmitter when calculating the BAT. More specifically, the range of sub-carriers of the BAT sent in the CE_ParamUpdate.req message shall be within the intersection of the sub-carrier ranges determined by the StartSubCarrier & StopSubCarrier of both the receiver and transmitter.

Once the channel estimation process is initiated, the receiver may request the transmitter to send one or more PROBE frames. The receiver can change parameters of a PROBE frame at each request. If the receiver requests a PROBE frame without specifying its parameters (e.g., request for PROBE frame transmission request via ACK_CE_CTRL as described in clause 8.11.1.4), the transmitter transmits the PROBE frame using parameters previously selected by the receiver. The receiver is not required to request PROBE frames if it chooses other means such as MSG frames or PROBE frames transmitted to other nodes to estimate the channel.

The receiver terminates the channel estimation process by sending the outcome of channel estimation to the transmitter. This includes, but is not limited to, the following parameters:

- Bit allocation table (BAT);
- FEC coding rate and block size;
- Guard interval for payload;
- PSD ceiling.

The receiver may cancel the channel estimation process without generating new channel estimation parameters.

The protocol provides several options to expedite the channel estimation process for faster channel adaptation. For example, the channel estimation initiation process (clause 8.11.1.1) can be omitted in case of channel adaptation where no new input parameter negotiation is necessary. The receiver can create a new BAT by sending an unsolicited CE_ParamUpdate.req (clause 8.11.3.1) or update the existing BAT by sending a CE_PartialBatUpdate.req (clause 8.11.3.2). The receiver can request PROBE frame transmission without going through the channel estimation initiation process (clause 8.11.4).

8.11.1 Receiver-initiated channel estimation

The following procedure describes the receiver-initiated channel estimation process:

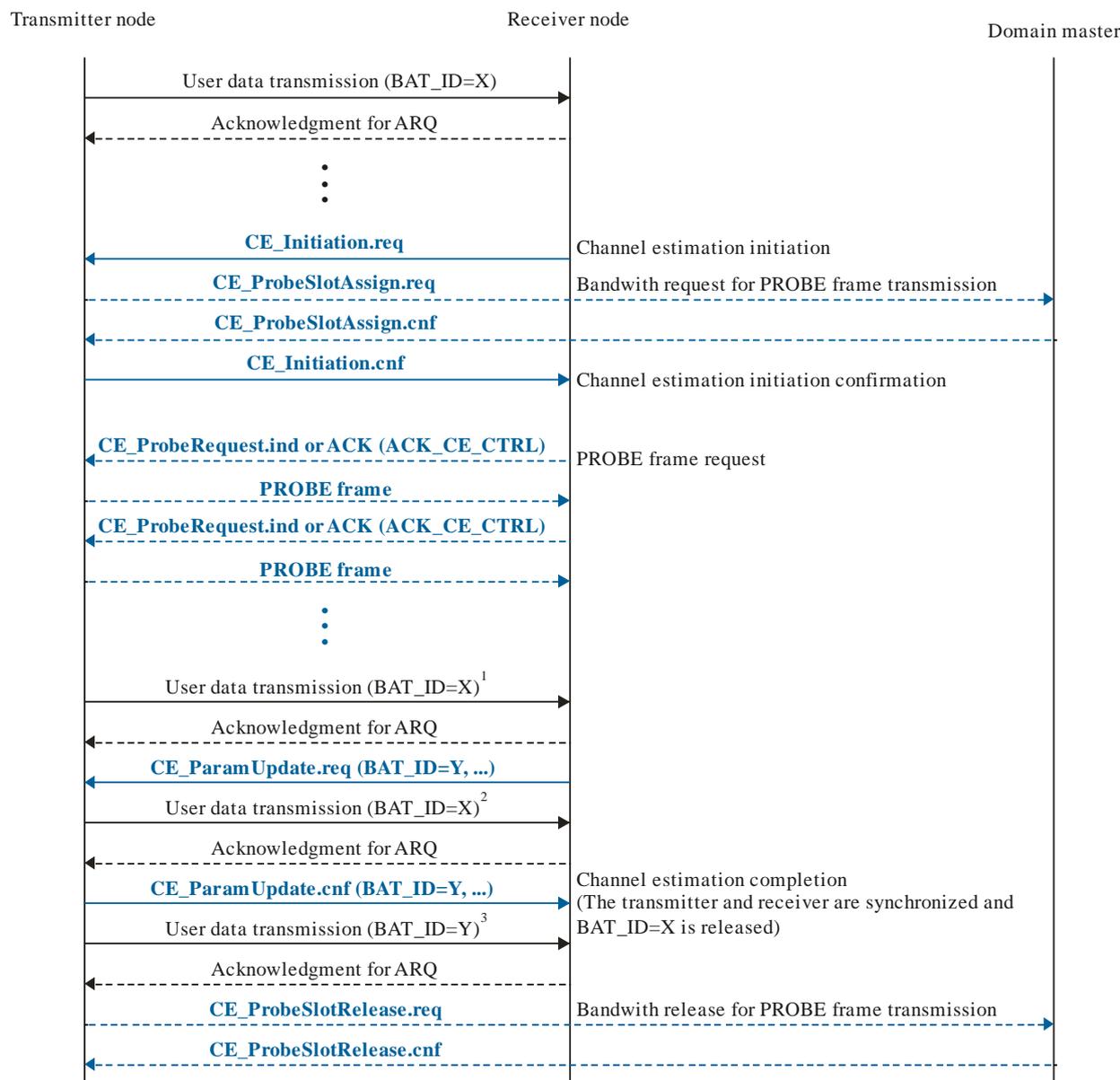
- 1) The receiver initiates the channel estimation process by sending the transmitter a CE_Initiation.req message. The receiver may request a PROBE frame transmission in this message (channel estimation initiation, see clause 8.11.1.1).
- 2) Upon reception of the channel estimation initiation request, if the transmitter does not have transmit opportunities for a given channel estimation window, it shall request the domain master to allocate bandwidth for PROBE frame transmission by sending a CE_ProbeSlotAssign.req message. The domain master shall confirm that it received the

bandwidth request by replying with the CE_ProbeSlotAssign.cnf message (bandwidth request, see clause 8.11.1.2).

- 3) Depending on the availability of the bandwidth, the transmitter may grant or reject the channel estimation initiation request by sending the receiver a CE_Initiation.cnf message (channel estimation initiation confirmation, see clause 8.11.1.3).
- 4) Upon reception of the CE_Initiation.cnf message indicating channel estimation initiation confirmation, the receiver may request the transmitter to send additional PROBE frames by sending a CE_ProbeRequest.ind message or through the ACK_CE_CTRL field in the PFH of an ACK frame (request for PROBE frame transmission, see clause 8.11.1.4).
- 5) Upon reception of the request for PROBE frame transmission, the transmitter shall transmit the PROBE frame as the receiver requested (PROBE frame transmission, see clause 8.11.1.5).
- 6) Steps 4 and 5 can repeat until the receiver sends the transmitter the final outcome of channel estimation using the CE_ParamUpdate.req message. The transmitter shall confirm reception of the new parameters by sending a CE_ParamUpdate.cnf message (channel estimation completion, see clause 8.11.1.6). Steps 4 and 5 may be skipped altogether if the receiver does not need additional PROBE frames.
- 7) The receiver may cancel the channel estimation process anytime after it receives the channel estimation initiation confirmation by sending a CE_Cancellation.req message (channel estimation cancellation, see clause 8.11.1.7).
- 8) Upon reception of a CE_ParamUpdate.req message, if the transmitter has been allocated extra bandwidth for the PROBE frame transmission, it shall send a CE_ProbeSlotRelease.req message to the domain master to release the bandwidth used for PROBE frame transmission. The domain master shall confirm the bandwidth release request by replying with a CE_ProbeSlotRelease.cnf message (bandwidth release, see clause 8.11.1.8).

The transmitter may send frames carrying payload with the existing settings (e.g., any valid runtime BAT or predefined BAT) at any time during this process.

The receiver-initiated channel estimation process is illustrated in Figure 8-56.



G.9961(10)-Cor.1(11)_F8-56

- 1) The transmitter can transmit data using the existing BAT anytime during channel estimation process.
- 2) 1st user data transmission after CE_ParamUpdate.ind may still use old channel estimation parameters.
- 3) The transmitter decides when to apply updated channel estimation parameters within a given constraint.

NOTE – Dotted-lines indicate optional communications.

Figure 8-56 – Receiver-initiated channel estimation

8.11.1.1 Channel estimation initiation

The receiver initiates the channel estimation process by sending the transmitter a CE_Initiation.req message.

The receiver shall select CE_GRP (*G*), which indicates the value of GRP_ID (*G*) associated with the BAT to be updated. The receiver shall select CE_STIME and CE_ETIME, which determines the start and end time of the channel estimation window. During the rest of channel estimation process, the transmitter shall send PROBE frames inside this window. The receiver shall select CE_BAT_ID from ones that are currently invalid. This value shall be used to differentiate multiple channel estimation processes being executed at the same time. The receiver may request PROBE frame transmission by setting CE_PRB_RQST field. The CE_PRB_PARM field specifies parameters for the default PROBE frame. If the CE_PRB_RQST field is not set to one, parameters for the default PROBE frame

shall be as follows: CE_PR_PRBTYPE = 0001₂; CE_PR_PRBFN = 0000₂; CE_PR_PRBSYM = 0011₂; CE_PR_PRBGI = 111₂, CE_PR_APSD_MAX = 31, and CE_PR_APSD_MAX_EXT = 0.

The receiver may resend the CE_Initiation.req message, if it does not receive the CE_Initiation.cnf message within 200 ms.

8.11.1.2 Channel estimation bandwidth request

If the transmitter does not have transmit opportunities inside a given channel estimation window, it shall request the domain master to allocate bandwidth for a PROBE frame transmission by sending a CE_ProbeSlotAssign.req message.

The transmitter shall provide the domain master the channel estimation identifier (i.e., CE_BAT_ID, Transmitter_ID, and Receiver_ID), channel estimation window (CE_STIME and CE_ETIME), and PROBE frame parameters (CE_PRB_PARM) as provided by CE_Initiation.req message.

The transmitter shall provide the priority of the bandwidth request in the CE_ProbeSlotAssign.req message by setting the CE_PRIORITY field to the highest priority of the user data traffic that the transmitter has to send to the specified receiver.

The domain master shall confirm the bandwidth request by replying to the transmitter with a CE_ProbeSlotAssign.cnf indicating whether or not the request is granted within 100 ms after it receives the CE_ProbeSlotAssign.req message.

The domain master should allocate bandwidth so that at least one PROBE frame with requested parameters can be transmitted during the channel estimation window. The additional TSs or TXOPs shall only be used for PROBE frame transmissions (see clause 8.8.4.1.1). If the domain master has granted extra bandwidth for PROBE frame transmission, it should keep this bandwidth until it receives the bandwidth release request from the transmitter (see clause 8.11.1.8).

If the transmitter does not receive the CE_ProbeSlotAssign.cnf message, it may resend the CE_ProbeSlotAssign.req message multiple times before it transmits the channel estimation initiation confirmation.

8.11.1.3 Channel estimation initiation confirmation

The transmitter shall confirm the channel estimation initiation request by sending the receiver a CE_Initiation.cnf message.

The transmitter shall indicate whether it grants or rejects the channel estimation initiation request. The transmitter shall set CE_BAT_ID to the value selected by the receiver in the CE_Initiation.req message, and shall set CE_GRP to be equal to the value indicated by the receiver.

The transmitter shall send a CE_Initiation.cnf message within 100 ms after it receives a CE_Initiation.req message. If the transmitter needs to request the bandwidth for PROBE frame transmission, the transmitter shall send a CE_Initiation.cnf message within 200 ms.

8.11.1.4 Request for PROBE frame transmission

Once a channel estimation initiation request has been confirmed, the receiver may request the transmitter to send additional PROBE frames by sending a CE_ProbeRequest.ind message.

The receiver can request specific parameters of the PROBE frame via the CE_PRB_PARM field of the CE_ProbeRequest.ind message.

Alternatively, the receiver may request PROBE frames by using the ACK_CE_CTRL field in the PHY-frame header of an ACK frame designated to the transmitter node (see clause 8.11.4).

The receiver may not request PROBE frames at all if it uses other frames carrying payload (e.g., MSG, BMSB, BACK) to estimate the channel.

8.11.1.5 PROBE frame transmission

Upon reception of a request for PROBE frame transmission, the transmitter shall transmit PROBE frames as soon as possible as described in clause 8.11.4.

8.11.1.6 Channel estimation completion

At any time after channel estimation initiation request has been confirmed, the receiver may send the transmitter the outcome of channel estimation using the CE_ParamUpdate.req message. The transmitter shall confirm reception of the new parameters by replying with the CE_ParamUpdate.cnf message within 100 ms.

Upon reception of the CE_ParamUpdate.req message, the transmitter shall incorporate the new channel estimation parameters (new BAT, etc.) as soon as possible.

If the transmitter does not receive a message that is related to channel estimation (i.e., CE_ProbeRequest.ind or CE_ParamUpdate.req), or does not receive a request for PROBE frame transmission via an ACK frame, within 200 ms after the channel estimation initiation request has been confirmed, it may send the receiver a CE_ParamUpdateRequest.ind message to request the receiver to resend the result of the specified channel estimation.

If the transmitter does not receive either CE_ParamUpdate.req or CE_Cancellation.req within 400 ms after the channel estimation initiation request has been confirmed, it shall abort the channel estimation process.

8.11.1.7 Channel estimation cancellation

At any time after channel estimation initiation request has been confirmed, the receiver may cancel the channel estimation process using CE_Cancellation.req message. The transmitter shall confirm receiving the cancellation request within 100 ms by replying with the CE_Cancellation.cnf message. If the receiver does not receive the CE_Cancellation.cnf message within 200 ms, it may resend the CE_Cancellation.req message.

If the receiver does not receive either CE_ParamUpdate.cnf or CE_Cancellation.cnf within 400 ms after the channel estimation initiation request has been confirmed, it shall abort the channel estimation process and consider the CE_BAT ID as invalid (see clause 8.11.5).

In this case, the channel estimation is finished without generating a new BAT.

8.11.1.8 Channel estimation bandwidth release

Upon reception of the CE_ParamUpdate.req or CE_Cancellation.req message, the transmitter shall request the domain master to release any bandwidth previously assigned for PROBE frame transmission by sending CE_ProbeSlotRelease.req message.

The transmitter shall provide the domain master the channel estimation identifier (i.e., CE_BAT_ID, Transmitter_ID, and Receiver_ID) and channel estimation window (CE_STIME and CE_ETIME) associated with the channel estimation process.

The domain master shall confirm receiving the CE_ProbeSlotRelease.req message within 100 ms by replying with the CE_ProbeSlotRelease.cnf message. If the domain master does not receive a CE_ProbeSlotRelease.req message from the transmitter within 800 ms after the bandwidth was assigned, it shall release the bandwidth allocated to the transmitter for PROBE frames. The domain master shall only release bandwidth additionally assigned to the transmitter for PROBE frame transmission for the associated channel estimation identifier.

8.11.2 Transmitter-requested channel estimation

The following procedure describes the transmitter-requested channel estimation process:

- 1) The transmitter requests channel estimation by sending the receiver CE_Request.ind message (channel estimation request, see clause 8.11.2.1).
- 2) The rest of the procedure is the same as described in clause 8.11.1 (step 1 through step 8).

The transmitter may send frames carrying payload with the existing settings (e.g., any valid runtime BAT or pre-defined BAT) any time during this process.

8.11.2.1 Channel estimation request

The transmitter triggers the channel estimation process by sending the receiver CE_Request.ind message.

The transmitter can specify the channel estimation window (CE_STIME and CE_ETIME). In this case the receiver shall use the same channel estimation window as the transmitter requested in the CE_Initiation.req message. Otherwise, the receiver can determine the channel estimation window at its own discretion.

The receiver shall respond to a CE_Request.ind message from the transmitter within 100 ms with either a CE_Initiation.req message or a CE_ParamUpdate.req message.

If the transmitter does not receive either the CE_Initiation.req message or the CE_ParamUpdate.req message within 200 ms after the CE_Request.ind message is sent, it may resend the channel estimation request message.

8.11.3 Shortened channel estimation processes

8.11.3.1 Unsolicited CE_ParamUpdate.req

It is not required to exchange PROBE frames between transmitter and receiver in order to exchange a new BAT between them. The receiver may send a new BAT at any time to the transmitter by sending a CE_ParamUpdate.req message, provided that the BAT_ID is invalid at the time of sending the new BAT. The receiver may use PROBE frames or other frames carrying payload (e.g., MSG, BMSG, BACK) transmitted to other nodes to estimate the channel.

Upon receiving the CE_ParamUpdate.req message, the transmitter shall reply by sending the CE_ParamUpdate.cnf message within 100 ms indicating whether the transmitter adopts the new BAT or rejects the new BAT due to lack of resources.

If the receiver does not receive CE_ParamUpdate.cnf within 200 ms, it may retry with the same or different CE_ParamUpdate.req message.

8.11.3.2 Partial BAT update

The transmitter and receiver that communicate with each other by establishing a common runtime BAT may update a portion of the BAT at any time during its usage. The receiver may initiate the partial BAT update (PBU) by sending PBU information in the management message.

The process of partial BAT update is described as follows:

- 1) At any time during communication, the receiver may send the PBU request for any valid BAT used by the transmitter. The PBU request contains the new valid BAT_ID (N_BAT_ID), old BAT_ID (O_BAT_ID) associated with the BAT to be updated, and bit allocation changes (see clause 8.11.3.2.1).
- 2) Upon reception of the PBU request, the transmitter shall update the BAT associated with the O_BAT_ID, and assign N_BAT_ID to the updated BAT and reply with the PBU confirmation. After receiving the first frame carrying payload using the N_BAT_ID, the receiver shall consider O_BAT_ID as invalid (see clause 8.11.5).
- 3) The receiver may send another PBU request after confirming that the transmitter incorporated the previous PBU request or after inferring that the previous PBU request was lost.

8.11.3.2.1 PBU request

The receiver may send the PBU request using the CE_PartialBatUpdate.req message in which the receiver can request bit allocation changes for up to 1024 sub-carriers. Figure 8-57 illustrates an example of partial BAT update using this approach. Note that acknowledgement is disabled in this example.

If the receiver does not receive CE_PartialBatUpdate.cnf within 200 ms, it may retry with the same or different CE_PartialBatUpdate.req message.

8.11.3.2.2 PBU confirmation

Upon reception of CE_PartialBatUpdate.req message, the transmitter should incorporate the new channel estimation parameters as soon as possible and then send the CE_PartialBatUpdate.cnf message to confirm the received CE_PartialBatUpdate.req message within 100 ms. The receiver shall infer loss of the PBU request if the PBU confirmation is not received within 200 ms after it transmits CE_PartialBatUpdate.req. The transmitter may switch to N_BAT_ID before sending the CE_PartialBatUpdate.cnf message.

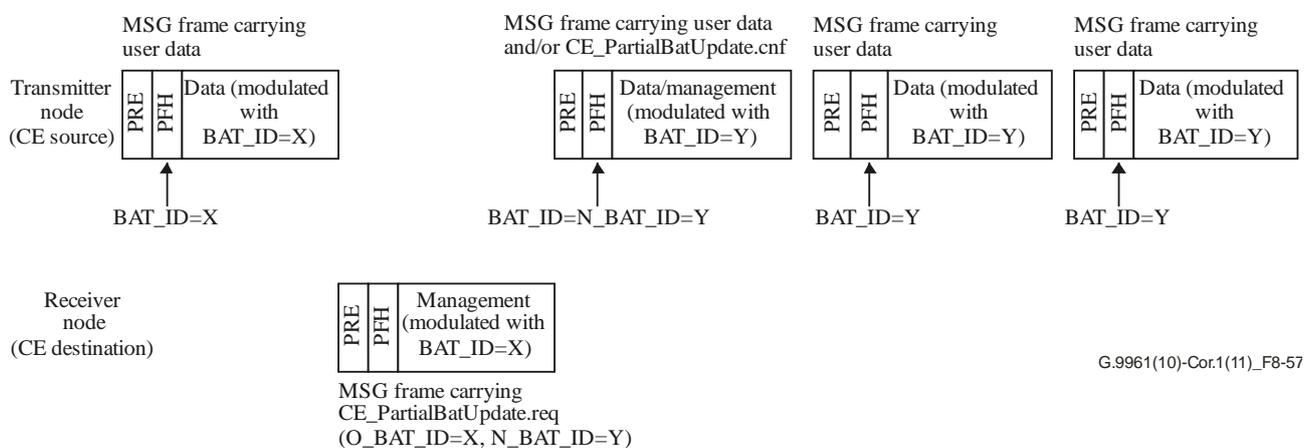


Figure 8-57 – Example of partial BAT update using management message

8.11.4 Channel estimation using PROBE frames

The receiver can request the transmitter for PROBE frame transmission at any time after registration without going through the channel estimation initiation process.

To request PROBE frames, the receiver may use CE_ProbeRequest.ind messages or the ACK_CE_CTRL field in the PFH of an ACK frame (see clause 7.1.2.3.2.3.8/[10.1.2.3.2.3.8](#) of [\[ITU-T G.9960\]](#)). Upon reception of a request for PROBE frame transmission, the transmitter should transmit PROBE frames as soon as possible.

If the receiver requests a PROBE frame through a specific management message, the transmitter shall transmit the PROBE frame using parameters selected by the receiver, that is, the parameters selected in the latest request for PROBE frame transmission (CE_ProbeRequest.ind) or channel estimation initiation (CE_Initiation.req).

If the receiver requests a PROBE frame through an ACK frame, the transmitter shall use the default PROBE frame. The transmitter shall use the default PROBE frame for all ACK frame-based requests for PROBE frame transmission by the receiver. In this case, the transmitter may transmit PROBE frames anywhere in a MAC cycle, regardless of a particular channel estimation window associated with the BAT_ID under channel estimation.

The parameters for the default PROBE frame are determined by the receiver through the CE_Initiation.req message as described in clause 8.11.1.1. Alternatively, they can be updated by setting a bit in the CE_ProbeRequest.ind message as described in Table 8-102.

When a transmitter receives a request for PROBE frame transmission from a receiver while handling previous requests for PROBE frame transmission from the same receiver, it should ignore the new request if the requested parameters are the same as the old ones, regardless of the value of the BAT_ID under estimation.

NOTE – The transmitter should try to cover as much of the channel estimation window as possible when generating PROBE frames.

When the receiver requests a PROBE frame via ACK frames, it may request multiple times by sending multiple ACK frames by setting ACK_CE_CTRL until it receives the PROBE frame. The transmitter should ignore new requests for PROBE frame transmission coming from the receiver in order to avoid unnecessary PROBE transmissions.

After PROBE transmissions, the receiver may send the outcome of channel estimation to the transmitter in case it is needed, using an unsolicited CE_ParamUpdate.req (clause 8.11.3.1) or a partial BAT update (clause 8.11.3.2).

A PROBE frame should be considered as having an MPDU priority equal to 7.

8.11.4.1 Channel estimation using PROBE frames transmitted by a node as requested by Domain Master

Since the DM is in charge of the scheduling and is also aware of the routes between any two nodes of the domain, DM may allocate a special CFTXOP assigned to a single source node and a minimum user priority identified by the tuple (SID, PRI) (see clause 8.3.4) with the value of PRI set by the DM where the assigned node shall transmit PROBE frames (if it has no data to transmit), which the neighbouring nodes can then use to estimate the SNR and measure throughput. This information can be useful for diagnostics. This allows measurement of SNRs and throughput even on the links that are not used for delivering traffic, where there is no need to generate and exchange BATs.

This special CFTXOP is indicated by the DM setting the "TXOP Limitations" field in the "Additional TXOP attributes" descriptor set to 2 (see Table 8-65.1) and the "Traffic Limitation" field in the "TXOP attributes" descriptor set to 0 (see Table 8-65).

The DM may allocate bandwidth to specific nodes (in the form of special TXOPs for probe transmissions, see clause 8.8.4.1.1) so that at least one PROBE frame with default parameters can be transmitted during the channel estimation window of interest. These special TXOPs shall be used for PROBE frame transmissions unless there is data that needs to be transmitted by the node that is assigned the special TXOP. The node that is assigned this special TXOP shall:

- transmit regular data traffic during the TXOP, if the node has data to transmit; or
- transmit PROBE frames if the node has no data to transmit.

Probe frames shall be transmitted at the maximum power level that the node uses for data transmission. The DID in the PHY frame header of the probe frame shall be set to FF₁₆.

8.11.5 BAT_ID maintenance

The receiver is responsible for tracking the list of valid and invalid BAT_IDs. The receiver informs the transmitter of the valid BAT_IDs in the VALID_BAT_ID field by sending a CE_BatIdMaintain.ind message. The transmitter shall stop using BAT_IDs that are marked as invalid by the receiver as soon as possible. If all the BAT_IDs are marked as invalid, the transmitter may use [robust communication mode \(RCM\)](#). In this case, the transmitter should use the parameters indicated in the CE_BatIdMaintain.ind message.

If a BAT_ID is marked as valid by the receiver but the transmitter does not have a BAT associated with it (e.g., the transmitter fails to receive CE_ParamUpdate.req), the transmitter shall send a CE_ParamUpdateRequest.ind message requesting the transmission of the BAT.

The receiver may instruct the transmitter to stop using a BAT_ID via the ACK_CE_CTRL field in the ACK frame (see clause 7.1.2.3.2.3.8/[10.1.2.3.2.3.8](#) of [\[ITU-T G.9960\]](#)). The transmitter shall then consider the BAT_ID as invalid.

The receiver may invalidate a BAT_ID as part of the channel estimation cancellation process (see clause 8.11.1.7).

8.11.6 ACE symbol insertion

The receiver may request the transmitter to attach up to seven ACE symbols (see clause 7.1.2.1/[10.1.2.1](#) of [\[ITU-T G.9960\]](#)) at any time after registration by sending a CE_ACESymbols.ind message. Within 100 ms after receiving this message, the transmitter shall attach ACE symbols as requested by the receiver to all frames sent to the receiver that are allowed to carry ACE symbols. The receiver may use the same procedure to remove or change the number of ACE symbols.

8.11.7 Management message formats for channel estimation

8.11.7.1 Format of CE_ProbeSlotAssign.req

The format of the MMPL of the CE_ProbeSlotAssign.req message shall be as shown in Table 8-91.

Table 8-91 – Format of the MMPL of the CE_ProbeSlotAssign.req

Field	Octet	Bits	Description
Transmitter ID	0	[7:0]	The DEVICE_ID of the node requesting the bandwidth allocation for probe transmissions.
Receiver ID	1	[7:0]	The DEVICE_ID of the receiver node in the channel estimation procedure.
CE_BAT_ID	2	[4:0]	This field indicates the BAT_ID associated with the runtime BAT to be updated by channel estimation. It shall be formatted as shown in Table 7-57/ 10-57 of [ITU-T G.9960] .
Reserved		[7:5]	Reserved by ITU-T (Note).
CE_STIME	3	[7:0]	This field indicates the time at which the transmitter can start PROBE frame transmissions, and it shall be coded as shown in Table 8-98.
CE_ETIME	4	[7:0]	This field indicates the time at which the transmitter shall end PROBE frame transmissions, and it shall be coded as shown in Table 8-99.
CE_PRB_PARM	5 to 7	[23:0]	This field specifies a set of parameters for PROBE frame. It shall be coded as shown in Table 8-102.1.
CE_PRIORITY	8	[2:0]	This field specifies the highest user priority of the traffic the transmitter has to transmit to the specified receiver.
Reserved		[7:3]	Reserved by ITU-T (Note).
NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

8.11.7.2 Format of CE_ProbeSlotRelease.req

The format of the MMPL of the CE_ProbeSlotRelease.req message shall be as shown in Table 8-92.

Table 8-92 – Format of the MMPL of the CE_ProbeSlotRelease.req message

Field	Octet	Bits	Description
Transmitter ID	0	[7:0]	The DEVICE_ID of the node requesting the bandwidth allocation for probe transmissions.
Receiver ID	1	[7:0]	The DEVICE_ID of the receiver node in the channel estimation procedure.
CE_BAT_ID	2	[4:0]	This field indicates the BAT_ID associated with the runtime BAT to be updated by channel estimation. It shall be formatted as shown in Table 7-57/10-57 of [ITU-T G.9960].
Reserved		[7:5]	Reserved by ITU-T (Note).
CE_STIME	3	[7:0]	This field indicates the time at which the transmitter can start PROBE frame transmissions, and it shall be coded as shown in Table 8-98.
CE_ETIME	4	[7:0]	This field indicates the time at which the transmitter shall end PROBE frame transmissions, and it shall be coded as shown in Table 8-99.
NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

8.11.7.3 Format of CE_ParamUpdate.req

The format of the MMPL of the CE_ParamUpdate.req message shall be as shown in Table 8-93.

Table 8-93 – Format of the MMPL of the CE_ParamUpdate.req message

Field	Octet	Bits	Description
New BAT ID	0	[4:0]	This field indicates the BAT_ID associated with a new BAT (CE_BAT_ID). It shall be formatted as shown in Table 7-57/10-57 of [ITU-T G.9960].
Bandplan Frequency range		[7:5]	This field indicates the type of bandplan-frequency range based on which the subsequent BAT entry is defined. It shall be formatted as shown in Table 7-10/10-10 of [ITU-T G.9960].
Group ID	1	[2:0]	This field indicates the GRP_ID (CE_GRP) associated with the new BAT (G). It shall be formatted as shown in Table 7-13/10-13 of [ITU-T G.9960].
BAT_COMPRESSION		[3]	This bit indicates if a run-length compression mechanism is used for conveying the BAT information (fields B ₁ to B _z) When this bit is set to zero, no run-length compression mechanism is used. When this bit is set to one, a run-length compression mechanism is used. This bit shall only be set to one if the BAT compression field (see Table 8-16.8) is also set to one.
NUM_VALID_DUR_EXT		[5:4]	This field contains the extended bits of NUM_VALID_DUR, and they're appended and used for the MSB of NUM_VALID_DUR. With this extension, the valid range of values for NUM_VALID_DUR is from 0 (V=1) to 31 (V=32) (Note 11).
Reserved		[7:6]	Reserved by ITU-T (Note 1).

Table 8-93 – Format of the MMPL of the CE_ParamUpdate.req message

Field	Octet	Bits	Description
VALID_BAT_ID	2 to 4	[23:0]	This field contains a bitmap indicating which runtime BATs are valid (including the New BAT ID) for this node (SID) when receiving from the destination node (DID). Each bit is associated with one runtime BAT. The LSB of the VALID_BAT_ID shall be set to one if runtime BAT 8 is valid. The MSB of the VALID_BAT_ID shall be set to one if runtime BAT 31 is valid.
NUM_TX_Avail_BATS	5	[4:0]	This field contains the number of runtime BATs, assuming $G = 1$, that this node (SID) can support when transmitting to the destination node (DID). Valid values are from 0 to 24.
APSD_MAX_EXT		[5]	This field is the value of APSD_MAX_EXT in the PHY-frame header associated with the new BAT. This field shall be formatted as shown in clause 7.1.2.3.2.2.25/ 10.1.2.3.2.2.25 of [ITU-T G.9960]. (Note 10)
Reserved		[7:6]	Reserved by ITU-T (Note 1)
New block size	6	[1:0]	This field indicates the proposed BLKSZ associated with the new BAT. It shall be formatted as shown in Table 7-7/ 10-7 of [ITU-T G.9960] (Note 2).
New FEC rate		[4:2]	This field indicates the proposed FEC_RATE associated with the new BAT. It shall be formatted as shown in Table 7-12/ 10-12 of [ITU-T G.9960] (Note 3).
New GI		[7:5]	This field indicates the proposed GI_ID associated with the new BAT. It shall be formatted as shown in Table 7-14/ 10-14 of [ITU-T G.9960] (Note 4).
APSD_MAX	7	[4:0]	This field is the value of APSD_MAX in the PHY-frame header associated with the new BAT. This field shall be formatted as shown in clause 7.1.2.3.2.2.11/ 10.1.2.3.2.2.11 of [ITU-T G.9960]. (Note 10)
NUM_VALID_DUR		[7:5]	This field indicates the number of valid durations specified for the new BAT (V). The valid range of values for this field is from 0 (V=1) to 7 (V=8) without extension (Note 5).
CE_STIME ₁	8	[7:0]	This field indicates the start time of the first duration in which the new BAT is valid. It shall be formatted as shown in Table 8-98.
CE_ETIME ₁	9	[7:0]	This field indicates the end time of the first duration in which the new BAT is valid. It shall be formatted as shown in Table 8-99.
...
CE_STIME _v	2V+6	[7:0]	This field indicates the start time of the last duration in which the new BAT is valid. It shall be formatted as shown in Table 8-98.
CE_ETIME _v	2V+7	[7:0]	This field indicates the end time of the last duration in which the new BAT is valid. It shall be formatted as shown in Table 8-99.
TIDX _{MIN} for Profile 1 OFBs	(2V+8) to	[11:0]	12-bit unsigned integer indicating the lowest sub-carrier index to which non-zero bits are assigned. It shall be an integer multiple of G (Note 6).

Table 8-93 – Format of the MMPL of the CE_ParamUpdate.req message

Field	Octet	Bits	Description
	(2V+10)		<u>This field is only valid when using Profile 1 OFBs</u> <u>When using Profile 2 OFBs, this field shall be set to zero.</u>
<u>TIDX_{MAX} for Profile 1 OFBs</u>		[23:12]	12-bit unsigned integer indicating the highest sub-carrier index to which non-zero bits are assigned. It shall be an integer multiple of G (Note 6) and if bit-loading grouping is used ($G > 1$) shall meet: $TIDX_{MAX} + G - 1 \leq \text{StopSubCarrier}$, where StopSubCarrier is specified in Table 8-16.6 (<u>Bandplan OFB Info Capability Value</u> field). Let W denote the number of BAT entries, which is $(TIDX_{MAX} - TIDX_{MIN}) / G + 1$. Let Z denote the smallest integer larger than or equal to $W/2$. <u>This field is only valid when using Profile 1 OFBs.</u> <u>When using Profile 2 OFBs, this field shall be set to zero.</u>
B ₁	2V+11	[3:0]	4-bit unsigned integer indicating the number of bits assigned to sub-carrier indices $TIDX_{MIN}$ to $TIDX_{MIN} + G - 1$ (Note 6). This field only exists if BAT_COMPRESSION field is set to zero.
		[7:4]	4-bit unsigned integer indicating the number of bits assigned to sub-carrier indices $TIDX_{MIN} + G$ to $TIDX_{MIN} + 2G - 1$ (Notes 6, 7, 8). This field only exists if BAT_COMPRESSION field is set to zero.
...
B _Z	2V+10 + Z	[3:0]	4-bit unsigned integer indicating the number of bits assigned to sub-carrier indices $TIDX_{MAX} - G$ to $TIDX_{MAX} - 1$ if W is even, or to sub-carrier indices $TIDX_{MAX}$ to $TIDX_{MAX} + G - 1$ if W is odd (Notes 6, 7). This field only exists if BAT_COMPRESSION field is set to zero.
		[7:4]	4-bit unsigned integer indicating the number of bits assigned to sub-carrier indices $TIDX_{MAX}$ to $TIDX_{MAX} + G - 1$ if W is even (Notes 6, 9). This field only exists if BAT_COMPRESSION field is set to zero.
COMPRESSED_BAT	Var	Var	Compressed information that includes the Z fields B _i . This field shall be filled using the mechanism described in clause 8.23 with a Value field formatted as field B _i of this message ($N=8$). This field only exists if BAT_COMPRESSION field is set to one.
<u>BitloadingTLV for Profile 2 OFBs</u>	<u>Var</u>	<u>Var</u>	<u>TLV including the estimated bitloading information for Profile 2 OFBs.</u> <u>It shall be coded following Table 8-140.6</u> <u>This TLV is only present if Profile 2 OFBs are used.</u>
NOTE 1 – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

Table 8-93 – Format of the MMPL of the CE_ParamUpdate.req message

Field	Octet	Bits	Description
NOTE 2 – The transmitter shall use the proposed block size or larger block size for a new connection. Once the block size is selected for a connection, it shall not be changed throughout the lifetime of the connection (clause 8.1.3.2).			
NOTE 3 – The transmitter shall use the proposed FEC rate or lower FEC rate.			
NOTE 4 – The transmitter shall use the proposed GI or longer GI value.			
NOTE 5 – A new BAT shall only be used over specified non-overlapping durations (up to 8) within a MAC cycle, defined by CE_STIME _i and CE_ETIME _i without extension.			
NOTE 6 – Sub-carrier index represents the physical index (clause 7.1.4.1/10.1.4.1 of [ITU-T G.9960]). All BAT entries outside [TIDX _{MIN} , TIDX _{MAX} + G – 1] shall be considered as unloaded.			
NOTE 7 – If a sub-carrier is not loaded, the field shall be set to zero.			
NOTE 8 – If W = 1, this field shall be set to zero.			
NOTE 9 – If W is an odd number, this field shall be set to zero.			
NOTE 10 – It is the responsibility of the receiver to set APSD_MAX and APSD_MAX_EXT taking into account the PSD limitations and capabilities of the transmitter that are known to the receiver (see clause 7.1.5/10.1.5 of [ITU-T G.9960]).			
NOTE 11 – A new BAT shall only be used over specified non-overlapping durations (up to 32) within a MAC cycle, defined by CE_STIME _i and CE_ETIME _i with extension of NUM_VALID_DUR.			

8.11.7.4 Format of CE_ParamUpdateRequest.ind

The format of the MMPL of the CE_ParamUpdateRequest.ind message shall be as shown in Table 8-94.

Table 8-94 – Format of the MMPL of the CE_ParamUpdateRequest.ind message

Field	Octet	Bits	Description
Requested BAT ID	0	[4:0]	This field indicates the BAT_ID for which the transmitter requests retransmission of the channel estimation result. It shall be formatted as shown in Table 7-57/10-57 of [ITU-T G.9960].
Reserved		[7:5]	Reserved by ITU-T (Note)
NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

8.11.7.5 Format of CE_PartialBatUpdate.req

The format of the MMPL of the CE_PartialBatUpdate.req message shall be as shown in Table 8-95.

Table 8-95 – Format of the MMPL of the CE_PartialBatUpdate.req message

Field	Octet	Bits	Description
O_BAT_ID	0	[4:0]	This field indicates the BAT_ID associated with the BAT to be updated by the PBU request. It shall be formatted as shown in Table 7-57/10-57 of [ITU-T G.9960].
Reserved		[7:5]	Reserved by ITU-T (Note 1).
N_BAT_ID	1	[4:0]	This field indicates the BAT_ID associated with the BAT updated by the PBU request. It shall be formatted as shown in Table 7-57/10-57 of [ITU-T G.9960].
Reserved		[7:5]	Reserved by ITU-T (Note 1).

Table 8-95 – Format of the MMPL of the CE_PartialBatUpdate.req message

Field	Octet	Bits	Description
NUM_BAT_ENT <u>for Profile 1 OFBs</u>	2 and 3	[9:0]	This field indicates the number of BAT entries to be updated (V). The valid range of this field is from 0 ($V=1$) to 1023 ($V=1024$). <u>For Profile 2 OFBs, this field shall be set to zero.</u>
GROUP_ID		[12:10]	This field indicates the current GRP_ID associated with the BAT corresponding to O_BAT_ID and N_BAT_ID (G). Partial BAT update shall not change the current GRP_ID. It shall be formatted as shown in Table 7-13/10-13 of [ITU-T G.9960].
Reserved		[15:13]	Reserved by ITU-T (Note 1).
T ₁	4 and 5	[11:0]	12-bit unsigned integer indicating the sub-carrier index (Note 2). It shall be an integer multiple of G .
B _{T1}		[15:12]	4-bit unsigned integer indicating the number of bits assigned to sub-carrier indices T ₁ to T ₁ + G -1.
...
T _V	(2 V +2) to (2 V +3)	[11:0]	12-bit unsigned integer indicating the sub-carrier index (Note 2). It shall be an integer multiple of G .
B _{Tv}		[15:12]	4-bit unsigned integer indicating the number of bits assigned to sub-carrier indices T _V to T _V + G -1.
<u>PartialBitloadingTLV</u>	<u>Var</u>	<u>Var</u>	<u>TLV including the partial bitloading information for Profile 2 OFBs.</u> <u>It shall be coded following Table 8-140.7</u> <u>This TLV is only present if Profile 2 OFBs are used.</u>
NOTE 1 – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			
NOTE 2 – Sub-carrier index represents the physical index (clause 7.1.4.1/10.1.4.1 of [ITU-T G.9960]).			

8.11.7.6 Format of CE_ACESymbols.ind

The format of the MMPL of the CE_ACESymbols.ind message shall be as shown in Table 8-96.

Table 8-96 – Format of the MMPL of the CE_ACESymbols.ind message

Field	Octet	Bits	Description
ACE symbols	0	[2:0]	This field indicates the number of ACE symbols added to the beginning of the payload of all frames sent to the receiver that are allowed to carry ACE symbols. It shall be formatted as shown in Table 7-16/10-16 of [ITU-T G.9960].
Reserved		[7:3]	Reserved by ITU-T (Note).
NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

8.11.7.7 Format of CE_ProbeSlotAssign.cnf

The format of the MMPL of the CE_ProbeSlotAssign.cnf message shall be as shown in Table 8-96.1.

Table 8-96.1 – Format of the MMPL of the CE_ProbeSlotAssign.cnf message

Field	Octet	Bits	Description
Transmitter ID	0	[7:0]	The DEVICE_ID of the node requesting the bandwidth allocation for probe transmissions.
Receiver ID	1	[7:0]	The DEVICE_ID of the receiver node in the channel estimation procedure.
CE_BAT_ID	2	[4:0]	This field indicates the BAT_ID associated with the runtime BAT to which bandwidth was required for probing.
StatusCode		[7:5]	0 – Bandwidth request is confirmed (Note). 1 – Request is rejected. 2 to 7 – Reserved by ITU-T.
NOTE – Bandwidth allocation will be identified in the MAP using the Transmitter_ID (SID), Receiver_ID (DID) and channel estimation only indication set in the TXOP attributes extension (see clause 8.8.4.1.1).			

8.11.7.8 Format of CE_ProbeSlotRelease.cnf

The format of the MMPL of the CE_ProbeSlotRelease.cnf message shall be as shown in Table 8-96.2.

Table 8-96.2 – Format of the MMPL of the CE_ProbeSlotRelease.cnf message

Field	Octet	Bits	Description
Transmitter ID	0	[7:0]	The DEVICE_ID of the node requesting the bandwidth allocation for probe transmissions.
Receiver ID	1	[7:0]	The DEVICE_ID of the receiver node in the channel estimation procedure.
CE_BAT_ID	2	[4:0]	This field indicates the BAT_ID associated with the runtime BAT for which the bandwidth has to be released.
StatusCode		[7:5]	0 – Request is confirmed. 1 – Request is rejected (unknown BAT identity) (Note). 2 to 7 – Reserved by ITU-T.
NOTE – There is no bandwidth allocated for the identified channel estimation procedure. The identification is defined by the Transmitter_ID, Receiver_ID, and CE_BAT_ID.			

8.11.7.9 Format of CE_ParamUpdate.cnf

The format of the MMPL of the CE_ParamUpdate.cnf message shall be as shown in Table 8-96.3.

Table 8-96.3 – Format of the MMPL of the CE_ParamUpdate.cnf message

Field	Octet	Bits	Description
New BAT ID	0	[4:0]	This field indicates the BAT_ID specified in the received CE_ParamUpdate.req message.
Reserved		[7:5]	Reserved by ITU-T (Note).
NUM_AVAIL_BATS	1	[4:0]	This field contains the number of available runtime BATs, assuming $G = 1$, that this node (SID) can support when transmitting to the destination node (DID). It excludes the BAT associated with the New BAT ID. Valid values are from 0 to 23.
StatusCode		[7:5]	0 – BAT successfully updated. 1 – Update is rejected (no more resources). 2 – Update is rejected (New BAT ID already exists). 3 to 7 – Reserved by ITU-T.
NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

8.11.7.10 Format of CE_PartialBatUpdate.cnf

The format of the MMPL of the CE_PartialBatUpdate.cnf message shall be as shown in Table 8-96.4.

Table 8-96.4 – Format of the MMPL of the CE_PartialBatUpdate.cnf message

Field	Octet	Bits	Description
CE_BAT_ID	0	[4:0]	This field indicates the CE_BAT_ID specified in the CE_PartialBatUpdate.req message
Reserved		[7:5]	Reserved by ITU-T (Note)
NUM_AVAIL_BATS	1	[4:0]	This field contains the number of available runtime BATs, assuming $G = 1$, that this node (SID) can support when transmitting to the destination node (DID). It excludes the BAT associated with the CE_BAT_ID. Valid values are from 0 to 23.
StatusCode		[7:5]	0 – BAT successfully updated 1 – Request rejected (no more resources) 2 – Request rejected (O_BAT_ID does not exist) 3 – Request rejected (N_BAT ID already exist) 4 to 7 – Reserved by ITU-T
NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

8.11.7.11 Format of CE_Request.ind

The format of the MMPL of the CE_Request.ind message shall be as shown in Table 8-97.

Table 8-97 – Format of the MMPL of the CE_Request.ind message

Field	Octet	Bits	Description
CE_WINDOW_SEL	0	[0]	This field shall be set to one if the transmitter selects the channel estimation window. It shall be set to zero, otherwise. If this field is set to zero, then CE_STIME and CE_ETIME shall be set to 00 ₁₆ , and these values shall be ignored by the receiver.
Reserved		[7:1]	Reserved by ITU-T (Note).
CE_STIME	1	[7:0]	This field indicates time at which the transmitter can start PROBE frame transmissions, and it shall be coded as shown in Table 8-98.
CE_ETIME	2	[7:0]	This field indicates time at which the transmitter shall end PROBE frame transmissions, and it shall be coded as shown in Table 8-99.
NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

Table 8-98 – CE_STIME field values

Value (b ₇ b ₆ b ₅ b ₄ b ₃ b ₂ b ₁ b ₀)	Interpretation
00000000	Start of MAC cycle (T_0).
00000001	$T_0 + 1/256$ of MAC cycle duration.
...	...
11111111	$T_0 + 255/256$ of MAC cycle duration.

Table 8-99 – CE_ETIME field values

Value (b ₇ b ₆ b ₅ b ₄ b ₃ b ₂ b ₁ b ₀)	Interpretation
00000000	End of MAC cycle.
00000001	$T_0 + 1/256$ of MAC cycle duration.
...	...
11111111	$T_0 + 255/256$ of MAC cycle duration.

8.11.7.12 Format of CE_Initiation.req

The format of the MMPL of the CE_Initiation.req message shall be as shown in Table 8-100.

Table 8-100 – Format of the MMPL of the CE_Initiation.req message

Field	Octet	Bits	Description
CE_BAT_ID	0	[4:0]	This field indicates the BAT_ID associated with the runtime BAT to be created by channel estimation. It shall be formatted as shown in Table 7-57/10-57 of [ITU-T G.9960].
CE_GRP		[7:5]	This field indicates the value of sub-carrier grouping. It shall be formatted as shown in Table 7-13/10-13 of [ITU-T G.9960].
CE_STIME	1	[7:0]	This field indicates the time at which the transmitter can start PROBE frame transmissions, and it shall be coded as shown in Table 8-98.
CE_ETIME	2	[7:0]	This field indicates the time at which the transmitter shall end PROBE frame transmissions, and it shall be coded as shown in Table 8-99.
CE_PRB_RQST	3	[0]	This field shall be set to one if the receiver wants PROBE frames along with channel estimation initiation confirmation. It shall be set to zero otherwise.
Reserved		[7:1]	Reserved by ITU-T (Note).
CE_PRB_PARM	4 to 6	[23:0]	This field specifies a set of parameters for PROBE frame. It shall be coded as shown in Table 8-102.1. This field shall be set to 000000 ₁₆ if CE_PRB_RQST is set to zero.

NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.

8.11.7.13 Format of CE_Initiation.cnf

The format of the MMPL of the CE_Initiation.cnf message is shown in Table 8-101.

Table 8-101 – Format of the MMPL of the CE_Initiation.cnf message

Field	Octet	Bits	Description
CE_BAT_ID	0	[4:0]	This field indicates the BAT_ID associated with the runtime BAT to be created by channel estimation. It shall be formatted as shown in Table 7-57/10-57 of [ITU-T G.9960].
CE_GRP		[7:5]	This field indicates the value of sub-carrier grouping. It shall be formatted as shown in Table 7-13/10-13 of [ITU-T G.9960].
NUM_AVAIL_BATS	1	[4:0]	This field contains the number of available runtime BATs, assuming $G = 1$, that this node (SID) can support when transmitting to the destination node (DID). It excludes the BAT associated with the CE_BAT_ID. Valid values are from 0 to 23.
StatusCode		[7:5]	0 – Channel estimation initiation is confirmed 1 – Rejected (CE_BAT_ID is valid and currently in use) 2 – Rejected (Bandwidth for PROBE frame transmission is not available) 3 – Rejected (Bandwidth request for probe frame transmission is pending) 4 – Rejected (Channel estimation window is currently not available) 5 to 7 – Reserved by ITU-T (Note)

NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.

8.11.7.14 Format of CE_ProbeRequest.ind

The format of the MMPL of the CE_ProbeRequest.ind message shall be as shown in Table 8-102.

Table 8-102 – Format of the MMPL of the CE_ProbeRequest.ind message

Field	Octet	Bits	Description
CE_BAT_ID	0	[4:0]	This field indicates the BAT_ID associated with the runtime BAT to be created by channel estimation. It shall be formatted as shown in Table 7-57/10-57 of [ITU-T G.9960] .
CE_PRB_DEFAULT_IND		[5]	When this field is set to one, the parameters provided in this message (CE_PRB_PARM) replace the existing parameters for the default PROBE frame for this node (SID) when receiving from the destination node (DID).
Reserved		[7:6]	Reserved by ITU-T (Note).
CE_PRB_PARM	1 to 3	[23:0]	See Table 8-102.1.

NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.

Table 8-102.1 – Format of the CE_PRB_PARM field

Field	Octet	Bits	Description
CE_PR_PRBTYPE	0	[3:0]	This field indicates the PRBTYPE requested by the receiver. It shall be formatted as shown in Table 7-40/10-40 of [ITU-T G.9960] .
CE_PR_PRBFN		[7:4]	This field indicates the number of PROBE frames that shall be sent by the transmitter at each request for PROBE frame transmission. The field shall be coded as shown in Table 8-103. The transmitter may send multiple PROBE frames within a single channel estimation window.
CE_PR_PRBSYM	1	[3:0]	This field indicates the PRBSYM requested by the receiver. It shall be formatted as shown in Table 7-41/10-41 of [ITU-T G.9960] .
CE_PR_PRBGI		[6:4]	This field indicates the PRBGI requested by the receiver. It shall be formatted as shown in Table 7-14/10-14 of [ITU-T G.9960] .
Reserved		[7]	Reserved by ITU-T (Note).
CE_PR_APSD_MAX	2	[4:0]	This field indicates the APSD_MAX-P requested by the receiver. It shall be formatted as described in clause 7.1.2.3.2.7.1.4/ 10.1.2.3.2.7.1.4 of [ITU-T G.9960] . (Note 2)
CE_PR_APSD_MAX_EXT		[5]	This field indicates the APSD_MAX_EXT-P requested by the receiver. It shall be formatted as described in clause 7.1.2.3.2.7.1.7/ 10.1.2.3.2.7.1.7 of [ITU-T G.9960] . (Note 2)
Reserved		[7:6]	Reserved by ITU-T (Note 1).

Table 8-102.1 – Format of the CE_PRB_PARM field

Field	Octet	Bits	Description
NOTE 1 – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			
NOTE 2 – It is the responsibility of the receiver to set CE_PR_APSD_MAX and CE_PR_APSD_MAX_EXT taking into account the PSD limitations and capabilities of the transmitter that are known to the receiver (see clause 7.1.5/10.1.5 of [ITU-T G.9960]).			

Table 8-103 – CE_PR_PRBFN field values

Value (b ₃ b ₂ b ₁ b ₀)	Interpretation
0000	One PROBE frame
0001	Two PROBE frames
...	...
1111	Sixteen PROBE frames

8.11.7.15 Format of CE_Cancellation.req

The format of the MMPL of the CE_Cancellation.req message shall be as shown in Table 8-104.

Table 8-104 – Format of the MMPL of the CE_Cancellation.req message

Field	Octet	Bits	Description
CE_BAT_ID	0	[4:0]	This field indicates the channel estimation identifier that is cancelled. It shall be formatted as shown in Table 7-57/10-57 of [ITU-T G.9960].
USE_RCM		[5]	When set to one it means the transmitter may use RCM with parameters communicated in the New block size, New FEC rate, Bandplan OFB_ID, and Repetitions fields. It shall be set to zero otherwise.
Reserved		[7:6]	Reserved by ITU-T (Note).
New block size	1	[1:0]	When USE_RCM is set to one this field indicates the proposed BLKSZ associated to RCM. It shall be formatted as shown in Table 7-7/10-7 of [ITU-T G.9960]. It shall be set to 00 ₂ otherwise.
New FEC rate		[4:2]	When USE_RCM is set to one this field indicates the proposed FEC_RATE associated to RCM. It shall be formatted as shown in Table 7-12/10-12 of [ITU-T G.9960]. It shall be set to 00 ₀₂ otherwise.
Bandplan OFB_ID		[7:5]	When USE_RCM is set to one this field indicates the BNDPL OFB based on which the RCM parameters are proposed. It shall be formatted as shown in Table 7-10/10-10 of [ITU-T G.9960]. It shall be set to 000 ₂ otherwise.
Repetitions	2	[2:0]	When USE_RCM is set to one this field indicates the proposed number of repetitions associated with RCM. It shall be formatted as shown in Table 7-8/10-8 of [ITU-T G.9960]. It shall be set to 000 ₂ otherwise.

Table 8-104 – Format of the MMPL of the CE_Cancellation.req message

Field	Octet	Bits	Description
RCM_BAT_ID		[3]	When USE_RCM is set to one, this field indicates the pre-defined BAT associated with RCM. It shall be set to the following value: zero, when pre-defined BAT Type 1 is used. one, when pre-defined BAT Type 2 is used.
Reserved		[7:4]	Reserved by ITU-T (Note).
RCM_GI_ID	3	[2:0]	When USE_RCM is set to one, this field indicates the GI_ID associated with RCM. It shall be formatted as shown in Table 7-14/ 10-14 of [ITU-T G.9960].
Reserved		[7:3]	Reserved by ITU-T (Note).
NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

8.11.7.16 Format of CE_BatIdMaintain.ind

The format of the MMPL of the CE_BatIdMaintain.ind message shall be as shown in Table 8-105.

Table 8-105 – Format of the MMPL of the CE_BatIdMaintain.ind message

Field	Octet	Bits	Description
VALID_BAT_ID	0 to 2	[23:0]	This field contains a bitmap indicating which runtime BATs are valid for this node (SID) when receiving from the destination node (DID). Each bit is associated with one runtime BAT. The LSB of the VALID_BAT_ID shall be set if runtime BAT 8 is valid. The MSB of the VALID_BAT_ID shall be set if runtime BAT 31 is valid.
NUM_TX_AVAIL_BATS	3	[4:0]	This field contains the number of available runtime BATs, assuming $G=1$, that this node (SID) can support when transmitting to the destination node (DID). Valid values are from 0 to 24.
Reserved		[7:5]	Reserved by ITU-T (Note 1)
New block size	4	[1:0]	This field indicates the proposed BLKSZ associated with RCM, if there is no available runtime BAT (Note 2). It shall be formatted as shown in Table 7-7/ 10-7 of [ITU-T G.9960] . It shall be set to 0 otherwise.
New FEC rate		[4:2]	This field indicates the proposed FEC_RATE associated with RCM, if there is no available runtime BAT (Note 2). It shall be formatted as shown in Table 7-12/ 10-12 of [ITU-T G.9960] . It shall be set to 0 otherwise.
Bandplan-OFB_ID		[7:5]	This field indicates the BNDPL-OFB based on which the RCM parameters are proposed, if there is no available runtime BAT (Note 2). It shall be formatted as shown in Table 7-10/ 10-10 of [ITU-T G.9960] . It shall be set to 0 otherwise.
Repetitions	5	[2:0]	This field indicates the proposed number of repetitions associated with RCM, if there is no available runtime BAT (Note 2). It shall be formatted as shown in Table 7-8/ 10-8 of [ITU-T G.9960] . It shall be set to 0 otherwise
RCM_BAT_ID		[3]	This field indicates the pre-defined BAT associated with RCM, if there is no available runtime BAT (Note 2). It shall be set to the following value:

Table 8-105 – Format of the MMPL of the CE_BatIdMaintain.ind message

Field	Octet	Bits	Description
			zero, when pre-defined BAT Type 1 is used one, when pre-defined BAT Type 2 is used
Reserved		[7:4]	Reserved by ITU-T (Note 1)
RCM_GI_ID	6	[2:0]	This field indicates the GI_ID associated with RCM, if there is no available runtime BAT (Note 2). It shall be formatted as shown in Table 7-14/ 10-14 of [ITU-T G.9960].
Reserved		[7:3]	Reserved by ITU-T (Note 1).
NOTE 1 – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			
NOTE 2 – Runtime BATs might only be available for specified time periods (see Table 8-93).			

8.11.7.17 Format of CE_Cancellation.cnf

The format of the MMPL of the CE_Cancellation.cnf message shall be as shown in Table 8-105.1.

Table 8-105.1 – Format of the MMPL of the CE_Cancellation.cnf message

Field	Octet	Bits	Description
CE_BAT_ID	0	[4:0]	This field indicates the BAT_ID specified in the received CE_Cancellation.req message.
StatusCode		[7:5]	0 – Channel estimation is successfully cancelled 1 – no ongoing channel estimation for this CE_BAT_ID. 2 to 7 – Reserved by ITU-T (Note)
NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

8.12 Connection management

Connection management is a mechanism used by the transmitter and the receiver to exchange information about the availability of resources to manage the communication. Connections may be established by the transmitter and may be released by the transmitter or the receiver.

Segments exchanged between devices shall be part of a connection except in the following cases:

- MAP or RMAP messages;
- APDUs and LCDUs conveyed in PHY-frames with the CNN_MNGMT field equal to 1111, for example, messages exchanged as part of the network admission protocol described in clause 8.6.1.1.1. In this case, the LLC frames contained in the MPDU shall be complete and the CONNECTION_ID shall be set to 255 (see clause 7.1.2.3.2.2.12/[10.1.2.3.2.2.12](#) of [ITU-T G.9960]). Segments contained in this MPDU shall not use the acknowledgement protocol defined in clause 8.9.

NOTE – More cases may be added in future versions of this Recommendation.

A connection shall be established prior to exchange of any segment associated with that connection.

A data connection associated with a priority queue is uniquely identified by the tuple:

(SID > 0, DID > 0, PRI-Q, MQF = 0) and is known as a prioritized data connection, where PRI-Q is specified in Table III.1 of [ITU-T G.9960] as a function of the user priority mapped to the connection and the number of priority queues (traffic classes) supported from the source node to the destination node (i.e., for user priority 6 and 3 priority queues, PRI-Q is equal to 2).

A data connection associated with a service flow is uniquely identified by the tuple:

(SID > 0, FLOW_ID, MQF = 0).

A management connection is uniquely identified by the tuple (SID > 0, DID > 0, MQF = 1).

Each connection shall use an independent SSN sequence numbering. At any given time, there may be multiple open "connections" between a transmitter and a receiver in a network:

- zero or one management connection (for exchange of LCDUs);
- between zero and eight prioritized data connections (for exchange of APDUs that have not been mapped to flows. LCDUs may be mixed with APDUs in a prioritized data connection; see clause 8.1.3.2);
- between zero and 243 unicast data connections, identified by CONNECTION_ID (see clauses 7.1.2.3.2.2.12/10.1.2.3.2.2.12 and 7.1.2.3.2.3.9.1.4/10.1.2.3.2.3.9.1.4 of [ITU-T G.9960]) value in the range 8 to 250, associated with service flows;
- between zero and 254 multicast connections, identified by MI=1, DID=MULTICAST_ID and CONNECTION_ID=252;
- two broadcast connections one for data and one for management.

Table 8-105.2 – Values of connection identifiers for different types of connections

Connection identifier	Type of connection
0 to 7	Prioritized data connection.
8 to 250	Service flow.
251	Management connection (unicast or broadcast).
252	Multicast connection.
253, 254	Reserved by ITU-T.
255	Broadcast data connection (Note).
NOTE – In the case where the data does not belong to any connection (i.e., the CNN_MNGMT field is set to 1111), CONNECTION_ID shall be set to 255.	

Only one connection (either the management connection or a prioritized data connection) for delivering LCDUs may be established between a source node and destination node. A multicast connection shall not be used for delivering LCDUs.

Connections are unidirectional.

Connections may be established either with or without acknowledgements. A connection with acknowledgements is a connection that uses the acknowledgement protocol described in clause 8.9. Establishment of two connections identified by the same tuple, one with ACKs and the other without ACKs, is not allowed.

A given PHY frame may carry segments from the management connection and from not more than one data connection. Data and management segments can be differentiated by the MQF field in the LPDU header.

The CONNECTION_ID field in the PHY-frame header identifies the connection. The CONNECTION_ID field shall be set to the FLOW_ID for connections associated to service flows, or it shall be set to the value of PRI-Q for prioritized data connections. The valid values of CONNECTION_ID field for different types of connections are as shown in Table 8-105.2.

The ~~FEC block~~LPDU size that the transmitter has selected for a connection shall be indicated in the PHY-frame header when the connection is established.

PHY frames carrying connection management information in which no payload is allowed (see Table 7-17/10-17 of [ITU-T G.9960]) shall have the MPDU priority equal to 7.

8.12.1 Establishment of a unicast connection with acknowledgements

Connections that require the use of the acknowledgement protocol shall be established as described in clauses 8.12.1.1 or 8.12.1.2.

8.12.1.1 Establishment of the management connection

The management connection shall be established using the protocol described in clause 8.9.5.3, where the transmitter shall send a PHY frame with FT=MSG, CNN_MNGMT=0001, START_SSN=ACK_TX_WINDOW_START, no payload and RPRQ=01.

If the receiver has resources to handle the new connection, it shall respond with a PHY frame with FT=ACK, RXRST_MNGMT=1. In this ACK frame, the receiver shall use the flow control fields FLCTRLT, FLCTRL and FLCTRL_CONN to provide additional flow control information, such as receiver buffer size or hold time. Once the protocol described in clause 8.9.5.3 is finished successfully, the transmitter may start sending PHY frames with segments belonging to the management connection.

Following the protocol described in clause 8.9.5.3, if the receiver temporarily does not have resources to handle the new connection, it shall respond with a PHY frame with FT=ACK, RXRST_MNGMT=1, FLCTRLT=<Hold Time>, FLCTRL_CONN=1 and FLCTRL equal to the amount of time desired by the receiver (see clauses 7.1.2.3.2.3.1/10.1.2.3.2.3.1, 7.1.2.3.2.3.2/10.1.2.3.2.3.2 and 7.1.2.3.2.3.3/10.1.2.3.2.3.3 of [ITU-T G.9960]).

If the receiver does not have resources to handle the new connection, it shall respond with a PHY frame with FT=ACK, RXRST_MNGMT=1, FLCTRLT=<Hold Time>, FLCTRL_CONN=1 and FLCTRL=31.

If the receiver has resources for the new connection, it shall respond with a PHY frame with FT=ACK, RXRST_MNGMT=1, FLCTRLT=<Status report>, FLCTRL_CONN=1 and FLCTRL equal to the number of LPDUs that the receiver can buffer for this connection. The transmitter shall set ACK_TX_CONF_WINDOW_SIZE (see clause 8.9.4.2) to the value received in the RX_CONN_WIN_SIZE field during connection set-up. The value of the RX_CONN_WIN_SIZE field shall not be changed during connection reset. The number of LPDUs that the receiver can buffer for this connection (indicated by the FLCTRL field during the lifetime of the connection) shall not exceed the maximum acknowledge window size that the receiver can support for the connection (indicated by RX_CONN_WIN_SIZE during connection set-up).

8.12.1.2 Establishment of a data connection

A data connection shall be established using the protocol described in clause 8.9.5.3, where the transmitter shall send a PHY frame with FT=MSG, CNN_MNGMT=0101, START_SSN=ACK_TX_WINDOW_START, no payload and RPRQ=01. The transmitter may advise the receiver about the required number of LPDUs that the receiver should buffer for this connection by setting the ADVISED_WIN_SIZE field.

If the receiver has resources to handle the new connection, it shall respond with a PHY frame with FT=ACK, RXRST_DATA=1. In this ACK frame, the receiver shall use the flow control fields FLCTRLT, FLCTRL and FLCTRL_CONN to provide additional flow control information, such as receiver buffer size or hold time. Once the protocol described in clause 8.9.5.3 is finished successfully, the transmitter may start sending PHY frames with data segments.

Following the protocol described in clause 8.9.5.3, if the receiver temporarily does not have resources to handle the new connection, it shall respond with a PHY frame with FT=ACK, RXRST_DATA=1, FLCTRLT=<Hold Time>, FLCTRL_CONN=0 and FLCTRL equal to the amount of time desired by the receiver.

If the receiver does not have resources to handle the new connection, it shall respond with a PHY frame with FT=ACK, RXRST_DATA=1, FLCTRLT=<Hold Time>, FLCTRL_CONN=0 and FLCTRL=31.

If the receiver has resources for the new connection, it shall respond with a PHY frame with FT=ACK, RXRST_DATA=1, FLCTRLT=<Status report>, FLCTRL_CONN=0 and FLCTRL equal to the number of LPDUs that the receiver can buffer for this connection. The receiver may consider the value of the ADVISED_WIN_SIZE as proposed by the transmitter to set its FLCTRL field and RX_CONN_WIN_SIZE field. The transmitter shall set ACK_TX_CONF_WINDOW_SIZE (see clause 8.9.4.2) to the minimum of the value indicated in the RX_CONN_WIN_SIZE field during connection set-up and its own available window size (see clause 7.1.2.3.2.3.8/10.1.2.3.2.3.8 of [ITU-T G.9960]). The value of the RX_CONN_WIN_SIZE field shall not be changed during connection reset. The number of LPDUs that the receiver can buffer for this connection, indicated by the FLCTRL field during the lifetime of the connection, shall not exceed the maximum acknowledge window size that the receiver can support for the connection indicated by RX_CONN_WIN_SIZE during connection set-up.

8.12.2 Establishment of a unicast connection without acknowledgements

Connections that do not require the use of the acknowledgement protocol shall be established as described in the following paragraphs.

The RPRQ field in the PHY-frame header of the MSG frames associated with these connections shall be set to zero except when the connection is being established or released.

8.12.2.1 Establishment of the management connection

The transmitter shall send to the receiver a PHY frame with FT=MSG, CNN_MNGMT=0010, no payload and RPRQ=01 to request the establishment of the connection.

If the receiver has resources to handle the new connection, it shall respond with a PHY frame with FT=ACK, FLCTRLT=<Hold Time/Management>, FLCTRL=30 and FLCTRL_CONN=1.

If the receiver temporarily does not have resources to handle the new connection, it shall respond with a PHY frame with FT=ACK, FLCTRLT=<Hold Time/Management>, FLCTRL_CONN=1 and FLCTRL equal to the amount of time desired by the receiver. The transmitter shall wait that time before resending the PHY frame requesting the establishment of the connection.

If the receiver does not have resources to handle the new connection, it shall respond with a PHY frame with FT=ACK, FLCTRLT=<Hold Time/Management>, FLCTRL_CONN=1 and FLCTRL=31.

If the transmitter node does not receive the answer from the receiver, the transmitter may resend the message to establish the connection. After resending this message twice without receiving the response from the receiver, the establishment of the connection is considered as failed.

Once the establishment of the connection is completed, the transmitter may start sending PHY-frames with segments belonging to that connection.

If the establishment of the connection is rejected or failed, the transmitter may discard the segments belonging to that connection.

8.12.2.2 Establishment of a data connection

The procedure to establish a data connection without acknowledgments is the same as described in clause 8.12.2.1 but with CNN_MNGMT=0110 and FLCTRL_CONN=0.

8.12.3 Establishment of a unicast data connection for a service flow

For data connections associated with service flows, the establishment process shall be:

- 1) First, a flow shall be established, following the procedure described in clause 8.6.2.
- 2) Once the flow has been established, the data connection shall be established following the procedure described in clauses 8.12.2 or 8.12.1.

8.12.4 Flow control of connections

Flow control is a mechanism that the receiver shall use to indicate the transmitter its runtime capabilities for re-assembly of LLC frames belonging to a given connection with acknowledgements.

The receiver may indicate the transmitter the number of LPDUs that the receiver can handle in the next burst of PHY frames or may indicate a period of time that the transmitter shall hold transmissions to the receiver node. This mechanism shall be used once the connection is established.

Use of the PHY-frame header fields FLCTRL_CONN, FLCTRLT and FLCTRL for flow control operation is described in clause 7.1.2.3.2.3/[10.1.2.3.2.3](#) of [\[ITU-T G.9960\]](#). The value of these fields may change in each transmission of acknowledgements.

8.12.5 Release of a unicast connection with acknowledgements

Connections that use the acknowledgement protocol shall be released as described in clauses 8.12.5.1 or 8.12.5.2. A connection may be released by the transmitter or by the receiver.

8.12.5.1 Release of the management connection

The receiver may release the management connection by sending a PHY frame with FT=ACK, RXRST_MNGMT=1, FLCTRLT=<Hold Time/Management>, FLCTRL_CONN=1 and FLCTRL=29. Upon reception of this frame, the transmitter shall discard the segments of the established connection and follow the procedure described in clause 8.9.5.3.

If the receiver receives a frame containing segments belonging to a connection that has been released or not established (the receiver is in RX_RESET state, see clause 8.9.5.3) it shall respond with a PHY frame with FT=ACK, RXRST_MNGMT=1, FLCTRL_CONN=1 indicating the availability of resources or a hold time by means of the FLCTRLT and FLCTRL fields.

The transmitter may release the connection by sending to the receiver a PHY frame with FT=MSG, CNN_MNGMT=0100, no payload and RPRQ = 01. Upon reception of this frame, the receiver shall reply with a PHY frame with FT=ACK, RXRST_MNGMT=1, FLCTRLT=<Hold Time/Management>, FLCTRL_CONN=1 and FLCTRL= 28 message acknowledging the release.

After receiving the release acknowledgement, the transmitter shall transition into the TX_RESET state.

If the transmitter node does not receive the release acknowledgement within the time period of CNM_TIMEOUT (see clause 8.4), the transmitter shall resend the message to release the connection. After resending this message twice without receiving the response from the receiver, the transmitter shall transition into the TX_RESET state.

8.12.5.2 Release of a data connection

The receiver may release a data connection by sending a PHY frame with FT=ACK, RXRST_DATA=1, FLCTRLT=<Hold Time/Management>, FLCTRL_CONN=0 and FLCTRL=29. Upon reception of this frame, the transmitter shall discard the segments of the established connection and follow the procedure described in clause 8.9.5.3.

If the receiver receives a frame containing segments belonging to a data connection that has been released or not established (the receiver is in RX_RESET state, see clause 8.9.5.3) it shall respond with a PHY frame with FT=ACK, RXRST_DATA=1, FLCTRL_CONN=0 indicating the availability of resources or a hold time by means of the FLCTRLT and FLCTRL fields.

The transmitter may release the connection by sending to the receiver a PHY frame with FT=MSG, CNN_MNGMT=1000, no payload and RPRQ = 01. Upon reception of this frame, the receiver shall reply with a PHY frame with FT=ACK, RXRST_DATA=1, FLCTRLT=<Hold Time/Management>, FLCTRL_CONN=0 and FLCTRL= 28 acknowledging the release.

After receiving the release acknowledgement, the transmitter shall transition into the TX_RESET state.

If the transmitter node does not receive the release acknowledgement within the time period of CNM_TIMEOUT (see clause 8.4), the transmitter shall resend the message to release the connection. After resending this message twice without receiving the response from the receiver, the transmitter shall transition into the TX_RESET state.

8.12.6 Release of a unicast connection without acknowledgements

Connections that do not require the use of the acknowledgement protocol shall be released as described in clauses 8.12.6.1 or 8.12.6.2. A connection may be released by the transmitter or by the receiver.

8.12.6.1 Release of the management connection

The transmitter may release a connection by sending to the receiver a PHY frame with FT=MSG, CNN_MNGMT=0100, no payload and RPRQ = 01. Upon reception of this frame, the receiver shall reply with a PHY frame with FT=ACK, FLCTRLT=<Hold Time/Management>, FLCTRL_CONN=1 and FLCTRL= 28 acknowledging the release.

If the transmitter node does not receive the release acknowledgement within the time period of CNM_TIMEOUT (see clause 8.4), the transmitter shall resend the message to release the connection. After resending twice this message without receiving the response from the receiver, the transmitter shall consider the connection as released.

The receiver may release a connection by sending to the transmitter a PHY frame with FT=ACK, FLCTRLT=<Hold Time/Management>, FLCTRL_CONN=1 and FLCTRL=29. Upon reception of this frame, the transmitter shall discard the segments of the established connection. This frame shall be sent in one of the TXOPs/TSs allocated to the receiver.

If the receiver receives a frame containing segments belonging to a management connection that has been released or not established it shall answer with a PHY frame with FT=ACK, FLCTRLT=<Hold Time/Management>, FLCTRL_CONN=1 and FLCTRL=29. This frame shall be sent in one of the TXOPs/TSs allocated to the receiver.

8.12.6.2 Release of a data connection

The procedure to release a data connection without acknowledgments is the same as described in clause 8.12.6.1 but with CNN_MNGMT=1000 and FLCTRL_CONN=0. In addition to this, the receiver shall identify the connection it refers to by means of the CONNECTION_ID field of the ACK frame used for releasing the connection.

8.12.7 Reset of a unicast connection with acknowledgements

Reset of a connection shall only be initiated by the transmitter. If a reset is received (see clause 8.9.5.3), the nodes shall follow the procedure described in clause 8.12.1.1 with CNN_MNGMT set to 0011₂ for a management connection, and the procedure described in clause 8.12.1.2 with CNN_MNGMT set to 0111₂ for a data connection.

8.12.8 Broadcast connections

A broadcast connection is a like a regular unicast or multicast connection except that it exists between a single source device and all other devices in the domain. A broadcast connection may be of type data or management.

A broadcast management connection is uniquely identified by the tuple (SID > 0, DID = BROADCAST_ID, MQF = 1, RPRQ = 00).

A broadcast data connection is uniquely identified by the tuple (SID > 0, DID = BROADCAST_ID, MQF = 0, RPRQ = 00).

Broadcast connection shall not use acknowledgement.

The following clauses describe the establishment and release of both data management and broadcast management connections.

8.12.8.1 Broadcast management connection

To establish a broadcast management connection the transmitter shall broadcast a PHY frame with FT=MSG, CNN_MNGMT=0010, no payload, DID = BROADCAST_ID and RPRQ = 00. The transmitter, after sending that frame, may then start sending PHY frames with segments belonging to that connection.

A receiver shall ensure that it always has sufficient resources available to establish a new broadcast management connection. It is implementation dependent how this is achieved. To release a broadcast management connection the transmitter shall send a PHY frame with FT=MSG, CNN_MNGT=0100, no payload, DID = BROADCAST_ID and RPRQ = 00. The transmitter shall consider the connection as released without waiting for any acknowledgment. Upon receiving this frame the receiver shall release the connection.

If a receiver receives frames of a broadcast management connection that it did not receive an explicit establishment request for, it shall allocate the required resources and implicitly establish the connection.

8.12.8.2 Broadcast data connection

To establish a broadcast data connection the transmitter shall broadcast a PHY frame with FT=MSG, CNN_MNGMT=0110, no payload, DID = BROADCAST_ID and RPRQ = 00. The transmitter, after sending that frame, may then start sending PHY frames with segments belonging to that connection.

To release a broadcast data connection the transmitter shall send a PHY frame with FT=MSG, CNN_MNGT=1000, no payload, DID = BROADCAST_ID and RPRQ = 00. The transmitter shall consider the connection as released without waiting any acknowledgment. Upon receiving this frame the receiver shall release the connection.

If a receiver receives frames of a broadcast data connection that was not explicitly established, it shall attempt to allocate the required resources and implicitly establish the connection. If it fails to allocate the resources for the connection it shall ignore the received frame.

8.12.9 Multicast data connection

For multicast data connections, see clause 8.16.

8.13 Message flooding

The goal of the flooding mechanism is to ensure that flooded messages are received by every node in the domain, regardless of the status of routing tables. The actual mechanism for flooding of messages is for further study.

8.14 Operation in the presence of neighbouring domains – Distributed domain coordination (NDIM)

8.14.1 General principles

The NDIM procedure shall be used in power lines. The usage of NDIM for other mediums is for further study.

The originating domain of the inter-domain messages used for NDIM shall be indicated using a 24-bit field called NDID. The NDID of a domain shall be set when the domain is established and shall never be changed during the operation of the domain, even if the domain master changes or the DNI changes. The 16 MSBs of NDID shall be the same as the DNI of the domain originating the inter-domain message and the 8 LSBs of NDID shall be a value randomly chosen by the DM when it generates the DNI (see clause 8.6.8.2) and establishes the domain. The NDIM_SYNCHRO_REF_NDID of the domain is then initialized to the value of the NDID.

NOTE – Since the DNI of two domains that are not directly interfering with each other, but are interfering with a third domain, could be the same, an additional randomly chosen byte is useful in differentiating between the messages originated by those domains and received by a third domain.

8.14.1.1 NDIM procedure

A node should be capable of detecting the presence of other neighbouring domains operating in the same medium, either directly or via information sent by other devices in its own domain, and should take all needed actions as detailed in the NDIM procedure to mitigate interference with the neighbouring domains' nodes including coordinating with them.

The NDIM procedure uses the following mechanisms:

- Inter-domain signalling that assists in detection of neighbouring domains;
- A mechanism to align the MAC cycles of all the domains that are present on the same medium and can detect each other (clause 8.14.4), independent of the level of interference among their nodes. The alignment procedure shall be triggered and executed whenever events specified in clause 8.14.4.1 happen;
- A mechanism allowing nodes to routinely update the information (clause 8.14.5) of the neighbouring domains that they detect;
- A mechanism specifying the usage of a near-orthogonal signal for generating and detecting the preamble, PR, INUSE and NACK signals in specific TXOPs. This mechanism may be applied in cases where the interference from neighbouring domains is low. With this mechanism the transmissions from other neighbouring domains are treated as noise;
- A mechanism allowing coordination between domains of transmissions from and to nodes that may interfere with each other (clause 8.14.7).

8.14.2 NDIM cluster

An NDIM cluster is defined as a group of domains that have their MAC cycles aligned. A cluster shall be uniquely identified by its 48-bit cluster identification (ClusterID).

When a new domain is created, the ClusterID of that domain shall be generated as follows:

- 1) A bit-reversed version of the REGID of the DM is generated, with the most significant bit masked with zero.
- 2) The two least significant bytes of this number are then replaced by a randomly generated 16-bit number.
- 3) If the DM is operating in coexistence mode ([[ITU-T G.9972](#)]) and is synchronized with an access system as specified in [[ITU-T G.9972](#)], the most significant bit of the resulting number shall then be set to one. This number is assigned as the ClusterID of the domain.

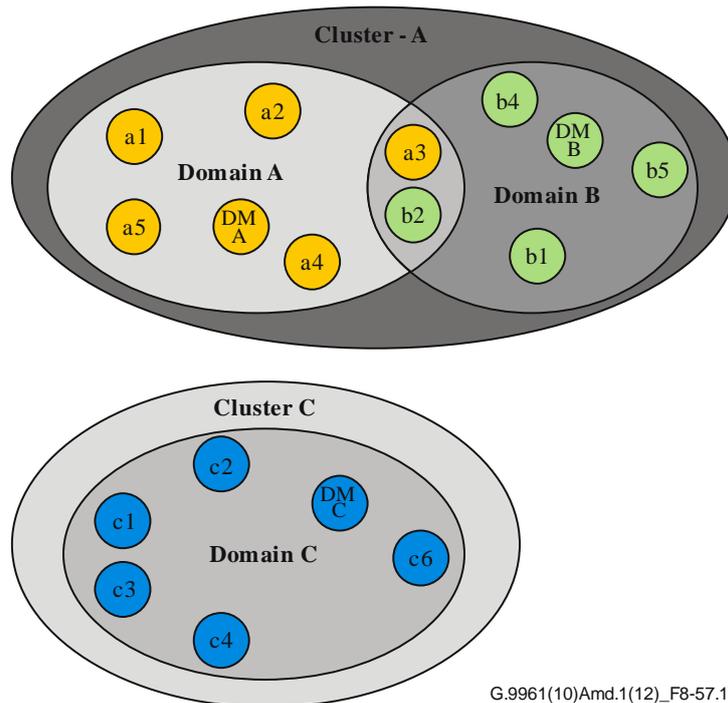
For example, if a DM has a REGID 00-B0-D0-86-BB-F7, and it is the only domain in the cluster, the initial ClusterID of that domain shall be generated as follows:

- 1) The REGID is bit-reversed and the MSB is masked with zero, resulting in the number 6F-DD-61-0B-0D-00.
- 2) The two least significant bytes are then replaced by a randomly generated 16-bit number ($xxxx_{16}$), resulting in the number 6F-DD-61-0B-xx-xx.

- 3) If the DM is not synchronized with an access system as specified in [ITU-T G.9972], this number is assigned as the ClusterID of the domain. If the DM is synchronized with an access system as specified in [ITU-T G.9972], the MSB is set to one and the number assigned as the ClusterID shall be EF-DD-61-0B-xx-xx.

NOTE – This ensures that DMs that are not operating in coexistence mode or those not aligned with an access system will align their MAC cycles with neighbouring domains that are operating in coexistence mode and are synchronized with an access system.

The ClusterID shall only be updated after either a MAC cycle alignment procedure as described in clause 8.14.4 or a routine maintenance procedure as described in clause 8.14.5. The new ClusterID to be used by the cluster results from the application of the rules described in clause 8.14.4.



G.9961(10)Amd.1(12)_F8-57.1

Figure 8-57.1 – Cluster definition

8.14.3 Inter-domain signalling and communication

The domain master shall reserve a fraction of the MAC cycle for an inter-domain signalling window (IDSW) and inter-domain communication channel (IDCC) as shown in Figure 8-57.2.

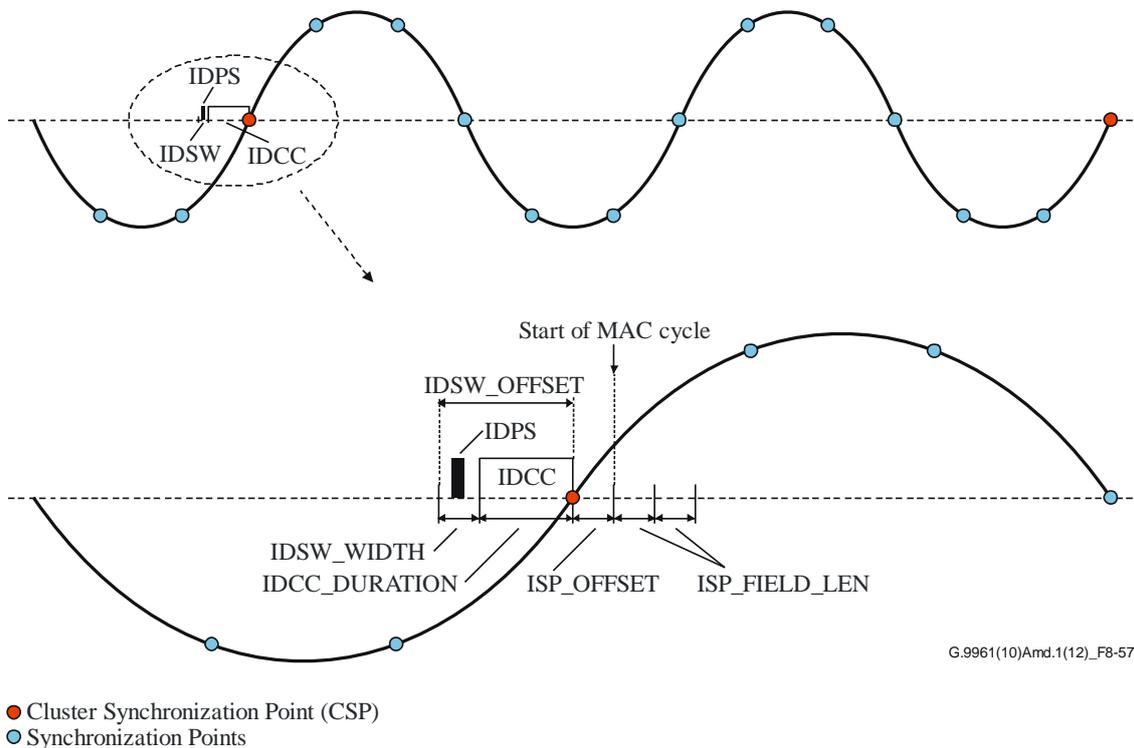


Figure 8-57.2 – Position of signalling information in the MAC cycle

The portion of the MAC cycle reserved for IDSW and IDCC is indicated by the DM using a special TXOP with the "TXOP Limitations" field in the "Additional TXOP attributes" descriptor set to 1 (see Table 8-65.1) and the "Traffic Limitation" field in the "TXOP attributes" descriptor set to 0 (see Table 8-65). Synchronization points are located at 0°, 60°, 120°, 180°, 240°, or 300° relative to a zero-crossing point of the AC mains.

In order to determine the synchronization point location, each node shall have a zero-crossing detector. The zero-crossing detector of the node shall detect the AC zero-crossing with a tolerance lower than $\pm 25 \mu\text{s}$ in 95% of the cases.

NOTE 1 – Additional filtering of the zero-crossing point detected by the circuit is recommended. However, any additional delay introduced by this filtering should be considered.

Each cluster shall use one of these synchronization points as a reference for the position of the inter-domain signalling window (IDSW) and the inter-domain communication channel (IDCC). This selected synchronization point is called the cluster synchronization point (CSP). The domain master shall ensure that the MAC cycle starts ISP_OFFSET after the CSP where ISP_OFFSET is defined in [ITU-T G.9972]. That is, the angular offset Δ defined in clause 8.6.3.1 shall be set to ISP_OFFSET .

The IDSW defines a region with width IDSW_WIDTH of the MAC cycle that shall be used by all the nodes of a cluster for transmitting their inter-domain presence signal (IDPS, see clause 7.2.2.2.5/10.2.2.2.5 of [ITU-T G.9960]). IDSW shall be present once every MAC cycle and shall be always located at a fixed offset IDSW_OFFSET before the CSP of that cluster as shown in Figure 8-57.2.

NOTE 2 – The IDSW is placed at the end of the MAC cycle where the start of the MAC cycle is aligned with TDMU defined in [ITU-T G.9972] so that NDIM works seamlessly regardless of whether ISP is enabled or not.

The IDPS shall be transmitted 50 μ s after the start of the IDSW by all nodes of all domains in the cluster in order to announce their presence to potential neighbouring domains. A node shall transmit IDPS signal as described in clause 7.2.2.2.5/10.2.2.2.5 of [ITU-T G.9960] with a periodicity of IDPS_PERIODICITY cycles. A node may choose any MAC cycle to transmit IDPS and continue transmitting IDPSs with the periodicity of IDPS_PERIODICITY.

Each DM shall set up the inter-domain communication channel (by allocating a CBTXOP containing CBTS with same attributes of minimum duration IDCC_DURATION, immediately after the IDSW, in each MAC cycle. Messages for inter-domain communication shall only be sent during the first IDCC_DURATION ms of this CBTXOP. This does not restrict this CBTXOP from other use. The DM shall indicate in the MAP that the seed to be used in the CBTXOP containing the IDCC is the 'default' seed.

NOTE 3 – The DM should use a CBTXOP of IDCC_DURATION whenever possible in order to avoid dividing the CBTXOP into two regions (an IDCC region and a non-IDCC region).

For all inter-domain management messages transmitted within IDCC, the reduced CW parameters shown in Table 8-105.3 shall be used to prioritize over intra-domain management messages that might be transmitted during IDCC. The CW parameters used for all transmissions (except transmissions of inter-domain management messages) in all CBTSs within the CBTXOP containing the IDCC shall be the default CW parameters specified in Table 8-7. INUSE, PR, and RTS/CTS shall not be used within the CBTXOP containing the IDCC.

NOTE 4 – The inter-domain management messages are the management messages starting with "IDM" in Table 8-88.

NOTE 5 – The DM and nodes should prioritize the use of this IDCC for inter-domain communications.

Table 8-105.3 – Valid $DC_{max}(BSC)$ and $NCW_{max}(BSC)$ values

BSC	$DC_{max}(BSC)$	$NCW_{max}(BSC)$
1	1	4
2	2	8
3	4	16
4	16	32

8.14.4 NDIM MAC cycle alignment procedures

MAC cycle alignment procedures consist of synchronizing the MAC cycles of two different clusters by:

- Creating a single cluster including all the nodes and domains belonging to the original clusters;
- Choosing a unique cluster synchronization point (CSP) with its associated IDSW and IDCC, to be used by all the nodes belonging to the newly created cluster for inter-domain communication;
- Assigning a unique ClusterID for identification of the resulting cluster.

At the end of the alignment procedures (any of the procedures described in clause 8.14.4) the domain master shall run the post-alignment procedure described in clause 8.14.4.4.

8.14.4.1 Full MAC cycle alignment procedure

A node shall initiate a full NDIM MAC cycle alignment procedure whenever any of the following events occur:

- On start-up, after the node is registered by the DM;
- On request by the DM via the message NDIM_StartAlignmentProcedure.ind;

- By its decision, based on vendor discretionary criteria.

When a node launches a MAC cycle alignment procedure, it shall try to detect the presence of a neighbouring domain's node. For this, it shall either try to detect IDPSs in the unaligned IDSWs for NDIM_IDPS_DETECT_CYCLES MAC cycles, or try to infer the position of the neighbouring domain's node in some other way, or both. The unaligned IDSWs are the IDSWs associated with all the synchronization points of the MAC cycle, excluding the one in use by the node's cluster. The exact algorithm, by which the node chooses the synchronization points within the MAC cycle in which to look for IDPSs, is vendor discretionary.

In order to enable searching for IDPSs in unaligned IDSWs, the DM should assign silent TXOPs, overlapping with the unaligned IDSWs, to ensure that no traffic would be transmitted by nodes of its domain.

The node shall select the IDCC associated with one of the unaligned IDSWs, for which the presence of a neighbouring domain's node has been detected, to be used for the rest of the MAC cycle alignment procedure. For the rest of the procedure, this IDSW will be designated "selected IDSW" and the IDCC designated "selected IDCC".

There are two cases of MAC cycle alignment procedure:

- 1) Local alignment: MAC cycle alignment procedure where the domain of the node that launches the MAC cycle alignment procedure has to be aligned with the neighbouring domain.
- 2) Remote alignment: MAC cycle alignment procedure where the neighbouring domain of the node that launches the MAC cycle alignment procedure has to be aligned with the local domain.

Figure 8-57.3 shows the local MAC cycle alignment procedure and Figure 8-57.4 shows the remote MAC cycle alignment procedure.

NOTE 1 – In the case that the local node is the local domain master or the remote node is the remote domain master, the message flows between the endpoint node (local or remote node that is the domain master) and the domain master are accomplished via internal communications.

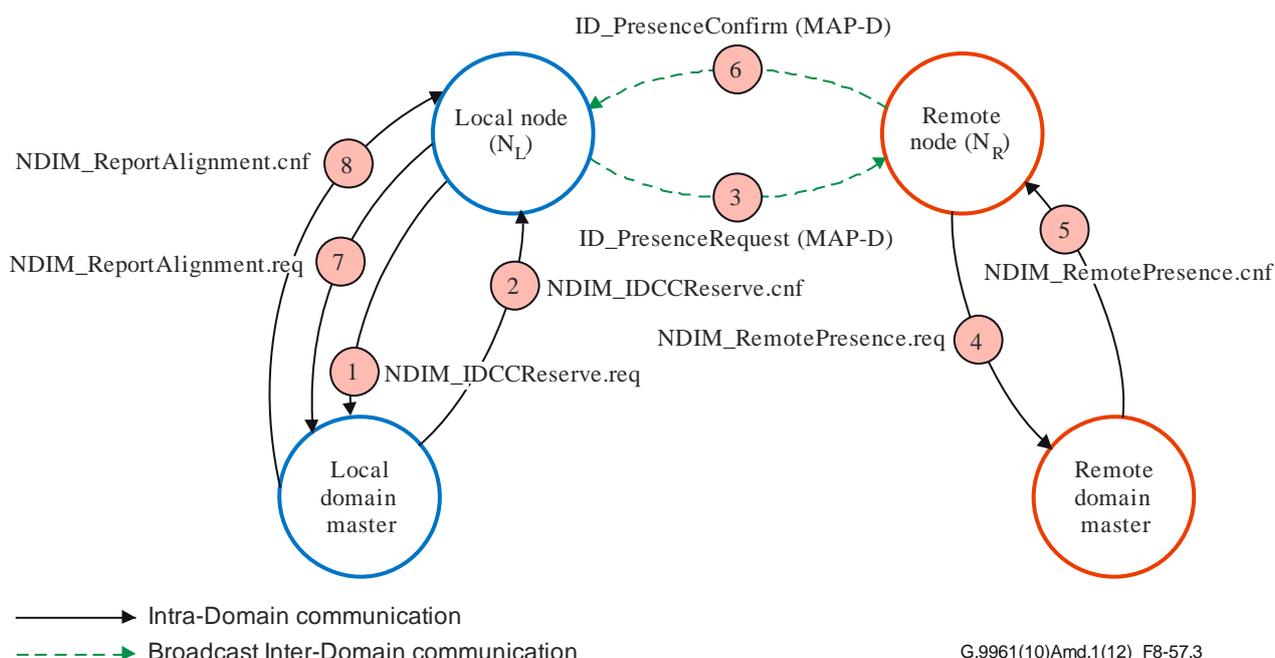


Figure 8-57.3 – Local MAC cycle alignment procedure

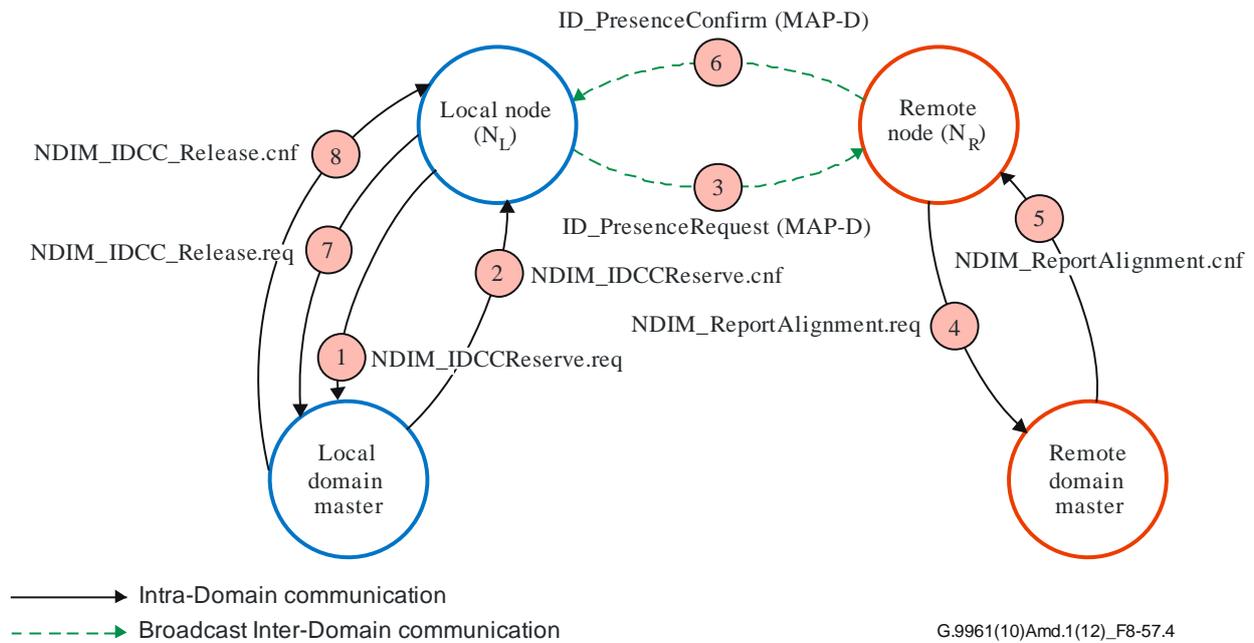


Figure 8-57.4 – Remote MAC cycle alignment procedure

The local node N_L that detected the presence of a neighbouring domain's node with a synchronization point corresponding to the selected IDSW shall start the MAC cycle alignment procedure by indicating its presence to any neighbouring domain's nodes in the selected IDCC. For this, N_L shall follow the procedure described in Figure 8-57.5.

In order to avoid any possible interference from transmissions of other nodes from the local domain during the MAC cycle alignment procedure, the detecting node shall send an `NDIM_IDCCReserve.req` (1) message to its DM requesting that it reserve time duration corresponding to the unaligned IDCC for the MAC cycle alignment procedure by assigning a CFTXOP to the node.

The DM shall respond with an `NDIM_IDCCReserve.cnf` (2) message to confirm that the reserved period shall be granted. This period shall only be used for inter-domain communication with the unaligned domain. A DM that granted bandwidth for inter-domain communication through a positive `NDIM_IDCCReserve.cnf` message shall use the granted node as a proxy for communication with the neighbouring domain the node is reporting, during the remaining MAC cycle alignment procedure, and shall answer any new request for MAC cycle alignment with an `NDIM_IDCCReserve.cnf` message with failure code `0216` until it receives an `NDIM_ReportAlignment.req` (7) message in case of local alignment or an `NDIM_IDCC_Release.req` (7) message in case of remote alignment from the node that launched the MAC cycle alignment procedure. If the DM does not receive one of these messages (7) within 600 ms, it shall terminate the alignment procedure, release the reserved "selected IDCC" slot, and can then accept other requests and allow new requests.

The requesting node shall wait for a positive response from its DM via an `NDIM_IDCCReserve.cnf` message before continuing further in the procedure. If a node receives a negative response from its DM via an `NDIM_IDCCReserve.cnf` message, or if no response is received in `RESERVE_TIMEOUT` ms, the node shall terminate the alignment procedure.

Upon reception of the positive `NDIM_IDCCReserve.cnf` message, the local node N_L that detected the presence of a neighbouring domain's node with the synchronization point corresponding to the selected IDSW shall indicate its presence to potential interfering nodes by broadcasting an `ID_PresenceRequest` (3) in the "selected IDCC". `ID_PresenceRequest` is a MAP-D frame containing the NDIM information sub-field with `ID_PRESENCE` set to zero.

After sending the ID_PresenceRequest, the local node N_L shall wait to receive an ID_PresenceConfirm (6) in the "selected IDCC". ID_PresenceConfirm is a MAP-D frame containing the NDIM information sub-field with ID_PRESENCE set to a non-zero value according to Table 8-85.5. If the node does not receive an answer from any neighbouring domain's node during NDIM_IDPS_DETECT_CYCLES MAC cycles, it shall not continue with the process and shall send its DM an NDIM_IDCC_Release.req message (7) with a failure code 01₂ to its DM.

NOTE 2 – Since ID_PresenceRequest and ID_PresenceConfirm are MAP-D frames as described in clause 8.8 (not IDMs as described in clause 8.14.6), they follow the same addressing scheme as MAP-D frames used by the domain master for distributing the information necessary for registration of new nodes.

NOTE 3 – The NDIM information sub-field contains all of the information necessary for MAC cycle alignment assessment (e.g., ClusterID, Domain ID, DeviceID of the local node, REGID of the domain's DM, NDID of the domain, CYCSTART).

When a node receives an ID_PresenceRequest in the IDCC corresponding to its cluster, it shall use the information from the received ID_PresenceRequest and apply the following rules in order to assess if a local alignment or remote alignment of the MAC cycle is needed:

- If the ClusterID of the node is higher than the ClusterID of the neighbouring node that sends the ID_PresenceRequest, the node shall send an NDIM_RemotePresence.req (4) message to its domain master including the information that the node received in the ID_PresenceRequest as described in Figure 8-57.3.
- If the ClusterID of the node is lower than the ClusterID of the neighbouring node that sends the ID_PresenceRequest, the node shall send an NDIM_ReportAlignment.req (4) message to its DM including the information that the node received in the ID_PresenceRequest, proposing an alignment with the remote cluster as described in Figure 8-57.4.

If the node sent the NDIM_ReportAlignment.req message (4) receives a negative reply (i.e., NDIM_ReportAlignment.cnf with StatusCode = 1) from its domain master, it shall not send the ID_PresenceConfirm. If the node does not receive any reply (NDIM_ReportAlignment.cnf) from its domain master within 200 ms, it shall retransmit the NDIM_ReportAlignment.req (4) message.

A domain master receiving an NDIM_ReportAlignment.req message from one of the nodes in its domain may decide to align its MAC cycle. If the domain master decides to align its MAC cycle, it shall respond with an NDIM_ReportAlignment.cnf (5) message indicating that the reporting node is allowed to report its presence to a neighbouring domain that requested to do so through an ID_PresenceConfirm.

If the node that sent the NDIM_RemotePresence.req (4) message receives a negative reply (i.e., NDIM_RemotePresence.cnf with StatusCode 02₁₆) from its domain master, it shall terminate the procedure. If the node does not receive any reply (NDIM_RemotePresence.cnf) from its domain master within 200 ms, it shall retransmit the NDIM_RemotePresence.req (4) message.

A domain master receiving an NDIM_RemotePresence.req message from one of the nodes in its domain and accepting to use that node as a proxy, shall respond with an NDIM_RemotePresence.cnf (5) message indicating that the reporting node is allowed to report its presence to a neighbouring domain that requested to do so through an ID_PresenceConfirm. If the decision is positive, the DM shall reject any new NDIM_IDCCReserve.req or NDIM_RemotePresence.req messages that contain a lower ClusterID than the ClusterID associated with the current alignment procedure until this alignment procedure has finished.

NOTE 4 – The domain master should include the proxy in the list of nodes that are authorized to transmit a MAP-D in the IDCC as soon as possible.

A node receiving an authorization from its DM via an NDIM_RemotePresence.cnf message shall respond to the previously received ID_PresenceRequest with a unicast ID_PresenceConfirm (6) containing the information necessary for MAC cycle alignment (receiving node's ClusterID, DomainID, DeviceID, REGID of the domain master, NDID of the domain) in the IDCC of its own

domain to inform the node that launched the alignment procedure of its presence. If the node does not succeed to transmit the ID_PresenceConfirm and meanwhile receives a new MAP from its DM with the new ClusterID, it shall stop trying to transmit the ID_PresenceConfirm.

Upon reception of the first ID_PresenceConfirm through the selected IDCC, the local node that launched the alignment procedure shall extract the necessary information to assess MAC cycle alignment. Any additional ID_PresenceConfirm received from other domains through the IDCC may be stored and used for further detections.

The local node shall use the information from the selected ID_PresenceConfirm and apply the following rules in order to assess if a local alignment (Local Alignment Procedure) or remote alignment (Remote Alignment Procedure) of the MAC cycle is needed:

- If the ClusterID of the local node is lower than the ClusterID of the neighbouring node, the local node shall launch a local alignment procedure as described in Figure 8-57.3, sending an NDIM_ReportAlignment.req message (7) to its DM proposing an alignment with the remote cluster;
- If the ClusterID of the local node is higher than the ClusterID of the remote node, the local node shall release the reserved TXOP of the "selected IDCC" by sending the NDIM_IDCC_Release.req message (7). The DM shall confirm receiving the message by sending an NDIM_IDCC_Release.cnf (8) message.

The alignment of the MAC cycle shall be based on the CYCSTART field of the ID_PresenceRequest or ID_PresenceConfirm received from the neighbouring domain or the NDID_CYCSTART field of the NDIM_ReportAlignment.req message transmitted by the proxy node.

To generate the NDID_CYCSTART field, NTR and CYCSTART fields received from neighbouring domains shall be converted so as to be in terms of its own domain master's clock.

NOTE 4 – For instance, the domain x's MAC cycle start can be interpreted by the domain y as $RX_Time(y) + CYCSTART(x) - NTR(x)$, taking into account wraparound, where $RX_Time(y)$ is the time according to the domain master y's transmit clock (see clause 7.1.2.3.2.1.2/10.1.2.3.2.1.2 of [ITU-T G.9960]) of the first received sample of the first OFDM symbol of the preamble of an ID_PresenceRequest or ID_PresenceConfirm transmitted by a node in domain x.

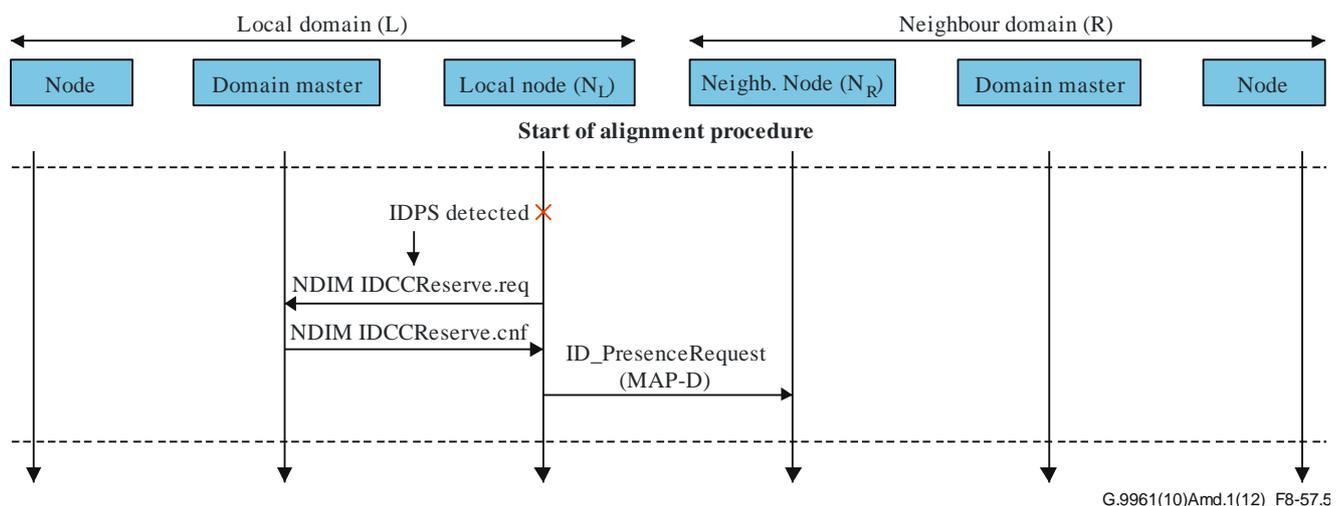


Figure 8-57.5 – Protocol diagram for start of MAC cycle alignment

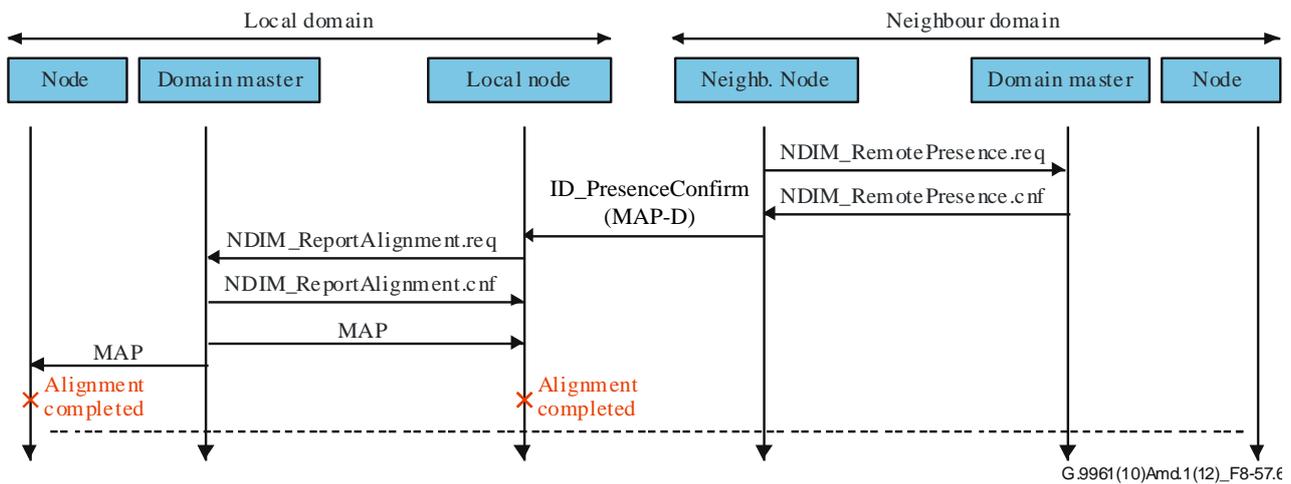


Figure 8-57.6 – Protocol diagram for local MAC cycle alignment

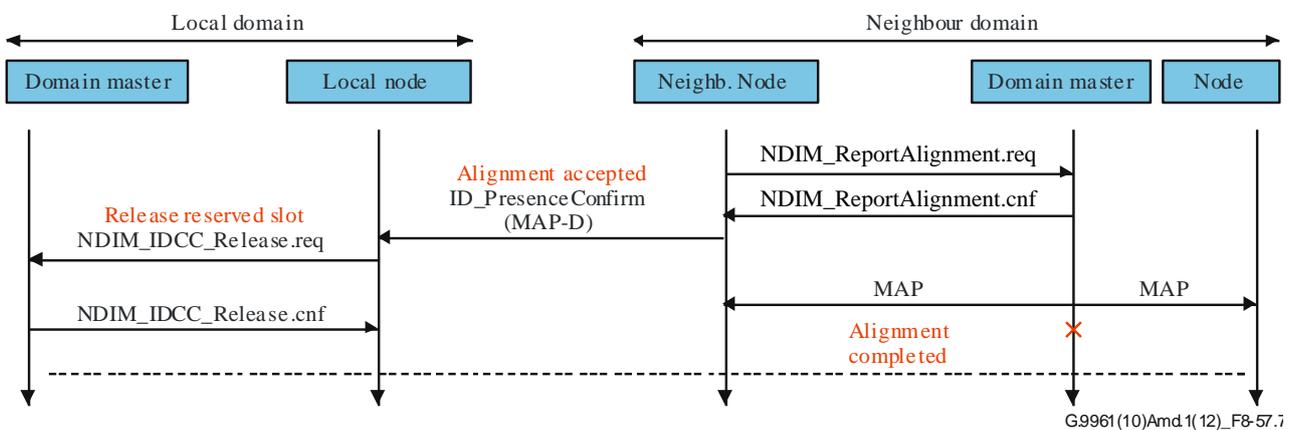


Figure 8-57.7 – Protocol diagram for remote MAC cycle alignment

8.14.4.2 Foreshortened MAC cycle alignment procedure

A node may use the messages collected during normal operation (e.g., MAPs from neighbouring domains, messages from on-going alignment procedures, MAP-D transmitted in IDCC, etc.) to align with neighbouring domains.

A node that decodes a message from a neighbouring domain that contains the ClusterID may use that for MAC cycle alignment. If the ClusterID of the local node is lower than the ClusterID of the neighbouring node, the local node shall send an NDIM_ReportAlignment.req (7) management message to its DM proposing an alignment with the remote cluster.

A node that determines that a remote alignment is necessary (i.e., if its ClusterID is higher than that of the remote node) may start a full MAC alignment procedure.

All MAP-D transmissions including ID_PresenceRequest and ID_PresenceConfirm shall contain NDIM information sub-field in addition to all parameters required for registration as defined in clause 8.8.1.

8.14.4.3 MAC cycle alignment procedure in presence of coexisting access system

The same MAC cycle alignment procedures specified in clause 8.14.4.2 or 8.14.4.3 shall be used in the presence of coexisting systems as specified in [ITU-T G.9972].

If a coexisting access system as specified in [ITU-T G.9972] is detected, the domain master shall set the MSB of its ClusterID to one to indicate the alignment of MAC cycle with a coexisting access system.

8.14.4.4 Post-alignment procedure

After MAC cycle alignment has completed, the domain master shall update NDIM information sub-field defined in Table 8-85.5.

Upon reception of an NDIM_ReportAlignment.req message with the necessary information for the alignment, the DM of the domain the node belongs to shall align its MAC cycle by adopting the received ClusterID and changing its MAP to match the start of the MAC cycle with the one associated with the new CSP. The only exception to this is for the case that the local DM is coexisting with an access system with different alignment, in which case it may decide not to align with the neighbouring cluster. The DM shall reply to the NDIM_ReportAlignment.req message with NDIM_ReportAlignment.cnf message.

A DM that received an NDIM_ReportAlignment.req message from a node of its domain shall select the node that sent the message as the proxy for inter-domain communication (e.g., coordination related messages) with the neighbouring domain reported. The selected proxy may be changed at any time based on updated interference information as described in clause 8.14.6.2.

In the case of a remote alignment procedure, the local DM that receives the NDIM_IDCC_Release.req message from its proxy node or directly receives the ID_PresenceConfirm concludes that the Remote Alignment Procedure is complete.

In the case of a local alignment procedure, the local DM that sends the NDIM_ReportAlignment.cnf message to its proxy node concludes that the local alignment procedure is complete.

A DM that aligns its MAC cycle with a neighbouring domain shall inform the rest of the DMs of the cluster that its domain has direct link with and that they should align their MAC cycle to the newly specified cluster. This is done by sending an IDM_ClusterAlignment.req message (via its former IDCC) containing the DOD of the domain requesting the alignment, the ClusterID of the new cluster to join, offset of the new CSP relative to the current CSP, and the list of the domains that this message is going to be sent to (i.e., the list of domains in the former cluster to which the domain has a direct link).

If a DM receives an IDM_ClusterAlignment.req message from another DM in its cluster while not running an alignment procedure, it shall align its MAC cycle as indicated in the IDM_ClusterAlignment.req message and shall forward the message to the other domains in its former cluster that its domain has direct links with. In order to prevent flooding of the message, the message shall be forwarded only to those domains that are not specified in the NDID list that was included in the received IDM_ClusterAlignment.req message.

Before forwarding the message, the DM shall update the list of domains included in the NDIDs list in the message by adding the ones that it will send the message to.

If a DM receives an IDM_ClusterAlignment.req message from another DM in its cluster while running an alignment procedure, it may abort this procedure in order to align its MAC cycle as indicated in the IDM_ClusterAlignment.req message.

A DM that receives an IDM_ClusterAlignment.req message shall confirm it by sending an IDM_ClusterAlignment.cnf message.

8.14.4.5 MAC cycle alignment procedure during startup of a domain

If, during initialization of a domain, the DM of this domain detects the presence of already existing clusters but does not detect the presence of a coexisting access system as specified in [ITU-T G.9972],

it shall align its MAC cycle with the domains of the cluster with the highest ClusterID that it has detected.

If, during initialization of a domain, the DM of this domain detects the presence of already existing clusters and a coexisting access system as specified in [ITU-T G.9972], it shall follow the procedure described in clause 8.14.4.3.

8.14.5 Routine maintenance procedure for a node to follow after MAC cycle alignment

Each node belonging to a cluster shall follow a routine maintenance procedure to keep track of the status of the neighbouring nodes that interfere with it. For this, the DM shall publish in the NDIM information sub field of the MAP the DEVICE_ID of the nodes that shall contend to signal their presence in the IDCC in the following MAC cycles until they receive an updated NDIM information sub field. DM shall designate the nodes of the group ensuring that they have direct visibility between them. This guarantees that only one of the nodes will win the contention and transmit a MAP-D.

If a node is designated by the DM to signal its presence in the IDCC, it shall contend for the channel. Upon successful contention, the node shall broadcast a MAP-D containing the NDIM information sub-field.

NOTE 1 – Nodes may use MAP-D messages sent in IDCC for interference level measurement.

Nodes shall continuously monitor the IDCC channel defined for their cluster for MAP-D or other messages. This procedure allows finding any new nodes in the cluster that may interfere with the monitoring node or to keep track of the neighbouring nodes that disappeared from the medium.

NOTE 2 – This procedure also enables detection of interfering nodes from a domain that is not in the cluster but whose MAC cycle is aligned initially (i.e., the same CSP was selected by both domains).

If the node detects a MAP-D belonging to a neighbouring node that is not in its list of interfering nodes, it shall update the list and notify it to its DM via NDIM_InterferenceReport.ind.

If a node does not detect a MAP-D from a neighbouring node for NDIM_IDLE_NODE MAC cycles, it shall delete it from its list of detected neighbouring nodes and notify its DM via the NDIM_InterferenceReport.ind.

8.14.6 Communication between neighbouring domains

An IDM can be transmitted directly between nodes from different domains through the IDCC or between DMs of different domains.

8.14.6.1 Node to node communication through IDCC

Node to node communication between nodes of different domains through the IDCC can be of two types: unicast node to node communication and broadcast node to multi-node communication. In either of these two cases, the OriginatingNode of the inter-domain LLC frames is set to 251 (see clause 8.1.3.1.16).

A node that transmits an MSG PHY frame that contains an IDM to a neighbouring domain's node shall follow these rules:

- It shall transmit the frame in an IDCC;
- It shall transmit the frame using the same transmission parameters as a PHY frame containing a MAP-D as described in clause 7.1.2.3.2.1.10/[10.1.2.3.2.1.10](#) of [ITU-T G.9960] and clause 8.8.2.;
- It shall use an SID 251;
- It shall include in the frame a complete IDM message;
- It shall use the MSG PHY frame type with the CNN_MNGMT field set to 1111 (payload does not belong to any connection);
- The TXOP used for IDCC shall use $T_{JFG\ MIN-D}$.

NOTE – This ensures that in case all LPDUs of a PHY frame containing an IDM are received with no error the LCDUs can be extracted completely from the carrying PHY frame.

8.14.6.1.1 Transmitting a unicast IDM to a neighbouring domain

For a unicast IDM, the destination MAC address of the LCDU shall be set to the MAC address of the neighbouring domain's node, while the source MAC address shall be set to the MAC address of the local node that generates the LCDU frame.

The REGID of the neighbouring domain master and NDID of the neighbouring domain shall be retrieved from the received ID_PresenceRequest if the neighbouring node was identified through a MAC cycle alignment procedure as described in clause 8.14.4.2 or from the MAP LCDU of the neighbouring domain's node obtained through the routine maintenance procedure as described in clause 8.14.5.

The DOD in the PHY frame header of a frame that contains a unicast IDM shall be set to the designated neighbouring domain's DOD and the DID shall be set to the DEVICE_ID of the neighbouring domain's node that is intended to receive the message. The DestinationNode of the corresponding LLC frame shall be set to the same value as the DID, and the BRCTI and MCSTI shall be both set to 0.

8.14.6.1.2 Transmitting a broadcast IDM to a neighbouring domain

For a broadcast IDM, the destination MAC address of the LCDU shall be set to the reserved MAC address 01-19-A7-52-76-96, whilst the source MAC address shall be set to the MAC address of the local node that generates the LCDU frame.

The DOD included in the PFH of the PHY frame containing the IDM shall be set to zero and the DID shall be set to the BROADCAST_ID. For the corresponding LLC frame, the DestinationNode, BRCTI and MCSTI shall be all set to 0.

8.14.6.2 Transmitting IDMs from domain master to neighbouring domain master

A DM shall select a proxy to communicate with a given neighbouring domain's DM. The selection of the proxy shall be done during the MAC cycle alignment phase and may be updated after the MAC cycle alignment phase with any interference information the DM gathers from nodes of its domain in order to make use of the best link to reach the neighbouring domain.

A DM that needs to send an LCDU to a neighbouring domain using an IDM shall embed the LCDU in an NDIM_Transmit.ind message EMBEDDED_MESSAGE field and send it to the chosen proxy in the local domain. The local domain proxy shall extract the IDM from the received LCDU and send it through the IDCC to the chosen neighbouring domain using the unicast node to node communication described in clause 8.14.6.1.

A node that receives an inter-domain management message with SID 251 shall forward it to its DM by extracting the payload of the received LCDU (MMH and MMPL) and embedding it in the EMBEDDED_MESSAGE field of an NDIM_Receive.ind message.

The LCDUs that can be exchanged between DMs using this procedure shall be the ones specified in clause 8.14.10 (Inter-domain protocol messages).

Whenever the DM of a domain changes, it shall immediately communicate the REGID of the new DM to all its neighbouring DMs using the unicast IDM_DmChange.ind message (see clause 8.14.10.17).

Whenever the DNI of a domain changes, it shall immediately communicate the new DNI to all its neighbouring DMs using the unicast IDM_DniChange.ind message (see clause 8.14.10.18).

8.14.7 Coordination of transmissions among nodes from neighbouring domains within a cluster

8.14.7.1 General Principles of neighbouring domain coordination

The neighbouring domain coordination mechanism is based on the following general principles:

- Input to the coordination algorithm consists of information on:
 - Nodes causing interference to nodes in neighbouring domains, and
 - Nodes suffering from interference coming from nodes in neighbouring domains.
- For the purpose of coordination, only nodes causing and suffering high interference, based on vendor discretionary criteria, are considered. Nodes with low interference can simply use different seeds for signals like preamble, INUSE and NACK, for interference mitigation, as described in clause 8.6.10;
- For the nodes in the domain that are not interfering, the domain master may allocate transmit opportunities anytime during the MAC cycle;
- All nodes in the domain are allowed to transmit anytime during the assigned (1/16th) of the MAC cycle within the TXOP assigned by the DM, subject to the scheduling constraints described in clause 8.14.7.2.3. This assigned time position in the MAC cycle depends on DOD of the domain;
- Permission for transmission for a node during the rest of the MAC cycle (i.e., (14/16th) of the MAC cycle, since the last (1/16th) of the MAC cycle is available for all of the domains), depends on the number of domains that this node needs to coordinate with and is based on the procedure defined in clause 8.14.7.2 that all domains follow;
- The permissions for transmission for a node that are based on the interference information may be changed by exchange of time resources between coordinating domains;
- Additional constraints to those discussed above may be applicable in some situations to deal with the time allocation corresponding to [\[ITU-T G.9972\]](#);
- Intra-domain receptions for nodes that suffer interference from neighbouring domains can be protected if the transmitting nodes of the domain are aware of the constraints applicable to the receiving nodes of the domain and transmit only during the time duration in which the receiving nodes are protected. This may be enforced by the domain master by using appropriate scheduling policy or by communicating each node's time allocation information to all other nodes in the domain.

8.14.7.2 Neighbouring domain coordination procedure

Once MAC cycles of neighbouring domains are aligned and clusters are formed following the procedure described in clause 8.14.4 or after the routine maintenance procedure described in clause 8.14.5, the procedure for coordination as described below is executed by those domains. The various steps in this procedure are as follows:

- Default MAC cycle time allocation based on DOD;
- Exchange of interference information between neighbouring domains;
- Default MAC cycle time allocation based on interference information;
- Exchange of time slots between neighbouring domains.

8.14.7.2.1 Default MAC cycle time allocation based on DOD

The MAC cycle is divided into 16 equal parts. Each of these parts is allocated to the domains with the corresponding DOD. The first part is allocated to domains with DOD 1, second part is allocated to domains with DOD 2 and so on up to the fifteenth part, which is allocated to domains with DOD 15. The sixteenth part is allocated to DOD 0 and it contains the IDSW and IDCC as described in

clause 8.14.3. At the end of the coordination process, any node in a coordinated domain can transmit or receive during the time of the MAC cycle allocated to that domain without fear of causing interference to, or suffering interference from nodes of coordinated neighbouring domains. This is called the default time allocation based on DOD. Since DOD 0 is reserved, part of that allocation is used for IDPS and IDCC. The rest of the allocation corresponding to DOD 0 is available for all the domains to share.

8.14.7.2.2 Exchange of interference information between neighbouring domains

The interference information is exchanged between neighbouring domains as follows:

- The domain master shall form lists of nodes that suffer high interference from at least one other node in a neighbouring domain (i.e., list of nodes that requires coordination). The criteria for determining the level of interference requiring coordination is vendor discretionary. It is the node's decision to determine the level of interference (either one of: no interference, possible interference or high interference) and report this to the DM to decide on further action. The node may compare relative power differences between transmissions from nodes in neighbouring domains and transmissions from nodes in its own domain. Each list consists of a node within the domain and all the node(s) in a neighbouring domain that cause high interference to this node. For example, node L_i in local domain (L) can form a list of nodes ($N_1, N_2, N_3, \dots, N_j$) in neighbouring domain (N) that cause high interference;
- This process is repeated for all nodes suffering high interference from nodes in neighbouring domains;
- The lists are then communicated to the neighbouring domains, by unicast transmission where the interfering nodes belong, via an `IDM_InterfNodesInfo.ind` message. For example, the list above is communicated by domain L to domain N. To reduce the size of this message, the nodes that are suffering interference from the same set of interfering nodes in the same set of neighbouring domains shall use a single interference table (see Table 8-105.24). The destination DM for the `IDM_InterfNodesInfo.ind` message shall send a confirmation `IDM_InterfNodesInfo.rsp` message upon receiving the `IDM_InterfNodesInfo.ind` message.
- The same process is executed by all the domains in the cluster;
- Once all the lists are communicated to respective neighbouring domains, each domain master has complete information on its nodes that cause interference to nodes in neighbouring domains (using received `IDM_InterfNodesInfo.ind` messages) and also its nodes that suffer interference from nodes in neighbouring domains (information used to generate the transmitted `IDM_InterfNodesInfo.ind` message);
- Using this information, for each node, the local domain master comes up with the list of DoDs of neighbouring domains that contain at least one node that causes interference or suffers interference or both;
- At the end of the previous step, for each node that needs coordination, the local DM has a list of DoDs that it needs to coordinate with;
- The list of domains that each node needs to coordinate with, are communicated to each one of the domains on that list via an `IDM_CoordDomainsInfo.ind` message. To reduce the size of this message, the nodes that are coordinating with the same set of domains shall use a single coordination table (see Table 8-105.29). The destination DM for the `IDM_CoordDomainsInfo.ind` message shall send a confirmation `IDM_CoordDomainsInfo.rsp` message upon receiving the `IDM_CoordDomainsInfo.ind` message;

NOTE – The `IDM_CoordDomainsInfo.ind` message does not contain list(s) of interfering nodes since that information is already communicated in the `IDM_InterfNodesInfo.ind` messages.

- At the end of the previous step, each node causing or suffering interference has complete knowledge of the allocation table for each one of its interfering nodes, in addition to its own allocation table. The domain master has this information for each node that requires

coordination. The domain master can then make scheduling decisions for each node, based on this information.

8.14.7.2.3 Default MAC cycle time allocation based on interference information

After the interference information is exchanged between domains, the rest of the (14/16th) of the MAC cycle that has not been allocated as per clause 8.14.7.2.1, is allocated using the interference information as per the rules described in this clause.

The basic principle of allocating a MAC cycle is to first assign the slot corresponding to the DOD of the local domain and then allocate slots corresponding to the DODs of the neighbouring domains that the node is coordinating with. The remaining unallocated slots are then divided into equal number of slots (with a 1 slot resolution) and allocated to each of the coordinating domains in sequential order. The allocation is done separately for each node that requires coordination with nodes in neighbouring domains.

The various steps in the allocation of time for each node that needs to coordinate with one or more neighbouring domains are as follows:

- First, the allocation table for a given node is generated depending on the number of domains that it needs to coordinate with. At this point each one of the coordinating domain masters has its own default slot allocated, including the local domain master of that node;
- The total number of unallocated slots is (15-D), where D is the number of coordinating domains with unique DODs that are interfering with or being interfered by this node, including the local and remote domains. Remote coordinating domains that have the same DOD can be considered as a single coordinating domain, for the purpose of allocation, without loss of generality. The unallocated slots are equally divided between each of the coordinating domains and assigned sequentially in time in the order of the DODs of the coordinating domains. At this point each of the domains, gets floor ((15-D)/D) additional time slots allocated to its node. The remaining (15-D) MODULO D, slots are marked as "unallocated";
- The above two steps are repeated for each one of the coordinating nodes in neighbouring domains, i.e., for each node in neighbouring domains that a given node is coordinating with. The allocation table for each one of the coordinating nodes in neighbouring domains is then compared with node's own allocation table and the slots with conflict may also be marked as "unallocated";
- The allocated slots can be used by the node as assigned by the domain master in the MAP. The domain master shall consider the allocation table of each individual node, while making its scheduling decisions. The scheduling by the domain master shall ensure that any transmissions by a node associated with the allocated slots shall end at least T_{IFG_MIN} before the end of the allocated slot if the next slot is not allocated to the node in the allocation table. The mandatory IFG at the end of frames transmitted inside a TXOP can be used to meet this requirement;

NOTE – This ensures that even if there is a skew of up to T_{IFG_MIN} in the MAC cycle alignment between coordinating domains, there is no interference between the coordinating nodes in those domains associated with the same allocation table.

- The usage of unallocated slots is explained in the next clause.

8.14.7.2.4 Static Exchange of time slots between neighbouring domains

The unallocated slots can be shared between nodes of the coordinating domains or can be used by one of the coordinating domains by negotiation using inter-domain messaging between those domains. The exchange of time slots in a given allocation period shall apply to the next allocation period.

The sharing, swapping and allocation requests (i.e., `IDM_ShareUnallocSlot.req`, `IDM_SwapAllocSlot.req` and `IDM_RequestUnallocSlot.req` messages respectively) are associated with a specific coordination table indicated in the corresponding message using the same ordinal number as the one used in the `CoordDomainsInfo.ind` for that coordination table (see Table 8-105.29). For example, for sharing request associated with the second coordination table, the `COORD_TABLE_ID` field is set to 2, indicating that the request is associated with the second coordination table from the `CoordDomainsInfo.ind`. The new transaction ID associated with this coordination table and the transmitting DM shall be generated for each request starting from a value of zero and incrementing for every new request. The status of eight previous consecutive associated transactions (with reference to current transaction) is indicated in each request message. If the status of a previous transaction is not available (for example, at the first transaction), the transaction status shall be set to "no status available" (see Tables 8-105.28, 8-105.31 and 8-105.34).

A domain master may propose sharing of one or more unallocated slots by sending the `IDM_ShareUnallocSlot.req` message to the coordinating domain masters. Sharing means nodes with high interference from neighbouring domains respect each other's signals and use default seeds for signal transmissions during this time. All coordinating domain masters may allocate CBTXOPs with default preamble to enable this kind of sharing. The slots are considered as shared after receiving an `IDM_ShareUnallocSlot.cnf` message with `StatusCode 0016` from all coordinating domain masters. If the confirmation is expected from multiple domain masters, the node that has sent the `IDM_ShareUnallocSlot.req` message shall indicate to the coordinating domain masters that sharing has been successful or failed by sending an `IDM_ShareUnallocSlot.ind` message. All coordinating domain masters associated with the shared slot shall allocate a CBTXOP containing CBTS with same attributes, whenever the nodes associated with the shared slot are allowed to transmit in that CBTXOP. The default CW parameters specified in Table 8-7 shall be used for those CBTXOPs. Also INUSE, PR, and RTS/CTS shall not be used within these CBTXOPs.

Any coordinating domain master that receives the `IDM_ShareUnallocSlot.req` message can indicate in the `IDM_ShareUnallocSlot.cnf` message that it wishes the requesting domain master to refrain from sending it additional `IDM_ShareUnallocSlot.req` messages for `RefrainMultiple` allocation periods.

A domain master may request allocation of one or more unallocated slots by sending the `IDM_RequestUnallocSlot.req` message to coordinating domain masters. The message shall contain an indication of the priority of the request, which is based on DM's vendor discretionary criteria. The slots are considered allocated after receiving an `IDM_RequestUnallocSlot.cnf` message with `StatusCode 0016` from all coordinating domain masters. If the confirmation is expected from multiple domain masters, the node sending the `IDM_RequestUnallocSlot.req` message shall indicate to the coordinating domain masters that the request has been successful or failed by sending an `IDM_RequestUnallocSlot.ind` message.

Any coordinating domain master that receives the `IDM_RequestUnallocSlot.req` message can indicate in the `IDM_RequestUnallocSlot.cnf` message that it wishes the requesting domain master to refrain from sending it additional `IDM_RequestUnallocSlot.req` messages for `RefrainMultiple` allocation periods.

A domain master may request swapping of allocated slots by sending an `IDM_SwapAllocSlot.req` message to coordinating domain masters. The slots are considered swapped after receiving an `IDM_SwapAllocSlot.cnf` message with `StatusCode 0016` from all coordinating domain masters. If the confirmation is expected from multiple domain masters, the node sending the `IDM_SwapAllocSlot.req` message shall indicate to the coordinating domain masters that the request has been successful or failed by sending an `IDM_SwapAllocSlot.ind` message. The swapping of allocated slots may only be done for the slots allocated based on interference information as described in clause 8.14.7.2.3 and not for the slots allocated based on DOD as described in clause 8.14.7.2.1.

Any coordinating domain master that receives the IDM_SwapAllocSlot.req message can indicate in the IDM_SwapAllocSlot.cnf message that it wishes the requesting domain master to refrain from sending it additional IDM_SwapAllocSlot.req messages for a RefrainMultiple allocation periods.

8.14.7.3 Impact of ITU-T G.9972 on neighbouring domain coordination

A node supporting [ITU-T G.9972] follows the allocation table from that Recommendation, which is dependent on its NetworkStatus. Supporting [ITU-T G.9972] may in general result in additional constraints in allocation of time slots to nodes for neighbouring domain coordination as described in this Recommendation. To mitigate the impact of supporting [ITU-T G.9972], a domain master should first try to change its DOD so that the default allocation of the MAC cycle as per clause 8.14.7.2.1 overlaps with the time allocation based on [ITU-T G.9972] for most of its nodes. Furthermore, using the procedure described in clause 8.14.7.2.4, domain masters of coordinating domains can further exchange time slots, taking into account the NetworkStatus of the interfering nodes, to mitigate the impact of supporting [ITU-T G.9972]. Even after executing the above steps, if the domain master ends up with no allowed time allocation for a node, it may allow un-coordinated transmissions from this node within the time allocated as per [ITU-T G.9972] NetworkStatus.

8.14.7.4 Examples of allocation

In these examples the DODs of domains A, B, C, D, and E are assumed to be 1, 2, 3, 4, and 5 respectively, without loss of generality. The nodes shown inside a circle are the nodes that belong to the domain. For example, in Figure 8-57.8 nodes b1, b2 and b3 belong to domain B with DOD 2.

The slots marked with 'X' in orange correspond to unallocated time slots. For example, slot No. 15 in Figure 8-57.9 is unallocated.

8.14.7.4.1 Example of simple neighbouring domain coordination between five domains

Consider five coordinating domains as shown in Figure 8-57.8.

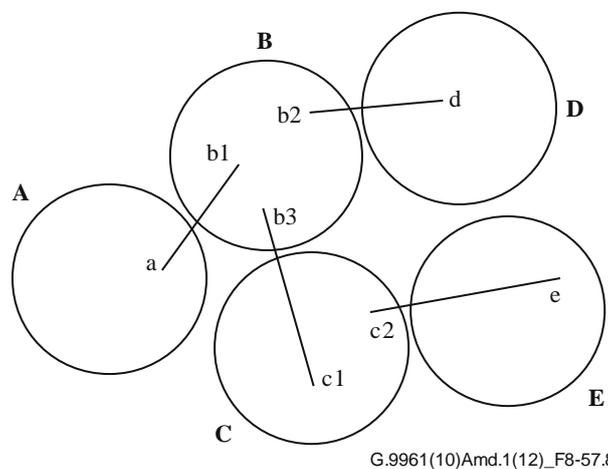


Figure 8-57.8 – Illustration of simple neighbouring domain coordination

This is a simple example of coordination because each coordinating node requires coordination with a node from just one other domain. The allocation of MAC cycle time slots based on the procedure explained in clause 8.14.7.2 is shown in Figure 8-57.9. The time slots allocated based on clause 8.14.7.2.1 are marked in yellow and those allocated based on clause 8.14.7.2.3 are marked in green.

	Slots		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Nodes		–																
a, b1		–	a	b1	a	a	a	a	a	a	b1	b1	b1	b1	b1	b1	X	IDCC
		–																
b2, d		–	b2	b2	b2	d	b2	b2	b2	b2	d	d	d	d	d	d	X	IDCC
		–																
b3, c1		–	b3	b3	c1	b3	b3	b3	b3	b3	c1	c1	c1	c1	c1	c1	X	IDCC
		–																
c2, e		–	c2	c2	c2	c2	e	c2	c2	c2	e	e	e	e	e	e	X	IDCC

Figure 8-57.9 – Allocation of MAC cycle time slots for example of simple neighbouring domain coordination

8.14.7.4.2 Example of neighbouring domain coordination between five domains with conflicts

Consider five coordinating domains as shown in Figure 8-57.10.

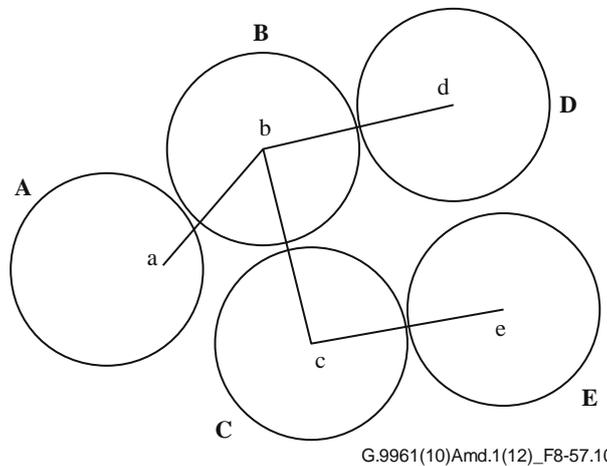


Figure 8-57.10 – Illustration of neighbouring domain coordination

This is an example of coordination where some coordinating nodes require coordination with nodes from more than one other domain. The allocation of MAC cycle time-slots based on the procedure explained in clause 8.14.7.2 is shown in Figure 8-57.11.

	Slot		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Node		–																
a		–	a	b	a	a	a	a	a	a	b	b	b	b	b	b	X	IDCC
		–																
b		–	a	b	c	d	a	a	b	b	c	c	d	d	X	X	X	IDCC
		–																
c		–	b	b	c	b	e	b	b	c	c	c	c	e	e	e	e	IDCC
		–																
d		–	b	b	b	d	b	b	b	b	d	d	d	d	d	d	X	IDCC
		–																
e		–	c	c	c	c	e	c	c	c	e	e	e	e	e	e	X	IDCC

	Slot		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Node		–																
a		–	a	b	a	a	a	a	X	X	b	b	b	b	b	b	X	IDCC
		–																
b		–	a	b	c	d	a	a	X	X	c	c	d	d	X	X	X	IDCC
		–																
c		–	b	b	c	b	e	b	b	X	X	X	X	e	e	e	e	IDCC
		–																
d		–	b	b	b	d	b	b	b	b	d	d	d	d	d	d	X	IDCC
		–																
e		–	c	c	c	c	e	c	c	c	X	X	X	e	e	e	X	IDCC

Figure 8-57.11 – Default allocation table with conflicts

The time slots allocated based on clause 8.14.7.2.1 are marked in yellow and those allocated based on clause 8.14.7.2.3 are marked in green. The slots with conflict are marked with red (see first allocation table in Figure 8-57.11). Some of the time slots are "unallocated" and can be used as per the procedure described in clause 8.14.7.2.4 (see slots marked with X in both tables in Figure 8-57.11). In the final allocation table, the slots with conflict are also considered as "unallocated" (see the second allocation table in Figure 8-57.11).

8.14.7.5 Synchronization of neighbouring domain coordination

The procedure described for neighbouring domain coordination needs to be synchronized so that the time allocations between neighbouring domains are based on the same set of information. Each domain master in a cluster maintains a synchronizing counter of MAC cycles that matches other domain masters in the cluster.

Whenever a new neighbouring domain in the cluster is detected, the domain master shall compare its NDIM_SYNCHRO_REF_NDID to that of the new domain. If it is lower, the domain master shall synchronize its synchronizing counter with that of the newly detected domain's synchronizing counter and set its NDIM_SYNCHRO_REF_NDID to that of the new domain. This procedure ensures that all the domains within a cluster have their synchronizing counter synchronized with that of the domain with the largest NDID.

The MAC cycle time allocation based on the interference information (see clause 8.14.7.2.3) and coordination messages exchanged during an allocation period shall be applied by all coordinating domains within a cluster, in the next allocation period. Additionally, a domain master may update its decision on participating in the coordination procedure every allocation period. It shall indicate its preferred coordination technique for the next period by sending the IDM_CoordPref.ind message to its neighbouring domain masters. If the coordination method indicated is "DM does not coordinate", the IDM_CoordPref.ind messages shall be the only IDM messages related to coordination sent in the allocation period.

During the routine maintenance procedure described in clause 8.14.5, new nodes or domains may be detected requiring re-alignment of the MAC cycle followed by neighbouring domain coordination.

Whenever a new domain joins a cluster:

- It shall first check the NDIM_SYNCHRO_REF_NDID field of its neighbouring domains in the cluster and determine the maximum NDIM_SYNCHRO_REF_NDID among those domains.
- It shall acquire (i.e., match) the synchronizing counter from this domain (see field NDIM_SYNCHRO_COUNTER in Table 8-85.5) whose NDIM_SYNCHRO_REF_NDID is the maximum.
- It shall then set its NDIM_SYNCHRO_REF_NDID to the maximum of its NDID and the maximum NDIM_SYNCHRO_REF_NDID among its neighbouring domains.

For neighbouring domain coordination, the exchange of interference information between neighbouring domains (see clause 8.14.7.2.2) shall be done within ALLOCATION_PERIOD MAC cycles after the previous coordination. The IDM_InterfNodesInfo.ind message transmissions shall be completed within the first $(\text{ALLOCATION_PERIOD}/2 - \text{ALLOCATION_MARGIN})$ MAC cycles and the IDM_CoordDomainsInfo.ind message transmissions shall be completed within the first $(\text{ALLOCATION_PERIOD}/2 - \text{ALLOCATION_MARGIN})$ MAC cycles of the second half of the allocation period. The ALLOCATION_MARGIN may be used for any retransmissions required due to lost messages, but shall not be used for transmitting new messages. Static exchange of time slots between neighbouring domains (see clause 8.14.7.2.4) may be initiated after the coordination tables for associated nodes are known by the coordinating domain masters.

The default MAC cycle time allocation based on the interference information (see clause 8.14.7.2.3) shall be applied by all coordinating domains within a cluster in the next allocation period. At the end of every allocation period, all DMs should check that the NDIM_SYNCHRO_REF_NDID of neighbouring domains is less than or equal to its own NDIM_SYNCHRO_REF_NDID. If it detects an NDIM_SYNCHRO_REF_NDID that is greater than its own, it shall acquire the synchronizing counter from the domain (see field NDIM_SYNCHRO_COUNTER in Table 8-85.5) whose NDIM_SYNCHRO_REF_NDID is the maximum among its neighbouring domains in the cluster and set its NDIM_SYNCHRO_REF_NDID to that of this domain.

8.14.8 NDIM parameters

The parameters needed in the neighbouring domains interference mitigation process are presented in Table 8-105.4.

Table 8-105.4 – Neighbouring domains coordination parameters

Notation	Parameter	Valid values or range
IDSW_OFFSET	Offset of IDSW window with respect to the cluster synchronization point	IDSW_WIDTH + IDCC_DURATION (Note)
IDSW_WIDTH	Width of the IDSW window	150 μs

Table 8-105.4 – Neighbouring domains coordination parameters

Notation	Parameter	Valid values or range
IDCC_DURATION	Duration of the IDCC channel	1 ms
NDIM_IDPS_DETECT_CYCLES	Number of MAC cycles during which a node has to look for IDPS signals during a MAC cycle alignment procedure	2
NDIM_PRESENCE_DETECT_CYCLES	Number of MAC cycles during which a node has to look for an ID_PresenceConfirm received in response to a transmitted ID_PresenceRequest in the IDCC	10
IDPS_PERIODICITY	Periodicity of transmission of IDPS by a node in units of MAC cycles	2
RESERVE_TIMEOUT	Timeout in ms for the reservation of resources for inter-domain communication	100
NDIM_IDLE_NODE	Number of MAC cycles without detecting a neighbouring node before inferring that the node has disappeared	100
ALLOCATION_PERIOD	Number of MAC cycles that an allocation table generated by coordinating domains in a cluster is valid. The allocation table generated in one allocation period applies to the next allocation period	100
ALLOCATION_MARGIN	Number of MAC cycles towards the end of the allocation period where exchange of new information between coordinating domains is disallowed and only retransmissions are allowed	10

NOTE – IDSW and IDCC are placed before the CSP. In coexistence mode [ITU-T G.9972], the CSP shall be aligned with the [ITU-T G.9972] synchronization point so that IDSW and IDCC are located in the last TDMS to ensure that the IDCC is guaranteed an allocation.

8.14.9 NDIM protocol messages

8.14.9.1 NDIM Start alignment procedure request

NDIM_StartAlignmentProcedure.ind is a management message that shall be sent by a DM to a node of its domain to instruct it to trigger a MAC cycle alignment procedure. The MMPL of this message is empty.

8.14.9.2 NDIM IDCC slot reserve request

NDIM_IDCCReserve.req is a management message that shall be sent by a node that intends to launch an NDIM MAC cycle alignment procedure to its DM in order to request a CFTXOP during the duration of the IDCC corresponding to a particular CSP.

The format of the MMPL of the NDIM_IDCCReserve.req message shall be as shown in Table 8-105.5.

Table 8-105.5 – Format of the MMPL of the NDIM_IDCCReserve.req message

Field	Octet	Bits	Description
OFFSET_DIR	0	[0]	This field contains the direction of the offset to be applied to the CSP to identify the required synchronization point to be protected. 0 if the CSP has to be delayed, 1 if the CSP has to be moved forward
OFFSET		[4:1]	Offset to be applied to CSP to obtain the synchronization point to be protected in increments of 60 degrees, where the value 0000 ₂ represents 0 degrees (see Figure 8-57.2). Valid values are from 0 to 11
Reserved		[7:5]	Reserved by ITU-T (Note)
Proposed_Start_Time	1 to 4	[31:0]	Proposed start time of the reservation period counted from the beginning of the MAC cycle in TIME_UNIT units where the size of a TIME_UNIT is equal to the base TICK size multiplied by a constant factor defined in the MAP header (see TICK_Factor in clause 8.8.3). The values of TICK are defined in clause 8.4.
Proposed_Stop_Time	5 to 8	[31:0]	Proposed stop time of the reservation period counted from the beginning of the MAC cycle in TIME_UNIT units where the size of a TIME_UNIT is equal to the base TICK size multiplied by a constant factor defined in the MAP header (see TICK_Factor in clause 8.8.3). The values of TICK are defined in clause 8.4.
NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

8.14.9.3 NDIM IDCC slot reserve confirmation

NDIM_IDCCReserve.cnf is a management message that shall be sent by the DM in response to an NDIM_IDCCReserve.req message.

The format of the MMPL of the NDIM_IDCCReserve.cnf message shall be as shown in Table 8-105.6.

Table 8-105.6 – Format of the MMPL of the NDIM_IDCCReserve.cnf message

Field	Octet	Bits	Description
StatusCode	0	[7:0]	Status of the request to reserve a CFTXOP: <ul style="list-style-type: none"> • 00₁₆ = Success. • 01₁₆ = Failure – Lack of resources.

Table 8-105.6 – Format of the MMPL of the NDIM_IDCCReserve.cnf message

Field	Octet	Bits	Description
			<ul style="list-style-type: none"> • 02₁₆ = Alignment process already started with another node • 03₁₆ = Failure due to other reason. • 04₁₆ – FF₁₆ = Reserved.
AlignmentclusterID	1 to 6	[47:0]	If StatusCode is 02 ₁₆ this field contains the ClusterID corresponding to the alignment process the domain is already involved in and that caused the failure of the request. In all other cases, this field shall not be present

8.14.9.4 NDIM_ReportAlignment.req format

NDIM_ReportAlignment.req is a management message that shall be sent by a node to inform its DM on the necessity to align the MAC cycle with a neighbouring domain.

The format of the MMPL of the NDIM_ReportAlignment.req message shall be as shown in Table 8-105.7.

Table 8-105.7 – Format of the MMPL of the NDIM_ReportAlignment.req message

Field	Octet	Bits	Description
NEW_CLID	0 to 5	[47:0]	This field contains the new cluster ID that has to be applied to the cluster after completion of MAC cycle alignment indicated by this message
Type of Alignment	6	[0]	This field contains: 0 for a local alignment, or 1 for a remote alignment (see clause 8.14.4.1)
OFFSET_DIR		[1]	This field contains: 0 if the CSP has to be delayed, or 1 if the CSP has to be moved forward
OFFSET		[5:2]	This field contains the offset in increments of 60 degrees, where the value 0000 ₂ represents 0 degrees (see Figure 8-57.2), to be applied to the CSP of the cluster to align the MAC cycle. Valid values are from 0 to 11
Reserved		[7:6]	Reserved by ITU-T (Note 1)
DM_REGID	7 to 12	[47:0]	The REGID of the DM corresponding to the domain to which the node has requested to align
NDID	13 to 15	[23:0]	The NDID of the domain to which the node has requested to align
NDID_CYCSTART	16 to 19	[31:0]	The start of the MAC cycle of the neighbouring domain indicated in the

Table 8-105.7 – Format of the MMPL of the NDIM_ReportAlignment.req message

Field	Octet	Bits	Description
			MAP-D received by a proxy node (Note 2)
NOTE 1 – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			
NOTE 2 – It represents the CYCSTART field of the MAP-D received from neighbouring domain in terms of its own domain master's clock count.			

8.14.9.5 NDIM_ReportAlignment.cnf format

NDIM_ReportAlignment.cnf is a management message that shall be sent by a DM as a response to a received NDIM_ReportAlignment.req message.

The format of the MMPL of the NDIM_ReportAlignment.cnf message shall be as shown in Table 8-105.8.

Table 8-105.8 – Format of the MMPL of the NDIM_ReportAlignment.cnf message

Field	Octet	Bits	Description
StatusCode	0	[7:0]	0 – MAC cycle alignment request is accepted 1 – MAC cycle alignment request is rejected 2-255: Reserved by ITU-T

8.14.9.6 NDIM Remote presence request

NDIM_RemotePresence.req is a management message that shall be sent by a node that received an ID_PresenceRequest in its cluster's IDCC to its DM in order to ask for permission to answer the request.

The format of the MMPL of the NDIM_RemotePresence.req message shall be as shown in Table 8-105.9.

Table 8-105.9 – Format of the MMPL of the NDIM_RemotePresence.req message

Field	Octet	Bits	Description
RemoteNode_DOD	0	[7:0]	The domain ID of the remote node
RemoteNode_CLID	1 to 6	[47:0]	The cluster ID of the remote node
RemoteNode	7	[7:0]	The DeviceID of the remote node
RemoteNode_NDID	8 to 10	[23:0]	The NDID of the remote node's domain

8.14.9.7 NDIM Remote presence confirmation

NDIM_RemotePresence.cnf is a management message that shall be sent by a DM in response to a node that sent an NDIM_RemotePresence.req message.

The format of the MMPL of the NDIM_RemotePresence.cnf message shall be as shown in Table 8-105.10.

Table 8-105.10 – Format of the MMPL of the NDIM_RemotePresence.cnf message

Field	Octet	Bits	Description
StatusCode	0	[7:0]	Status of the request: <ul style="list-style-type: none"> • 00₁₆ = Success. The node is allowed to answer the ID_PresenceRequest and the domain shall align with the neighbouring domain • 01₁₆ = Success. The node is allowed to answer the ID_PresenceRequest and the neighbouring domain shall align with this domain • 02₁₆ = Failure. The node is not allowed to respond to the ID_PresenceRequest • 03₁₆ – FF₁₆ = Reserved

8.14.9.8 NDIM Transmit indication

NDIM_Transmit.ind is a management message that shall be sent by a DM to a node of its domain used as a proxy for DM to DM communication between different domains.

The format of the MMPL of the NDIM_Transmit.ind message shall be as shown in Table 8-105.11.

Table 8-105.11 – Format of the MMPL of the NDIM_Transmit.ind message

Field	Octet	Bits	Description
DM_REGID	0 to 5	[47:0]	The REGID of the DM corresponding to the domain to which the inter-domain message is destined
NDID	6 to 8	[23:0]	The NDID of the domain to which the inter-domain message is destined
EMBEDDED_MESSAGE	Variable	<u>Variable</u>	LCDU to be transmitted through the IDCC. Only IDM may be embedded

8.14.9.9 NDIM Receive indication

NDIM_Receive.ind is a management message that shall be sent by a node used as a proxy for DM to DM communication between different domains to its DM.

The format of the MMPL of the NDIM_Receive.ind message shall be as shown in Table 8-105.12.

Table 8-105.12 – Format of the MMPL of the NDIM_Receive.ind message

Field	Octet	Bits	Description
DM_REGID	0 to 5	[47:0]	The REGID of the DM corresponding to the domain that originated the inter-domain message
NDID	6 to 8	[23:0]	The NDID of the domain that originated the inter-domain message
EMBEDDED_MESSAGE	Variable	<u>Variable</u>	LCDU transmitted through the IDCC. Only IDM may be embedded

8.14.9.10 NDIM IDCC_Release message format

NDIM_IDCC_Release.req is a management message that shall be sent by a node that has received a positive NDIM_IDCCReserve.cnf after the node has finished using this reserved allocation.

The format of the MMPL of the NDIM_IDCC_Release.req message shall be as shown in Table 8-105.13.

Table 8-105.13 – Format of the MMPL of the NDIM_IDCC_Release.req message

Field	Octet	Bits	Description
RemoteNode_DOD	0	[7:0]	The domain ID of the remote node
RemoteNode_CLID	1 to 6	[47:0]	The cluster ID of the remote node
RemoteNode	7	[7:0]	The DeviceID of the remote node
StatusCode	8	[7:0]	0 – The reserved slot can be released because a neighbouring node has replied with an ID_PresenceConfirm 1 – The reserved slot can be released because no neighbouring node has replied with an ID_PresenceConfirm 2-255 is reserved by ITU-T (Note)

NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.

8.14.9.11 NDIM IDCC_Release confirmation

The NDIM_IDCC_Release.cnf is a management message that shall be sent by the DM to confirm receiving an NDIM_IDCC_Release.req message.

The format of the MMPL of the NDIM_IDCC_Release.cnf message shall be as shown in Table 8-105.14.

Table 8-105.14 – Format of the MMPL of the NDIM_IDCC_Release.cnf message

Field	Octet	Bits	Description
StatusCode	0	[7:0]	0 – The reserved slot is released 1-255 is reserved by ITU-T (Note)

NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.

8.14.9.12 NDIM interference report indication

NDIM_InterferenceReport.ind is a management message that shall be sent by a node that detected a new interfering source, or a change in an already existing interfering source, to indicate to its DM the level of the interference.

The format of the MMPL of the NDIM_InterferenceReport.ind message shall be as shown in Table 8-105.15.

Table 8-105.15 – Format of the MMPL of the NDIM_InterferenceReport.ind message

Field	Octet	Bits	Description
DEVICE_ID	0	[7:0]	DEVICE_ID of the node whose interference information is conveyed in this message
Type	1	[7:0]	Shall be set to: <ul style="list-style-type: none"> – 00₁₆ if report includes a record for every neighbouring domain the reporting node detects – 01₁₆ if the report is an update about the neighbouring domain the reporting node detects (Note 8) Other values are reserved by ITU-T
InterferenceInfo	2	[1:0]	Node's estimation of level of interference that it suffers from neighbouring domains. <ul style="list-style-type: none"> 00₂ – No interference 01₂ – Low interference 10₂ – High interference Other values are reserved by ITU-T
Reserved		[7:2]	Reserved by ITU-T (Note 1)
Reserved	3 and 4	[15:0]	Reserved by ITU-T (Note 1)
NumDomRecs (Note 2, Note 5)	5	[7:0]	Number of records (n) on neighbouring domains represented as an unsigned integer in the range from 0 to 255
NeighbDom_Size[0] (Note 5)	6 and 7	[15:0]	Size of the first neighbouring domain information field in bytes represented as an unsigned integer
NeighbDom_Rec[0] (Note 5)	Variable	See Table 8-105.16	First reported neighbouring domain information field, with a format as defined in Table 8-105.16
...
NeighbDom_Size[n-1] (Note 5)	Variable	[15:0]	Size of the nth neighbouring domain information field in bytes represented as an unsigned integer
NeighbDom_Rec[n-1] (Note 5)	Variable	See Table 8-105.16	nth reported neighbouring domain information field, with a format as defined in Table 8-105.16
NumRemovedDomains (Note 4)	Variable	[7:0]	Number of domains that the reporting node has stopped to detect (r)
RemovedDomain_NDID [0] (Note 4)	Variable	[23:0]	NDID of the first Domain that does not interfere anymore

**Table 8-105.15 – Format of the MMPL of the
NDIM_InterferenceReport.ind message**

Field	Octet	Bits	Description
...
RemovedDomain_NDID [r-1] (Note 4)	Variable	[23:0]	NDID of the rth Domain that does not interfere anymore
LinkPowerReportType	Variable	[1:0]	Type of link power report <ul style="list-style-type: none"> – 00₂: No report is included – 01₂: Report by bands: The received power level is divided in bands of 512 subcarriers. – 10₂: The value indicated in the report represents the received power level for the full band Other values are for further study
NumReportedLinkPower Bands (Note 6)	Variable	[4:2]	Number of reported bands minus one included in the link power bands report (p)
Reserved	Variable	[7:5]	Reserved by ITU-T (Note 1)
LinkPowerLevel[0] (Note 6)	Variable	[7:0]	Received power level in the worst link for this reporting node in the first reported band (lowest band), excluding masked sub-carriers, in dBm, represented as 0.5 dBm per unit with FF ₁₆ representing 25 dBm, FE ₁₆ representing 24.5 dBm, and so on (Note 3)
LinkPowerLevel[1] (Note 7)	Variable	[7:0]	Received power level in the worst link for this reporting node in the 2nd reported band, excluding masked sub-carriers, in dBm, represented as 0.5 dBm per unit with FF ₁₆ representing 25 dBm, FE ₁₆ representing 24.5 dBm, and so on (Note 3)
...	

**Table 8-105.15 – Format of the MMPL of the
NDIM_InterferenceReport.ind message**

Field	Octet	Bits	Description
LinkPowerLevel[6] (Note 7)	Variable	[7:0]	Received power level in the worst link for this reporting node in the 7th reported band (highest band), excluding masked sub-carriers, in dBm, represented as 0.5 dBm per unit with FF ₁₆ representing 25 dBm, FE ₁₆ representing 24.5 dBm, and so on (Note 3)
<p>NOTE 1 – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.</p> <p>NOTE 2 – The value of 0 indicates that no information on neighbouring domains is available. The value of 255 indicates that no record on neighbouring domains is attached (while information on neighbouring domains is available).</p> <p>NOTE 3 – The criteria for selecting the worst link (less power received for a transmission within a domain) power level is vendor discretionary.</p> <p>NOTE 4 – This field only exists if Type is 01₁₆.</p> <p>NOTE 5 – If Type = 01₁₆ this field represents additional information to be taken into account by the DM.</p> <p>NOTE 6 – This field only exists if LinkPowerReportType = 01₂ or 10₂.</p> <p>NOTE 7 – This field only exist if LinkPowerReportType = 10₂.</p> <p>NOTE 8 – The update refers only to the presence or not of a record for a given neighbouring domain. The type of update within the record is specified in the field NeighbDomRecType in Table 8-105.16.</p>			

**Table 8-105.16 – Format of a NeighbDomRec field of the
NDIM_InterferenceReport.ind message**

Field	Octet	Bits	Description
NeighbDomRecType	0	[7:0]	<p>Shall be set to:</p> <ul style="list-style-type: none"> – 00₁₆ if the record includes full information for this node – 01₁₆ if the record includes an update on the information relative to the neighbouring node's characteristics (Note 4) <p>Other values are for further study</p>
NeighbNode_NDID	1 to 3	[23:0]	NDID of the neighbouring domain reported in this record
NeighbNode_CLID	4 to 9	[47:0]	ClusterID of the neighbouring domain reported in this record
NeighbNode_DM_REGID	10 to 15	[47:0]	DM REGID of the neighbouring domain reported in this record
NeighbNode_DOD	16	[3:0]	DOD of the neighbouring domain reported in this record
Reserved		[7:4]	Reserved by ITU-T (Note 1)

**Table 8-105.16 – Format of a NeighbDomRec field of the
NDIM_InterferenceReport.ind message**

Field	Octet	Bits	Description
NumNodeRecs (Note 2)	17	[7:0]	Number of neighbouring domain's nodes (m) reported in this record for this reported neighbouring domain represented as an unsigned integer in the range from 0 to 255
NeighbNode_Size[0]	18 and 19	[15:0]	Size of the first neighbouring domain's node information field in bytes represented as an unsigned integer
NeighbNode_Rec[0]	Variable	See Table 8-105.17	First neighbouring domain's node information field, with a format as defined in Table 8-105.17
...
NeighbNode_Size[m-1]	Variable	[15:0]	Size of the m-th neighbouring domain's node information field in bytes represented as an unsigned integer
NeighbNode_Rec[m-1]	Variable	See Table 8-105.17	Neighbouring domain's node information field, with a format as defined in Table 8-105.17
NumRemovedNeighb Nodes (Note 3)	Variable	[7:0]	Number of nodes that the reporting node has stopped to detect (s)
RemovedNeighbNodes[0] (Note 3)	Variable	[7:0]	DEVICE_ID of the first node that is not detected anymore
...
RemovedNeighbNodes [s-1] (Note 3)	Variable	[7:0]	DEVICE_ID of the s-th node that is not detected anymore
NeighbNode_NDomainName	Variable	[255:0]	Domain Name of the neighbouring domain reported in this record
NeighbNode_Defined_Seed	Variable	[31:0]	Value of the initialization seed of the neighbouring domain reported in this record

NOTE 1 – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.

NOTE 2 – If NeighbDomRecType = 01₁₆ this field represents additional information to be taken into account by the DM.

NOTE 3 – This field only exists if NeighbDomRecType is 01₁₆.

NOTE 4 – If the information relative to a neighbouring domain's node record is indicated as updated, the full information for this node record shall be reported, replacing (if it existed) the previous record about this node.

**Table 8-105.17 – Format of a NeighbNode_Rec field of the
NDIM_InterferenceReport.ind message**

Field	Octet	Bits	Description
NeighbNode_DeviceID	0	[7:0]	DeviceID of the interfering node this record is reporting
NeighbNode_MAC	1 to 6	[47:0]	MAC Address of the interfering node this record is reporting
NeighbNode_InterfEstim	7	[1:0]	Node's estimation of level of interference that it suffers from this neighbouring node. 00 ₂ – No interference 01 ₂ – Low interference 10 ₂ – High interference
Reserved		[7:2]	Reserved by ITU-T (Note 1)
InterfLevelReportType	8	[1:0]	Type of interference level report – 00 ₂ : No report is included – 01 ₂ : Report by bands: The received power level is divided in bands of 512 subcarriers. – 10 ₂ : The value indicated in the report represents the received power level for the full band Other values are for further study
NumReportedBands (Note 2)		[4:2]	Number of reported interference level bands minus one included in this report (q)
Reserved		[7:5]	Reserved by ITU-T (Note 1)
InterfLevel[0] (Note 2)	9	[7:0]	Average Power level received as interference from this node in the first reported band (lowest band), excluding masked sub-carriers, in dBm, represented as 0.5 dBm per unit unit with FF ₁₆ representing 25 dBm, FE ₁₆ representing 24.5 dBm, and so on
InterfLevel[1] (Note 2)	10	[7:0]	Average Power level received as interference from this node in the second reported band, excluding masked sub-carriers, in dBm, represented as 0.5 dBm per unit unit with FF ₁₆ representing 25 dBm, FE ₁₆ representing 24.5 dBm, and so on
...
InterfLevel[6] (Note 3)	Variable	[7:0]	Average Power level received as interference from this node in the 7th reported band (highest band), excluding masked sub-carriers, in dBm, represented as 0.5 dBm per unit unit with FF ₁₆ representing 25 dBm, FE ₁₆ representing 24.5 dBm, and so on
<p>NOTE 1 – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.</p> <p>NOTE 2 – This field only exist if InterfLevelReportType = 01₂ or 10₂.</p> <p>NOTE 3 – This field only exist if InterfLevelReportType = 10₂.</p>			

8.14.10 Inter-domain protocol messages

8.14.10.1 IDM_ClusterAlignment.req format

IDM_ClusterAlignment.req is a management message that is sent by a DM that has just completed MAC cycle alignment to a new cluster, to all the domains in its previous cluster that its domain has a direct link with, to inform them of the necessity of performing a MAC cycle alignment to the new cluster. This message shall include a list of NDIDs of the domains to whom the DM is going to send this message.

Each DM that receives this message sends an IDM_ClusterAlignment.req message to all the domains in its cluster that its domain has direct link with, and that are not included in the NDID list in the received message. Each DM that sends this message adds to the NDID list the NDIDs of the domains whither it will send this message.

The format of the MMPL of the IDM_ClusterAlignment.req message shall be as shown in Table 8-105.18.

**Table 8-105.18 – Format of the MMPL of the
IDM_ClusterAlignment.req message**

Field	Octet	Bits	Description
IDM_DEVICE_ID	0	[7:0]	The 8-bit DEVICE_ID of the node sending this message
IDM_DOD	1	[7:0]	The DOD of the domain to which the node sending this message belongs
IDM_CLID	2 to 7	[47:0]	The Cluster ID of the sending node
IDM_DM_REGID	8 to 13	[47:0]	The REGID of the DM corresponding to the domain to which the node sending the message belongs
IDM_NDID	14 to 16	[23:0]	The NDID of the domain to which the node sending the message belongs
OFFSET_DIR	17	[0]	This field contains the direction of the offset to be applied to the CSP of the cluster of the node receiving the message. 0 if the CSP has to be delayed, 1 if the CSP has to be moved forward
OFFSET		[4:1]	This field contains the offset in increments of 60 degrees, where the value 0000 ₂ represents 0 degrees (see Figure 8-57.2), to be applied to the CSP of the cluster of the node receiving the message. Valid values are from 0 to 11
Reserved		[7:5]	Reserved by ITU-T (Note 1)
Num_Domains	18	[7:0]	Number (n) of domains to whom this message will be sent
NDID_LIST[0]	19 to 21	[23:0]	Contains the NDID of the first domain in the list of domains to whom this message will be sent
...	

**Table 8-105.18 – Format of the MMPL of the
IDM_ClusterAlignment.req message**

Field	Octet	Bits	Description
NDID_LIST[n-1]	Variable	[23:0]	Contains the NDID of the last domain in the list of domains to whom this message will be sent
NDID_CYCSTART_DELTA_DIRECTION	Variable	[0]	This field contains the direction of the offset to be applied to the NDID_CYCSTART_DELTA. 0 if the NDID_CYCSTART_DELTA has to be delayed, 1 if the NDID_CYCSTART_DELTA has to be moved forward
NDID_CYCSTART_DELTA		[31:1]	The delta time to be applied to the start of the MAC cycle of the domain receiving this message.
NOTE 1 – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

8.14.10.2 IDM_ClusterAlignment.cnf message format

IDM_ClusterAlignment.cnf is a management message that is sent to confirm receiving the IDM_ClusterAlignment.req message.

The format of the MMPL of the IDM_ClusterAlignment.cnf message shall be as shown in Table 8-105.19.

Table 8-105.19 – Format of the MMPL of the IDM_ClusterAlignment.cnf message

Field	Octet	Bits	Description
IDM_DEVICE_ID	0	[7:0]	The 8-bit DEVICE_ID of the node sending this message
IDM_DOD	1	[7:0]	The DOD of the domain to which the node sending this message belongs
IDM_CLID	2 to 7	[47:0]	The Cluster ID of the sending node
IDM_DM_REGID	8 to 13	[47:0]	The REGID of the DM corresponding to the domain to which the node sending the message belongs
IDM_NDID	14 to 16	[23:0]	The NDID of the domain to which the node sending the message belongs
StatusCode	17	[7:0]	Status of the alignment request: 0 – The domain will align to the new cluster 1 – The domain will not align to the new cluster 2-255: Reserved by ITU-T (Note)
NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

8.14.10.3 IDM interfering nodes information

IDM_InterfNodesInfo.ind is a management message that is sent by a node in its IDCC to indicate the nodes suffering interference in its domain to the neighbouring domains that contain the nodes causing interference.

The format of the MMPL of the IDM_InterfNodesInfo.ind message shall be as shown in Table 8-105.20.

Table 8-105.20 – Format of the MMPL of the IDM_InterfNodesInfo.ind message

Field	Octet	Bits	Description
IDM_DEVICE_ID	0	[7:0]	The 8-bit DEVICE_ID of the node sending this message
IDM_DOD	1	[7:0]	The DOD of the domain to which the node sending this message belongs
IDM_CLID	2 to 7	[47:0]	The Cluster ID of the sending node
IDM_DM_REGID	8 to 13	[47:0]	The REGID of the DM corresponding to the domain to which the node sending the message belongs
IDM_NDID	14 to 16	[23:0]	The NDID of the domain to which the node sending the message belongs.
NUM_INTERF_TABLES	17	[7:0]	Number of interference tables (<i>n</i>) in the domain associated with nodes suffering interference from the nodes in the destination domain for this message, and therefore requiring coordination
NODE_INTF_TABLE_LEN ₁	18 and 19	[15:0]	Length of the first interference table in octets
NODE_INTF_TABLE ₁	Variable	[7:0]	First interference table as per Table 8-105.21
NODE_INTF_TABLE_LEN ₂	Variable	[15:0]	Length of the second interference table in octets
NODE_INTF_TABLE ₂	Variable	[7:0]	Second interference table as per Table 8-105.21
...
NODE_INTF_TABLE_LEN _{<i>n</i>}	Variable	[15:0]	Length of the <i>n</i> th interference table in octets
NODE_INTF_TABLE _{<i>n</i>}	Variable	[7:0]	<i>n</i> th interference table as per Table 8-105.21

Table 8-105.21 – Format of interference table for a group of nodes

Field	Octet	Bits	Description
NUM_NODES	0	[7:0]	This field contains the number (<i>m</i>) of nodes that are suffering interference from the same set of interfering nodes in the same set of neighbouring domains (i.e., they have the same interference table)
NODE_ID ₁	1	[7:0]	This field contains the DEVICE_ID of the first node for which this interference table applies
...
NODE_ID _{<i>m</i>}	<i>m</i>	[7:0]	This field contains the DEVICE_ID of the last node for which this interference table applies
INTERFERENCE_TABLE	variable	[7:0]	Table that contains the list of domains and their nodes that cause interference to the nodes listed above. The format of this table is as specified in Table 8-105.22
INTERFERENCE_TABLE_VALIDITY	variable	[7:0]	<p>Bitmap representing (Note 2)</p> <p>Bit 0, if set to one, indicates that all nodes of this domain are considered as suffering interference from the same set of interfering nodes in the same set of neighbouring domains (i.e., they have the same interference table).</p> <p>Bit 1, if set to one, indicates that the domain is treating all the nodes of the interfering domain(s) corresponding to this interference table as interfering. In this case the list of nodes provided is an estimation based on currently available information</p> <p>Bits 2-7 are reserved by ITU-T (Note 1)</p>
<p>NOTE 1 – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.</p> <p>NOTE 2 – Bits 0 and 1 shall both be set to one or both be set to zero.</p>			

Table 8-105.22 – Format of the interference table

Field	Octet	Bits	Description
NUM_INTF_DOMAINS	0	[7:0]	Number of interfering domains (<i>k</i>)
INTF_DOMAIN_TABLE ₁	Variable	[7:0]	The interference table entry corresponding to interfering nodes from the first interfering domain. The format of this field is as specified in Table 8-105.23

Table 8-105.22 – Format of the interference table

Field	Octet	Bits	Description
INTF_DOMAIN_TABLE ₂	Variable	[7:0]	The interference table entry corresponding to interfering nodes from the second interfering domain. The format of this field is as specified in Table 8-105.23
...
INTF_DOMAIN_TABLE _k	Variable	[7:0]	The interference table entry corresponding to interfering nodes from the last interfering domain. The format of this field is as specified in Table 8-105.23

Table 8-105.23 – Format of the interference table entry corresponding to interfering nodes from a specific domain

Field	Octet	Bits	Description
INTF_DOMAIN_NDID	0 to 2	[23:0]	The NDID of the interfering domain <i>D</i>
INTF_DOMAIN_DOD	3	[7:0]	The DOD of the interfering domain <i>D</i>
NUM_INTF_NODES	4	[7:0]	Number of interfering nodes (<i>n</i>) in the interfering domain <i>D</i>
INTF_NODE_ID ₁	5	[7:0]	DEVICE_ID of first interfering node in <i>D</i>
INTF_NODE_ID ₂	6	[7:0]	DEVICE_ID of second interfering node in <i>D</i>
...
INTF_NODE_ID _n	4+ <i>n</i>	[7:0]	DEVICE_ID of last interfering node in <i>D</i>

8.14.10.4 IDM interference information response

IDM_InterfNodesInfo.rsp is a management message that shall be sent to confirm that an IDM_InterfNodesInfo.ind has been received.

The format of the MMPL of the IDM_InterfNodesInfo.rsp message shall be as shown in Table 8-105.24.

Table 8-105.24 – Format of the MMPL of the IDM_InterfNodesInfo.rsp message

Field	Octet	Bits	Description
IDM_DEVICE_ID	0	[7:0]	The 8-bit DEVICE_ID of the node sending this message
IDM_DOD	1	[7:0]	The DOD of the domain to which the node sending this message belongs

Table 8-105.24 – Format of the MMPL of the IDM_InterfNodesInfo.rsp message

Field	Octet	Bits	Description
IDM_CLID	2 to 7	[47:0]	The Cluster ID of the sending node
IDM_DM_REGID	8 to 13	[47:0]	The REGID of the DM corresponding to the domain to which the node sending the message belongs
IDM_NDID	14 to 16	[23:0]	The NDID of the domain to which the node sending the message belongs
StatusCode	17	[7:0]	Status of the request: <ul style="list-style-type: none"> • 00₁₆ = Success • 01₁₆ – FF₁₆ = Reserved

8.14.10.5 IDM coordinating domains information

IDM_CoordDomainsInfo.ind is a management message that is sent by a node in its IDCC to indicate the nodes in its domain that are coordinating with other domain masters and also contains the DODs of the coordinating domains for each node.

The format of the MMPL of the IDM_CoordDomainsInfo.ind message shall be as shown in Table 8-105.25.

Table 8-105.25 – Format of the MMPL of the IDM_CoordDomainsInfo.ind message

Field	Octet	Bits	Description
IDM_DEVICE_ID	0	[7:0]	The 8-bit DEVICE_ID of the node sending this message
IDM_DOD	1	[7:0]	The DOD of the domain the node sending this message belongs to
IDM_CLID	2 to 7	[47:0]	The Cluster ID of the sending node
IDM_DM_REGID	8 to 13	[47:0]	The REGID of the DM corresponding to the domain to which the node sending the message belongs
IDM_NDID	14 to 16	[23:0]	The NDID of the domain to which the node sending the message belongs
NUM_COORD_TABLES	17	[7:0]	Number of coordination tables (<i>n</i>) included in the message
COORD_TABLE ₁	Variable	[7:0]	First coordination table as per Table 8-105.26
COORD_TABLE ₂	Variable	[7:0]	Second coordination table as per Table 8-105.26
...

Table 8-105.25 – Format of the MMPL of the IDM_CoordDomainsInfo.ind message

Field	Octet	Bits	Description
COORD_TABLE _{<i>n</i>}	Variable	[7:0]	Last coordination table as per Table 8-105.26
COORD_TABLE_VALIDITY	Variable	[7:0]	Bitmap representing Bit 0, if set to one, indicates that the coordination table applies to all nodes of the domain. In this case NUM_COORD_TABLES shall be set to 1. Bits 1-7 are reserved by ITU-T (Note 1)
NOTE 1– Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

Table 8-105.26 – Format of the coordination table for a group of nodes

Field	Octet	Bits	Description
NUM_NODES_COORD	0	[7:0]	Number of nodes (<i>k</i>) for which this coordination table applies
NODE_ID ₁	1	[7:0]	DEVICE_ID of the first node for which this coordination table applies
NODE_ID ₂	2	[7:0]	DEVICE_ID of the second node for which this coordination table applies
...
NODE_ID _{<i>k</i>}	<i>k</i>	[7:0]	DEVICE_ID of the last node for which this coordination table applies
NUM_COORD_DOMAINS	<i>k</i> + 1	[7:0]	Number of unique DODs (<i>n</i>) this group of nodes coordinates with
DOD ₁	<i>k</i> + 2	[3:0]	First unique DOD – DOD ₁
DOD ₂		[7:4]	Second unique DOD – DOD ₂
...
DOD _{<i>n</i>}	<i>k</i> + 1 + ROUNDUP(<i>n</i> /2)	[3:0]/[7:4]	Last unique DOD – DOD _{<i>n</i>}
Reserved	<i>k</i> + 1 + ROUNDUP(<i>n</i> /2)	[7:4]	Padding set to 0, if required to end on a byte boundary

8.14.10.6 IDM coordination information response

IDM_CoordDomainsInfo.rsp is a management message that shall be sent to confirm that an IDM_CoordDomainsInfo.ind has been received.

The format of the MMPL of the IDM_CoordDomainsInfo.rsp message shall be as shown in Table 8-105.27.

Table 8-105.27 – Format of the MMPL of the IDM_CoordDomainsInfo.rsp message

Field	Octet	Bits	Description
IDM_DEVICE_ID	0	[7:0]	The 8-bit DEVICE_ID of the node sending this message
IDM_DOD	1	[7:0]	The DOD of the domain to which the node sending this message belongs
IDM_CLID	2 to 7	[47:0]	The Cluster ID of the sending node
IDM_DM_REGID	8 to 13	[47:0]	The REGID of the DM corresponding to the domain to which the node sending the message belongs
IDM_NDID	14 to 16	[23:0]	The NDID of the domain to which the node sending the message belongs
StatusCode	17	[7:0]	Status of the request: <ul style="list-style-type: none"> • 00₁₆ = Success • 01₁₆ – FF₁₆ = Reserved

8.14.10.7 IDM share unallocated slot request

IDM_ShareUnallocSlot.req is a management message that is sent by a node in its IDCC to request sharing of unallocated slots with coordinating domain masters.

The format of the MMPL of the IDM_ShareUnallocSlot.req message shall be as shown in Table 8-105.28.

Table 8-105.28 – Format of the MMPL of the IDM_ShareUnallocSlot.req message

Field	Octet	Bits	Description
IDM_DEVICE_ID	0	[7:0]	The 8-bit DEVICE_ID of the node sending this message
IDM_DOD	1	[7:0]	The DOD of the domain to which the node sending this message belongs
IDM_CLID	2 to 7	[47:0]	The Cluster ID of the sending node
IDM_DM_REGID	8 to 13	[47:0]	The REGID of the DM corresponding to the domain the node sending the message belongs to

Table 8-105.28 – Format of the MMPL of the IDM_ShareUnallocSlot.req message

Field	Octet	Bits	Description
IDM_NDID	14 to 16	[23:0]	The NDID of the domain to which the node sending the message belongs
COORD_TABLE_ID	17	[7:0]	The coordination table associated with this request
TRANSACTION_ID	18	[7:0]	The transaction ID associated with this request
TRANSACTION_STATUS	19 and 20	[15:0]	This field indicates the status of eight previous transactions. The LSB corresponds to the oldest and MSB corresponds to the newest transaction prior to the transaction proposed in this message. The status of each transaction is indicated by a two bit field as follows: 00 ₂ – no status available 01 ₂ – transaction in progress 10 ₂ – transaction succeeded 11 ₂ – transaction failed
NUM_SLOTS_REQ	21	[3:0]	Number of MAC cycle slots requested for sharing minus 1. Valid values are 0 to 14
Reserved		[7:4]	Reserved by ITU-T (Note)
SLOT_ID0	22	[3:0]	Identification of first unallocated slot requested for sharing. Valid values are 1 to 15
SLOT_ID1		[7:4]	Identification of second unallocated slot requested for sharing. Valid values are 1 to 15
SLOT_IDn	21+ ROUNDUP(n/2)	..	Identification of nth unallocated slot requested for sharing. Valid values are 1 to 15
NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

8.14.10.8 IDM share unallocated Slot confirmation

IDM_ShareUnallocSlot.cnf is a management message that shall be sent as a confirmation of IDM_ShareUnallocSlot.req.

The format of the MMPL of the IDM_ShareUnallocSlot.cnf message shall be as shown in Table 8-105.29.

Table 8-105.29 – Format of the MMPL of the IDM_ShareUnallocSlot.cnf message

Field	Octet	Bits	Description
IDM_DEVICE_ID	0	[7:0]	The 8-bit DEVICE_ID of the node sending this message
IDM_DOD	1	[7:0]	The DOD of the domain to which the node sending this message belongs
IDM_CLID	2 to 7	[47:0]	The Cluster ID of the sending node
IDM_DM_REGID	8 to 13	[47:0]	The REGID of the DM corresponding to the domain to which the node sending the message belongs
IDM_NDID	14 to 16	[23:0]	The NDID of the domain to which the node sending the message belongs
COORD_TABLE_ID	17	[7:0]	The coordination table associated with this confirmation
TRANSACTION_ID	18	[7:0]	The transaction ID associated with this confirmation
StatusCode	19	[7:0]	Status of the request: <ul style="list-style-type: none"> • 00₁₆ = Success. • 01₁₆ = Failure • 02₁₆ = Failure and request to refrain from sending additional requests for RefrainMultiple allocation periods • 03₁₆ – FF₁₆ = Reserved
RefrainMultiple	20	[7:0]	Number of allocation periods for which the requesting domain master is asked to refrain from sending additional requests starting from the next allocation period (Note)
NOTE – This field exists only if the StatusCode field is set to 02 ₁₆ .			

8.14.10.9 IDM share unallocated Slot indication

IDM_ShareUnallocSlot.ind is a management message that shall be sent as an indication of the status of the share unallocated slots request.

The format of the MMPL of the IDM_ShareUnallocSlot.ind message shall be as shown in Table 8-105.30.

Table 8-105.30 – Format of the MMPL of the IDM_ShareUnallocSlot.ind message

Field	Octet	Bits	Description
IDM_DEVICE_ID	0	[7:0]	The 8-bit DEVICE_ID of the node sending this message
IDM_DOD	1	[7:0]	The DOD of the domain to which the node sending this message belongs
IDM_CLID	2 to 7	[47:0]	The Cluster ID of the sending node
IDM_DM_REGID	8 to 13	[47:0]	The REGID of the DM corresponding to the domain to which the node sending the message belongs
IDM_NDID	14 to 16	[23:0]	The NDID of the domain to which the node sending the message belongs
COORD_TABLE_ID	17	[7:0]	The coordination table associated with this indication
TRANSACTION_ID	18	[7:0]	The transaction ID associated with this indication
StatusCode	19	[7:0]	Status of the request: <ul style="list-style-type: none"> • 00₁₆ = Success. • 01₁₆ = Failure • 02₁₆ – FF₁₆ = Reserved.

8.14.10.10 IDM request unallocated slot request

IDM_RequestUnallocSlot.req is a management message that is sent by a node in its IDCC to request assignment of unallocated slots with coordinating domain masters. The request shall include an indication of the priority of the request for the requested slot, which is based on DM's vendor discretionary criteria.

The format of the MMPL of the IDM_RequestUnallocSlot.req message shall be as shown in Table 8-105.31.

Table 8-105.31 – Format of the MMPL of the IDM_RequestUnallocSlot.req message

Field	Octet	Bits	Description
IDM_DEVICE_ID	0	[7:0]	The 8-bit DEVICE_ID of the node sending this message
IDM_DOD	1	[7:0]	The DOD of the domain to which the node sending this message belongs
IDM_CLID	2 to 7	[47:0]	The Cluster ID of the sending node
IDM_DM_REGID	8 to 13	[47:0]	The REGID of the DM corresponding to the domain to which the node sending the message belongs

Table 8-105.31 – Format of the MMPL of the IDM_RequestUnallocSlot.req message

Field	Octet	Bits	Description
IDM_NDID	14 to 16	[23:0]	The NDID of the domain to which the node sending the message belongs
COORD_TABLE_ID	17	[7:0]	The coordination table associated with this request
TRANSACTION_ID	18	[7:0]	The transaction ID associated with this request
TRANSACTION_STATUS	19 and 20	[15:0]	This field indicates the status of eight previous transactions. The LSB corresponds to the oldest and MSB corresponds to the newest transaction prior to the transaction proposed in this message. The status of each transaction is indicated by a two bit field as follows: 00 ₂ – no status available 01 ₂ – transaction in progress 10 ₂ – transaction succeeded 11 ₂ – transaction failed
NUM_SLOTS_REQ	21	[3:0]	Number of MAC cycle slots requested for assignment minus 1 (<i>n</i>). Valid values are 0 to 14
Reserved		[7:4]	Reserved by ITU-T (Note)
SLOT_ID1	22	[3:0]	Identification of first unallocated slot requested for assignment. Valid values are 1 to 15
PRIORITY_SLOT_ID1		[7:4]	Priority of the first unallocated slot requested for assignment. Valid values are 0 to 7
SLOT_ID2	23	[3:0]	Identification of second unallocated slot requested for assignment. Valid values are 1 to 15
PRIORITY_SLOT_ID2		[7:4]	Priority of the second unallocated slot requested for assignment. Valid values are 0 to 7
...

Table 8-105.31 – Format of the MMPL of the IDM_RequestUnallocSlot.req message

Field	Octet	Bits	Description
SLOT_ID n	22 + n	[3:0]	Identification of the n th unallocated slot requested for assignment. Valid values are 1 to 15
PRIORITY_SLOT_ID n		[7:4]	Priority of the n th unallocated slot requested for assignment. Valid values are 0 to 7
NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

8.14.10.11 IDM request unallocated slot confirmation

IDM_RequestUnallocSlot.cnf is a management message that shall be sent as a confirmation of IDM_RequestUnallocSlot.req.

The format of the MMPL of the IDM_RequestUnallocSlot.cnf message shall be as shown in Table 8-105.32.

Table 8-105.32 – Format of the MMPL of the IDM_RequestUnallocSlot.cnf message

Field	Octet	Bits	Description
IDM_DEVICE_ID	0	[7:0]	The 8-bit DEVICE_ID of the node sending this message
IDM_DOD	1	[7:0]	The DOD of the domain to which the node sending this message belongs
IDM_CLID	2 to 7	[47:0]	The Cluster ID of the sending node
IDM_DM_REGID	8 to 13	[47:0]	The REGID of the DM corresponding to the domain to which the node sending the message belongs
IDM_NDID	14 to 16	[23:0]	The NDID of the domain to which the node sending the message belongs
COORD_TABLE_ID	17	[7:0]	The coordination table associated with this confirmation
TRANSACTION_ID	18	[7:0]	The transaction ID associated with this confirmation
StatusCode	19	[7:0]	Status of the request: <ul style="list-style-type: none"> • 00₁₆ = Success. • 01₁₆ = Failure • 02₁₆ = Failure and request to refrain from sending additional requests for RefrainMultiple allocation periods 03 ₁₆ – FF ₁₆ = Reserved

Table 8-105.32 – Format of the MMPL of the IDM_RequestUnallocSlot.cnf message

Field	Octet	Bits	Description
RefrainMultiple	20	[7:0]	Number of allocation periods for which the requesting domain master is asked to refrain from sending additional requests starting from the next allocation period (Note 1)
NOTE 1 – This field exists only if the StatusCode field is set to 02 ₁₆ .			

8.14.10.12 IDM request unallocated slot indication

IDM_RequestUnallocSlot.ind is a management message that shall be sent as an indication of the status of the unallocated slots request.

The format of the MMPL of the IDM_RequestUnallocSlot.ind message shall be as shown in Table 8-105.33.

Table 8-105.33 – Format of the MMPL of the IDM_RequestUnallocSlot.ind message

Field	Octet	Bits	Description
IDM_DEVICE_ID	0	[7:0]	The 8-bit DEVICE_ID of the node sending this message
IDM_DOD	1	[7:0]	The DOD of the domain to which the node sending this message belongs
IDM_CLID	2 to 7	[47:0]	The Cluster ID of the sending node
IDM_DM_REGID	8 to 13	[47:0]	The REGID of the DM corresponding to the domain to which the node sending the message belongs
IDM_NDID	14 to 16	[23:0]	The NDID of the domain to which the node sending the message belongs
COORD_TABLE_ID	17	[7:0]	The coordination table associated with this indication
TRANSACTION_ID	18	[7:0]	The transaction ID associated with this indication
StatusCode	19	[7:0]	Status of the request: <ul style="list-style-type: none"> • 00₁₆ = Success. • 01₁₆ = Failure 02₁₆ – FF₁₆ = Reserved

8.14.10.13 IDM swap allocated slot request

IDM_SwapAllocSlot.req is a management message that is sent by a node in its IDCC to request swapping of allocated slots with coordinating domain masters.

The format of the MMPL of the IDM_SwapAllocSlot.req message shall be as shown in Table 8-105.34.

Table 8-105.34 – Format of the MMPL of the IDM_SwapAllocSlot.req message

Field	Octet	Bits	Description
IDM_DEVICE_ID	0	[7:0]	The 8-bit DEVICE_ID of the node sending this message
IDM_DOD	1	[7:0]	The DOD of the domain to which the node sending this message belongs
IDM_CLID	2 to 7	[47:0]	The Cluster ID of the sending node
IDM_DM_REGID	8 to 13	[47:0]	The REGID of the DM corresponding to the domain to which the node sending the message belongs
IDM_NDID	14 to 16	[23:0]	The NDID of the domain to which the node sending the message belongs
COORD_TABLE_ID	17	[7:0]	The coordination table associated with this request
TRANSACTION_ID	18	[7:0]	The transaction ID associated with this request
TRANSACTION_STATUS	19 and 20	[15:0]	This field indicates the status of eight previous transactions. The LSB corresponds to the oldest and MSB corresponds to the newest transaction prior to the transaction proposed in this message. The status of each transaction is indicated by a two bit field as follows: 00 ₂ – no status available 01 ₂ – transaction in progress 10 ₂ – transaction succeeded 11 ₂ – transaction failed
NUM_SLOTS_REQ	21	[3:0]	Number of MAC cycle slots requested for assignment minus 1 (<i>n</i>). Valid values are 0 to 14 (Note 1)
Reserved		[7:4]	Reserved by ITU-T (Note 2)
SLOT_ID1	22	[3:0]	Identification of first slot requested for assignment. Valid values are 1 to 15
SLOT_ID2		[7:4]	Identification of second slot requested for assignment. Valid values are 1 to 15
SLOT_ID _{<i>n</i>}	Identification of <i>n</i> th slot requested for assignment. Valid values are 1 to 15
NUM_SLOTS_OFFERED	Variable	[3:0]	Number of MAC cycle slots offered for assignment minus 1 (<i>k</i>). Valid values are 0 to 15
Reserved		[7:4]	Reserved by ITU-T (Note 2)
SLOT_ID1	Variable	[3:0]	Identification of first slot offered for assignment. Valid values are 1 to 15

Table 8-105.34 – Format of the MMPL of the IDM_SwapAllocSlot.req message

Field	Octet	Bits	Description
SLOT_ID2		[7:4]	Identification of second slot offered for assignment. Valid values are 1 to 15
SLOT_ID k	Variable	.. <u>Variable</u>	Identification of k -th slot offered for assignment. Valid values are 1 to 15
NOTE 1 – The number of requested slots need not be the same as the number of offered slots. NOTE 2 – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

8.14.10.14IDM swap allocated slot confirmation

IDM_SwapAllocSlot.cnf is a management message that shall be sent to confirm that an IDM_SwapAllocSlot.req has been received.

The format of the MMPL of the IDM_SwapAllocSlot.cnf message shall be as shown in Table 8-105.35.

Table 8-105.35 – Format of the MMPL of the IDM_SwapAllocSlot.cnf message

Field	Octet	Bits	Description
IDM_DEVICE_ID	0	[7:0]	The 8-bit DEVICE_ID of the node sending this message
IDM_DOD	1	[7:0]	The DOD of the domain to which the node sending this message belongs
IDM_CLID	2 to 7	[47:0]	The Cluster ID of the sending node
IDM_DM_REGID	8 to 13	[47:0]	The REGID of the DM corresponding to the domain to which the node sending the message belongs
IDM_NDID	14 to 16	[23:0]	The NDID of the domain to which the node sending the message belongs
COORD_TABLE_ID	17	[7:0]	The coordination table associated with this confirmation
TRANSACTION_ID	18	[7:0]	The transaction ID associated with this confirmation
StatusCode	19	[7:0]	Status of the request: <ul style="list-style-type: none"> • 00₁₆ = Success. • 01₁₆ = Failure • 02₁₆ = Failure and request to refrain from sending additional requests for RefrainMultiple allocation periods 03₁₆ – FF₁₆ = Reserved
RefrainMultiple	20	[7:0]	Number of allocation periods for which the requesting domain master is asked to refrain from sending additional requests starting from the next allocation period (Note)
NOTE – This field exists only if the StatusCode field is set to 02 ₁₆ .			

8.14.10.15 IDM swap allocated slot indication

IDM_SwapAllocSlot.ind is a management message that shall be sent as an indication of the status of the proposed swap of allocated slots.

The format of the MMPL of the IDM_SwapAllocSlot.ind message shall be as shown in Table 8-105.36.

Table 8-105.36 – Format of the MMPL of the IDM_SwapAllocSlot.ind message

Field	Octet	Bits	Description
IDM_DEVICE_ID	0	[7:0]	The 8-bit DEVICE_ID of the node sending this message
IDM_DOD	1	[7:0]	The DOD of the domain to which the node sending this message belongs
IDM_CLID	2 to 7	[47:0]	The Cluster ID of the sending node
IDM_DM_REGID	8 to 13	[47:0]	The REGID of the DM corresponding to the domain to which the node sending the message belongs
IDM_NDID	14 to 16	[23:0]	The NDID of the domain to which the node sending the message belongs
COORD_TABLE_ID	17	[7:0]	The coordination table associated with this indication
TRANSACTION_ID	18	[7:0]	The transaction ID associated with this indication
StatusCode	19	[7:0]	Status of the request: <ul style="list-style-type: none"> • 00₁₆ = Success. • 01₁₆ = Failure 02₁₆ – FF₁₆ = Reserved

8.14.10.16 IDM coordination preference indication

IDM_CoordPref.ind is a management message that shall be sent as an indication of a DM's preference of coordination method to its neighbouring domains.

The format of the MMPL of the IDM_CoordPref.ind message shall be as shown in Table 8-105.37.

Table 8-105.37 – Format of the MMPL of the IDM_CoordPref.ind message

Field	Octet	Bits	Description
IDM_DEVICE_ID	0	[7:0]	The 8-bit DEVICE_ID of the node sending this message
IDM_DOD	1	[7:0]	The DOD of the domain to which the node sending this message belongs
IDM_CLID	2 to 7	[47:0]	The Cluster ID of the sending node
IDM_DM_REGID	8 to 13	[47:0]	The REGID of the DM corresponding to the domain to which the node sending the message belongs
IDM_NDID	14 to 16	[23:0]	The NDID of the domain to which the node sending the message belongs
CoordinationMethod	17	[7:0]	Coordination method: <ul style="list-style-type: none"> • 00₁₆ = method specified in clause 8.14.7 • FF₁₆ = DM does not coordinate 01₁₆ – FE₁₆ = Reserved

8.14.10.17 IDM domain master change indication

IDM_DmChange.ind is a management message that shall be sent as an indication to the neighbouring domains that the DM of a domain has changed.

The format of the MMPL of the IDM_DmChange.ind message shall be as shown in Table 8-105.38.

Table 8-105.38 – Format of the MMPL of the IDM_DmChange.ind message

Field	Octet	Bits	Description
IDM_DEVICE_ID	0	[7:0]	The 8-bit DEVICE_ID of the node sending this message
IDM_DOD	1	[7:0]	The DOD of the domain to which the node sending this message belongs
IDM_CID	2 to 7	[47:0]	The Cluster ID of the sending node
IDM_DM_REGID	8 to 13	[47:0]	The REGID of the DM corresponding to the domain to which the node sending the message belongs
IDM_NDID	14 to 16	[23:0]	The NDID of the domain to which the node sending the message belongs
IDM_OLD_DM_REGID	17 to 22	[47:0]	The REGID of the earlier DM corresponding to the domain to which the node sending the message belongs

8.14.10.18 IDM DNI change indication

IDM_DniChange.ind is a management message that shall be sent as an indication to the neighbouring domains that the DNI of a domain has changed.

The format of the MMPL of the IDM_DniChange.ind message shall be as shown in Table 8-105.39.

Table 8-105.39 – Format of the MMPL of the IDM_DniChange.ind message

Field	Octet	Bits	Description
IDM_DEVICE_ID	0	[7:0]	The 8-bit DEVICE_ID of the node sending this message
IDM_DOD	1	[7:0]	The DOD of the domain to which the node sending this message belongs
IDM_CID	2 to 7	[47:0]	The Cluster ID of the sending node
IDM_DM_REGID	8 to 13	[47:0]	The REGID of the DM corresponding to the domain to which the node sending the message belongs
IDM_NDID	14 to 16	[23:0]	The NDID of the domain to which the node sending the message belongs
IDM_DNI	17 and 18	[15:0]	The current DNI of the domain to which the node sending the message belongs
IDM_OLD_DNI	19 and 20	[15:0]	The previous DNI of the domain to which the node sending the message belongs

8.15 Coexistence with alien power-line networks

The [ITU-T G.9972] coexistence protocol mitigates interference to ITU-T G.9960/1 nodes from alien networks, thus enabling coexistence with other, non-interoperable, networks (alien networks). [ITU-T G.9972] provides MMPLs for the ITU-T G.9960/1 coexistence related management messages as specified in clause 8.10.1. When mitigation using [ITU-T G.9972] is unnecessary, [ITU-T G.9972] provides a management message that is communicated between ITU-T G.9960/1 nodes of the domain to cease transmission of ITU-T G.9972 signals.

NOTE – Powerline communication devices may suffer interference from and create interference to alien power-line networks when operating over the same frequency range. Therefore, when there is a chance that multiple non-interoperable power line technologies are simultaneously using the same power line cables in the same frequency range, it is strongly recommended that ITU-T G.9960/1 and alien devices use [ITU-T G.9972] to avoid performance degradation.

8.16 PHY multicast binding protocol

The PHY multicast binding protocol enables a transmitter node to transmit the same PHY frames to several nodes that might share a common BAT. The multicast binding protocol enables the creation of a PHY multicast group and the management of the membership of nodes in the PHY multicast group. The PHY multicast transmissions are identified by (SID, MULTICAST_ID) tuples (see clause 8.7.1.2) which identify the transmitter node and the receiver nodes that receive directly the PHY frame from the transmitter node.

8.16.1 Initialization of a PHY multicast group

A transmitting node of a PHY multicast stream may initiate the PHY multicast binding protocol when it needs to transmit the same data to several nodes directly (in the same hop).

When a node initiates the multicast binding protocol, it shall compute the common BATs to be used for the PHY multicast group based on the BATs (see clause 8.11) reported by the receiver nodes in the group. The transmitter shall also consider its own StartSubCarrier & StopSubCarrier and that of the receivers of the PHY multicast group. More specifically, the range of sub-carriers of the BAT sent in the MC_GrpInfoUpdate.ind message shall be within the intersection of the sub-carrier ranges determined by the node's StartSubCarrier & StopSubCarrier of both the transmitter and all receivers in the PHY multicast group (also see clause 8.18.5).

BATs to be used for multicast transmission shall not include values of 5, 7, 9 or 11 bits.

When Mc-ACK is used for a PHY multicast group, the transmitter shall assign receivers to the Mc-ACK/NACK slots. The acknowledgement protocol state machine for PHY multicast transmission shall be initialized as specified in clause 8.9.5.4.

NOTE 1 – The actual method for deciding on the number of PHY multicast groups and the BATs used for each group and the assignment of nodes to the Mc-ACK slots is beyond the scope of this Recommendation.

The transmitter shall then send an MC_GrpInfoUpdate.ind message to all the nodes that will be part of the PHY multicast group including information about the BATs to be used within the PHY multicast group and the receiver nodes that are members of the group. Upon reception of a MC_GrpInfoUpdate.ind message, each receiver that appears as a receiver of a PHY multicast group shall confirm the message by sending an MC_GrpInfoUpdate.cnf message to the transmitter.

In case the MC_GrpInfoUpdate.cnf message is not received from all of the receiving devices within T_{MCST} the transmitter shall retransmit the request until N_{MCST} retries are exhausted.

The transmitter may control whether flow-control is enabled or not for a PHY multicast group by setting the appropriate value of the FlowControlInd field in the MC_GrpInfoUpdate.ind message. The decision as to whether flow control should be enabled or not is beyond the scope of this Recommendation.

NOTE 2 – Flow-control may be disabled if Mc-ACK slots have not been allocated to all members of the PHY multicast group.

When flow-control is not used on a PHY multicast group a transmitter shall advertise the recommended receive buffer size in the MC_GrpInfoUpdate.ind message. The initial recommended receive buffer size (ACK_RX_CONF_WINDOW_SIZE) for a PHY multicast group shall be specified by the transmitter to have a maximum value (set in the MinRxBufSize field in Table 8-107). Upon reception of the MC_GrpInfoUpdate.ind message receivers shall respond by specifying their available receive buffer sizes (ACK_RX_CONF_WINDOW_SIZE) in the MC_GrpInfoUpdate.cnf message. The transmitter shall collect all the receive buffer sizes advertised by all PHY multicast group members and shall adjust the recommended receive buffer size advertised in the MC_GrpInfoUpdate.ind message. The adjusted value of the MinRxBufSize field (see MinRxBufSize in Table 8-107) in the MC_GrpInfoUpdate.ind message shall be set to the minimum of the receive buffer size of all members of the PHY multicast group. Upon reception of the adjusted MC_GrpInfoUpdate.ind message receivers may reduce the size of their receive buffers to the specified value. The new receive buffer size used by the receiver shall be reported in the corresponding MC_GrpInfoUpdate.cnf message.

NOTE 3 – Based on the advertised receive buffer sizes of members of the PHY multicast group the transmitter may decide to reassign PHY multicast group members to different groups.

When flow-control is not used the value of FLCTRL specified in the ACK frames shall be set to the value advertised by the receiver in the last MC_GrpInfoUpdate.cnf message.

When flow control is used the recommended receive buffer size specified in the MC_GrpInfoUpdate.ind message shall be ignored by the receiving nodes that are assigned a Mc-ACK slot. The initial recommended receive buffer size (ACK_RX_CONF_WINDOW_SIZE) for a PHY multicast group shall be specified by the transmitter to have a maximum value (set in the MinRxBufSize field in Table 8-107). The receiving nodes that are not assigned a Mc-ACK slot shall respond by specifying their available buffer sizes (ACK_RX_CONF_WINDOW_SIZE) in the MC_GrpInfoUpdate.cnf message. The transmitter shall collect the receive buffer sizes advertised by all receiving nodes that are not assigned a Mc-ACK slot and shall adjust the recommended receive buffer size advertised in the MC_GrpInfoUpdate.ind message. The adjusted value of the MinRxBufSize field (see MinRxBufSize in Table 8-107) in the MC_GrpInfoUpdate.ind message shall be set to the minimum of the receive buffer size of those receiving nodes of the PHY multicast group. Upon reception of the adjusted MC_GrpInfoUpdate.ind message these receivers may reduce the size of their receive buffers to the specified value. The new receive buffer size used by these receivers shall be reported in the corresponding MC_GrpInfoUpdate.cnf message. The transmitter shall limit the number of LPDUs transmitted within each PHY frame according to the transmit window corresponding to this group, to the minimum of the receive buffer size indicated in the MC_GrpInfoUpdate.cnf message by the receivers that are not assigned an Mc-ACK slot and the values indicated in the FLCTRL field by the receivers that are assigned Mc-ACK slots.

Before the multicast binding is completed for a new PHY multicast group the transmitter may send the multicast stream traffic using the BROADCAST_ID as DID, or by making unicast transmissions to the multicast receivers.

During initialization of a PHY multicast group or when a change in the membership of nodes of an existing PHY multicast group occurs the transmitter may use broadcast DID when sending the protocol messages. The reserved MAC address 01-19-A7-52-76-96 shall be used as the DA in the LCDU delivering the MC_GrpInfoUpdate.ind message. The DestinationNode of the LLC frame corresponding to the LCDU delivering the MC_GrpInfoUpdate.ind message shall be set to zero.

8.16.2 Maintenance of multicast binding information

The transmitter shall send MC_GrpInfoUpdate.ind message as specified in this clause to update receivers of a PHY multicast group when there is a change in BATs, or in the membership of receiver nodes, or in Mc-ACK slot assignment.

Changes in the Mc-ACK slots assignments shall take effect only when the number of Mc-ACK slots following a multicast transmission changes, as reflected in the NUM_MCACK_SLOTS field of the PHY-frame header. The transmitter shall not indicate a different number of Mc-ACK slots in the NUM_MCACK_SLOTS field until all receivers assigned to acknowledge have confirmed their status in MC_GrpInfoUpdate.ind message by sending an MC_GrpInfoUpdate.cnf message. The transmitter shall not change the Mc-ACK slot assignment for an existing node if the number of Mc-ACK slots remains same.

A receiver that was assigned a Mc-ACK slot of a PHY multicast group associated with a multicast stream shall continue acknowledging in its assigned slot until its assignment for this Mc-ACK slot is terminated by an MC_GrpInfoUpdate.ind message.

The transmitter of a PHY multicast group may remove any receiver that is a member of the PHY multicast group at any time by sending an MC_GrpRemove.req message to that receiver. The receiver shall send the MC_GrpRemove.cnf message to the transmitter, confirming that it is no longer a member of the multicast group. If the receiver is assigned an Mc-ACK slot of the PHY multicast group, it shall continue acknowledging in its assigned slot until its assignment for this Mc-ACK slot is terminated by an MC_GrpInfoUpdate.ind message. From the time of receiving MC_GrpRemove.req until the assignment of its Mc-ACK slot is terminated by an MC_GrpInfoUpdate.ind message, the receiver shall set the FACK field to 111₂ (see clause 7.1.2.3.2.3.9.1.5/[10.1.2.3.2.3.9.1.5](#) of [\[ITU-T G.9960\]](#)).

The transmitter may split an existing PHY multicast group into several PHY multicast groups, for example, when new receivers with very different BAT join. The transmitter shall assign a new multicast DID to each of the newly created PHY multicast groups and shall send MC_GrpInfoUpdate.ind, which includes the information describing the new PHY multicast groups, to all nodes associated with that PHY multicast group, using either separate unicast DIDs, broadcast DID or other multicast group DIDs.

The transmitter shall follow the actions described in clause 8.16.1 each time it sends MC_GrpInfoUpdate.ind for informing on new PHY multicast groups or for updating existing PHY multicast group information.

The transmitter shall allocate a new BAT ID for a PHY multicast group when a change is required in any of the active BAT IDs of the PHY multicast group.

The MC_GrpInfoUpdate.ind message sent by the transmitter shall include the list of all BAT_IDs that are to be active in the PHY multicast group (inside the McstGroupInfo field, see Table 8-107). This consists of those BAT_IDs that are to be retained and new BAT_IDs to be added. BAT_IDs to be removed shall be excluded from the list. New BAT_IDs in this list are accompanied by BATInfo fields (see Table 8-109). The transmitter shall not start using the new BATs until all the PHY multicast group receivers have confirmed the change. Once the transmitter actually uses a new BAT in transmission, the receivers of the PHY multicast group shall invalidate the old BATs assigned to that PHY multicast group that were excluded from the list in the McstGroupInfo field of the last received MC_GrpInfoUpdate.ind message.

In case the transmitter detects a change in the PHY multicast group information while awaiting confirmation from the receivers, it shall restart the procedure generating full binding information and retransmitting an MC_GrpInfoUpdate.ind message with a higher sequence number.

8.16.3 Termination of a multicast group

When a transmitter wishes to terminate a PHY multicast group it shall send MC_GrpInfoUpdate.ind to the PHY multicast group members to release the PHY multicast DID. The receivers shall respond with an MC_GrpInfoUpdate.cnf message. Upon receiving the MC_GrpInfoUpdate.cnf messages from all the PHY multicast group members the transmitter shall terminate the PHY multicast group and release the multicast DID.

When all members have left a PHY multicast group a transmitter shall terminate the PHY multicast group and release the multicast DID.

8.16.4 PHY multicast binding protocol flow

8.16.4.1 Message sequence – Initialization of a PHY multicast group for a new multicast stream

Figure 8-58 shows an example of initialization of a PHY multicast group when the multicast stream is not active. The PHY multicast binding protocol messages are marked in grey arrows.

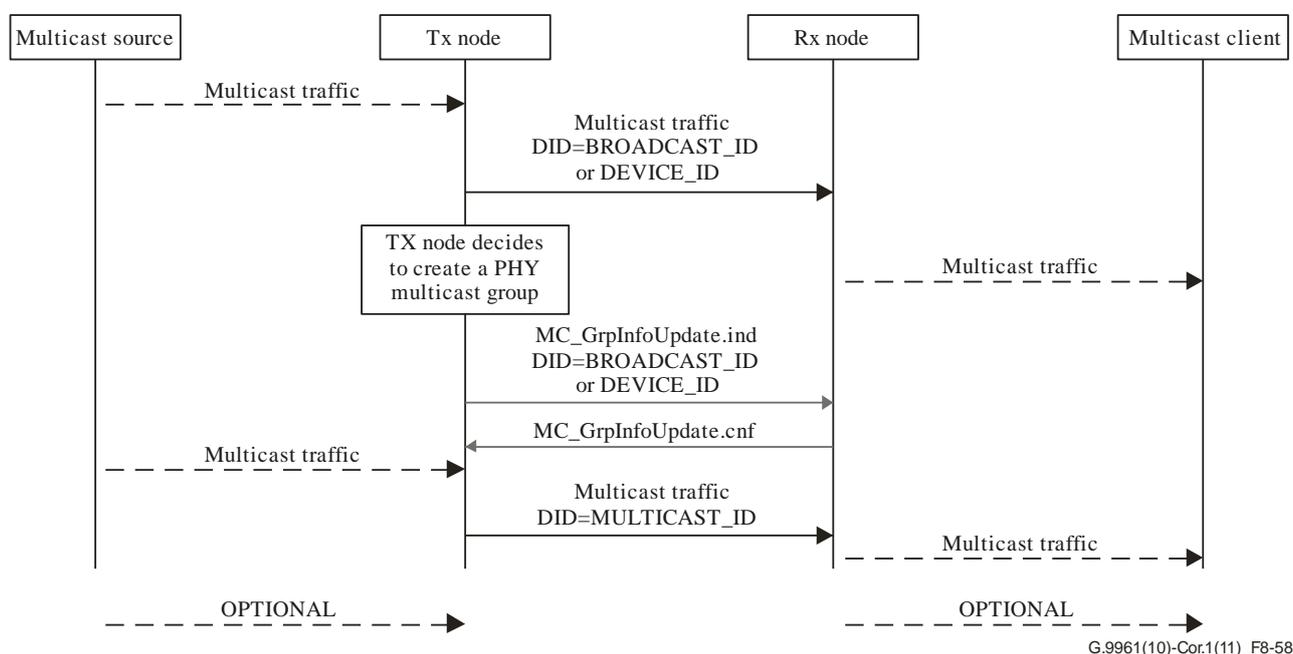


Figure 8-58 – Example of initializing a PHY multicast group

In this example the MC_GrpInfoUpdate.ind is sent after the transmitter decides to use the PHY multicast group mechanism to deliver a multicast stream to a set of receiver nodes.

8.16.4.2 Message sequence – Split of a multicast stream into several PHY multicast groups

Figure 8-59 shows an example of an existing multicast stream that is transmitted using a single PHY multicast group to two receivers. When the transmitter needs to add node number three to join the multicast stream, the transmitter decides that it is better to allocate a new PHY multicast group for this node. Hence, it creates another PHY multicast DID and informs node number three on the new group via the MC_GrpInfoUpdate.ind message. When node number three confirms the received message, the transmitter starts using the new PHY multicast group in addition to the existing multicast group.

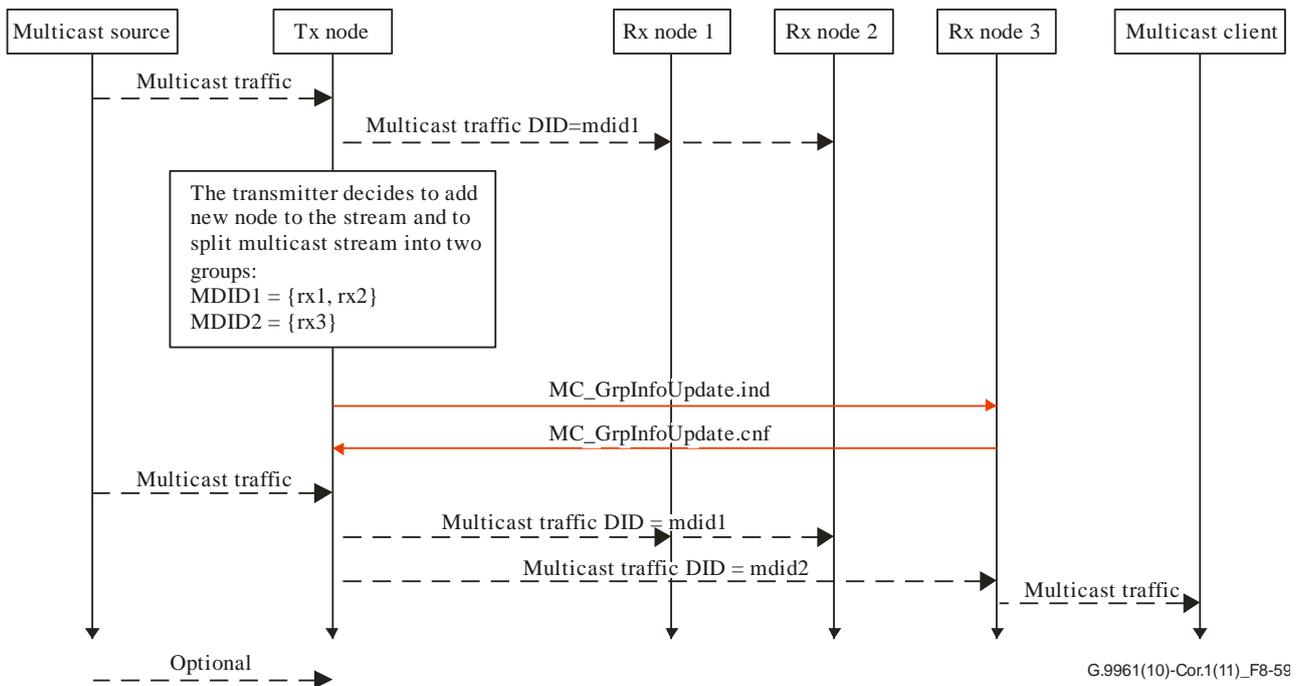
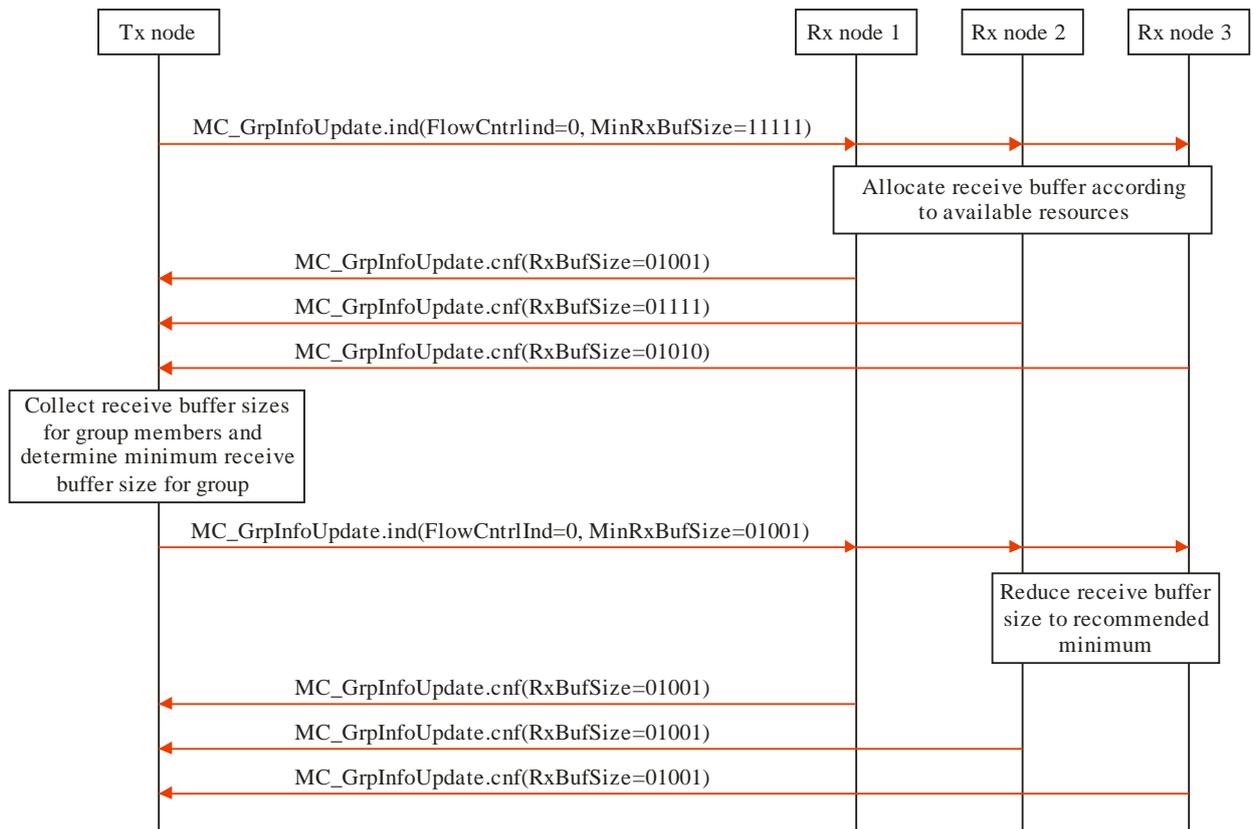


Figure 8-59 – Example of split of a PHY multicast group

8.16.4.3 Message sequence – Establish a PHY multicast group with flow-control disabled

Figure 8-60 shows an example of the establishment of a PHY multicast group with flow-control disabled. In this example the transmitter initiates the sequence by sending a MC_GroupInfoUpdate.ind message to the three PHY multicast group members specifying the recommended minimum receive buffer size. The PHY multicast group members allocate receive buffers for the multicast flow and respond with a MC_GroupInfoUpdate.cnf specifying the actual size of the allocated receive buffer. The transmitter collects the results from the MC_GroupInfoUpdate.cnf messages, calculates the new minimum receive buffer size and advertises it to the group using another MC_GroupInfoUpdate.ind message. Finally, the receivers adjust their receive buffer allocations according to the specified minimum receiver buffer size and reply to the transmitter using the updated MC_GroupInfoUpdate.cnf messages.

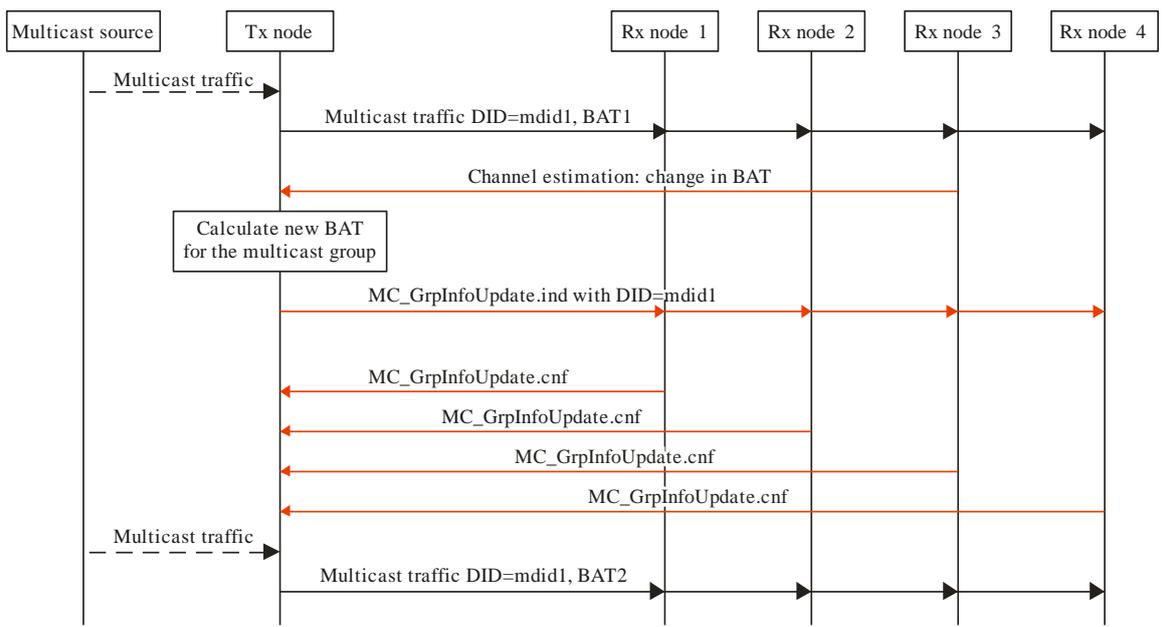


G.9961(10)_F8-60

Figure 8-60 – Example of establishment of a PHY multicast group with flow-control disabled

8.16.4.4 Message sequence – Maintenance of multicast binding information

Figure 8-61 shows an example of an existing multicast stream that is transmitted using a single PHY multicast group to its four receivers. When receiver number three reports on change in its recommended BAT, the transmitter decides to update the BAT of the PHY multicast group. Hence, it informs the receivers on the change via the MC_GrpInfoUpdate.ind message. After the change is confirmed by all the receivers the transmitter can start using the new BAT.



G.9961(10)-Cor.1(11)_F8-61

Figure 8-61 – Example of maintenance of PHY multicast binding information

NOTE – The transmitter sends MC_GrpInfoUpdate.ind to the multicast DID. Also, the transmitter may continue transmitting the multicast stream using the original BAT until the multicast binding is completed.

8.16.5 PHY multicast binding protocol messages

8.16.5.1 PHY multicast group information update indication

MC_GrpInfoUpdate.ind is a management message sent by the transmitter node to create a new PHY multicast group or to update all the receivers of a PHY multicast group about any change in the multicast binding information (e.g., update of existing group parameters). The McstGroupInfo within the message identifies a PHY multicast group uniquely by its multicast DID and the source ID of the transmitter.

Each PHY multicast group contains the list of all receivers of the group each uniquely identified by its device ID. Each receiver information contains the McAckSlot assignment if Mc-ACK is enabled for the PHY multicast group.

The format of the MMPL of the MC_GrpInfoUpdate.ind message shall be as shown in Table 8-106.

Table 8-106 – Format of the MMPL of the MC_GrpInfoUpdate.ind message

Field	Octet	Bits	Description
Source ID	0	[7:0]	The Device ID of the transmitter.
McstGroupInfo	<u>1 Variable</u>	<u>Variable See Table 8-107</u>	<u>Refer to See Table 8-107.</u>
...	..		
NumBATs	Variable	[7:0]	Number of BATs described. Zero indicates no BAT is described.
BATInfo[0]	..	<u>Variable See Table 8-109</u>	<u>Refer to See Table 8-109.</u>
...	..		
BATInfo[N]	..	Variable	<u>See Table 8-109.</u>

Table 8-107 – Format of McstGroupInfo Field

Field	Octet	Bits	Description
MulticastDID	0	[7:0]	Multicast DID for the group
RPRQ	1	[1:0]	As per RPRQ value defined in Table 8-86 – Types of multicast acknowledgement
NUM_MCAACK_SLOTS		[4:2]	This field shall contain the number of Mc-ACK slots
FlowControlInd		[5]	Flow control mechanism indication, indicating usage of the flow control mechanism by the receivers of the PHY multicast group: 0 – The flow control mechanism shall not be used. 1 – The flow control mechanism shall be used
Reserved		[7:6]	Reserved by ITU-T (Note 1)

Table 8-107 – Format of McstGroupInfo Field

Field	Octet	Bits	Description
MinRxBufSize	2	[4:0]	Recommended minimum receiver buffer size expressed in LPDUs to be buffered by receivers in the PHY multicast group. (Note 2) The values of this field shall be the same as FLCTRL field for status report (Table 7-21/10-21 of ITU-T G.9960)
MinRxBufSize_ BLKSZ unit		[6:5]	LPDU_size units for the MinRxBufSize field: 0 – 120 bytes 1 – 540 bytes 2 – 2160 bytes 3 – Reserved for use by ITU-T
Reserved		[7:6]	Reserved by ITU-T (Note 1)
NumBatIds	3	[7:0]	Number of BAT_IDs minus one used for this PHY multicast group as allocated by the transmitter. The number of BAT_IDs shall not exceed n=32.
BAT_ID	4	[7:0]	The first of n BAT IDs used for this PHY multicast group.
...
BAT_ID	n+3	[7:0]	The last of n BAT IDs used for this PHY multicast group.
NumRxNode	n+4	[7:0]	Number of receive nodes m that are members of the PHY multicast group. Zero indicates that this multicast DID is released.
RxNodeInfo	n+5 and n+6	[15:0]	Info for the first of m receive nodes of the PHY multicast group.
...
RxNodeInfo	n+m+4 and n+m+5	[15:0]	Info for the last of m receive nodes of the PHY multicast group.

NOTE 1 – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.

NOTE 2 – The receiver buffer size (MinRxBufSize) specified by a transmitter for a PHY multicast group may be reduced over time by sending another MC_GrpInfoUpdate.ind message with a new MinRxBufSize value. When the flow control mechanism is not used, a receiver may reduce the size of the receiver buffer to the new value specified by the transmitter.

When the flow control mechanism is used, a receiver shall ignore the value of this field.

Table 8-108 – Format of RxNodeInfo field

Field	Octet	Bits	Description
RxDeviceID	0	[7:0]	Device ID of a receive node of the PHY multicast group
McAckSlot	1	[2:0]	Mc-Ack Slot assigned to this node 0 – Use NACK Slot if NACK is enabled according to RPRQ of the PHY multicast group 1-7 – Mc-ACK Slot ID if Mc-ACK is enabled according to the RPRQ of the PHY multicast group
Reserved		[7:3]	Reserved by ITU-T (Note)
NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

Table 8-109 – Format of BATInfo field

Field	Octet	Bits	Description
New BAT_ID	0	[4:0]	The BAT ID this BATInfo describes. It shall be formatted as shown in Table 7-55/ 10-55 of [ITU-T G.9960] .
Bandplan-OFB_ID		[7:5]	This field indicates the type of bandplan-OFB used by the transmitter based on which the subsequent BAT entry is defined. It shall be formatted as shown in Table 7-10/ 10-10 of [ITU-T G.9960] .
Minimum group ID	1	[2:0]	This field indicates the minimum GRP_ID associated with the new BAT_ID. It shall be formatted as shown in Table 7-13/ 10-13 of [ITU-T G.9960] .
Reserved		[7:3]	Reserved by ITU-T (Note 1).
Number of valid durations	2	[2:0]	This field indicates the number of durations (n) specified for the new BAT_ID minus one. The valid range of this field is from 0 (n=1) to 7 (n=8) (Note 2).
Reserved		[7:3]	Reserved by ITU-T (Note 1).
CE_STIME ₁	3	[7:0]	This field indicates the start time of the first duration in which a new BAT is valid. It shall be formatted as shown in Table 8-98.
CE_ETIME ₁	4	[7:0]	This field indicates the end time of the first duration in which a new BAT is valid. It shall be formatted as shown in Table 8-99.
...
CE_STIME _n	2n+1	[7:0]	This field indicates the start time of the last duration in which a new BAT is valid. It shall be formatted as shown in Table 8-98.
CE_ETIME _n	2n+2	[7:0]	This field indicates the end time of the last duration in which a new BAT is valid. It shall be formatted as shown in Table 8-99.

Table 8-109 – Format of BATInfo field

Field	Octet	Bits	Description
NumBATEntries	2n+3 to 2n+4	[15:0]	Number of sub-carrier entries minus one contained in this message. Valid values are $0 \leq m \leq 4095$ 32767 .
B ₀	2n+5	[3:0]	4-bit unsigned integer indicating the number of bits assigned to sub-carrier index 0 (Note 3).
B ₁		[7:4]	4-bit unsigned integer indicating the number of bits assigned to sub-carrier index 1 (Note 3).
...
B _{m-1}	2n+4 + $\lfloor m/2 \rfloor$	[3:0]	4-bit unsigned integer indicating the number of bits assigned to sub-carrier index m-1 (Note 3).
B _m		[7:4]	4-bit unsigned integer indicating the number of bits assigned to sub-carrier index m (Note 3).
NOTE 1 – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			
NOTE 2 – A new BAT shall only be used over specified non-overlapping durations (up to 8) within a MAC cycle, defined by CE_STIME _i and CE_ETIME _i .			
NOTE 3 – If a sub-carrier is not loaded, this field shall be set to zero.			

8.16.5.2 Multicast binding information confirmation from receiver

Message MC_GrpInfoUpdate.cnf is a management message that shall be sent by a receiver node in response to the MC_MulticastGrpInfoUpdate.ind.

The format of the MMPL of the MC_GrpInfoUpdate.cnf message shall be as shown in Table 8-110.

Table 8-110 – Format of the MMPL of the MC_GrpInfoUpdate.cnf message

Field	Octet	Bits	Description
Sequence number	0 and 1	[15:0]	Sequence number (see Table 8-87) of the MC_GrpInfoUpdate.ind message that is confirmed.
StatusCode	2	[7:0]	Status for the response to MC_MulticastGrpInfoUpdate.ind: <ul style="list-style-type: none"> • 00₁₆ = Success (indicating MC_GrpInfoUpdate.ind has been accepted). • 01₁₆ = Failure – lack of resources. • 02₁₆ – FF₁₆ = Reserved by ITU-T.
RXBufSize	3	[4:0]	Available receiver buffer size. This field shall indicate the number of LPDUs that the receiver can buffer for this connection. The values of this field shall be the same as FLCTRL field for status report (Table 7-21/10-21 of [ITU-T G.9960]) (Note 1).
Reserved		[7:5]	Reserved by ITU-T (Note 2).
NOTE 1 – When the flow control mechanism is not used, as indicated by the FlowControlInd field in the MC_GrpInfoUpdate.ind message, all receivers of the PHY multicast group shall report the value for the FLCTRL field (Table 7-21/10-21 of [ITU-T G.9960]) as they reported in the RXBufSize field in the corresponding MC_GrpInfoUpdate.cnf message.			
NOTE 2 – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

8.16.5.3 Format of MC_GrpRemove.req

Message MC_GrpRemove.req is a management message that shall be sent by the transmitter of the PHY multicast group to a receiver of the PHY multicast group to request the receiver to leave the PHY multicast group.

The format of the MMPL of the MC_GrpRemove.req message shall be as shown in Table 8-111.

Table 8-111 – Format of the MMPL of the MC_GrpRemove.req message

Field	Octet	Bits	Description
RxDeviceID	0	[7:0]	DEVICE_ID of the receiver node that is requested to leave the PHY multicast group.
TxDeviceID	1	[7:0]	DEVICE_ID of the transmitter of the PHY multicast group.
MulticastID	2	[7:0]	MULTICAST_ID of the group.

8.16.5.4 Format of MC_GrpRemove.cnf

Message MC_GrpRemove.cnf is a management message that shall be sent by the receiver of a PHY multicast group to the transmitter of the PHY multicast group to confirm the reception of the MC_GrpRemove.req message.

The format of the MMPL of the MC_GrpRemove.cnf message shall be as shown in Table 8-112.

Table 8-112 – Format of the MMPL of the MC_GrpRemove.cnf message

Field	Octet	Bits	Description
RxDeviceID	0	[7:0]	DEVICE_ID of the receiver node that confirmed the request to leave the PHY multicast group.
TxDeviceID	1	[7:0]	DEVICE_ID of the transmitter of the PHY multicast group.
MulticastID	2	[7:0]	MULTICAST_ID for the group.

8.17 DLL multicast stream

A source node that decides to establish a DLL multicast stream shall establish a multicast path towards each client of the DLL multicast stream. The paths towards the client nodes may include relay nodes that are bound to the path and the DLL multicast stream identifier (MSID). The source node that establishes a DLL multicast group shall generate the DLL multicast stream identifier (MSID) that together with the DEVICE_ID of the source of the DLL multicast stream uniquely identifies the DLL multicast stream. The members of a DLL multicast group are identified by the source node of the DLL multicast stream. The source of a DLL multicast stream, shall transmit the traffic of the DLL multicast stream to the members of the DLL multicast group according to established paths as described in the following clauses.

8.17.1 DLL multicast stream establishment

A node that determines that it has to transmit a multicast stream to client nodes in the domain, shall establish a path to each one of the client nodes. The source node that generates the DLL multicast stream shall first allocate an MSID that together with the DEVICE_ID of the source node, shall uniquely identify the DLL multicast stream. Valid values of MSID are from 1 to 250. The source node shall also initialize the Transaction ID for that DLL multicast stream to zero. The source node shall increment the Transaction ID for each new DLL multicast path it establishes for that DLL multicast stream.

The source node shall establish the path towards a client node as follows:

If the source node has a direct link to the client node according to the current unicast routing table, it shall send a DMC_Path.req message to the client node to bind it with the specified MSID multicast stream and the multicast stream MAC Address. The client node shall reply with a DMC_Path.cnf message that contains the same Transaction_ID that was specified in the DMC_Path.req message and shall bind itself to the established path identified by the MSID, the DEVICE_ID of the source node and the multicast MAC address (DA). The source node upon receiving the DMC_Path.cnf message shall bind the path and complete the path establishment procedure. If the source node does not receive a DMC_Path.cnf message after a vendor discretionary period, which is larger than MAX_WAIT_TIME, it may repeat the request through a new DMC_Path.req with a different Transaction_ID.

NOTE – DMC_Path.req should be sent using connections with acknowledgements in order to avoid long set-up times for DLL multicast trees because of lost messages.

If the source node does not have a direct link to the client node, it shall determine the first relay node towards the client node according to the current unicast routing table and send a DMC_Path.req message to that node.

The DMC_Path.req message shall contain the following information: the DEVICE_ID of the source node of the DLL multicast, the allocated MSID, the DEVICE_ID of the client node, the MAC address of the multicast stream, the Transaction_ID, and the DEVICE_ID of the first relay node (Relay ID). If there is a direct link between the source node and the client node, the Relay ID field shall be set to the DEVICE_ID of the client node. The source node of the multicast stream shall address the DMC_Path.req message to the first relay node by setting the DA to the MAC address of that node.

A relay node that receives a DMC_Path.req message and has a direct link with the client node shall bind the DEVICE_ID of the source of the DLL multicast stream, the MSID, and the sender node's DEVICE_ID with the DEVICE_ID of the client endpoint node, and shall replace the DA of the LCDU of the DMC_Path.req message with the client node's MAC address and transmit the DMC_Path.req message to the client node.

A relay node that receives a DMC_Path.req message and does not have a direct link to the client node shall bind the DEVICE_ID of the source of the DLL multicast stream, the MSID, and the sender node's DEVICE_ID with the DEVICE_ID of the next relay node towards the client node according to the unicast routing table. It shall then replace the DA of the LCDU of the DMC_Path.req message with the MAC address of the next relay node and replace the Relay ID field by the DEVICE_ID of the next relay node and send the updated DMC_Path.req message to that node.

Upon reception of the DMC_Path.req message, the client node shall reply to the node that sent this message with a DMC_Path.cnf message and shall bind itself to the specified DLL multicast stream identified by the DEVICE_ID of the source DLL multicast stream, the MSID, and the sender node's DEVICE_ID.

A relay node that receives the DMC_Path.cnf message shall mark the binding of the DLL multicast stream path identified by the DEVICE ID of the source DLL multicast stream node and the MSID as valid. The relay node shall then append its DEVICE_ID to the Path_List field in the MMPL of the

received DMC_Path.cnf message. The relay node shall transmit the updated DMC_Path.cnf message to the node from which it has received the DMC_Path.req message, which can be either a relay node or the node originating the DLL multicast stream.

Once the originating node receives the DMC_Path.cnf message, it has the complete path of this bound client from the received DMC_Path.cnf message. This completes the path establishment procedure. The source node may then start sending the multicast stream packets towards the client node(s) either directly or via the first relay node according to the established path.

Each relay node shall identify LLC frames corresponding to a DLL multicast stream according to the OriginatingNode and the MSID specified in the LFH. The relay node shall then relay any received LLC frames of that DLL multicast stream to all the nodes it has bound to this DLL multicast stream according to the binding information that it has configured during the DLL multicast stream path establishment. The relay node shall only relay LLC frames corresponding to a DLL multicast stream path for which its binding is marked as valid.

When the multicast source node or any other relay node in the DLL multicast paths receives an updated routing table, it shall not update the current multicast paths. A relay node shall correct an established multicast path only by explicit order received from the multicast source node as defined in clause 8.17.3.

The procedure described above assumes that each node involved in the DLL multicast stream path establishment has sufficient resources to support the DLL multicast stream. However in some cases, it is possible that a node that is requested to establish a link in the DLL multicast stream path by sending a DMC_Path.req, does not have sufficient resources to support the link. Each node (which can be either a relay node or the client node) that receives a DMC_Path.req message and does not have sufficient resources to support that DLL multicast stream shall send a DMC_Path.cnf message with the status set to "the request is rejected due to lack of resources" towards the node (which can be either a relay node or the node originating the DLL multicast stream) from which it has received the DMC_Path.req message. Each relay node in the path that receives the DMC_Path.cnf message with the rejection indication shall forward the DMC_Path.cnf message with the rejection indication towards the node (which can be either a relay node or the node originating the DLL multicast stream) from which it has received the DMC_Path.req message and release any resources allocated to support the path to the specified client.

Once the originating node receives the DMC_Path.cnf message with the rejection indication it shall delete the client node from the DLL multicast stream as specified in clause 8.17.4.

8.17.2 DLL multicast stream establishment with bandwidth reservation

An originating node that decides to establish a DLL multicast stream shall also decide whether to have it 'BW reserved' or not. The 'BW reserved' attribute shall be carried in the DLL multicast stream protocol messages so all participating nodes will have that knowledge as well.

It is the responsibility of the originating node and the relay nodes in the stream to make sure new non-leaf nodes added to a bandwidth reserved stream support this feature (each node reports its node information via NodeInformationTLVs (see Table 8-16.1) to the DM, which then broadcasts this information to the domain, see clause 8.6.4).

DLL multicast stream establishment with BW reservation includes the same steps as specified in the previous clause.

In addition to these steps, the actions specified in the following clauses enable the nodes to reserve BW from the DM.

8.17.2.1 Bandwidth reservation when direct link from originating node to client node

Once the originating node receives the DMC_Path.cnf message from the client node, it shall decide whether bandwidth reservation is required for the stream.

If BW reservation from the DM is required, the originating node will ask the DM to reserve BW for the stream by sending a DMC_BWReserve.req message. The DMC_BWReserve.req message will include the BW required for the DLL multicast stream, and the rate from the originating node to the client node. The rate enables the domain master to estimate the time allocation needed to serve the DLL multicast transmission based on the number of bytes needed to be transmitted and the rate of the node.

Based on its calculations, the DM shall reply with a DMC_BWReserve.cnf message, and if bandwidth reservation was approved, the DM will then allocate TXOPs for this hop in the DLL multicast stream (identified by the MSID of the stream and the SID of the node) if this is the first bandwidth reservation for this DLL multicast stream, or change the allocation for existing TXOPs for this hop in the multicast stream if this is a bandwidth update for an existing stream.

In case that the DM concludes that it does not have enough bandwidth to reserve for this DLL multicast stream to support the specified client node it shall respond to the originating node with a DMC_BWReserve.cnf message with the StatusCode set to requested bandwidth is denied indication. Upon receiving the DMC_BWReserve.cnf message with the rejection indication, the originating node shall release the client node from the DLL multicast stream as specified in clause 8.17.4.

8.17.2.2 Bandwidth reservation through relay

The source node will indicate that the DLL multicast stream bandwidth reserved and that transmission should occur only in allocated TXOPs (this is done through the 'Is BW managed' field in the DMC_Path.req message).

A relay node that receives a DMC_Path.req message shall make the bandwidth reservation only when both of the following criteria are met:

- the 'Is BW managed' field in the DMC_Path.req message is set to one; and
- the relay node has not already made a bandwidth reservation for the same hop and the same DLL multicast stream.

In addition to the steps described in clause 8.17.1 – DLL multicast stream establishment, the relay node shall send a DMC_BWReserve.req message to DM. The DMC_BWReserve.req message will include the SID of the relay node, the MSID of the stream, the DEVICE ID of the originating node of the stream and the requested bandwidth.

Based on this information the DM will calculate whether requested bandwidth can be allocated and reply with a DMC_BWReserve.cnf message. If bandwidth reservation was approved, the DM will then allocate resources to serve the hop in the DLL multicast stream.

If the bandwidth reservation was not approved, the DM shall respond to the relay node with the DMC_BWReserve.cnf message with the StatusCode set to "the requested bandwidth is denied". Upon receiving the DMC_BWReserve.cnf message with the rejection indication the relay node shall send a DMC_Path.cnf message with the status set to "the request is rejected due to rejection of bandwidth reservation from the DM" towards the node (which can be either a relay node or the node originating the DLL multicast stream) from which it has received the DMC_Path.req message. Each relay node in the path that receives the DMC_Path.cnf message with the rejection indication shall forward the DMC_Path.cnf message with the rejection indication towards the node (which can be either a relay node or the node originating the DLL multicast stream) from which it has received the DMC_Path.req message, and release any bandwidth reservation related resources allocated to support the path to the specified client by sending DMC_BWRelease.req to the DM. In this case, the DM shall reply with a DMC_BWRelease.cnf and completely remove the TXOPs assigned for the indicated hop in the bandwidth reserved multicast stream.

Once the originating node receives the DMC_Path.cnf message with the rejection indication it shall delete the client node from the DLL multicast stream as specified in clause 8.17.4.

8.17.3 Preventing loops and packet duplications

The paths of a specific DLL multicast stream shall be established in a tree topology that ensures that a node shall not receive duplicate multicast packets from different paths and prevent the source node or any relay node from unnecessarily duplicating transmissions. The topology of the DLL multicast stream tree is built under a principal rule that each node shall receive packets of a specific (OriginatingNode, MSID) only from one node. The DLL multicast stream paths tree shall be built according to this rule by executing the following procedure in path establishment: When a source node binds a new client node to an existing DLL multicast stream, it shall send towards it the DMC_Path.req message as defined in the previous clause. Any relay node on the path towards the newly joined client node shall verify that it always receives the DMC_Path.req for this specific (OriginatingNode, MSID) from the same sender node. In case it receives a DMC_Path.req message from a node different from the sender node to which it is currently bound, it shall reply with the DMC_PathReject.cnf message towards the source node. The DMC_PathReject.cnf message shall contain the rejecting node's DEVICE_ID, the DEVICE_ID of the node that sent it DMC_Path.req message and the rejection reason (duplication source).

When the source multicast node receives the DMC_PathReject.cnf message, it may decide to release the entire tree or the branch and rebuild it again, or to enforce establishment of the path up to the rejecting relay node based on the existing path. If the source node decides to enforce the existing path, it shall send the DMC_EnforcePath.req towards the relay node that encountered the problem via the original path. The source node shall address the DMC_EnforcePath.req message to the first relay node in the path toward the rejecting node. The DMC_EnforcePath.req message shall contain the full path up to the rejecting node and the client node. Each relay node that receives the DMC_EnforcePath.req message shall forward the message to the next relay node according to the specified path toward the rejecting node. When the rejecting node receives the DMC_EnforcePath.req message it shall create a DMC_Path.req message, filling it with the information received in the DMC_EnforcePath.req message, and forward the message to the next relay node according to the current routing table. From this phase, the path establishment procedure towards the client node shall continue as specified in the previous clause. The client node shall reply with the DMC_Path.cnf message to the relay node that sent it the DMC_Path.req message. All the relay nodes on the path towards the source node upon receiving the DMC_Path.cnf message, shall execute the bind, update the Path_List field in the DMC_Path.cnf message with their own DEVICE_ID and forward the DMC_Path.cnf message towards the source node.

A specific relay that has to forward a specific MSID stream traffic to several nodes that are bound with this MSID may establish a PHY multicast group. In this case, the node may create or update PHY multicast groups when it receives a DMC_Path.req message to transmit the data to the next relay nodes or client nodes. In that case, the PHY multicast group shall only include bound client nodes and relay nodes that are in its bind list for this MSID in its current hop.

Node A decides to add node N to the '*MSID*' DLL multicast stream.
According to the current routing table, node A sends the *DMC_Path.req* message to node B.

According to the current routing table, node B sends the *DMC_Path.req* message to node R.

According to the current routing table Relay node R sends the *DMC_Path.req* message to node E.

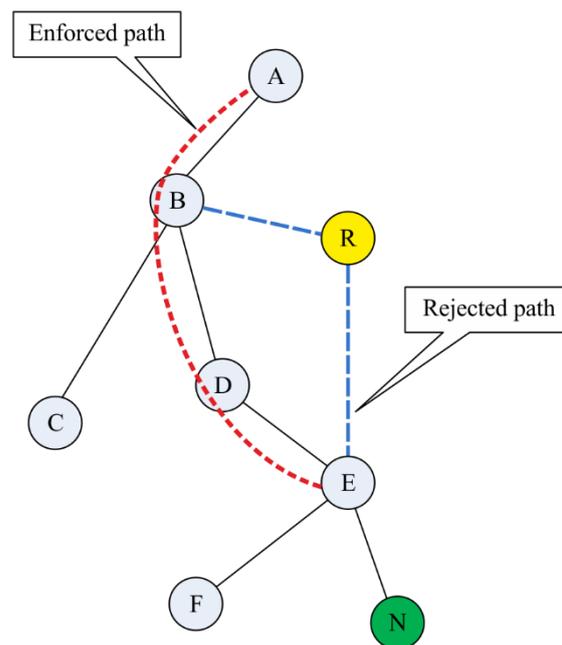
Node E knows that the legitimate source node for the specified '*MSID*' is node D, therefore it rejects the *DMC_Path.req* message by sending the *DMC_PathReject.cnf* message to node R.

Node R receives the *DMC_PathReject.cnf* message and sends a *DMC_PathReject.cnf* message node B.

Node B receives the *DMC_PathReject.cnf* message and sends a *DMC_PathReject.cnf* message node A.

Node A sends the *DMC_EnforcePath.req* message to node B to establish the path towards node N with a specified path until node E. Node B sends the *DMC_EnforcePath.req* message to node D (and not to node R) according to the specified path in the *DMC_EnforcePath.req* message.

Node D sends the *DMC_EnforcePath.req* message to node E and node E sends *DMC_Path.req* message to node N.



G.9961(15)_F8-62

Figure 8-62 – Example of the mechanism for preventing loops in the DLL multicast tree

8.17.4 Releasing client node from MSID

When the source node of a DLL multicast stream decides to release a client node from its MSID, it shall increment the Transaction ID of that DLL multicast stream and shall send a *DMC_ReleasePath.req* message to the respective client node or to its first relay node in case the client node is accessed via relay node(s). Each node that receives the *DMC_ReleasePath.req* shall release the specified node from this bind list and forward the message towards the client node. Each node that received the message shall reply with the *DMC_ReleasePath.cnf* message to the node that sent it the *DMC_ReleasePath.req* message. Each relay node that does not have any nodes in its bind list shall release itself from the MSID multicast stream and indicate it in the replied *DMC_ReleasePath.cnf* message.

8.17.5 Recovery from a DLL multicast broken path

In the case where one of the relay nodes determines that the path of a specific MSID is broken, it shall inform the domain master via a normal topology update message and the source node of the DLL multicast stream via a *DMC_BrokenLink.ind* message.

The multicast source node may correct the broken path according to a newly received updated routing table from the domain master. The source node may correct an existing path by sending *DMC_ReleasePath.req* to the relevant nodes and then it shall send new *DMC_Path.req* to the relevant nodes.

8.17.6 Aging DLL multicast path process

In order to prevent a situation where a multicast source node leaves the network and all the respective nodes in the multicast stream path are still holding MSID resources, an aging mechanism shall be used. The source node of each DLL multicast stream (MSID) shall send a management message, *DMC_PathAlive.ind*, via the established MSID DLL multicast stream paths tree at least twice within *DMC_PATH_AGING_PERIOD* (1 second) to the first node of each path (client node or relay node).

Each node in the tree that receives this message shall reset its aging timer for that DLL multicast stream and shall transmit the DMC_PathAlive.ind message to each of the nodes that are bound to this DLL multicast stream, identified by (OriginatingNode, MSID), according to the binding information that it has configured during the DLL multicast stream paths establishment. Each node in the path that does not receive a DMC_PathAlive.ind message within a period of DMC_PATH_AGING_PERIOD shall remove itself from this DLL multicast stream and release all of its MSID resources.

8.17.7 DLL Multicast stream bandwidth maintenance

Each node participating in a DLL multicast stream that is bandwidth reserved (whether it is the originating node or a relay node) is responsible for modifying its hop's requested bandwidth according to changes in the stream (e.g., if nodes are removed from a binding list, or if the hop's rate is changed, or if the hop transmission method is changed from using unicast connection to a PHY multicast connection).

If a node is no longer acting as a relay node in a DLL multicast stream, it is this node's responsibility to ask the DM to release all bandwidth reserved for its hop in the stream.

The modification of the bandwidth reservation is done by sending DMC_BWReserve.req to the DM. In this case, the DM shall reply with a DMC_BWReserve.cnf and change the allocation for the existing TXOPs for the indicated hop in the bandwidth reserved multicast stream.

The allocated bandwidth can be released by sending DMC_BWRelease.req to the DM. In this case, the DM shall reply with a DMC_BWRelease.cnf and completely remove the TXOPs assigned for the indicated hop in the bandwidth reserved multicast stream.

The internal rules used by the domain master to decide whether an allocation should be expanded or contracted due to ongoing flow maintenance done by the bandwidth management function are out of the scope of this Recommendation.

NOTE – If the bandwidth allocation for the DLL multicast stream is changed, this will be reflected in the MAP describing the following MAC cycles.

8.17.8 Transmission of bandwidth managed DLL multicast stream

Originating node and relay nodes participating in a bandwidth managed DLL multicast streams should transmit the traffic associated with the multicast stream only in TXOPs assigned for that stream. This assignment is indicated via the 'Multicast Indication' flag, SID and MSID fields in the MAP's TXOP basic descriptor (see clause 8.8.4)

8.17.9 DLL Multicast protocol messages

8.17.9.1 DMC_Path.req message format

The format of the DMC_Path.req management message shall be as shown in Table 8-113.

Table 8-113 – Format of the MMPL of the DMC_Path.req message

Field	Octet	Bits	Description
Source ID	0	[7:0]	The DEVICE_ID of the source node of the DLL multicast stream.
MSID	1	[7:0]	The multicast identification allocated by the source of the DLL multicast stream.
ClientID	2	[7:0]	The DEVICE_ID of the client node of the DLL multicast stream source node.
MulticastAddress	3 to 8	[47:0]	MAC address of the multicast stream.
Transaction_ID	9	[7:0]	Identifies this path transaction.

Table 8-113 – Format of the MMPL of the DMC_Path.req message

Field	Octet	Bits	Description
Relay ID	10	[7:0]	The DEVICE_ID of the next hop relay node towards the client node (Note 1)
Is BW Managed	11	[0]	0 – If this DLL multicast stream is not bandwidth reserved 1 – If this DLL multicast stream is bandwidth reserved (and transmission should occur only in allocated TXOPs)
Reserved		[7:1]	Reserved by ITU-T. (Note 2)
NOTE 1 – If there is a direct link between the source node and the client node, this field shall be set with the DEVICE_ID of the client node			
NOTE 2 – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

8.17.9.2 DMC_Path.cnf message format

The format of the DMC_Path.cnf management message shall be as shown in Table 8-114.

Table 8-114 – Format of the MMPL of the DMC_Path.cnf message

Field	Octet	Bits	Description
Source ID	0	[7:0]	The DEVICE_ID of the source node of the DLL multicast stream.
MSID	1	[7:0]	The multicast identification allocated by the source of the DLL multicast stream.
ClientID	2	[7:0]	The DEVICE_ID of the client node of the DLL multicast stream source node.
Transaction_ID	3	[7:0]	Identifies this path establishment transaction. It shall contain the same value that was specified in the corresponding DMC_Path.req message.
NumOfNodes	4	[7:0]	Specifies the number of relay nodes (n) in the Path_List from the source node towards the client node.
Path_List[0]	5	[7:0]	This entry in the list contains the DEVICE_ID of the last relay node in the established path from the source node towards the client node.
...
Path_List[n-1]	4+n	[7:0]	This entry in the list contains the DEVICE_ID of the first relay in the established path from the source node towards the client node.
StatusCode	5+n	[7:0]	Indicates if the request in the DMC_Path.req is confirmed or rejected. 0: the request is confirmed 1: the request is rejected due to lack of resources 2: the request is rejected due to rejection of bandwidth reservation from the DM. 3-255: reserved by ITU-T.

8.17.9.3 DMC_PathReject.cnf message format

The format of the DMC_PathReject.cnf management message shall be as shown in Table 8-115.

Table 8-115 – Format of the MMPL of the DMC_PathReject.cnf message

Field	Octet	Bits	Description
Source ID	0	[7:0]	The DEVICE_ID of the source node of the DLL multicast stream.
MSID	1	[7:0]	The multicast identification allocated by the source of the DLL multicast stream.
ClientID	2	[7:0]	The DEVICE_ID of the client node of the DLL multicast stream source node
Transaction_ID	3	[7:0]	Identifies this path establishment transaction specified in the DMC_Path.req message.
RejectingNodeId	4	[7:0]	The DEVICE_ID of the relay node that rejects the DMC_Path.req message
Rejection_code	5	[7:0]	00 ₁₆ – The request path is conflicted because there is already a path established for the specified DLL multicast stream with a different relay node. 01 ₁₆ – The node is not able to support additional DLL multicast streams. 02 ₁₆ to FF ₁₆ – Reserved by ITU-T.

8.17.9.4 DMC_EnforcePath.req message format

The format of the DMC_EnforcePath.req management message shall be as shown in Table 8-116.

Table 8-116 – Format of the MMPL of the DMC_EnforcePath.req message

Field	Octet	Bits	Description
Source ID	0	[7:0]	The DEVICE_ID of the source node of the DLL multicast stream.
MSID	1	[7:0]	The multicast identification allocated by the source of the DLL multicast stream.
ClientID	2	[7:0]	The DEVICE_ID of the client node of the DLL multicast stream source node.
MulticastAddress	3 to 8	[47:0]	MAC address of the multicast stream
Transaction_ID	9	[7:0]	Identifies this path establishment transaction. It shall contain the same value as in the original DMC_Path.req for this path.
NumOfNodes	10	[7:0]	Specifies the number of relay nodes (n) in the Path_List from the source node towards the rejecting node.
Path_List[0]	11	[7:0]	This entry in the list contains the DEVICE_ID of the first relay node in the established path from the source node towards the rejecting node.
...
Path_List[n-1]	10+n	[7:0]	This entry in the list contains the DEVICE_ID of the last relay in the established path from the source node towards the rejecting node.

8.17.9.5 DMC_ReleasePath.req message format

The format of the DMC_ReleasePath.req management message shall be as shown in Table 8-117.

Table 8-117 – Format of the MMPL of the DMC_ReleasePath.req message

Field	Octet	Bits	Description
Source ID	0	[7:0]	The DEVICE_ID of the source node of the DLL multicast stream.
MSID	1	[7:0]	The multicast identification allocated by the source of the DLL multicast stream.
ClientID	2	[7:0]	The DEVICE_ID of the client node of the DLL multicast stream source node to be release from the path.
Transaction_ID	3	[7:0]	Identifies this path transaction.
NumOfNodes	4	[7:0]	Specifies the number of relay nodes (n) in the Path_List from the source node towards the client node.
Path_List[0]	5	[7:0]	This entry in the list contains the DEVICE_ID of the first relay node in the established path from the source node towards the client node.
...
Path_List[n-1]	4+n	[7:0]	This entry in the list contains the DEVICE_ID of the last relay node in the established path from the source node towards the client node.

8.17.9.6 DMC_ReleasePath.cnf message format

The format of the DMC_ReleasePath.cnf management message shall be as shown in Table 8-118.

Table 8-118 – Format of the MMPL of the DMC_ReleasePath.cnf message

Field	Octet	Bits	Description
Source ID	0	[7:0]	The DEVICE_ID of the source node of the DLL multicast stream.
MSID	1	[7:0]	The multicast identification allocated by the source of the DLL multicast stream.
ClientID	2	[7:0]	The DEVICE_ID of the client node of the DLL multicast stream source node to be release from the path.
Transaction_ID	3	[7:0]	Identifies this path establishment transaction. It shall contain the same value as in the corresponding DMC_ReleasePath.req for this path.
RelayNodeStatus	4	[7:0]	Specifies the status of the relay node that sends this message. 0: The relay node released itself from the specified DLL multicast stream. 1: The relay node still belongs to the specified DLL multicast stream. 2 to 255: Reserved by ITU-T.

8.17.9.7 DMC_PathAlive.ind message format

The format of the DMC_PathAlive.ind management message shall be as shown in Table 8-119.

Table 8-119 – Format of the MMPL of the DMC_PathAlive.ind message

Field	Octet	Bits	Description
Source ID	0	[7:0]	The DEVICE_ID of the source node of the DLL multicast stream.
MSID	1	[7:0]	The multicast identification allocated by the source of the DLL multicast stream.

8.17.9.8 DMC_BrokenLink.ind message format

This message is sent by a node that needs to report to the source of a DLL multicast stream that a link towards a multicast client node is broken.

The format of the DMC_BrokenLink.ind message shall be as shown in Table 8-120.

Table 8-120 – Format of the MMPL of the DMC_BrokenLink.ind message

Field	Octet	Bits	Description
Source ID	0	[7:0]	The DEVICE_ID of the source node of the DLL multicast stream.
Reporting_DeviceID	1	[7:0]	The DEVICE_ID of the node reporting the broken link.
Broken_DeviceID	2	[7:0]	DEVICE_ID of the node with which the link is broken.
StatusCode	3	[7:0]	0: the reporting node experienced a broken link. 1: the reporting node has no bind information for this MSID. 2 to 255: Reserved by ITU-T.
NumberAffectedMSID	4	[7:0]	Number n of MSIDs affected by the broken link.
MSID0	5	[7:0]	The multicast identification of the first affected MSID.
...
MSIDn	Variable	[7:0]	The multicast identification of the nth affected MSID.

8.17.9.9 DMC_BWReserve.req message format

The format of the DMC_BWReserve.req management message shall be as shown in Table 8-121.

Table 8-121 – Format of the MMPL of the DMC_BWReserve.req message

Field	Octet	Bits	Description
Originating Node	0	[7:0]	The DEVICE_ID of the originating node of the DLL Multicast stream.
MSID	1	[7:0]	The multicast identification allocated by the source of the DLL multicast stream.
Requested BW	3 to 6	[39:8]	Specifies the requested bandwidth rate in bit/s, represented as a 32-bit unsigned integer.
Rate	7 and 8	[55:40]	The PHY data rate in bits per second for this hop in steps of 32 kbit/s.

8.17.9.10 DMC_BWReserve.cnf message format

The format of the DMC_BWReserve.cnf management message shall be as shown in Table 8-122.

Table 8-122 – Format of the MMPL of the DMC_BWReserve.cnf message

Field	Octet	Bits	Description
StatusCode	0	[7:0]	00 ₁₆ – The requested bandwidth is approved. 01 ₁₆ – The requested bandwidth is denied. 02 ₁₆ to FF ₁₆ – reserved by ITU-T
Originating Node	1	[7:0]	The DEVICE_ID of the originating node of the DLL Multicast stream.
MSID	2	[7:0]	The multicast identification allocated by the source of the DLL multicast stream.

8.17.9.11 DMC_BWRelease.req message format

The format of the DMC_BWRelease.req management message shall be as shown in Table 8-123.

Table 8-123 – Format of the MMPL of the DMC_BWRelease.req message

Field	Octet	Bits	Description
Originating Node	0	[7:0]	The DEVICE_ID of the originating node of the DLL Multicast stream.
MSID	1	[7:0]	The multicast identification allocated by the source of the DLL multicast stream.

8.17.9.12 DMC_BWRelease.cnf message format

The format of the DMC_BWRelease.cnf management message shall be as shown in Table 8-124.

Table 8-124 – Format of the MMPL of the DMC_BWRelease.cnf message

Field	Octet	Bits	Description
MSID	0	[7:0]	The multicast identification allocated by the source of the DLL multicast stream.
Originating Node	1	[7:0]	The DEVICE_ID of the originating node of the DLL Multicast stream.

8.18 Inter-bandplanOFB interoperability

ITU-T G.9960/1 specifies, for each of the supported mediums, transceivers capable of operating with different bandplansOFBs. Transceivers of different bandplansOFBs, specified for the mediums that the domains work on, may be used in the same domain and shall be capable of interoperating with transceivers of other bandplansOFBs, specified for the same medium. This requirement does not apply to transceivers working on disjoint frequency bands since they cannot interoperate. This clause specifies the means provided by this Recommendation by which transceivers of different bandplansOFBs can interoperate, by either specifying these means or referencing other clauses of the ITU-T G.9960/1 Recommendations that specify them.

8.18.1 BandplanOFB-related information

For allowing nodes of different bandplansOFBs to communicate, the following parameters are used in this Recommendation:

- BandplanOFB-related capabilities of a node:
 - The node's maximum and minimum reported bandplanOFB: This is the maximum (and minimum) bandplanOFB that a node supports, and is reported by the node during its registration. ~~This~~ These maximum bandplanOFBs cannot be changed during the time the node is registered to the domain. A node is allowed to register to a domain only if ~~its bandplanit supports and OFB that~~ is within the range indicated by the Minimum bandplanOFB and Maximum bandplanOFB allowed by the domain (see Table 8-77);
 - StartSubCarrier & StopSubCarrier: The first and last sub-carrier indexes supported by the node. The StartSubCarrier & StopSubCarrier are within the node's maximum reported bandplanOFB. For Profile 1 OFBs (see Table 6-1 of [ITU-T G.9960]), ~~T~~the bandwidth determined by the StartSubCarrier & StopSubCarrier shall be equal to or lower than the bandwidth associated with the node's maximum reported bandplanOFB. If it is lower, it shall be lower than the bandwidth of this maximum bandplanOFB by no more than 15% of the non-masked sub-carriers. The StartSubCarrier & StopSubCarrier cannot be changed while the node is registered to the domain;
 - The bandplanOFB related information (i.e., the node's maximum reported bandplanOFB and StartSubCarrier & StopSubCarrier) of each node in the domain is made available to all nodes in the domain, as specified in clause 8.18.3.
- Configured minimum & maximum bandplansOFBs for the domain: Values configured by a service provider (or user) to determine the minimum and maximum bandplanOFBs allowed in the domain. See Table 8-77 and clause 8.18.2;
- BandplanOFB-related information for payloads using pre-defined BATs: The BNDDL-OFB_ID (bandplanOFB identifier) field in the PFH of the MSG PHY frame and the MAP PHY frame is used to identify the bandplanOFB used for transmitting this frame's payload (whenever the payload is transmitted using pre-defined BATs). The bandplanOFB indicated by the BNDDL-OFB_ID field may be lower than the node's maximum reported bandplanOFB indicated by the node during registration and it may be lower than the minimum bandplanOFB configured for the domain (but it shall not be lower than the minimum configured OFB for the domain). For PHY frames transmitted by a node, this bandplanOFB may be different from one frame to another (even if the destination node is the same node). The BNDDL-OFB_ID field is not relevant for payloads that are transmitted using runtime BATs;
- BandplanOFB-related information for payloads using runtime BATs: For payload transmissions using runtime BATs, the CE_ParamUpdate.req and the MC_GrpInfoUpdate.ind messages include the following set of relevant parameters for each BAT_ID associated with a specific runtime BAT:
 - BandplanOFB_ID: the bandplanOFB of the specific BAT associated with the BAT_ID;

- $TIDX_{MIN}$ & $TIDX_{MAX}$ of the specific BAT associated with the BAT_ID .

NOTE – Given $TIDX_{MIN}$ & $TIDX_{MAX}$, the Bandplan-OFB ID information is redundant.

NOTE – For the sake of computation of minimum and maximum OFBs, Profile 1 OFBs are always considered as lower than Profile 2 OFBs.

8.18.2 Configuring the minimum and maximum bandplan-OFBs for the domain

A service provider (or user) configuring the home network may configure the minimum and maximum bandplans-OFBs allowed in the domain. These parameters are specified in [ITU-T G.9962] (See clauses 7.4.9 and 7.4.10 of [ITU-T G.9962]) and are published in the MAP's PSD-related domain info sub-field (see Table 8-77). The default values for the minimum and maximum configured bandplans-OFBs shall be set to the lowest and highest bandplan-OFBs specified in the Recommendation for the specific medium, respectively (for example, for powerline, the default minimum bandplan-OFB is 25 MHz, and the default maximum bandplan-OFB is may be set to 100 MHz).

A node trying to join the domain shall only try to join when its bandplan-OFB is within the range set by the minimum and maximum configured bandplans-OFBs. The DM shall reject registration requests from nodes which indicate (in the ADM_NodeRegisterRequest.req and the TM_NodeTopologyChange.ind messages) that their maximum reported bandplans supported OFBs do not overlap is outside the minimum and maximum bandplans-OFB range configured for the domain.

NOTE – An end point that does not understand the Minimum OFB indicated in the MAP-D should not try to register into the domain.

A node may transmit any type of PHY frame (when the payload is either transmitted with a pre-defined or runtime BAT) in a bandplan which is lower than the minimum bandplan configured for the domain.

8.18.3 Conveying bandplan-OFB-related information to all nodes in the domain

In order to allow inter-bandplan-OFB operation over a domain, every node communicating with another node within the domain needs to know the bandplan-OFB-related information (the node's maximum reported bandplan-OFB, StartSubCarrier & StopSubCarrier) of that node. In order to achieve that, the Recommendation specifies the following mechanism:

- 1) **Registration:** The registering node reports its minimum and maximum supported bandplan-OFB (referred to as the "node's minimum reported OFB" and "node's maximum reported bandplan-OFB", respectively) and the StartSubCarrier and StopSubCarrier in the ADM_NodeRegisterRequest.req message.
- 2) **Conveying the routing information from the DM to all nodes of the domain:** The DM reports the minimum and maximum reported bandplan-OFB and StartSubCarrier and StopSubCarrier of all nodes of the domain in the TM_DomainRoutingChange.ind message.
- 3) **Topology information report of a node to the DM:** A node reports its minimum and maximum bandplan-OFB and its StartSubCarrier and StopSubCarrier whenever it sends its TM_NodeTopologyChange.ind message to the DM.

The bandplan-OFB related information (the node's minimum and maximum reported-supported bandplan-OFB and StartSubCarrier & StopSubCarrier) is carried in the above mentioned messages in the "Bandplan-OFB Info Capability Value" fields (both Profile 1 and Profile 2), which is part of the NodeInformationTLVs.

8.18.4 Transmissions of MAPs to guarantee inter-bandplan-OFB interoperability

To facilitate operation of domains with devices of different bandplansOFBs, the MAP PHY frames are transmitted using transmission parameters as specified in clause 7.1.2.3.2.1.10/10.1.2.3.2.1.10 of [ITU-T G.9960] and clause 8.8.2.

8.18.5 Inter-bandplan-OFB payload transmissions

When using run-time BATs for the communication between nodes, the following rules shall be followed by the transmitter:

- Both transmitter and receiver are able to support Profile 1 OFBs only: In this case, the transmitter shall use a Profile 1 OFB.
- Combination of nodes supporting Profile 2 and nodes just supporting Profile 1: In this case, transmission is done using a framing understood by both transmitter and receiver (Profile 1 transmission).
- Both transmitter and receiver are able to support Profile 2 OFBs: In this case, transmission may be done using a Profile 2 OFB.

Once the OFB is decided, in order for two nodes to communicate with each other using runtime BATs, the following rules shall apply:

- A receiver performing channel estimation with a transmitter shall consider its own bandplan frequency information (namely the StartSubCarrier & StopSubCarrier) and that of the transmitter. More specifically, the range of sub-carriers of the BAT sent in the CE_ParamUpdate.req message shall be within the intersection of the sub-carrier ranges determined by the StartSubCarrier & StopSubCarrier of both the receiver and transmitter;
- A transmitter sending BAT information to receivers belonging to a PHY multicast group, shall consider its own bandplan frequency information (namely the StartSubCarrier & StopSubCarrier) and that of the receivers of the PHY multicast group. More specifically, the range of sub-carriers of the BAT sent in the MC_GrpInfoUpdate.ind message shall be within the intersection of the sub-carrier ranges determined by the StartSubCarrier & StopSubCarrier of both the transmitter and all receivers in the PHY multicast group.

When using RCM transmission for communication between nodes, the following rules shall be followed by the transmitter:

- When the transmitter knows the capabilities of all the receiver nodes: The transmitter shall send the frame in any RCM mode that the receiver(s) can understand (e.g., using the lowest OFB among the ones supported by the receiver(s) and the transmitter).
- When the destination node(s) is(are) unknown: The frame should be sent using the lowest OFB supported by all nodes in the domain.

~~For MAP transmission see clause 8.18.4. For other transmissions using pre-defined BATs the transmitter may choose either to:~~

- ~~• Transmit using the lowest bandplan between the receiver and transmitter bandplans; or~~
- When using Profile 1 OFBs, the transmitter may also transmit using its own bandplan-OFB using a sufficient repetition factor (e.g., a node of bandplan-OFB_100 MHz can transmit a frame using 100 MHz bandplanOFB with repetition factor of 2, to be received by a node of 50 MHz bandplanOFB).

NOTE – Whenever the transmitter is of a higher bandplanOFB than that of the receiver, transmissions outside the frequency band supported by the receiver (i.e., transmitting power on sub-carriers higher than the receiver's StopSubCarrier) might cause degradation of the receiver's performance in some cases depending on the receiver implementation. It is therefore recommended, if the transmitter's implementation allows it, to not output any power on the sub-carriers outside the frequency band supported by the receiver (i.e., on sub-carriers beyond the receiver's StopSubCarrier).

8.19 Node information and capabilities exchange

Each node that enters the domain shall inform its domain master about the features it implements and the capabilities it supports. It shall include at least one NodeInformationTLV (see Table 8-16.1) in the ADM_NodeRegistrRequest.req message. The first NodeInformationTLV included in the message shall correspond to ITU-T information (see Table 8-16.2). In addition, if a node's features or capabilities change (i.e., any of the fields of any of its NodeInformationTLVs change) during its operation, it shall inform its DM via the TM_NodeTopologyChange.ind message, by including the relevant NodeInformationTLVs.

The DM informs registering nodes about the domain capabilities via a NodeInformationTLV by using a MAP auxiliary Domain information sub-field (see Table 8-74).

The DM shall inform the rest of the nodes in its domain whenever it receives new capability information via a NodeInformationTLV from a node by using the TM_DomainRoutingChange.ind message.

8.20 Metrics acquisition

The goal of the metrics acquisition mechanism is to provide a node a way to calculate the throughput metrics between itself and a destination node without the need to establish a flow.

A node that wants to obtain the achievable throughput with another node of the domain may request metric information to the DM by sending the message MA_AcquireMetrics.req, which includes the information on the destination node.

Upon reception of MA_AcquireMetrics.req, the DM shall respond to the requesting node with an MA_AcquireMetrics.cnf message.

8.20.1 Metrics acquisition protocol messages

8.20.1.1 Format of MA_AcquireMetrics.req

This message is sent from a node requesting an update of metrics information to the DM.

The format of the MMPL of the MA_AcquireMetrics.req shall be as shown in Table 8-125.

Table 8-125 – Format of the MMPL of the MA_AcquireMetrics.req message

Field	Octet	Bits	Description
OriginDeviceID	0	[7:0]	DEVICE_ID of the node origin of the link for which the metrics are requested
DestinationDeviceID	1	[7:0]	DEVICE_ID of the node destination of the link for which the metrics are requested

8.20.1.2 Format of MA_AcquireMetrics.cnf

This message is sent by the DM to inform a requesting node about the metrics of a particular link.

The format of the MMPL of the MA_AcquireMetrics.cnf shall be as shown in Table 8-126.

Table 8-126 – Format of the MMPL of the MA_AcquireMetrics.cnf message

Field	Octet	Bits	Description
OriginDeviceID	0	[7:0]	DEVICE_ID of the node origin of the link for which the metrics are requested
DestinationDeviceID	1	[7:0]	DEVICE_ID of the node destination of the link for which the metrics are requested

Table 8-126 – Format of the MMPL of the MA_AcquireMetrics.cnf message

Field	Octet	Bits	Description
MetricsRouteList	Variable	See Table 8-127	Routing list from OriginDeviceID toward DestinationDeviceID. This field is only present when there is at least one relay node ($N \geq 1$)
MaxBitsPerSecond	Variable	[15:0]	Maximum data rate in bits per second from OriginDeviceID to DestinationDeviceID. (Note 1)
AttainableBitsPerSecond	Variable	[15:0]	The attainable data rate in bits per second from OriginDeviceID to DestinationDeviceID that a DM can allocate for a flow between these two nodes under current domain traffic conditions.

NOTE 1 – This value is calculated from the values reported to the DM by each of the nodes of the domain in the BitsPerSecond field of the Visibility_List field of the TM_NodeTopologyChange.ind message (See Table 8-48) using the following formula: $MaxBitsPerSecond = \frac{1}{\sum_{i=1}^{N+1} \frac{1}{BitsPerSecond_i}}$ where N is the number of relays between OriginDeviceID and DestinationDeviceID.

Table 8-127 – Format of MetricsRouteList

Field	Octet	Bits	Description
NumRelays	0	[7:0]	Number of relay nodes (N) in the MetricsRouteList
RelayNode[1]	1	[7:0]	DEVICE_ID of the first relay node in the list
...
RelayNode[N]	N	[7:0]	DEVICE_ID of the Nth relay node in the list
LinkMetrics[1]	N + 1 and N + 2	[15:0]	PHY data rate in bits per second from the originating node in the route to the first relay. (Note 1)
....
LinkMetrics[N+1]	3*N + 1 and 3*N + 2	[15:0]	PHY data rate in bits per second from the Nth relay in the route to the destination node. (Note 1)

NOTE 1 – This value corresponds to the value reported to the DM by each of the nodes of the domain in the BitsPerSecond field of the Visibility_List field of the TM_NodeTopologyChange.ind message (See Table 8-48)

8.21 Operation in power saving modes

Five modes of operation are defined with the intention of reducing the total power consumption in home networks. When switched on, a node shall operate in one of the following modes:

- Normal mode (L0): This is the mode in which transmission up to the maximum data rate is possible;
- Reduced-power mode (L1): In this mode, power consumption is reduced by limiting medium access for transmission and reception only to a portion of a MAC cycle (see short inactivity schedule);
- Low-power mode (L2): In this mode, power consumption is reduced by suppressing medium access over multiple MAC cycles up to 36000 MAC cycles. Only a limited data rate is supported (see long inactivity schedule);

- Standby mode (L3): In this mode, power consumption is reduced by suppressing medium access over an undetermined multiple MAC cycles. Only a limited data rate is supported (see long inactivity schedule). In this mode, the node can be awakened by any node of the domain or by new transmissions through its A interface;
- Idle mode (L4): In this mode, power consumption is minimized by suppressing any activity related to the domain. No data or control messages are transmitted or received. A node may request to enter this mode when no data can be transmitted/received through its A interface for a vendor-discretionary period of time.

During L0 to L3 power saving modes and transitions between them, the node shall maintain its original DEVICE_ID.

8.21.1 Power saving mode transitions

ITU-T G.hn nodes shall be capable of operating in normal power saving mode (L0). In addition, ITU-T G.hn nodes may be capable of operating in one or more of the other power saving modes defined in clause 8.21 (L1, L2, L3, and L4).

Figure 8-63 shows the allowed transitions between node power saving modes.

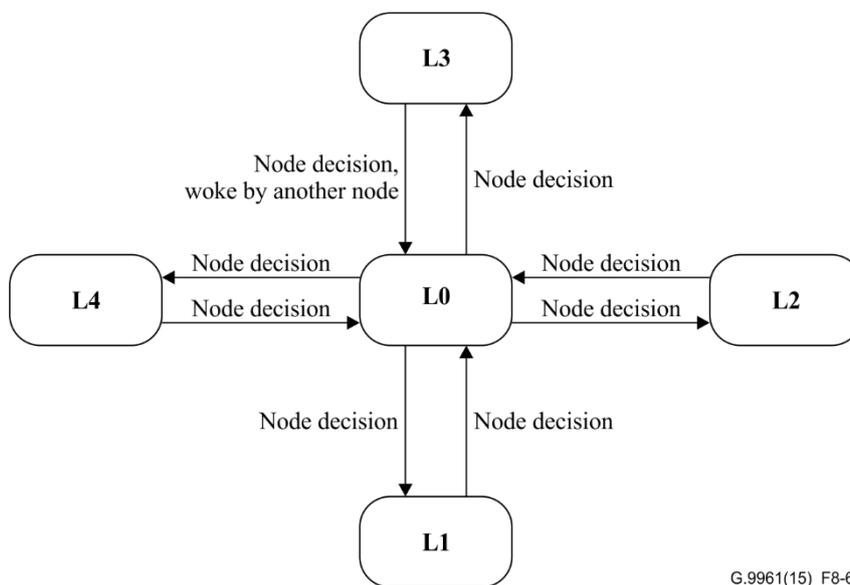


Figure 8-63 – Allowed transitions for node power saving modes

Transitions are only allowed between L0 (normal power saving mode) and each of the other power saving modes. When a node leaves one of the power saving modes, it shall come back to L0 mode.

The transition between L0 and any other power saving modes may triggered by the node requesting the power saving mode.

The transition between L1 power saving mode and L0 power saving mode may be triggered by the node in L1 power saving mode

The transition between L2 power saving mode and L0 power saving mode may be triggered by the node in L2 power saving mode

The transition between L3 power saving mode and L0 power saving mode may be triggered by the node in L3 power saving mode or upon request from a different node in the domain.

The transition between L4 power saving mode and L0 power saving mode may be triggered by the node in L4 power saving mode

Direct transitions between node power saving modes are for further study.

8.21.2 Operation in L1 power saving mode

A node may request the domain master to enter L1 power saving mode through the short inactivity schedule mechanism (see clause 8.3.6.2), indicating the desired periods of activity/inactivity.

A DM that receives a request for transition to L1 power saving mode shall analyse the impact of such request on the topology of the domain (e.g., relays, MAP relays). If no impact is expected, it shall grant the transition by sending a positive answer to the requesting node. If the DM foresees an impact on the topology, it may reject the request and inform the requesting node about the reason.

The mechanism through which the DM assesses the impact of the request on the topology of the domain is vendor discretionary.

The schedule for nodes in the L1 power saving mode should synchronize with each other. That is, every node in L1 power saving mode should have the same inactive periods within a MAC cycle. This may be realized by using the default schedule for all nodes in L1 power saving mode sent in the power saving auxiliary information field of the MAP.

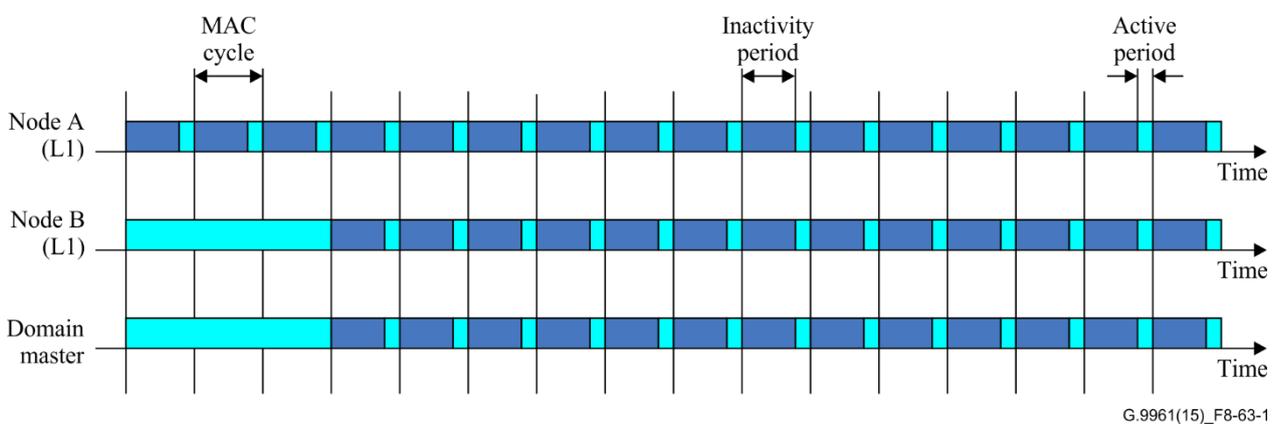


Figure 8-63.1 – Example of domain with nodes in L1 power saving mode

The default schedule provided by the domain master is vendor discretionary. The default L1 schedule shall be published in the power saving auxiliary information field of MAP-Ds used for registration. The DM should periodically announce the default schedule through a power saving auxiliary information field of the MAP-As.

As an alternative, after analysing the request, the domain master may also send a positive answer to the requesting mode indicating that the proposed schedule has been replaced by a modified one that takes into account the power-down requests from the rest of the nodes and any additional constraints on the scheduling of the domain (e.g., NDIM).

A node in L1 power saving mode shall follow the behaviour described in clause 8.3.6 (scheduled inactivity)

A node in L1 power saving mode may terminate the low power state and return to L0 power saving mode using the mechanisms described in the short inactivity schedule mechanism.

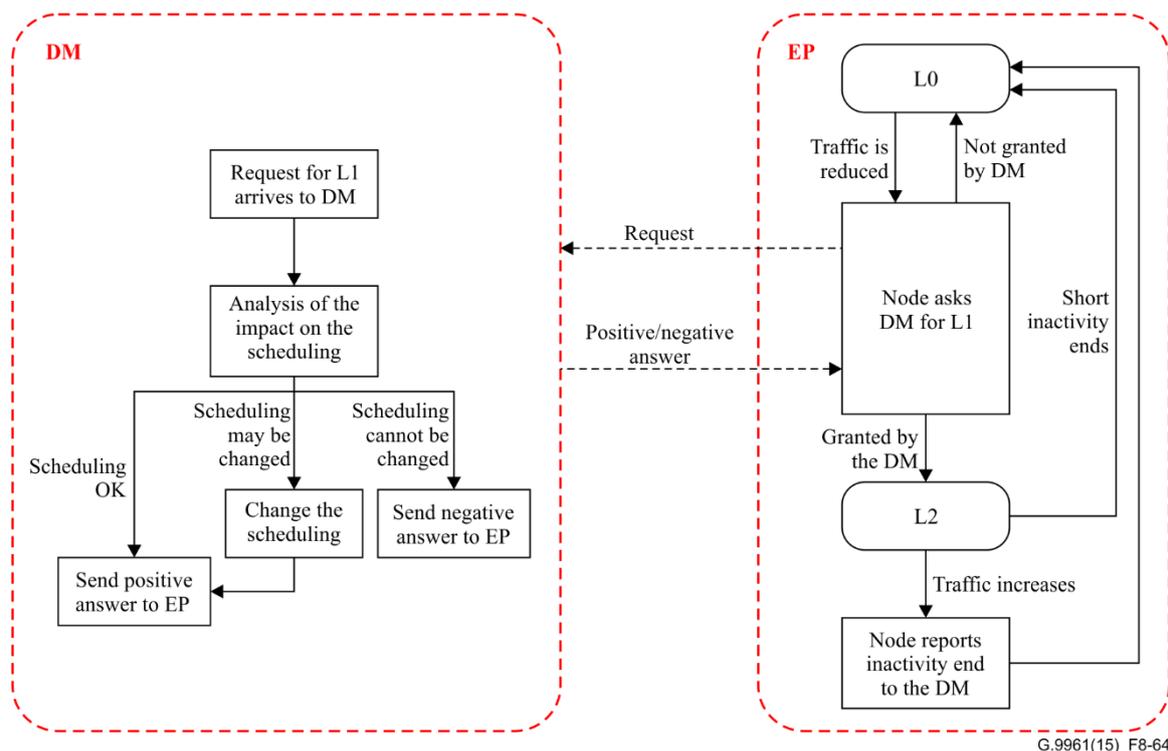


Figure 8-64 – Operation in L1 power saving mode

8.21.3 Operation in L2 power saving mode

A node may request the domain master to enter L2 power saving mode after a reduction of traffic through its A interface through the long inactivity schedule mechanism (see clause 8.3.6.2), indicating the type of long inactivity.

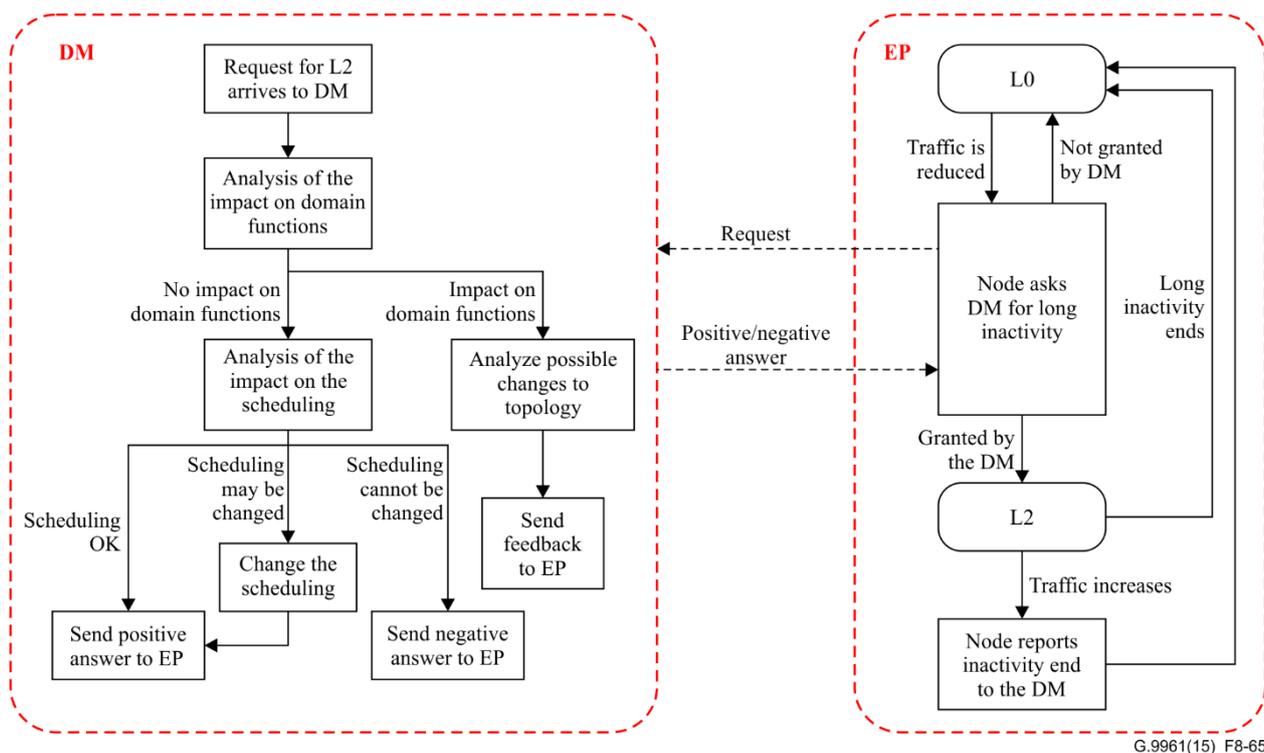


Figure 8-65 – Operation in L2 power saving mode

The schedule for nodes in the L2 power saving mode should synchronize with each other such that they have at least one common active period during the maximum sleep cycle. In this situation, any two nodes in L2 mode should have the same sleep cycle, or one have the sleep cycle multiple of the sleep cycle of the other node. This can be realized by selecting sleep cycles that are 2^k multiples (i.e., 1^* , 2^* , 4^* , 8^* , ...) of the default sleep cycle for L2 power saving mode.

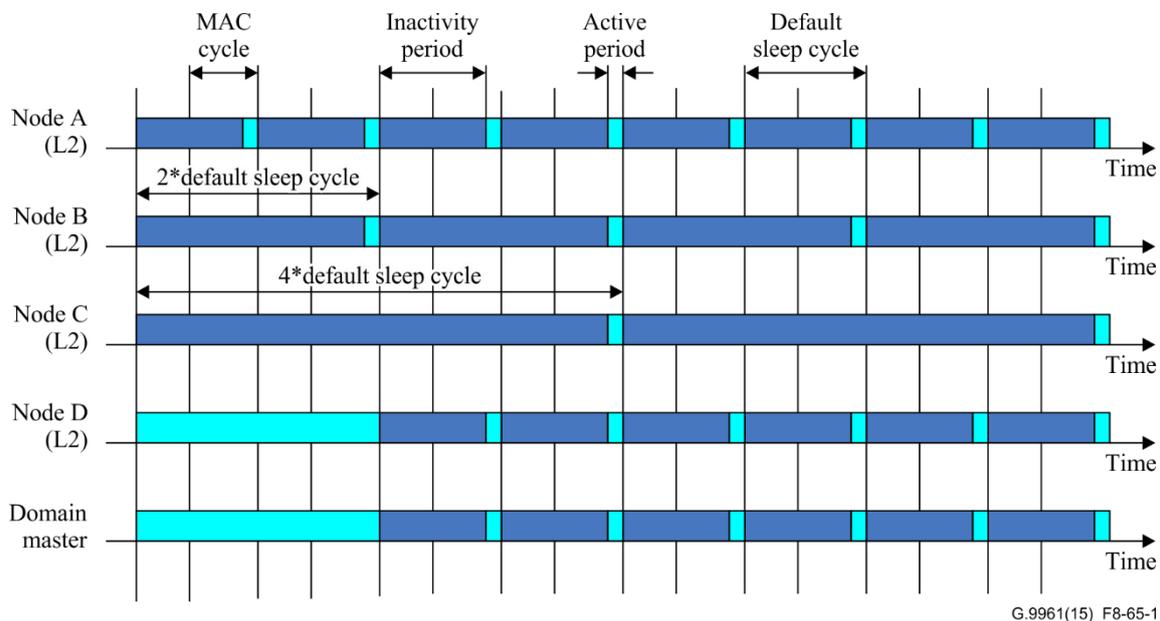


Figure 8-65.1 – Example of domain with nodes in L2 power saving mode

The default schedule provided by the domain master is vendor discretionary. The default L2 schedule shall be published in the power saving auxiliary information field of MAP-Ds used for registration. The DM should periodically announce the default schedule through a power saving auxiliary information field of the MAP-As.

Upon reception of a long inactivity scheduling request by a node, the DM shall analyse the impact of such request on the topology of the domain (i.e., if the basic functions would still be guaranteed during the inactivity of the node).

The mechanism through which the DM assesses the impact of the request on the topology of the domain is vendor discretionary.

Two possible results can be obtained after the analysis:

- **No impact on the domain functions:** The inactivity of the requesting node is transparent for the rest of the nodes and all the domain functions are guaranteed. In this case, DM shall proceed with the following analysis;
- **Impact on the domain functions:** If the domain functions cannot be guaranteed when the requesting node is inactive the DM shall reject the request. However, if by changing the topology of the domain the functions could be guaranteed (e.g., a DM handover, etc..), the DM may indicate it in the rejection code and perform these actions in order to prepare for a new request from the requesting node.

The domain functions that shall be guaranteed are:

- Registration of new nodes;
- MAP relaying;
- SC;
- DM functionality;

- Relaying;
- NDIM with a neighbouring domain.

If no impact on the domain functions is expected, the DM shall analyse the impact of the request on the overall scheduling of the domain and try to group inactivity scheduling

If an impact on the domain functions is expected, the DM may:

- Reject the request and indicate the reason to the requesting node if the node entering power down might disturb the scheduling dramatically;
- Adapt the scheduling to the request by reorganizing transmissions within the domain.

8.21.4 Operation in L3 power saving mode

A node may transition to L3 power saving mode when it detects that no traffic is flowing across its A interface. For this, the node shall request a long inactivity scheduling (see clause 8.3.6.1.1) for an unlimited time to its DM specifying the type of long inactivity.

A node may transition to L3 power saving mode if:

- The node has no data available for transmitting but data can be available again unexpectedly in a short timeframe (A-interface active but with no traffic);
- Expects no data from the ITU-T G.hn medium, but it has to be available for receiving in a short timeframe.

During the operation in L3 power saving mode, the node shall be able to maintain synchronization and decode MAPs sent by the DM. When a node of the domain is operating in L3 power saving mode, the DM should not change the position of the MAP-A.

A node in L3 power saving mode can be awakened by any other node of the domain if this last node has data to transmit. A node may wake up another node in L3 power saving mode by requesting it to the DM through the message LP_RequestWakeUp.req. DM shall wake up the node in L3 power saving mode by means of the power saving update auxiliary information field of the MAP. DM shall include for at least 4 MAC cycles the information necessary to wake up a node in L3 in the power saving update auxiliary information field.

NOTE 1 – The requesting node may monitor the status of the request through the contents of the MAP.

NOTE 2 – In order to be able to be awakened, the node has to be able to decode MAP-s in a particular region of the MAC cycle, decided by the DM. (in this sense, it is similar to L1 power saving mode, but only decoding MAPs and not any kind of frames). The node in L3 power saving mode needs to be able to decode at least one MAP over 4 MAC cycles.

A node may leave L3 mode at any moment and transition to L0 power saving mode when traffic appears again over its A-interface (e.g., Ethernet port activity).

A domain master shall not be able to enter L3 power saving mode since it will not be able to maintain synchronization of the domain.

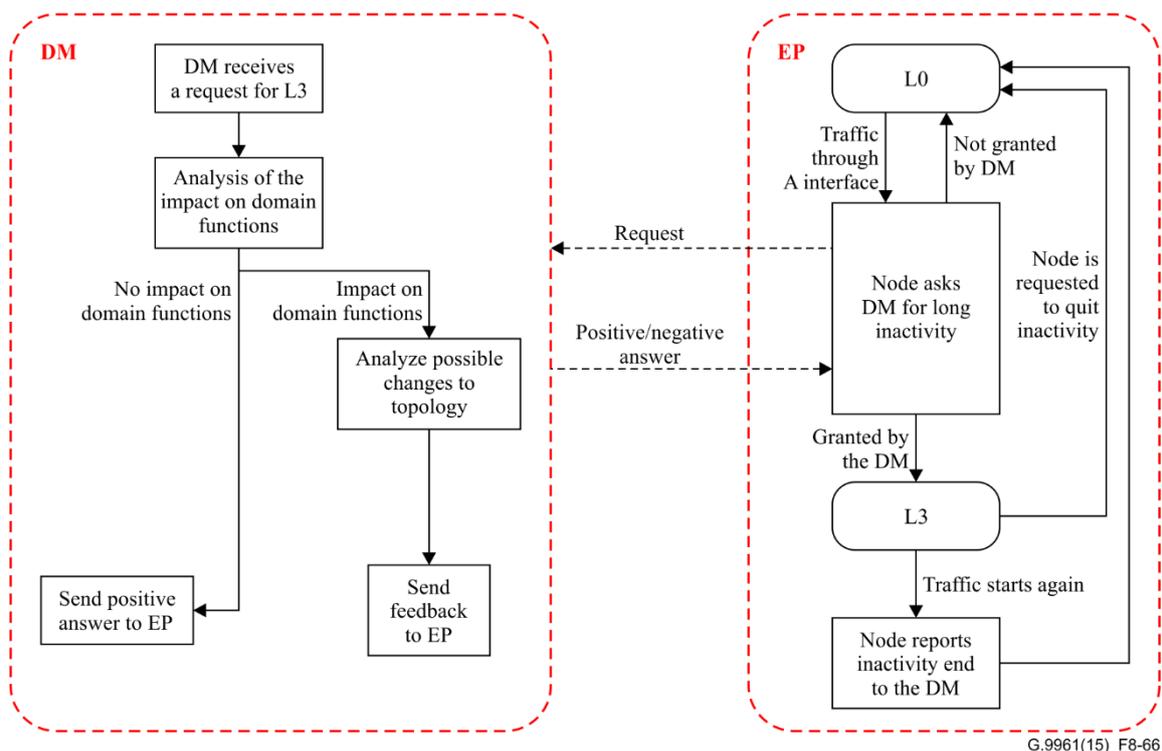


Figure 8-66 – Operation in L3 power saving mode

8.21.5 Operation in L4 power saving mode

A node may transition to L4 power saving mode when it detects that its A interface is inactive (i.e., no physical possibility to transmit/receive information). For this, it shall request to resign from the domain, indicating that the reason for such resignation is the transition to L4 power saving mode. Unlike normal resignation, the node shall not transition to L4 if it is not authorized by the DM (in order to maintain domain functions).

When a node moves to L4 power saving mode, it loses its original DEVICE_ID.

Upon reception of a resignation request by a node, the DM shall analyse the impact of such request on the topology of the domain (i.e., if the basic functions would still be guaranteed during the inactivity of the node).

Two possible results can be obtained after the analysis:

- **No impact on the domain functions:** The inactivity of the requesting node is transparent for the rest of the nodes and all the domain functions are guaranteed. In this case, DM shall grant the transition to L4 power saving mode;
- **Impact on the domain functions:** If the domain functions cannot be guaranteed when the requesting node is inactive the DM shall reject the request. However, if by changing the topology of the domain the functions could be guaranteed (e.g., a DM handover), the DM may indicate it in the rejection code and perform these actions in order to prepare for a new request from the requesting node.

The domain functions that shall be guaranteed are:

- Registration of new nodes
- MAP relaying
- SC
- DM functionality
- Relaying
- NDIM with a neighbouring domain

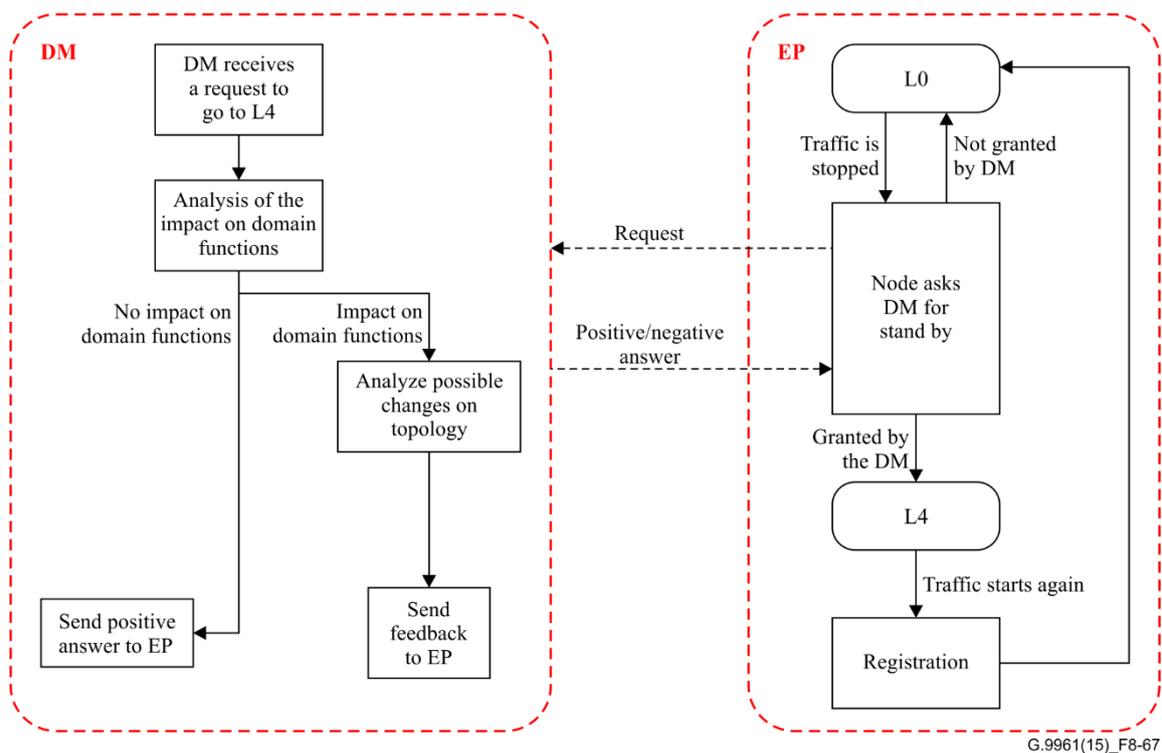


Figure 8-67 – Operation in L4 power saving mode

8.21.6 Domain master in power saving mode

A domain master may transition to L1 power saving mode if all other nodes in the domain are in L1 power saving mode.

A domain master may transition to L2 power saving mode if all other nodes in the domain are in L2 power saving mode.

The domain master in power saving modes L1 and L2 should guarantee the following functions: Registration of new nodes, MAP relaying, SC, DM functionality, Relaying, NDIM with a neighbouring domain.

NOTE – A domain master should transition to L1 or L2 power saving mode whenever possible.

8.22 Layer 2 Configuration and Management Protocol (LCMP)

8.22.1 LCMP behaviour description

LCMP protocol allows an external application entity (AE) to interact with the management of one or more ITU-T G.hn nodes in order to read/write parameters internal to the node.

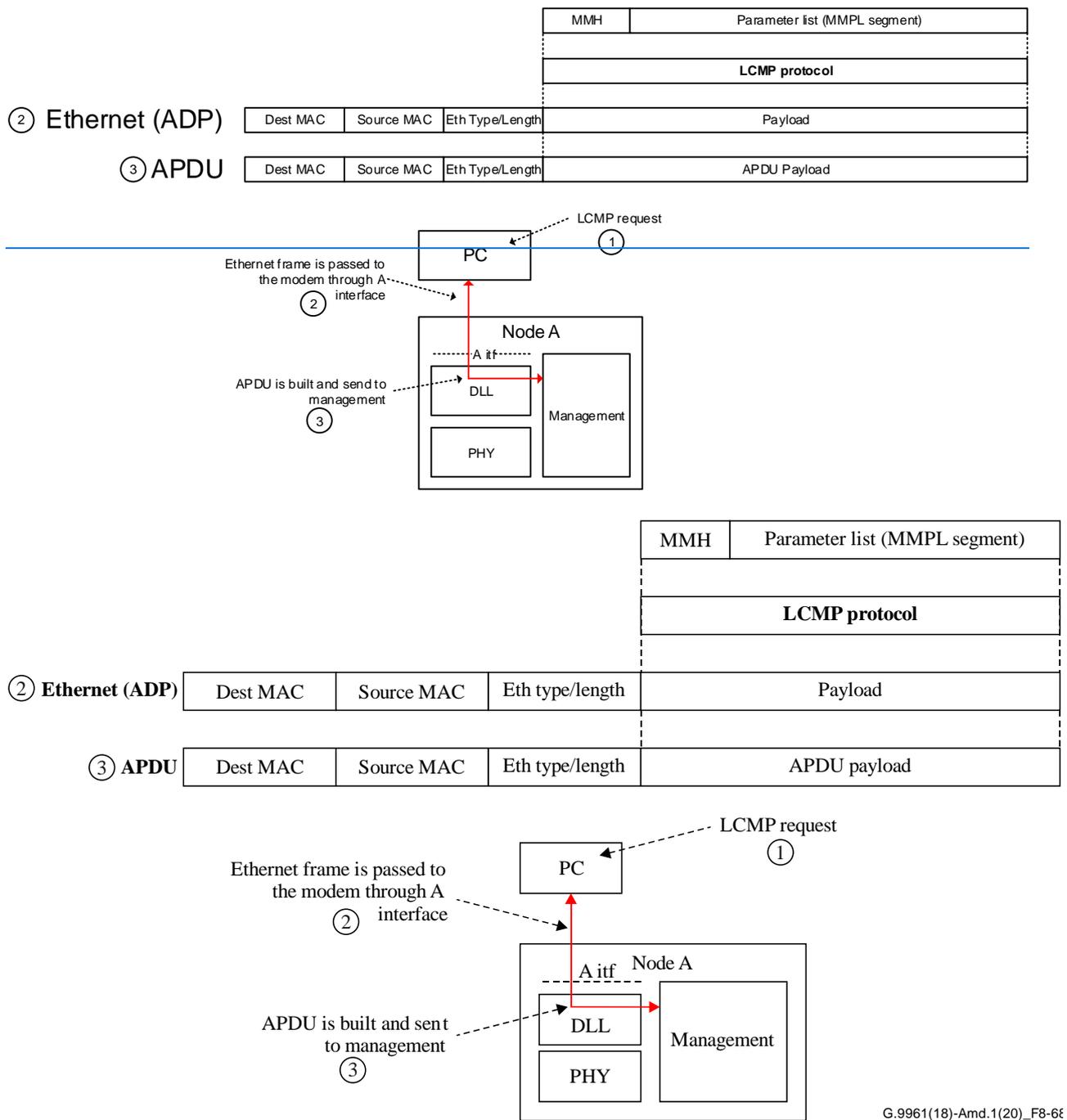
NOTE 1 – The use of LCMP between management entities of nodes of a domain is for further study.

LCMP management messages shall be mapped into *in-band management messages* to implement the protocol (i.e., Encapsulate management information within ADP/APDUs):

- Local LCMP messages (i.e., messages that are used to communicate from an external AE to an ITU-T G.hn node through its A interface without going through the ITU-T G.hn medium) are mapped into local in-band management messages;
- Remote LCMP messages (i.e., messages that are used to communicate from an external AE to an ITU-T G.hn node through the ITU-T G.hn medium) are mapped into remote in-band management messages.

NOTE 2 – An LCMP message follows the same format as any other management message (see clause 8.10.1). It is therefore composed of an MMH and an MMPL segment.

LCMP messages are embedded into in-band management messages in order to establish a link between an external AE (e.g., PC) and the modems (either directly or remotely through the ITU-T G.hn medium). For this, LCMP messages shall be embedded in an Ethernet frame (ADP) as a normal data packet and, after that, into an APDU that can be treated as an in-band management message. This process is described in Figure 8-68, where a local LCMP message is depicted.



G.9961(18)-Amd.1(20)_F8-6f

Figure 8-68 – Example of encapsulation of LCMP messages into ITU-T G.hn in-band management messages

8.22.2 Transmission scheme

LCMP messages use the same addressing scheme as the rest of the in-band management messages unless additionally restricted by the message type (e.g., limitations to the use of broadcast).

In case of unicast transmission, the Destination MAC of the Ethernet frame conveying the LCMP message shall be set to the MAC address of the destination node.

In case of transmissions addressed to all ITU-T G.hn nodes (i.e., G.hn broadcast), the Destination MAC of the Ethernet frame conveying the LCMP message shall be set to the broadcast address.

In all LCMP messages, including G.hn broadcast messages, EtherType shall be set to 22E3₁₆.

8.22.3 LCMP messages

Since LCMP is mapped into ITU-T G.hn in-band management messages, it makes use of ITU-T G.hn management messages.

LCMP uses the same transmission mechanism as any other management message (e.g., fragmentation).

LCMP defines ten new management messages:

- A READ operation composed of
 - A "READ" message (**LCMP_Read.req**) asking a particular ITU-T G.hn node for the value of one or more particular parameters specified in the request;
 - A "READ confirmation" message (**LCMP_Read.cnf**) used by an ITU-T G.hn node to answer a LCMP_Read request from an entity outside the domain.
- A WRITE operation composed of
 - A "WRITE" message (**LCMP_Write.req**) asking a particular ITU-T G.hn node to use the value sent within the message for one or more particular parameters specified in the message;
 - A "WRITE confirmation" message (**LCMP_Write.cnf**) used by an ITU-T G.hn node to confirm an LCMP_Write request.
- A CONTROL operation composed of
 - A "CTRL" message (**LCMP_Ctrl.req**) asking a particular ITU-T G.hn node to use the value sent within the message to execute one or more specific actions of the modem;
 - A "CTRL confirmation" message (**LCMP_Ctrl.cnf**) used by an ITU-T G.hn node to confirm an LCMP_Ctrl request.
- A NOTIFY operation composed of
 - A "NOTIFY" message (**LCMP_Notify.ind**) used to notify the receiving entity of an action that occurred in the transmitter of the message;
 - An "NOTIFY response" message (**LCMP_Notify.rsp**) used by an ITU-T G.hn node to confirm the reception of a NOTIFY indication.
- A LCMP operation composed of
 - An indication message (LCMP.ind). The contents of this message are for further study;
 - A response message (LCMP.rsp). The contents of this message are for further study.

8.22.3.1 LCMP READ request message (LCMP_Read.req)

The LCMP_Read.req message is a unicast management message sent by an application entity to a node or an ITU-T G.hn broadcast management message sent to one or more nodes. It is intended to be used for reading parameters from the management entity of a node. The format of the MMPL of the LCMP_Read.req message shall be as shown in Table 8-128.

Table 8-128 – Format of the MMPL of the LCMP_Read.req message

Field	Octet	Bits	Description
LCMP_CONTROL	0	[7:0]	LCMP Control octet, formatted as described in Table 8-129
AE_MAC	1 to 6	[47:0]	MAC address of the AE that is requesting to read the parameters listed in this message. If the origin of the message is a DLL management entity of a node, it shall be set to the REGID of the requesting node.
LCMPLength	7 and 8	[15:0]	Length in bytes of the LCMPValue field
Transaction ID	9 and 10	[15:0]	Transaction ID of the LCMP_Read.req message. This field shall be initialized to 0000 ₁₆ for the first message and shall be incremented by 0001 ₁₆ for each new message. It shall be expressed as a 16-bit unsigned integer and shall wrap around (goes back to value 0000 ₁₆ after FFFF ₁₆).
LCMPValue	Variable	See Annex C of [ITU-T G.9962] Variable	The contents of this field depends on the type of request indicated in LCMP_CONTROL field (see Table 8-129) and shall be filled according to Annex C of [ITU-T G.9962].

Table 8-129 – Format of the LCMP Control Field

Field	Octet	Bits	Description
RequestType	0	[3:0]	of request. 0 – Communication through L1 interface (see clause 6.1 of [ITU-T G.9962] and Annex A of [ITU-T G.9962]) 1 ₁₆ to 4 ₁₆ – Reserved for HGF Data models 5 ₁₆ – Communication through L6 interface (see clause 6.1 of [ITU-T G.9962] and Annex B of [ITU-T G.9962]) All other values of this field are reserved by ITU-T.
Reserved		[7:4]	Reserved by ITU-T (Note 1)
NOTE 1 – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

8.22.3.2 LCMP READ confirmation message (LCMP_Read.cnf)

The LCMP_Read.cnf message is a unicast management message sent by a node to an external AE, and is intended to be used for reading parameters from the management entity of the node. The format of the MMPL of the LCMP_Read.cnf message shall be as shown in Table 8-130.

Table 8-130 – Format of the MMPL of the LCMP_Read.cnf message

Field	Octet	Bits	Description
LCMP_CONTROL	0	[7:0]	LCMP Control octet, formatted as described in Table 8-129
LCMPLength	1 and 2	[15:0]	Length in bytes of the LCMPValue field
Transaction ID	3 and 4	[15:0]	Transaction ID of the LCMP_Read.req message corresponding to this confirmation message.
RequestsType	Variable	VariableSee Table 8-129	The contents of this field depends on the type of request indicated in LCMP_CONTROL field (See Table 8-129)

8.22.3.3 LCMP WRITE request message (LCMP_Write.req)

The LCMP_Write.req message is a unicast or multicast management message sent by an application entity to a node or an ITU-T G.hn broadcast management message sent to one or more nodes. It is intended to be used for writing parameters to the management entity of a node. The format of the MMPL of the LCMP_Write.req message shall be as shown in Table 8-131.

Table 8-131 – Format of the MMPL of the LCMP_Write.req message

Field	Octet	Bits	Description
LCMP_CONTROL	0	[7:0]	LCMP Control octet, formatted as described in Table 8-129
AE_MAC	1 to 6	[47:0]	MAC address of the AE that is requesting to write the parameters listed in this message. If the origin of the message is a DLL management entity of a node, it shall be set to the REGID of the requesting node.
LCMPLength	7 and 8	[15:0]	Length in bytes of the LCMPValue field
Transaction ID	9 and 10	[15:0]	Transaction ID of the LCMP_Read.req message corresponding to this confirmation message.
LCMPValue	Variable	See Annex C of [ITU-T G.9962] Variable	The contents of this field depends on the type of request indicated in LCMP_CONTROL field (see Table 8-129) and shall be filled according to Annex C of [ITU-T G.9962].

8.22.3.4 LCMP WRITE confirmation message (LCMP_Write.cnf)

The LCMP_Write.cnf message is a unicast management message sent by a node to an external AE, and is intended to be used for confirming the writing of parameters to the management entity of the node. The format of the MMPL of the LCMP_Write.cnf message shall be as shown in Table 8-132.

Table 8-132 –Format of the MMPL of the LCMP_Write.cnf message

Field	Octet	Bits	Description
LCMP_CONTROL	0	[7:0]	LCMP Control octet, formatted as described in Table 8-129
LCMPLength	1 and 2	[15:0]	Length in bytes of the LCMPValue field
Transaction ID	3 and 4	[15:0]	Transaction ID of the LCMP_Read.req message corresponding to this confirmation message.
LCMPValue	Variable	See Annex C of [ITU-T G.9962] Variable	The contents of this field depends on the type of request indicated in LCMP_CONTROL field (see Table 8-129) and shall be filled according to Annex C of [ITU-T G.9962].

8.22.3.5 LCMP_Ctrl request message (LCMP_Ctrl.req)

The LCMP_Ctrl.req message is a unicast management message sent by an application entity to a node or an ITU-T G.hn broadcast management message sent to one or more nodes. It is intended to be used for writing a control action to the management entity of the node. The format of the MMPL of the LCMP_Ctrl.req message shall be as shown in Table 8-133.

Table 8-133 – Format of the MMPL of the LCMP_Ctrl.req message

Field	Octet	Bits	Description
LCMP_CONTROL	0	[7:0]	LCMP Control octet, formatted as described in Table 8-129
AE_MAC	1 to 6	[47:0]	MAC address of the AE that is requesting to write the parameters listed in this message. If the origin of the message is a DLL management entity of a node, it shall be set to the REGID of the requesting node.
LCMPLength	7 and 8	[15:0]	Length in bytes of the LCMPValue field
Transaction ID	9 and 10	[15:0]	Transaction ID of the LCMP_Read.req message corresponding to this confirmation message.
LCMPValue	Variable	See Annex C of [ITU-T G.9962] Variable	The contents of this field depends on the type of request indicated in LCMP_CONTROL field (see Table 8-129) and shall be filled according to Annex C of [ITU-T G.9962].

8.22.3.6 LCMP_Ctrl confirmation message (LCMP_Ctrl.cnf)

The LCMP_Ctrl.cnf message is a unicast management message sent by a node to an external AE, and is intended to be used for confirming an action sent to the management entity of the node. The format of the MMPL of the LCMP_Ctrl.cnf message shall be as shown in Table 8-134.

Table 8-134 – Format of the MMPL of the LCMP_Ctrl.cnf message

Field	Octet	Bits	Description
LCMP_CONTROL	0	[7:0]	LCMP Control octet, formatted as described in Table 8-129
LCMPLength	1 and 2	[15:0]	Length in bytes of the LCMPValue field
Transaction ID	3 and 4	[15:0]	Transaction ID of the LCMP_Read.req message corresponding to this confirmation message.
LCMPValue	Variable	Variable See Table 8-129	The contents of this field depends on the type of request indicated in LCMP_CONTROL field (see Table 8-129) and shall be filled according to Annex C of [ITU-T G.9962].

8.22.3.7 LCMP NOTIFY indication message (LCMP_Notify.ind)

The LCMP_Notify.ind message is a unicast or ITU-T G.hn broadcast management message for indicating to one or more nodes an occurrence of an event at the transmitter side management entity. The format of the MMPL of the LCMP_Notify.ind message shall be as shown in Table 8-135.

Table 8-135 – Format of the MMPL of the LCMP_Notify.ind message

Field	Octet	Bits	Description
LCMP_CONTROL	0	[7:0]	LCMP Control octet, formatted as described in Table 8-129
LCMPLength	1 and 2	[15:0]	Length in bytes of the LCMPValue field
Transaction ID	3 and 4	[15:0]	Transaction ID of the LCMP_Read.req message corresponding to this confirmation message.
NotificationAck	5	[0]	If set to one, this notification message shall be acknowledged by the receiver of the management message by sending an LCMP_Notify.cnf. Zero otherwise.
Reserved		[7:1]	Reserved by ITU (Note 1)
LCMPValue	Variable	See Annex C of [ITU-T G.9962]Variable	The contents of this field depends on the type of request indicated in LCMP_CONTROL field (see Table 8-129) and shall be filled according to Annex C of [ITU-T G.9962].

NOTE 1 – Bits that are reserved by ITU shall be set to zero by the transmitter and ignored by the receiver.

8.22.3.8 LCMP NOTIFY response message (LCMP_Notify.rsp)

The LCMP_Notify.rsp message is a unicast management message sent by a node to an external AE, and is intended to be used for confirming the reception of a NOTIFY.ind management message. The format of the MMPL of the LCMP_Notify.rsp message shall be as shown in Table 8-136.

Table 8-136 – Format of the MMPL of the LCMP_Notify.rsp message

Field	Octet	Bits	Description
LCMP_CONTROL	0	[7:0]	LCMP Control octet, formatted as described in Table 8-129
LCMPLength	1 and 2	[15:0]	Length in bytes of the LCMPValue field
Transaction ID	3 and 4	[15:0]	Transaction ID of the LCMP_Read.req message corresponding to this confirmation message.
LCMPValue	Variable	See Annex C of [ITU-T G.9962] Variable	The contents of this field depends on the type of request indicated in LCMP_CONTROL field (see Table 8-129) and shall be filled according to Annex C of [ITU-T G.9962].

8.22.3.9 LCMP indication message (LCMP.ind)

The contents and operation of this message are for further study.

8.22.3.10 LCMP response message (LCMP.rsp)

The contents and operation of this message are for further study.

8.22.4 ITU-T LCMP messages

This section is for further study.

8.23 Run-length compression of fields of management messages

Some of the management messages included in this Recommendation may use a run-length compression mechanism.

The run-length compression mechanism shall only be used when both transmitter of the message and receiver of the message support this capability.

Whenever a field of a management message is compressed using this mechanism it shall be formatted as described in Table 8-137.

Table 8-137 – Format of the compressed field

Field	Octet	Bits	Description
M	0	[3:0]	Number of bits (M) used to code the NumInstances field of each group. M is chosen by the transmitter of the message.
N	0	[7:4]	Number of bits (N) used to code the Pattern field of each group. N is message-dependent.
Group[0]	Variable	Variable	M+N bit long field that includes a compressed group following the format presented in Table 8-138.
...	
Group[i]	M+N bit long field that includes a compressed group following the format presented in Table 8-138

Table 8-138 – Format of a compressed group

Field	Octet	Bits	Description
NumInstances/GroupType	0	Variable	M-bit long field. When this field is set to a non zero value, the GroupData field describes a compressed group. In this case, this field represents the number of consecutive instances of the Pattern field that are being compressed into one group. When this field is set to zero, the GroupData field describes a special group. In this case, this field represents a special behaviour of the compression mechanism.
GroupData	Variable	Variable	When NumInstances/GroupType is set to a non-zero value, this field is filled as described in Table 8-139. When NumInstances/GroupType is set to a zero value, this field is filled as described in Table 8-140.

Table 8-139 – Format of a compressed group

Field	Octet	Bits	Description
Pattern	Variable	Variable	N-bit long field representing the series of patterns being compressed. When uncompressing the field, the Pattern field shall be generated the number of times specified in the NumInstances field. The format of Pattern field is message-specific.

Table 8-140 – Format of a special group

Field	Octet	Bits	Description
EOS/NumUncompressed	Variable	Variable	8 bit long field. When set to a zero value, it indicates the end of the compressed field. When set to a non-zero value (P), it indicates that P groups of N bits indicated in the field UncompressedPattern are not compressed.
Uncompressed Pattern[0]	Variable	Variable	First non-compressed Pattern. This field exists only if EOS/NumUncompressed field is set to a non-zero value.
...
UncompressedPattern[P-1]			(P-1)th non-compressed Pattern This field exists only if EOS/NumUncompressed field to a non-zero value.

Figure 8-69 shows an example of how the different fields of this mechanism are used.

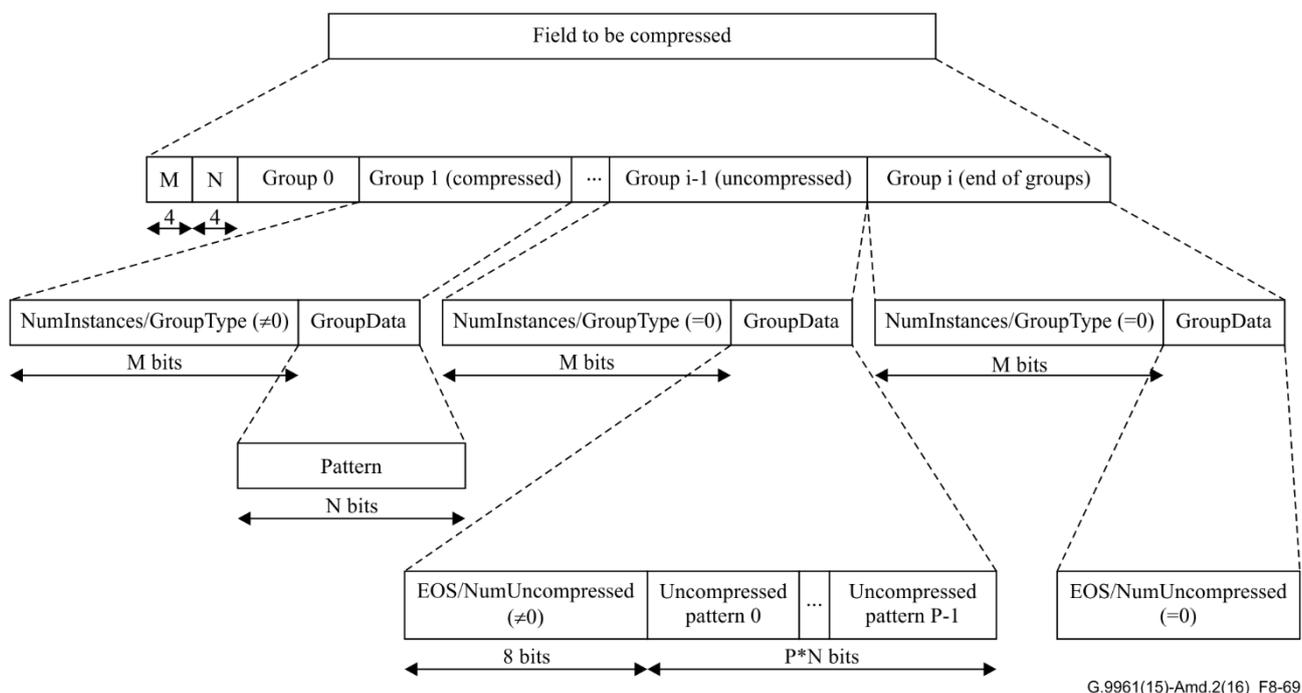


Figure 8-69 – Example of compressing a field of a management message using the run-length compression mechanism

8.24 Payload compression in management messages

When a node detects that the transmitter and receiver in a particular link support the compression of management frames ("MMPL Compression support" capability, see clause 8.6.1.1.4.1), it can compress the payload (MMPL) of a management frame using one of the allowed compression algorithms to create a new compressed version of the MMPL (CMMPL – Compressed MMPL).

The compressed management frame uses the CMMPL instead of the MMPL to generate the MMH information (e.g., Length field) and for segmentation purposes.

8.24.1 Huffman coding of MMPL

When using Huffman coding, the transmitter node shall set the MMPL Compression field of the MMH of management messages to 01₂ (see Table 8-87 in clause 8.10.1).

The procedure to build the Huffman tree shall be as follows:

- Do a pre-order traversal, writing each node visited.
- Differentiate leaf nodes from internal/non-leaf nodes.
- Write a single bit for each node, 1 for leaf and 0 for non-leaf.
- For leaf nodes, write the character stored.
- For non-leaf nodes there's no information that needs to be written, just the bit that indicates there's an internal node.
- Huffman tree and Huffman coded message bits are written from MSB to LSB. The MSB of byte 0 is the first bit of the Huffman tree.

If by the application of the compression algorithm the resulting compressed message length is not byte-aligned (i.e., not a multiple of 8 bits), the compressed message shall include padding bits (set to zero) to complete a whole byte.

NOTE – An implementation of this mechanism can be found at [b-BCL].

8.25 TLV pool

This clause describes the TLVs that may be used by the different management messages exchanged between two nodes.

The general structure of a TLV field is described in Table 8-140.1.

Table 8-140.1 – General structure of a TLV field

<u>Field</u>	<u>Octet</u>	<u>Bits</u>	<u>Description</u>
<u>TLVType</u>	<u>0</u>	<u>[7:0]</u>	<u>Set to the value corresponding to the TLV being transmitted (see Table 8-140.2).</u>
<u>Length</u>	<u>1</u>	<u>[7:0]</u>	<u>Length in bytes of information value field.</u>
<u>Value</u>	<u>Variable</u>	<u>See Table 8-140.2</u>	<u>Value of information field. This field shall be populated with the value field corresponding to the TLV type specified in TLVType (see Table 8-140.2).</u>

Table 8-140.2 – List of TLVs

<u>TLV Type</u>	<u>TLV Name</u>	<u>Value field for the TLV</u>
<u>00₁₆-01₁₆</u>	<u>Reserved</u>	<u>Reserved by ITU-T (See Table 8-16.1)</u>
<u>02₁₆</u>	<u>NodeVisibilityTLV</u>	<u>See Table 8-140.3</u>
<u>03₁₆</u>	<u>PSDShapingTLV</u>	<u>See Table 8-140.4</u>
<u>04₁₆</u>	<u>SMDDescriptorTLV</u>	<u>See Table 8-140.5</u>
<u>05₁₆</u>	<u>BitLoadingTLV</u>	<u>See Table 8-140.6</u>
<u>06₁₆</u>	<u>PartialBitLoadingTLV</u>	<u>See Table 8-140.7</u>
<u>07₁₆-FF₁₆</u>	<u>Reserved</u>	<u>Reserved by ITU-T</u>

8.25.1 TLV Value fields

8.25.1.1 NodeVisibilityTLV

The NodeVisibilityTLV field includes the visibility information reported by a node for a link.

Table 8-140.3 – Format of NodeVisibilityTLV field

<u>Field</u>	<u>Octet</u>	<u>Bits</u>	<u>Description</u>
<u>DeviceID</u>	<u>0</u>	<u>[7:0]</u>	<u>DEVICE ID of a node that the reporting node detected.</u>
<u>BitsPerSecond</u>	<u>1 to 6</u>	<u>[47:0]</u>	<u>Bits [23:0] indicate the PHY data rate in bits per second from the reporting node to the detected node; Bits [47:24] indicate the PHY data rate from the detected node to the reporting node. (Notes 1, 2)</u>
<u>OFB_ID</u>	<u>7</u>	<u>[2:0]</u>	<u>OFB used to estimate the BitsPerSecond field See clause 7/10.1.2.3.2.1.9 of [ITU-T G.9960]</u>
<u>Reserved</u>		<u>[7:3]</u>	<u>Reserved by ITU-T (Note 3)</u>

Table 8-140.3 – Format of Node Visibility TLV field

<u>Field</u>	<u>Octet</u>	<u>Bits</u>	<u>Description</u>
<p><u>NOTE 1 – Both data rates shall follow the formula $\sum_{i=1}^N \text{BitsPerSecond}_i * \frac{T_i}{T_{\text{Cycle}}}$ where N is the number of channel estimation windows in the MAC cycle, BitsPerSecond_i is the result of applying the same formula as Note 2 in Table 8-33 to the i-th channel estimation window, T_i is the duration of the i-th channel estimation window and T_{Cycle} is the duration of the MAC cycle. If no channel estimation is available for a particular window, RCM parameters shall be used for that window. If the data rate with the particular detected node is not available, the value of this parameter shall be set to FFFF_{16}.</u></p> <p><u>NOTE 2 – In case the DM receives conflicting information in the BitsPerSecond field from the Visibility List of two different nodes for a given link, the DM shall take into account the minimum value between both.</u></p> <p><u>NOTE 3 – Bits that are reserved by ITU shall be set to zero by the transmitter and ignored by the receiver.</u></p>			

8.25.1.2 PSDShaping TLV

The PSDShaping TLV field includes information related to a PSD mask.

Table 8-140.4 – Format of PSDShaping TLV field

<u>Field</u>	<u>Octet</u>	<u>Bits</u>	<u>Description</u>
<u>Breakpoints</u>	<u>0</u>	<u>[4:0]</u>	<u>Number of breakpoints (B_p). The valid range of this field is 0 ($B_p=1$) to 31 ($B_p=32$).</u>
<u>Reserved</u>		<u>[7:5]</u>	<u>Reserved by ITU-T (Note 1).</u>
<u>PSDBreakpoint</u>	<u>1 and 2</u>	<u>[15:0]</u>	<u>Sub-carrier index of the first breakpoint being described (Note 2).</u>
<u>PSDLevel</u>	<u>3 and 4</u>	<u>[12:0]</u>	<u>PSD level on this sub-carrier in steps of 0.1dB with an offset of -140 dBm/Hz (Notes 3 and 4).</u>
<u>Reserved</u>		<u>[15:13]</u>	<u>Reserved by ITU-T (Note 1).</u>
<u>...</u>	<u>...</u>	<u>...</u>	<u>...</u>
<u>PSDBreakpoint</u>	<u>Variable</u>	<u>[15:0]</u>	<u>Sub-carrier index of the last breakpoint being described (Note 2).</u>
<u>PSDLevel</u>	<u>Variable</u>	<u>[12:0]</u>	<u>PSD level on this sub-carrier in steps of 0.1dB with an offset of -140 dBm/Hz (Notes 3 and 4).</u>
<u>Reserved</u>		<u>[15:13]</u>	<u>Reserved by ITU-T (Note 1).</u>
<p><u>NOTE 1 – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.</u></p> <p><u>NOTE 2 – The sub-carrier index shall be put in ascending order. The PSD level for the lowest sub-carrier index shall also apply to lower sub-carrier indexes. The PSD level for the highest sub-carrier index shall also apply to higher sub-carrier indexes.</u></p> <p><u>NOTE 3 – The dynamic range of the PSD level specified in this descriptor shall be 30 dB (see clause 7.1.5.3/10.1.5.3 of [ITU-T G.9960]).</u></p> <p><u>NOTE 4 – Example: A value of 320_{16} represents a breakpoint with PSD of $320_{16} \times 0.1 - 140 = -60$ dBm/Hz.</u></p>			

8.25.1.3 SMDescriptorTLV

The SMDescriptorTLV field includes data related to an SM mask.

Table 8-140.5 – Format of SMDescriptorTLV field

<u>Field</u>	<u>Octet</u>	<u>Bits</u>	<u>Description</u>
<u>Breakpoints</u>	<u>0</u>	<u>[4:0]</u>	<u>Number of breakpoints (B_p). The valid range of this field is 0 ($B_p=1$) to 31 ($B_p=32$).</u>
<u>Reserved</u>		<u>[7:5]</u>	<u>Reserved by ITU-T (Note 1).</u>
<u>LowestFreqSubcarrier[0]</u>	<u>1 to 2</u>	<u>[15:0]</u>	<u>Index of the lowest frequency sub-carrier in the first band to be masked.</u>
<u>HighestFreqSubcarrier[0]</u>	<u>3 to 4</u>	<u>[15:0]</u>	<u>Index of the highest frequency sub-carrier in the first band to be masked.</u>
<u>...</u>	<u>...</u>	<u>...</u>	<u>...</u>
<u>LowestFreqSubcarrier[B_p-1]</u>	<u>1 to 2</u>	<u>[15:0]</u>	<u>Index of the lowest frequency sub-carrier in the last band to be masked.</u>
<u>HighestFreqSubcarrier[B_p-1]</u>	<u>3 to 4</u>	<u>[15:0]</u>	<u>Index of the highest frequency sub-carrier in the last band to be masked.</u>
<u>NOTE 1 – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.</u>			

8.25.1.4 BitloadingTLV

The BitloadingTLV field includes data related to the bitloading to be used in a link.

Table 8-140.6 – Format of BitloadingTLV field

<u>Field</u>	<u>Octet</u>	<u>Bits</u>	<u>Description</u>
<u>TIDX_{MIN}</u>	<u>0 to 1</u>	<u>[15:0]</u>	<u>16-bit unsigned integer indicating the lowest sub-carrier index to which non-zero bits are assigned. It shall be an integer multiple of G (Note 1).</u>
<u>TIDX_{MAX}</u>	<u>2 to 3</u>	<u>[15:0]</u>	<u>16-bit unsigned integer indicating the highest sub-carrier index to which non-zero bits are assigned. It shall be an integer multiple of G (Note 1) and if bit-loading grouping is used ($G>1$) shall meet: <u>$TIDX_{MAX}+G-1 \leq \text{StopSubCarrier}$, where StopSubCarrier is specified in Table 8-16.6 ("OFB Info Capability Value field).</u> <u>Let W denote the number of BAT entries, which is $(TIDX_{MAX} - TIDX_{MIN}) / G + 1$.</u> <u>Let Z denote the smallest integer larger than or equal to $W/2$.</u></u>
<u>B₁</u>	<u>2V+11</u>	<u>[3:0]</u>	<u>4-bit unsigned integer indicating the number of bits assigned to sub-carrier indices $TIDX_{MIN}$ to $TIDX_{MIN} + G - 1$ (Note 1).</u> <u>This field only exists if BAT_COMPRESSION field is set to zero.</u>
		<u>[7:4]</u>	<u>4-bit unsigned integer indicating the number of bits assigned to sub-carrier indices $TIDX_{MIN} + G$ to $TIDX_{MIN} + 2G - 1$ (Notes 1, 2, 3).</u>

Table 8-140.6 – Format of BitloadingTLV field

<u>Field</u>	<u>Octet</u>	<u>Bits</u>	<u>Description</u>
			<u>This field only exists if BAT_COMPRESSION field is set to zero.</u>
...
<u>B_Z</u>	<u>2V+10 + Z</u>	<u>[3:0]</u>	<u>4-bit unsigned integer indicating the number of bits assigned to sub-carrier indices TID_X_{MAX} – G to TID_X_{MAX} – 1 if W is even, or to sub-carrier indices TID_X_{MAX} to TID_X_{MAX} + G – 1 if W is odd (Notes 1, 2).</u> <u>This field only exists if BAT_COMPRESSION field is set to zero.</u>
<p><u>NOTE 1 – Sub-carrier index represents the physical index (clause 7.1.4.1/10.1.4.1 of [ITU-T G.9960]). All BAT entries outside [TID_X_{MIN}, TID_X_{MAX} + G – 1] shall be considered as unloaded.</u></p> <p><u>NOTE 2 – If a sub-carrier is not loaded, the field shall be set to zero.</u></p> <p><u>NOTE 3 – If W = 1, this field shall be set to zero.</u></p>			

8.25.1.5 PartialBitloadingTLV

The PartialBitloadingTLV field includes partial information on the bitloading to be used in a link.

Table 8-140.7 – Format of PartialBitloadingTLV field

<u>Field</u>	<u>Octet</u>	<u>Bits</u>	<u>Description</u>
<u>NUM_BAT_ENT</u> <u>for Profile 2 OFBs</u>	<u>0 and 1</u>	<u>[9:0]</u>	<u>This field indicates the number of BAT entries to be updated (V). The valid range of this field is from 0 (V=1) to 1023 (V=1024).</u>
<u>GROUP_ID</u>		<u>[12:10]</u>	<u>This field indicates the current GRP_ID associated with the BAT corresponding to O_BAT_ID and N_BAT_ID (G). Partial BAT update shall not change the current GRP_ID. It shall be formatted as shown in Table 7-13/10-13 of [ITU-T G.9960].</u>
<u>Reserved</u>		<u>[15:13]</u>	<u>Reserved by ITU-T (Note 1).</u>
<u>T₁</u>	<u>2 and 3</u>	<u>[15:0]</u>	<u>16-bit unsigned integer indicating the sub-carrier index (Note 2). It shall be an integer multiple of G.</u>
<u>B_{T1}</u>	<u>4</u>	<u>[3:0]</u>	<u>4-bit unsigned integer indicating the number of bits assigned to sub-carrier indices T₁ to T₁+G–1.</u>
<u>Reserved</u>		<u>[7:4]</u>	<u>Reserved by ITU-T (Note 1).</u>
...
<u>T_V</u>	<u>Var</u>	<u>[15:0]</u>	<u>16-bit unsigned integer indicating the sub-carrier index (Note 2). It shall be an integer multiple of G.</u>
<u>B_{Tv}</u>	<u>Var</u>	<u>[3:0]</u>	<u>4-bit unsigned integer indicating the number of bits assigned to sub-carrier indices T_v to T_v+G–1.</u>
<u>Reserved</u>		<u>[7:4]</u>	<u>Reserved by ITU-T (Note 1).</u>
<p><u>NOTE 1 – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.</u></p> <p><u>NOTE 2 – Sub-carrier index represents the physical index (clause 7.1.4.1/10.1.4.1 of [ITU-T G.9960]).</u></p>			

9 Security

Security inside a domain is provided by encryption of the relevant LLC frames communicated between the nodes of the domain. The encryption method used is based on AES-128 and described in clause 9.1. Every pair of nodes in unicast and nodes of every multicast group communicating in a secure mode may use a unique encryption key.

Authentication, generation, distribution of encryption keys between nodes, and periodic key and authentication updates are provided by a set of authentication and key management (AKM) procedures, described in clause 9.2.

Security of a network containing more than one domain is provided by setting all the domains of the network in secure mode. Inter-domain bridges are considered to be secure, while security measures protecting inter-domain bridges against outside intrusion are beyond the scope of this Recommendation.

Confidentiality between clients associated with the same node is considered to be resolved at the higher layers of the client protocol stack and is beyond the scope of this Recommendation.

9.1 Encryption

The encryption is based on the advanced encryption standard (AES) according to [NIST FIPS 197] and the counter with cipher block chaining message authentication code (CCM) algorithm recommended by [NIST 800-38C]. The CCM protocol (CCMP) includes the CCM encryption mechanism and a particular format the encrypted LLC frame shall be communicated to facilitate decryption.

9.1.1 Description of the CCMP

9.1.1.1 CCM encryption

The CCM encryption algorithm complies with [NIST SP 800-38C] except that some variables are expressed in bytes instead of bits.

Prerequisite:

- Block-cipher algorithm AES-128 [NIST FIPS 197];
- Encryption key K: 128 bits (16 bytes) long;
- Counter-generation function: produces 128-bit (16 bytes) counter blocks (Ctr);
- Length of the message integrity code (MIC), Tlen bytes.

Input:

- Nonce N: a bit-string of less than 128 bits (16 bytes) long;
- Payload P of length Plen bytes: the part of the data unit (APDU or LCDU) to be both encrypted and protected by the MIC;
- Associated data A of length Alen bytes: the unencrypted part of the data unit and additional data to be protected by the MIC.

Output:

- cipher text (encrypted payload) C;
- MIC of the length Tlen bytes.

Steps of the algorithm:

- 1) Apply the formatting function, as described in clause 9.1.1.3 to the input variables N , A , and P to produce the 128-bit blocks B_0, B_1, \dots, B_r .
- 2) Set $Y_0 = \text{CIPH}_K(B_0)$: apply the block-cipher algorithm [NIST FIPS 197] with the key K .
- 3) For $i = 1$ to r , do $Y_i = \text{CIPH}_K(B_i \oplus Y_{i-1})$: chaining the blocks.
- 4) Set $T = \text{MSB}_{Tlen}(Y_r)$: the $Tlen$ most significant bits of the final round of this computation.

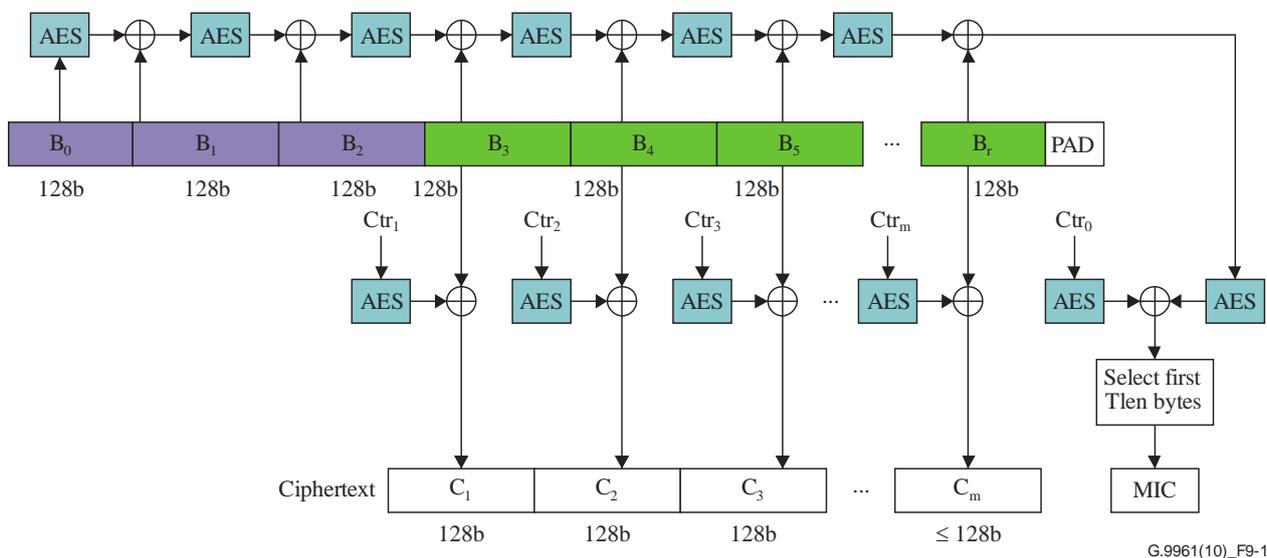
NOTE 1 – These first four steps constitute the cipher-block chaining that calculates the value of T to generate MIC. If the contents of the encrypted blocks have been altered before reception, it is extremely unlikely that the received T value will still match the received MIC. Agreement therefore constitutes assurance of message authenticity (integrity).

- 5) Generate the counter blocks $Ctr_0, Ctr_1, \dots, Ctr_m$, where $m = \text{ceiling}(Plen/128)$.
- 6) For $j = 0$ to m , do $S_j = \text{CIPH}_K(Ctr_j)$: apply the block-cipher algorithm with the key K .
- 7) Set $S = S_1 \parallel S_2 \parallel \dots \parallel S_m$: this defines the string of encrypted counter blocks. Note that S_0 is skipped.
- 8) Compute $C = (P \oplus \text{MSB}_{Plen}(S)) \parallel (T \oplus \text{MSB}_{Tlen}(S_0))$: the cipher text is the string of counter blocks XOR'd with the payload data; the MIC is produced by XOR'ing T with S_0 .

NOTE 2 – The second four steps constitute generation of the actual cipher text of encrypted data concatenated with the MIC. The associated data A are not incorporated into the cipher text C : the relevant part of the data are sent unencrypted, as described in clause 9.1.2.1. The A -data are incorporated in the calculation of the MIC, and thus are protected against undetected alteration.

A block diagram illustrating the CCM encryption and MIC generation algorithm described above is presented in Figure 9-1.

The B -blocks from B_3 onwards contain payload bits (P). B_0 contains a nonce (N). B_1 and B_2 contain associated data bits (A). The AES-blocks stand for AES-128 functions. Those are fed by 128-bit counter blocks (Ctr_0 - Ctr_m). The PAD complements the last payload block to 128 bits.



G.9961(10)_F9-1

Figure 9-1 – Functional diagram of CCM encryption and message-authentication

9.1.1.2 Parameters

Valid values of the CCM encryption parameters are presented in Table 9-1.

Table 9-1 – CCM parameters

Parameter	Valid values
MIC size (<i>Tlen</i>), bytes	4, 8, 16
Payload size (<i>Plen</i>), bytes	$\leq (2^{14} - 1)$
Associated data size (<i>Alen</i>), bytes	See Table 9-5

NOTE – Selection of MIC size is vendor discretionary but should be based on the guidance provided in [NIST FIPS 197].

9.1.1.3 Input variables

The input variables to support CCM encryption are:

- counter blocks (*Ctrn*);
- nonce block (*B0*);
- associated data blocks (*B1* and *B2*);
- payload blocks (*B3* to *Br*);
- encryption key.

The 16-byte counter blocks *Ctr₀*, *Ctr₁*, ..., *Ctr_m* shall have the format presented in Table 9-2. Each block shall comprise a 1-byte flag, a 13-byte nonce, and a 2-byte counter block number (in the range from 0 to *m*). All bytes of the counter block shall be formatted MSB first: the first bit of the byte 0 is the MSB (bit 7) and the last bit of the byte 15 is the LSB (bit 0). The counter block number shall be represented as a 16-bit binary integer where the LSB is the LSB of byte 15.

Table 9-2 – Format of the Ctr blocks

Byte number	0	1, 2 ..., 13	14, 15
Contents	Flags (Note)	Nonce	Counter block number
NOTE – The content of the Flags byte is: bits [7:6] – reserved by ITU-T for NIST, shall be set to 00 ₂ . bits [5:3] – shall be set to 000 ₂ . bits [2:0] – shall be set to 001 ₂ .			

The 13-byte nonce shall be constructed as presented in Table 9-3. The MSB of byte 0 of the nonce in Table 9-3 shall be mapped to the MSB of byte 1 of the Ctr block. The value and format of the frame number (FN) shall be as specified in clause 9.1.2 (Table 9-6). The LSB of the FN shall be mapped to the LSB of byte 12 of the nonce, and byte 7 of the nonce shall be set to 00₁₆. The source MAC address of the APDU or LCDU shall have a standard IEEE 802.3 format where the MSB shall be mapped to the MSB of byte 1 of the nonce. All bytes of the nonce shall be formatted MSB first: the first bit of the byte 0 is MSB (bit 7) and the last bit of the byte 12 is LSB (bit 0).

Table 9-3 – Format of the nonce

Byte number	0	1 – 6	7	8 – 12
Contents	Flags (Note)	Source MAC address	00 ₁₆	Frame number (FN)
NOTE – The content of the Flags byte is: Bits [7:3] – the same bits of Byte 0 of the CCMP header. Bits [2:0] – reserved by ITU-T. All reserved bits of the Flags byte shall be set to zero.				

The value of the nonce (for the given key) shall never be the same for different encrypted payloads, and shall always be the same for identical encrypted payloads (e.g., when APDU or LCDU is retransmitted or relayed). The encryption key shall be changed promptly to avoid repetition of the nonce (see clause 9.1.2.3).

The 16-byte nonce block B_0 shall have a format as presented in Table 9-4. The length of the encrypted payload in octets ($Plen$) shall be represented as a 16-bit unsigned integer with the LSB mapped to the LSB of byte 15 of B_0 .

Table 9-4 – Format of block B_0

Byte number	0	1, 2..., 13	14, 15
Content	Flags (Note)	Nonce	Length of the payload ($Plen$)
NOTE – The content of the Flags byte is: Bit [7] – Reserved by ITU-T for NIST, shall be set to zero. Bit [6] – Shall be set to one. Bits [5:3] – Shall indicate the length of the MIC encoded as: 001 ₂ – 4-byte MIC. 011 ₂ – 8-byte MIC. 111 ₂ – 16-byte MIC. All other values are reserved by ITU-T. Bits [2:0] – Shall be set to 001 ₂			

The two 16-byte associated data blocks B_1 and B_2 shall have a format as presented in Table 9-5. Byte 0 is the first byte and byte 15 is the last byte.

Table 9-5 – Format of blocks B_1, B_2

Block	Bytes	Contents (Note 1)
B_1	0 and 1	Length of associated data in bytes ($Alen$), expressed as an unsigned integer (Note 2).
	2 and 3	Reserved by ITU-T (Note 3).
	4 to 9	Destination MAC address.
	10 to 15	Source MAC address.
B_2	0 to 4	Portion of LFH excluding bytes containing TTL and TSMP fields (Note 4).
	5 to (4 + V)	Additional unencrypted field. APDU (EAPC): TG bytes of VLAN TAGs plus 2 bytes of MAC client length/type ($V = TG + 2$, See Figure A.1). LCDU: 2 bytes of EtherType ($V = 2$, or equivalent).
	(5 + V) to 15	Zero padding as specified in [NIST 800-38C].
NOTE 1 – All fields are mapped so that the most significant byte of the value associated with a particular field is mapped onto the byte with the smaller sequential number. NOTE 2: For APDU (EAPC), $Alen$ shall include byte 2 of B_1 to byte 6+ TG of B_2 (21+ TG bytes). For LCDU, $Alen$ shall include byte 2 of B_1 to byte 6 of B_2 (21 bytes). NOTE 3 – Bits that are reserved by ITU-T shall be set to zero. NOTE 4 – Byte 0 to byte 4 of B_2 shall correspond to byte 0 to byte 4 of LFH, respectively (Table 8-1). The bit corresponding to the CCMPI field of LFH shall be set to 1.		

All bytes of the nonce and the associated data blocks shall be formatted MSB first: the first bit of the byte 0 is MSB (bit 7) and the last bit of the byte 15 is LSB (bit 0).

Payload blocks (B_3 to B_r) are 16-byte long and shall contain bytes of the APDU or LCDU to be encrypted (see clause 9.1.2.2, encrypted part of APDU or LCDU). The APDU or LCDU bytes shall be mapped to payload blocks in sequential order, so that the first byte of the APDU or LCDU to be encrypted is mapped to byte 0 of B_3 , the second byte of the payload is mapped to byte 1 of B_3 , the 17th byte of the APDU or LCDU is mapped to byte 0 of B_4 , and so on. If the last byte of the payload

does not fall on byte 15 of B_r , the payload shall be padded to fill the last block by appending zero bytes (00_{16}). All bytes of the payload blocks shall be formatted MSB first: the first bit of byte 0 of block B_3 is the MSB (bit 7) and the last bit of byte 15 of block B_r is LSB (bit 0).

The encryption key is 128 bits long and shall be generated and assigned as described in clause 9.2.

9.1.2 CCM encryption protocol (CCMP)

9.1.2.1 Functional description

The functional model of the CCMP is presented in Figure 9-2. The incoming APDU (or LCDU) is encrypted by the CCM encryption function, performing as described in clauses 9.1.1 and 9.1.2.2. The LFH is sent unencrypted. Both the LFH and the unencrypted part of the APDU (or LCDU) are protected by the MIC as a part of associated data. If the encrypted LLC frame cannot be authenticated, it shall be dropped by the receiver.

The keyID, the frame number (FN), and the length of the MIC associated with the encrypted LLC frame are conveyed to the receive side in the CCMP header to assist decryption; CCMP header is sent unencrypted and described in clause 9.1.2.3, but is also protected by the MIC. Construction of the nonce (N) and the Associated data is as described in clause 9.1.1.

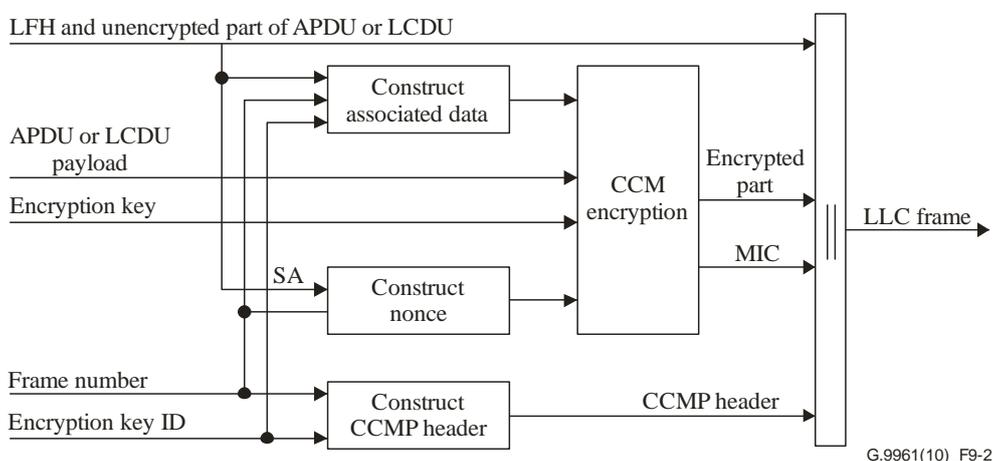


Figure 9-2 – Functional diagram of CCMP encryption

9.1.2.2 CCMP encryption format

The format of the encrypted LLC frame consists of five parts: LFH, CCMP header, unencrypted part, encrypted part (cipher text), and MIC as shown in Figure 9-3.

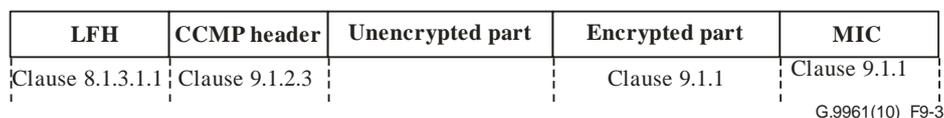


Figure 9-3 – Format of CCMP-encrypted LLC frame

The format of the LFH shall be as described in clause 8.1.3.1.1. The format of CCMP header shall be as described in clause 9.1.2.3. Generation of the cipher text (encrypted part of the APDU or LCDU) and MIC shall be as described in clause 9.1.1.

The unencrypted part of the APDU may be defined differently based on the type of APC. For Ethernet APC, the unencrypted part is defined in Annex A. The unencrypted part of the LCDU shall include all bytes starting from the first byte of the LCDU and ending by the last byte of the EtherType field of the LCDU (see clause 8.1.3.4). The length of the unencrypted part of the LCDU is 14 bytes.

9.1.2.3 CCMP header

The CCMP header consists of six bytes and shall have a format as presented in Table 9-6. It carries the encryption key identification number (keyID), the type of the encryption key, the length of the MIC, and the security frame number (FN). These four parameters are necessary for decryption.

The length of the MIC shall be selected according to the procedure defined in clause 9.2.3.

Table 9-6 – CCMP header format

Field	Octet	Bits	Description
CCMP header	0	[2:0]	Length of the MIC encoded as: 001 – 4-byte MIC. 011 – 8-byte MIC. 111 – 16-byte MIC. All other values are reserved by ITU-T.
		[3]	Reserved by ITU-T (Note).
		[5:4]	The type of encryption key: 00 – NN key or NMK. 10 – DB key. 01 – NSC key. 11 – Reserved by ITU-T.
		[6]	Encryption key ID, formatted as an unsigned binary integer.
	[7]	Reserved by ITU-T (Note).	
	1 to 5	[39:0]	40-bit FN, formatted as an unsigned binary integer.
NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

The keyID identifies the used encryption key among those assigned to the communicating nodes during the AKM procedure, as described in clause 9.2.5.2. Keys assigned for communication with different peers may have the same key IDs. The keyID shall be formatted as a 1-bit unsigned binary integer. Valid values of the keyID are 0 and 1.

The FN is a serial number of the encrypted LLC frame and shall be represented as a 40-bit unsigned binary integer. The FN shall be set to one when a new encryption key is established and increased by one with every encrypted LLC frame passed using this key. FN shall never be repeated for the same value of the key: the key shall be changed prior to FN reaching its maximum value. FN expiration is used as a trigger for keys update (see clause 9.2.4).

In order to allow some time for the FN update procedure before the FN repeats itself, whenever an FN covers 95% of its maximum value, the key update procedure for the corresponding key shall be started. This would apply to all the keys, namely NSC, NN, NMK, DB and multicast keys. In the case of NMK, DB and multicast keys it is possible that multiple nodes realize that the keys need to be updated at the same time, so a random delay in the range 0 to 5s is added between when a node realizes that it needs the key to be updated and when it communicates this to the SC. The node shall communicate to the SC that the key needs to be updated only if the key is not updated by the time its random delay timer expires.

For the DB, the NMK and multicast keys the FN is initialized to a value that is composed of the node's DEVICE_ID in the most significant byte of the FN field and zeros in all other bytes of the FN field. This will ensure that a node will repeat an FN value already used by another node for the same key with a very low probability.

NOTE – On the receive side, the FN may not appear to be sequential, if the order in which packets are encrypted and transmitted is different.

9.2 Authentication and key management procedures

9.2.1 Overview

Authentication and key management (AKM) defines a set of procedures allowing a node to join a secure domain and to operate in it with point-to-point and point-to-multipoint security. AKM includes the following main procedures:

- authentication to the domain in secure mode;
- establishing point-to-point encryption keys for unicast communication;
- establishing point to multi point encryption keys for multicast communication;
- periodic re-authentication and updating point-to-point and point-to-multipoint encryption keys.

To set a node for secure operation, it shall be provided with a password. The node password shall comply with the characteristics presented in clause 9.2.2.1. Passwords shall never be communicated, even if encrypted. A particular way to establish a node password is vendor discretionary and beyond the scope of this Recommendation.

Prior to authentication to the domain in secure mode, the node shall first register with the domain master using the admission procedure described in clause 8.6.1. The domain master shall indicate to the registering node that the domain operates in secure mode by setting the security field of the ADM_DmRegistrResponse.cnf message to "Secure". A registered node shall then apply for authentication to operate in secure mode. Authentication shall be performed as described in clause 9.2.2. A node that is not authenticated shall not attempt to communicate with other nodes of the secure domain. In a domain operating in secure mode, all the communications within the domain between authenticated nodes (except for MAPs) shall be encrypted with the appropriate key.

NOTE – A node that is not yet authenticated can communicate without encryption with the DM and SC (either directly or via proxy node) for the purpose of registration and authentication respectively.

An authenticated node can establish encryption keys for secure unicast, multicast, and broadcast communications inside the domain. Point-to-point encryption keys shall be established using the procedures described in clause 9.2.3.

The procedures described in clause 9.2.4 shall be used for periodic re-authentication and updates of encryption keys.

A flowchart of AKM procedures is presented in Figure 9-4.

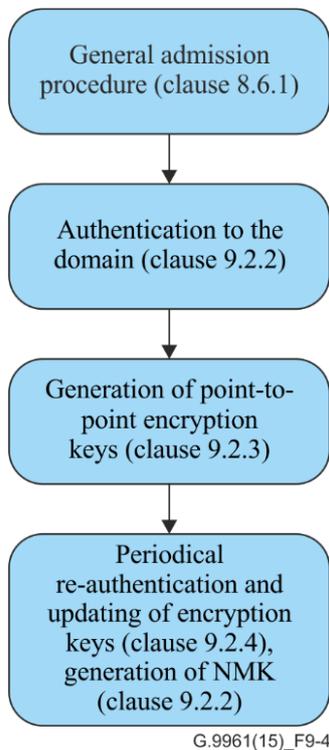


Figure 9-4 – Flowchart of AKM

AKM procedures in the domain are managed by the security controller (SC), which may be an additional function of the domain master or an endpoint node. Only the following two configurations for the SC shall be allowed:

- SC function is assigned to a specific node and can only be changed by user/operator control (may be non-co-located with the DM). In this case, a backup SC function may also be assigned to another specific node by user/operator control.
- SC and DM are always co-located and may change autonomously (i.e., without user/operator control)

When the SC is implemented by an endpoint node, the DM shall start authentication as soon as possible, after this endpoint node has completed its registration with the DM. In this case, the authentication of this endpoint node is done internally.

A DM in a secure domain that is not functioning as the SC is allowed to transmit MAPs, registration confirmation messages to the SC and authentication requests to the SC, before being authenticated. In a secure domain, if a node sends a registration request when the DM is not authenticated yet, the DM shall reject the registration request by registration response with status 'DM not authenticated' (see Table 8-15).

If the DM and SC are not co-located then the DM shall be configured with the REGID of the SC and the DM shall allow registration of only the SC until the DM has been authenticated by the SC. If a backup SC is assigned by user/operator control, the DM shall also be configured with the REGID of the backup SC. In this case, if the SC has not registered with the DM within 60 seconds, the DM shall allow registration of the backup SC as the SC.

The SC can be configured to use:

- a single encryption key per domain/network using the network membership key (NMK); or
- point-to-point or point-to-multipoint keys (NN).

NMK is granted to the node during its authentication, as specified in clause 9.2.2.1. In case of using NMK, the AKM procedures intended for generating point-to-point encryption keys (clause 9.2.3) are skipped.

9.2.2 Authentication to the domain

For operation in secure mode, a registered node shall authenticate itself to the security controller (SC) as described in this clause. A node that is not authenticated by the SC shall not attempt to communicate (except for the purpose of registration and authentication) with any other node in the secure domain, until it is authenticated. After authentication, the node can be a part of the domain operating in secure mode.

A registered node that attempts to authenticate itself to the security controller shall first establish a connection with the SC (or with the proxy node, if necessary) for transmission and reception of management messages (i.e., management connection or data connection with mixed management messages). This connection should be established with acknowledgement enabled. The SC shall verify (or the proxy node shall verify, see Table 8-20.3) that the node requesting the establishment of connections for transmission and reception of management messages is already registered before setting up these connections. Once established, these connections shall be used to exchange the management messages of the AKM procedure described in this clause.

NOTE – Authentication of the devices joining the domain with a remote facility (e.g., a broadband service provider) requires a trusted channel between the remote facility and the user or between the remote facility and the SC. Set up of this channel and related communication protocols is beyond the scope of this Recommendation. In this case, it is assumed that a remote authenticator, as necessary, may perform some SC functions and it controls operation of the SC.

NOTE – Since a node that is trying to authenticate itself has already a DEVICE_ID assigned, it cannot use the registration RCBTS transmission opportunity. Therefore, the DM should allocate transmission opportunities to this node to exchange the messages that are necessary for the authentication procedure.

9.2.2.1 Authentication

Authentication to the SC shall use the Password-Authenticated Key Exchange (PAK) protocol defined in [\[ITU-T X.1035\]](#) with protocol parameters specified in clause 9.2.2.2. The procedure is described in Figure 9-5. It assumes two nodes, called Supplicant (Node A in Figure 9-5, the node requesting authentication) and Authenticator (Node B in Figure 9-5, the SC), which both share the node password PW. The Supplicant shall initiate a Diffie-Hellman handshake with the Authenticator specified in [\[ITU-T X.1035\]](#). The handshake results that the Supplicant and the Authenticator co-generate a Node-to-SC (NSC) encryption key, which shall only be used for encryption of secure communications between the node and the SC.

NOTE 1 – The NSC key is used only for communication with the SC function of the node. For secure communications with other clients associated with a node containing the SC function, NMK, DB, or NN keys are used, as defined in clause 9.2.3.

NOTE 2 – The PAK protocol, with very high probability, returns a new encryption key after each run.

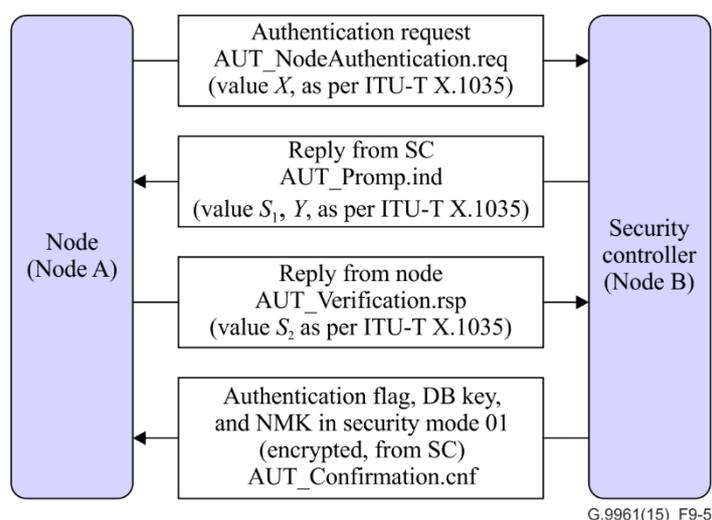


Figure 9-5 – PAK handshake procedure

The procedure shall include the following steps presented in Figure 9-5. The format of the authentication messages supporting the procedure shall be as described in clause 9.2.5.1.

- 1) The Supplicant shall initiate the authentication procedure with the SC by sending to the SC an authentication request (AUT_NodeAuthentication.req) message which includes the REGID of the node, the REGID of the SC as specified in the registration response, and the node password known to the SC (values A, B, PW, respectively, as per [ITU-T X.1035]), hashed into value of X, [ITU-T X.1035]. The values of A, B, and PW shall be as defined in Table 9-7. The AUT_NodeAuthentication.req message shall be sent unencrypted.
- 2) The SC shall verify the received value of X and reply to the Supplicant with an authentication prompt (AUT_Prompt.ind) message, including values S_1 and Y, as per [ITU-T X.1035]. The message shall be sent unencrypted within 800 ms after reception of the AUT_NodeAuthentication.req message. If the SC determines that the value of X is invalid, it shall reply with AUT_Prompt.ind with the status set to one.

NOTE – The SC identifies the node providing the value of X using the node MAC address (SA of the LCDU carrying the AUT_NodeAuthentication.req message).

- 3) The node shall verify the prompt, compute the value S_2 , as per [ITU-T X.1035], and send it to the SC in the authentication prompt verification (AUT_Verification.rsp) message. The message shall be sent unencrypted within 800 ms after reception of the AUT_Prompt.ind. If the node determines that the value of S_1 or the value of Y is invalid, it shall reply with AUT_Verification.rsp with the status set to one.
- 4) This step of the process applies only in case of authentication of the node to the SC (i.e., it does not apply to re-authentication). Upon receiving the AUT_Verification.rsp message, if the SC concludes that the authenticating node is authenticated, it shall inform the DM by sending an AUT_NodeAuthenticated.req message that the authenticating node has been authenticated and may be joined to the secured domain. The AUT_NodeAuthenticated.req message is encrypted by the NSC key that it shares with the DM. The AUT_NodeAuthenticated.req message includes the DEVICE_IDs of the authenticating node and the proxy node. The DM shall confirm receiving the AUT_NodeAuthenticated.req message by replying with an AUT_NodeAuthenticated.cnf message and shall then update the routing table to include the authenticating node. The updated routing table is then advertised by the DM by broadcasting the updated TM_DomainRoutingChange.ind message. In case of authentication directly with SC without going through a proxy node, the proxy field of the AUT_NodeAuthenticated.req and AUT_NodeAuthenticated.cnf messages shall be set to 0.

- 5) Using the exchanged variables S_1 and S_2 both nodes shall compute independently the 128-bit NSC encryption key (value of K as per [ITU-T X.1035]).
- 6) The SC shall send to the Supplicant the authentication confirmation (AUT_Confirmation.cnf) message, which includes the confirmation flag, NMK, and the in-domain broadcast (DB) encryption key. The AUT_Confirmation.cnf message shall be sent encrypted by NSC within 800 ms after reception of the AUT_Verification.rsp message.

9.2.2.1.1 Authentication failure

The node shall consider the authentication process failed in the following conditions:

- The node does not receive the AUT_Prompt.ind message within 1 s after it sent the AUT_NodeAuthentication.req message; or
- The node receives the failure indication in the AUT_Prompt.ind message; or
- The node does not receive the AUT_Confirmation.cnf within 1 s after it sent the AUT_Verification.rsp message; or

The node receives the failure indication in the AUT_Confirmation.cnf message.

If the SC does not receive the AUT_Verification.rsp message within 1 s after it sent the AUT_Prompt.ind message, it shall abort the authentication process.

In case the authentication process failed, the Supplicant may start re-authentication in time period greater than 1 s from the instant it detects the failure, and shall not transmit any data until it starts re-authentication. After four unsuccessful re-authentication attempts, the SC shall request the domain master by sending the SC_DMRes.req message to resign the node (Supplicant) from the domain using forced resignation, as described in clause 8.6.1.1.3.2. The domain master, upon receiving the SC_DMRes.req message, shall reply with an SC_DMRes.cnf message within 100 ms. If the SC does not receive an SC_DMRes.cnf message within 200 ms, it shall retry sending the SC_DMRes.req message.

9.2.2.1.2 Successful authentication

The node whose authentication was confirmed is allowed to broadcast and receive broadcast messages from other secure nodes of the domain using the DB encryption key. If the domain is configured to operate in point-to-point mode, the node can request from the SC point-to-point encryption keys to communicate with other nodes operating in secure mode as described in clause 9.2.3.

If the SC is configured to operate with NMK, it shall send the NMK and DB encryption key to the authenticated node in an AUT_Confirmation.cnf message (see clause 9.2.5.1.4). The authenticated node is allowed to broadcast and receive broadcast messages using the DB encryption key and may communicate with other nodes using the NMK encryption key in this mode.

If the SC is configured to operate with NN key, it shall send only the DB encryption key to the authenticated node in AUT_Confirmation.cnf message.

9.2.2.1.3 Authentication via proxy

If a node cannot communicate with the SC directly, it shall authenticate itself with the SC using other nodes as relays. In a secure domain, a node is not allowed to send its topology update messages and cannot read topology updates sent by other nodes prior to authentication. Thus, the registration proxy shall be used as the first relay between the SC and the node.

The authenticating node shall start authentication by sending an AUT_NodeAuthentication.req message with the ProxyAuth field set to 1 and the addressing fields as defined in Table 9-6.1.

Table 9-6.1 – Addressing fields of the AUT_NodeAuthentication.req from authenticating node to proxy node

Field	Value
DA of the LCDU carrying the message	REGID of the SC
SA of the LCDU carrying the message	REGID of the authenticating node
OriginatingNode of the LLC frame carrying the LCDU	DEVICE_ID of the authenticating node
DestinationNode of the LLC frame carrying the LCDU	DEVICE_ID of the proxy node
SID of the PHY frame carrying the message	DEVICE_ID of the authenticating node
DID of the PHY frame carrying the message	DEVICE_ID of the proxy node

Upon receiving the AUT_NodeAuthentication.req message the proxy node shall relay the AUT_NodeAuthentication.req message to the SC by setting the addressing fields as defined in Table 9-6.2.

Table 9-6.2 – Addressing fields of the AUT_NodeAuthentication.req from proxy to the SC

Field	Value
DA of the LCDU carrying the message	REGID of the SC
SA of the LCDU carrying the message	REGID of the authenticating node
OriginatingNode of the LLC frame carrying the LCDU	DEVICE_ID of the proxy node
DestinationNode of the LLC frame carrying the LCDU	DEVICE_ID of the SC
SID of the PHY frame carrying the message	DEVICE_ID of the proxy node
DID of the PHY frame carrying the message	DEVICE_ID of the SC or the next relay node towards the SC in case where the proxy does not have a direct link with the SC

Upon receiving the AUT_NodeAuthentication.req message, the SC shall detect that the received AUT_NodeAuthentication.req message was sent by a proxy node. It shall extract the REGID of the authenticating node from the SA of the LCDU carrying the AUT_NodeAuthentication.req message to identify the authenticating node. The SC shall validate the AUT_NodeAuthentication.req message as described in the regular authentication procedure (see clause 9.2.2.1) and build accordingly the AUT_Prompt.ind message with the ProxyAuth set to 1. The SC shall set the addressing fields as defined in Table 9-6.3.

The AUT_Prompt.ind message shall be sent unencrypted within 800 ms after receipt of the AUT_NodeAuthentication.req message. If the SC determines that the value of X is invalid, it shall reply with an AUT_Prompt.ind message with the status set to one.

**Table 9-6.3 – Addressing fields of the AUT_Prompt.ind
from SC to the proxy node**

Field	Value
DA of the LCDU carrying the message	REGID of the authenticating node
SA of the LCDU carrying the message	REGID of the SC
OriginatingNode of the LLC frame carrying the LCDU	DEVICE_ID of the SC
DestinationNode of the LLC frame carrying the LCDU	DEVICE_ID of the proxy node
SID of the PHY frame carrying the message	DEVICE_ID of the SC
DID of the PHY frame carrying the message	DEVICE_ID of the proxy node or the next relay node towards the proxy in case where the SC has not direct link with the proxy

Upon receiving the AUT_Prompt.ind message the proxy node shall then unicast the received AUT_Prompt.ind message to the authenticating node using the addressing scheme shown in Table 9-6.4.

**Table 9-6.4 – Addressing fields of the AUT_Prompt.ind
from the proxy node to the authenticating node**

Field	Value
DA of the LCDU carrying the message	REGID of the authenticating node
SA of the LCDU carrying the message	REGID of the SC
OriginatingNode of the LLC frame carrying the LCDU	DEVICE_ID of the proxy node
DestinationNode of the LLC frame carrying the LCDU	DEVICE_ID of the authenticating node
SID of the PHY frame carrying the message	DEVICE_ID of the proxy node
DID of the PHY frame carrying the message	DEVICE_ID of the authenticating node

Upon receiving the AUT_Prompt.ind message, the authenticating node shall verify the prompt, compute the value S_2 as per [ITU-T X.1035], and send to the SC in the AUT_Verification.rsp message by again using the proxy node as the first relay towards the SC with the addressing scheme defined in Table 9-6.1 and wait 1 second for the AUT_Confirmation.cnf message to complete the authentication process.

Upon receiving the AUT_Verification.rsp message the proxy node shall forward the received message to the SC with the addressing scheme defined in Table 9-6.2.

Upon receiving the AUT_Verification.rsp message, the SC shall verify that the authenticating node is authenticated. If the SC concludes that the authenticating node is authenticated, it shall inform the DM by sending an AUT_NodeAuthenticated.req message that the authenticating node has been authenticated and may be joined to the secured domain. The AUT_NodeAuthenticated.req message is encrypted by the NSC key that it shares with the DM. The AUT_NodeAuthenticated.req message includes the DEVICE_IDs of the authenticating node and the proxy node. The DM shall confirm receiving the AUT_NodeAuthenticated.req message by replying with an AUT_NodeAuthenticated.cnf message and shall then update the routing table to include the authenticating node, and indicate that all routes to the authenticating node go via the proxy node. The updated routing table is then advertised by the DM by broadcasting the updated TM_DomainRoutingChange.ind message. After the SC receives the updated TM_DomainRoutingChange.ind message that includes the authenticating node, it shall send the AUT_Confirmation.cnf message to the authenticating node via the proxy node using the normal unicast routing procedure described in clause 8.5.7. The addressing fields of the AUT_Confirmation.cnf message from the SC to the authenticating node are as shown in Table 9-6.5.

**Table 9-6.5 – Addressing fields of the AUT_Confirmation.cnf
from SC to the authenticating node**

Field	Value
DA of the LCDU carrying the message	REGID of the authenticating node
SA of the LCDU carrying the message	REGID of the SC
OriginatingNode of the LLC frame carrying the LCDU	DEVICE_ID of the SC
DestinationNode of the LLC frame carrying the LCDU	DEVICE_ID of the authenticating node
SID of the PHY frame carrying the message	DEVICE_ID of the SC
DID of the PHY frame carrying the message	DEVICE_ID of the proxy node or the next relay node towards the proxy node in case the SC does not have direct link to the proxy node

The failure of a node's authentication and related actions by the node and the SC are as described in clause 9.2.2.1.1.

The steps that the authenticating node takes after its successful authentication to become part of a secure domain are as described in clause 9.2.2.1.2.

9.2.2.2 The PAK protocol parameters

The PAK parameters used for node authentication shall comply with the requirements listed in Table 9-7.

Table 9-7 – X.1035 – PAK parameters

X.1035 parameter	Description	Length (bits)	Notes
A, B	Node identifiers of the supplicant and authenticator	48	Clause 9.2.2.2.1
PW	Node password of the supplicant	96	Clause 9.2.2.2.2
p	Diffie-Hellman prime	1024	Clause 9.2.2.2.3
g	Diffie-Hellman generator	8	Clause 9.2.2.2.4
R_A, R_B	Secret exponents of the supplicant and authenticator	384	Clause 9.2.2.2.5
H_1	Hash functions of SHA-256 type	1152	Clause 9.2.2.2.6
H_2		1152	Clause 9.2.2.2.6
H_3, H_4, H_5		128	Clause 9.2.2.2.6
K	NSC key	128	Clause 9.2.2.2.7

9.2.2.2.1 Node identifier

The parameters A and B defined in [\[ITU-T X.1035\]](#) represent the identifiers of the supplicant and the authenticator of the PAK protocol, respectively. A shall be set to the 48-bit MAC address of the supplicant. B shall be set to the 48-bit MAC address of the security controller.

9.2.2.2.2 Node password

The parameter PW defined in [ITU-T X.1035] represents the secret password shared by the supplicant and the authenticator. The size of PW shall be 96 bits. Generation of the node password is out of scope of this Recommendation.

9.2.2.2.3 Diffie-Hellman prime

The parameter p defined in [ITU-T X.1035] represents the Diffie-Hellman prime, which is a 1024-bit predefined constant. It shall be set to the following number (MSB first):

```
p = FFFF FFFF FFFF FFFF C90F DAA2 2168 C234 C4C6 628B 80DC 1CD1
    2902 4E08 8A67 CC74 020B BEA6 3B13 9B22 514A 0879 8E34 04DD
    EF95 19B3 CD3A 431B 302B 0A6D F25F 1437 4FE1 356D 6D51 C245
    E485 B576 625E 7EC6 F44C 42E9 A637 ED6B 0BFF 5CB6 F406 B7ED
    EE38 6BFB 5A89 9FA5 AE9F 2411 7C4B 1FE6 4928 6651 ECE6 5381
    FFFF FFFF FFFF FFFF16
```

9.2.2.2.4 Diffie-Hellman generator

The parameter g defined in [ITU-T X.1035] represents the Diffie-Hellman generator, which is an 8-bit predefined constant. It shall be set to the following number (MSB first):

$$g = 0D_{16}$$

9.2.2.2.5 Secret exponents

The parameters R_A and R_B defined in [ITU-T X.1035] represent the secret exponents selected by the supplicant and the authenticator of the PAK protocol, respectively. They shall be selected randomly as follows:

- The number generated shall be 384-bits in length;
- The number generated shall not be less than 4;
- The number generated shall have a uniform statistical distribution over its range $[4, 2^{384}-1]$.

9.2.2.2.6 Hash functions

The parameters, H_1 , H_2 , H_3 , H_4 , and H_5 defined in [ITU-T X.1035] represent the hashing functions used by the supplicant and authenticator in various stages of PAK protocol. They shall be defined as follows:

```
 $H_1(u_1) =$  SHA-256(00 00 00 0116 | 00 00 00 0116 |  $u_1$ ) |
    SHA-256(00 00 00 0116 | 00 00 00 0216 |  $u_1$ ) |
    SHA-256(00 00 00 0116 | 00 00 00 0316 |  $u_1$ ) |
    SHA-256(00 00 00 0116 | 00 00 00 0416 |  $u_1$ ) |
    TRC(SHA-256(00 00 00 0116 | 00 00 00 0516 |  $u_1$ ), 128);
 $H_2(u_2) =$  SHA-256(00 00 00 0216 | 00 00 00 0116 |  $u_2$ ) |
    SHA-256(00 00 00 0216 | 00 00 00 0216 |  $u_2$ ) |
    SHA-256(00 00 00 0216 | 00 00 00 0316 |  $u_2$ ) |
    SHA-256(00 00 00 0216 | 00 00 00 0416 |  $u_2$ ) |
    TRC(SHA-256(00 00 00 0216 | 00 00 00 0516 |  $u_2$ ), 128);
 $H_3(u_3) =$  TRC(SHA-256(00 00 00 0316 |  $u_3$ ), 128);
```

$H_4(u_4) = \text{TRC}(\text{SHA-256}(00\ 00\ 00\ 04_{16} \mid u_4), 128)$; and

$H_5(u_5) = \text{TRC}(\text{SHA-256}(00\ 00\ 00\ 05_{16} \mid u_5), 128)$.

SHA-256 is defined in [NIST FIPS 180-3]. $x \mid y$ denotes the concatenation of strings x and y (MSB first) as defined in [ITU-T X.1035]. 00000001_{16} denotes a 32-bit constant in a hexadecimal form. $\text{TRC}(z, a)$ denotes the first a MSBs of z (i.e., truncated string). u_1 and u_2 denote the 192-bit inputs to $H_1(\cdot)$ and $H_2(\cdot)$, respectively. u_3 , u_4 , and u_5 denote the 3264-bit inputs to $H_3(\cdot)$, $H_4(\cdot)$, and $H_5(\cdot)$, respectively.

9.2.2.2.7 NSC key

The parameter K defined in [ITU-T X.1035] represents the 128-bit NSC key, which is the output of the PAK protocol generated independently by the supplicant and the authenticator.

9.2.3 Pair-wise authentication and generation of point-to-point keys

The node authenticated by the SC is authorized to communicate with other nodes in the domain. In order to establish a point-to-point secure communication with another node (for unicast) or with several nodes (for multicast), all nodes intended to be involved in communications (both the supplicant and all the addressees) shall be:

- authenticated to the SC, as described in clause 9.2.2, and granted with a unique NSC key;
- granted with a pair of node-to-node (NN) encryption keys, one per each direction of communication.

The NN encryption keys shall be established as described in this clause and shall be used for all secure communications between the nodes (unicast or multicast, respectively).

9.2.3.1 Generation of point-to-point encryption keys

The procedure to establish NN keys shall include the following steps, also presented in Figure 9-6. The format of the messages supporting the described procedure is defined in clause 9.2.5.2.

- 1) The supplicant shall send an `AKM_KeyRequest.req` message to the SC which includes the `DEVICE_ID(s)` of the addressee node(s) it intends to communicate with. The message shall be encrypted with NSC of the supplicant. In the case of requesting point-to-multipoint keys, the supplicant shall set the Multicast Stream Identifier to the MSID of the associated DLL multicast group. In the case of requesting point-to-point keys, the Multicast Stream Identifier field shall be set to 0.
- 2) Upon reception of the `AKM_KeyRequest.req` message, the SC shall accept the request and shall generate a pair of NN keys (NN_{SA} to be used for supplicant towards the addressee(s), and NN_{AS} to be used by each addressee towards the supplicant) if at least one of the addressees is authenticated. Keys shall not be generated if none of the addressees in the supplicant request are authenticated. In the case of multicast keys the NN_{SA} and NN_{AS} keys shall be same.
- 3) The SC shall send the generated pair of NN keys to each of the authenticated addressees using the `AKM_NewKey.req` message; no key shall be sent to addressees that are not authenticated. The `AKM_NewKey.req` message shall be encrypted using the NSC key of the addressee. The addressee shall acknowledge the `AKM_NewKey.req` message by sending an `AKM_NewKey.cnf` message to the SC. In case no `AKM_NewKey.cnf` message is received from a particular addressee during the time period of 100 ms, the SC shall retransmit the message up to four times, and shall remove the addressee from the list if no `AKM_NewKey.cnf` message arrives after the last attempt or an `AKM_NewKey.cnf` message is received with a rejection code (NACK).
- 4) After receiving confirmation messages from all the addressees or expiration of all attempts, the SC shall reply to the supplicant with the `AKM_KeyConfirmation.req` message, which

includes the generated pair of NN keys and DEVICE_ID(s) of the addressee(s) that acknowledged receipt of the AKM_NewKey.req message without a rejection code in the AKM_NewKey.cnf message. The AKM_KeyConfirmation.req message shall be encrypted using the NSC key of the supplicant.

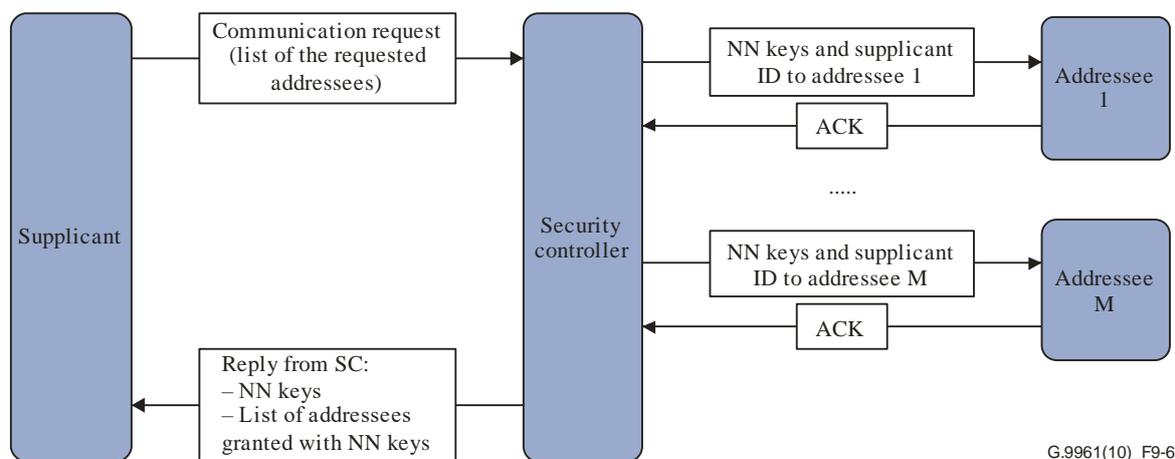


Figure 9-6 – Procedure for NN key generation for unicast (M=1) and multicast (M > 1)

Upon receiving the AKM_KeyConfirmation.req message, the supplicant shall send an AKM_NewKey.ind message to the addressee(s) indicating that new NN keys are established. The AKM_NewKey.ind message shall be sent encrypted using the new NN key.

If the supplicant does not receive the reply from the SC (AKM_KeyConfirmation.req message) within 5 seconds, it shall consider the procedure failed and may re-start it again at the first opportunity. The maximum number of attempts shall be four. After four unsuccessful attempts, the supplicant shall resign from the network (since it is improperly configured) using the resignation procedure defined in clause 8.6.1.1.3.

In case a supplicant intends to join an additional addressee to the existing multicast group, the following steps shall be taken:

- 1) The supplicant shall send to the SC an AKM_AddClient.req message that includes the NN key already established for the multicast group and the DEVICE_ID of the addressee node it intends to join. The message shall be encrypted with NSC of the supplicant.
- 2) Upon reception of the AKM_AddClient.req message, the SC shall check whether the addressee is authenticated, and shall send the NN key supplied by the supplicant to the authenticated addressee using the AKM_NewKey.req message, encrypted using the NSC key of the addressee. The addressee shall acknowledge the received AKM_NewKey.req message by sending an AKM_NewKey.cnf message to the SC. In case no AKM_NewKey.cnf message is received from the addressee during the time period of 200 ms, the SC shall retransmit the message up to four times, and shall remove the addressee from the list if no AKM_NewKey.cnf message arrives after the last attempt or if the AKM_NewKey.cnf message brings a rejection code (NACK).
- 3) After receiving the AKM_NewKey.cnf message, the SC replies to the supplicant with the AKM_KeyConfirmation.req message that includes the NN key and DEVICE_ID of the addressee. If no addressee name is communicated in the AKM_KeyConfirmation.req message, the addressee is not added to the group. The AKM_KeyConfirmation.req message shall be encrypted using the NSC key of the supplicant.
- 4) Upon receiving the AKM_KeyConfirmation.req message, the supplicant shall transmit AKM_NewKey.ind message (encrypted using the NN key) to the Addressee indicating that the NN key is confirmed.

9.2.4 Updating and termination of encryption keys

From time to time the SC shall initiate a routine update of encryption keys. The frequency of routine updates is vendor discretionary, but the interval between the updates shall be:

- Longer than 30 minutes and not exceed 1 hour for NMK.
- Longer than 1 hour and not exceed 6 hours for NN/DB keys.

In addition, the key shall be updated to prevent repetition of FN for the same key (see clause 9.1.2.3). In case the SC suspects a security breach, it may update the security keys immediately.

A transmitting node shall not use an old key to encrypt APDUs that arrived at the A-interface, or LCDUs that were generated after the key was updated.

In case that the SC is allowed to change autonomously, an endpoint node shall not update the encryption keys during the handover process of DM.

9.2.4.1 Updating of NSC using the re-authentication procedure

The NSC key is updated by using the re-authentication procedure.

The frequency of running the routine re-authentication procedure is vendor discretionary, but the interval between the two consecutive procedures shall be longer than 6 hours and not exceed 24 hours.

When an SC determines that NSC key should be updated (due to the routine authentication), it shall send an AKM_KeyUpdate.req message to the node. The AKM_KeyUpdate.req message shall indicate the NSC key update request and 'routine update' request reason (see Table 9-19). The AKM_KeyUpdate.req message shall be encrypted using the NSC key from the SC to the node. The node receiving the AKM_KeyUpdate.req message shall then initiate an authentication procedure with the SC, as described in clause 9.2.2. The 'Re-authentication flag' in the AUT_NodeAuthentication.req message used to initiate the authentication shall be set to 1, to signal that the request is for re-authentication (see Table 9-8). All messages exchanged between the node and the SC for re-authentication shall be encrypted using the NSC key. The final message sent by the SC during the re-authentication procedure AUT_Confirmation.cnf shall be encrypted using the newly generated NSC.

If the SC does not receive the reply from the requested node in a time period of 200 ms, it shall repeat the request. If after four attempts the node does not start the process to re-establish the key, the SC shall terminate the NSC key associated with this node, and initiate forced resignation of the node from the domain using the procedures described in clause 8.6.1.1.3.2 (by sending to the domain master the SC_DMRes.req message). The resigned node can further request to be admitted back using the standard admission procedure described in clause 8.6.1.

9.2.4.2 Updating of NN keys

When a node detects FN expiration for one of its point-to-point/point-to-multipoint keys, it shall initiate a point-to-point/point-to-multipoint key generation procedure to establish NN keys with the relevant addressees, as described in clause 9.2.3. The 'Request Reason' field in the AKM_KeyRequest.req message used to initiate the key generation shall be set to 'key update due to FN expiration' (see Table 9-12).

NOTE – Setting the 'Request Reason' field to 'key update due to FN expiration' can help the SC to refrain from multiple key updates in a point-to-multipoint scenario when multiple nodes detect FN expiration at the same time and send multiple AKM_KeyRequest.req messages to the SC to update the same key.

When an SC detects that a point-to-point/point-to-multipoint key should be updated (due to routine update, see clause 9.2.4), it shall send a key update request AKM_KeyUpdate.req message to the node that initiated generation of the key(s) to be updated. The AKM_KeyUpdate.req message shall indicate NN keys update request and 'routine update' request reason (see Table 9-19). The node shall then initiate a point-to-point/point-to-multipoint key generation procedure as described in the previous paragraph.

9.2.4.3 Termination of NSC and NN keys

The SC shall terminate all NSC keys associated with a node upon node resignation from the domain, as indicated in the TM_DomainRoutingChange.ind message. The node shall terminate NN keys if the node-suppllicant for these keys resigns from the domain or its re-registration is unsuccessful. Old values of NSC and NN keys shall be terminated after the corresponding key update procedures.

The NSC and NN keys associated with a node shall not be terminated and are not required to be updated after a successful re-registration of the node.

The NN keys associated with a node shall not be terminated and are not required to be updated after a successful re-authentication of the node.

The domain master may resign any node from the domain based on security considerations using the forced resignation procedure described in clause 8.6.1.1.3.2. The SC shall use the SC_DMRes.req message to request resignation of the node from the domain.

9.2.4.4 Updating of the DB keys and NMK

Whenever the DB key or NMK has expired, the SC shall update the DB key or NMK, respectively, and communicate the updated keys to all authenticated nodes in the domain except the domain master, by unicasting the AKM_DomainKeyUpdate.req message. This message shall always be sent encrypted with the NSC key of the corresponding destination node. Each node that receives the request AKM_DomainKeyUpdate.req message shall confirm with the AKM_DomainKeyUpdate.cnf message. If the SC does not receive the confirmation AKM_DomainKeyUpdate.cnf message within 800 ms it shall resend the AKM_DomainKeyUpdate.req message to each node from which it has not received an AKM_DomainKeyUpdate.cnf message.

The SC shall send the AKM_DomainKeyUpdate.req message to the domain master after receiving an AKM_DomainKeyUpdate.cnf message from all the other nodes in the domain. Upon receiving the AKM_DomainKeyUpdate.req message, the domain master shall advertise that the updated NMK or DB key is going to take effect starting from the MAC cycle that is specified in the UpdateMacCycle field in the auxiliary NMK_DB_update sub-field in the MAP. A node that concludes according to the advertisement in the MAP that it did not receive the updated key shall request the updated key from the SC by sending the request AKM_KeyUpdate.req message.

Any authenticated node can also detect FN expiration of the NMK or DB and request an updated set of DB or NMK by sending to the SC an AKM_KeyUpdate.req message. The 'Request Reason' field in the AKM_KeyRequest.req message used to initiate the key generation shall be set to 'key update due to FN expiration' (see Table 9-12). The SC shall then run the procedure for updating the expired DB key or NMK as described in this clause..

9.2.5 Messages supporting AKM procedures

9.2.5.1 Authentication messages

9.2.5.1.1 Format of AUT_NodeAuthentication.req

The AUT_NodeAuthentication.req message is a unicast management message intended to be used for authentication request only. The format of the MMPL of the AUT_NodeAuthentication.req message shall be as shown in Table 9-8.

Table 9-8 – Format of the MMPL of the AUT_NodeAuthentication.req message

Field	Octet	Bits	Description
Value of X	0 to 271	[2175:0]	Value of X as per [ITU-T X.1035] .
Re-authentication flag	272	[0]	Shall be set to zero for first authentication request and to one if the request is for re-authentication.
Attempt number		[2:1]	Shall be set to 00 ₂ for the initial request and incremented for every next attempt.
ProxyAuth		[3]	Proxy authentication flag; shall be set to one for authentication through proxy and zero otherwise.
Reserved		[7:4]	Reserved by ITU-T (Note 1).
ProxyDevID	273	[7:0]	Device ID of the authentication proxy (Note 2).
NOTE 1 – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			
NOTE 2 – This field shall be set to zero by the transmitter and ignored by the receiver when the ProxyAuth field is set to zero.			

9.2.5.1.2 Format of AUT_Prompt.ind

The AUT_Prompt.ind message is a unicast management message intended to be used for communication of the prompt computed by the Authenticator. The format of the MMPL of the AUT_Prompt.ind message shall be as shown in Table 9-9.

Table 9-9 – Format of the MMPL of the AUT_Prompt.ind message

Field	Octet	Bits	Description
Value S ₁	0 to 15	[127:0]	Value of S ₁ as per [ITU-T X.1035]
Value Y	16 to 287	[2175:0]	Value of Y as per [ITU-T X.1035]
StatusCode	288	[0]	Shall be set to zero if the X-value was accepted or one otherwise
Reserved		[7:1]	Reserved by ITU-T (Note)
NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

9.2.5.1.3 Format of AUT_Verification.rsp

The AUT_Verification.rsp message is a unicast management message that is intended to communicate to the Authenticator the variables computed for prompt verification by the Supplicant. The format of the MMPL of the AUT_Verification.rsp message shall be as shown in Table 9-10.

Table 9-10 – Format of the MMPL of the AUT_Verification.rsp message

Field	Octet	Bits	Description
Value S ₂	0 to 15	[127:0]	Value of S ₂ as per [ITU-T X.1035]
StatusCode	16	[0]	Shall be set to zero if both the S ₁ -value and Y-value were accepted or one otherwise
Reserved		[7:1]	Reserved by ITU-T (Note)
NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

9.2.5.1.4 Format of AUT_Confirmation.cnf

The AUT_Confirmation.cnf message is a unicast management message intended to communicate confirmation of authentication from the Authenticator to the Supplicant, and grant the Supplicant the DB key and the NMK. The format of the MMPL of the AUT_Confirmation.cnf message shall be as shown in Table 9-11.

Table 9-11 – Format of the MMPL of the AUT_Confirmation.cnf message

Field	Octet	Bits	Description
Security mode	0	[1:0]	00 ₂ – point-to-point (Note 1) 01 ₂ – single key per domain (NMK) (Note 1) 10 ₂ , 11 ₂ – reserved by ITU-T (Note 2)
Confirmation flag		[3:2]	00 ₂ – Success 01 ₂ – Failure 10 ₂ , 11 ₂ – Reserved by ITU-T
DB Key Present		[4]	0 if DB Key is not present 1 if DB Key is present
NMK Present		[5]	0 if NMK is not present 1 if NMK is present
DB Key ID		[6]	The current DB key ID to use for encryption. This field shall be ignored if DB Key Present is set to 0.
NMK ID		[7]	The current NMK ID to use for encryption. This field shall be ignored if NMK Present is set to 0.
DB0 key		Variable	[127:0]
DB1 key	Variable	[127:0]	Encryption key for broadcast communications with keyId 1. This field only exists if DB Key Present is set to 1.
NMK0	Variable	[127:0]	NMK with keyID 0, if security mode is set to 01 ₂ . This field shall be skipped if security mode is set to 00 ₂ . This field only exists if NMK Present is set to 1.
NMK1	Variable	[127:0]	NMK with keyID 1, if security mode is set to 01 ₂ . This field shall be skipped if security mode is set to 00 ₂ . This field only exists if NMK Present is set to 1.

NOTE 1 – For the first key exchange with the security controller, if the security mode is a point-to-point, DB Key Present shall be set to one, and NMK Present shall be set to zero; and if the security mode is a single key per domain, both DB Key Present and NMK Present shall be set to one.
NOTE 2 – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.

9.2.5.1.5 Format of AUT_NodeAuthenticated.req

The AUT_NodeAuthenticated.req message is a unicast management message intended to inform the DM that the specified authenticating node is authenticated by the SC and the DM shall include it in the routing table. The format of the MMPL of the AUT_NodeAuthenticated.req message shall be as shown in Table 9-11.1.

Table 9-11.1 – Format of the MMPL of the AUT_NodeAuthenticated.req message

Field	Octet	Bits	Description
SC	0	[7:0]	DEVICE ID of the SC
AUTH	1	[7:0]	DEVICE ID of the authenticated node
Proxy	2	[7:0]	DEVICE ID of the proxy node

9.2.5.1.6 Format of AUT_NodeAuthenticated.cnf

The AUT_NodeAuthenticated.cnf message is a unicast management message intended to inform the SC that the DM received successfully the AUT_NodeAuthenticated.req message. The format of the MMPL of the AUT_NodeAuthenticated.cnf message shall be as shown in Table 9-11.2.

Table 9-11.2 – Format of the MMPL of the AUT_NodeAuthenticated.cnf message

Field	Octet	Bits	Description
AUTH	0	[7:0]	DEVICE ID of the authenticated node
Proxy	1	[7:0]	DEVICE ID of the proxy node

9.2.5.2 Pair-wise authentication messages

9.2.5.2.1 Format of AKM_KeyRequest.req

The AKM_KeyRequest.req message is a unicast management message intended to be used for communication request by the supplicant only. It is limited to 248 addressees. The format of the MMPL of the AKM_KeyRequest.req message shall be as shown in Table 9-12.

Table 9-12 – Format of the MMPL of the AKM_KeyRequest.req message

Field	Octet	Bits	Description
Number of Addressees	0	[7:0]	Number of addressees N: – for unicast transmission, N=1 – for multicast transmission, valid values of N are from 1 to 248.
Multicast stream identifier	1	[7:0]	Shall be set to the multicast stream identifier (MSID) for multicast keys. Otherwise it shall be set to 0.
Addressee name	2	[7:0]	First addressee unicast DEVICE_ID.
Addressee name	3	[7:0]	Second addressee unicast DEVICE_ID.
...
Addressee name	N+1	[7:0]	N-th addressee unicast DEVICE_ID.
Attempt number	N+2	[1:0]	Shall be set to 00 ₂ for the initial request and incremented for every next attempt.
KeyID		[2]	Set to zero to request key with ID = 0 or set to one to request key with ID = 1.
Request Reason		[4:3]	00 for first key generation, 01 for key update due to FN expiration, 10-11 are reserved by ITU-T
Reserved		[7:5]	Reserved by ITU-T (Note).

NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.

9.2.5.2.1.1 Format of AKM_AddClient.req

The AKM_AddClient.req message is a unicast management message intended to be used for joining a node to a multicast group originated by the supplicant only. It is limited to one addressee. The MMPL of the AKM_AddClient.req message is presented in Table 9-13.

Table 9-13 – Format of the MMPL of the AKM_AddClient.req message

Field	Octet	Bits	Description
Addressee name	0	[7:0]	The addressee unicast DEVICE_ID.
Multicast stream identifier	1	[7:0]	Shall be set to the multicast stream identifier (MSID) for multicast keys.
NN _{SA} key 0/1	2 to 17	[127:0]	Encryption key for Supplicant-to-Addressee direction with ID=0 if KeyID=0 and with ID=1 if KeyID=1
Attempt number	18	[1:0]	Shall be set to 00 ₂ for the initial request and incremented for every next attempt.
KeyID		[2]	Set to zero for key with ID = 0 and set to one for key with ID = 1
Reserved		[7:3]	Reserved by ITU-T (Note).
NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

9.2.5.2.2 Format of AKM_NewKey.req

The AKM_NewKey.req message is a unicast management message intended to be used for communication of the NN key from the SC to the addressee only. The format of the MMPL of the AKM_NewKey.req message shall be as shown in Table 9-14.

Table 9-14 – Format of the MMPL of the AKM_NewKey.req message

Field	Octet	Bits	Description
Supplicant name	0	[7:0]	Supplicant's unicast DEVICE_ID.
Reserved	1	[7:1]	Reserved by ITU-T (Note).
Multicast stream identifier	2	[7:0]	Shall be set to the multicast stream identifier (MSID) for multicast keys. Otherwise it shall be set to 0
NN _{SA} key 0/1	3 to 18	[127:0]	Encryption key for supplicant-to-addressee direction with ID=0 if KeyID=0 and with ID=1 if KeyID=1
NN _{AS} key 0/1	19 to 34	[127:0]	Encryption key for addressee-to-supplicant direction with ID=0 if KeyID=0 and with ID=1 if KeyID=1
KeyID	35	[0]	Set to zero for key with ID = 0 and set to one for key with ID = 1
Reserved		[7:1]	Reserved by ITU-T (Note)
NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

9.2.5.2.3 Format of AKM_NewKey.cnf

The AKM_NewKey.cnf message is a unicast management message intended to be used to confirm delivery of the new encryption key to the SC or to reject the communication request. The format of the MMPL of the AKM_NewKey.cnf message shall be as shown in Table 9-15.

Table 9-15 – Format of the MMPL of the AKM_NewKey.cnf message

Field	Octet	Bits	Description
Supplicant	0	[7:0]	Device ID of the supplicant associated with this key
Multicast Stream Identifier	1	[7:0]	Shall be set to the multicast stream identifier (MSID) for multicast keys. Otherwise it shall be set to 0
ACK	2	[1:0]	00 – If the addressee successfully received the new encryption key. 01 – If the addressee successfully received the new encryption key, but denies communication with supplicant (NACK). 10, 11 – Reserved by ITU-T.
Reserved		[7:2]	Reserved by ITU-T (Note).
NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

9.2.5.2.4 Format of AKM_KeyConfirmation.req

The AKM_KeyConfirmation.req message is a unicast management message intended to communicate the NN key with the actual list of addressees or the NMK from the SC to the supplicant only. The format of the MMPL of the AKM_KeyConfirmation.req message is shown in Table 9-16.

Table 9-16 – Format of the MMPL of the AKM_KeyConfirmation.req message

Field	Octet	Bits	Description
Security mode	0	[1:0]	00 – Point-to-Point. 01 – Single key per domain (NMK). 10 – Update of the DB keys. 11 – Reserved by ITU-T.
KeyID		[2]	Set to zero for key with ID = 0 or set to one for key with ID = 1
Reserved		[7:3]	Reserved by ITU-T (Note 1).
DB 0/1	1 to 16	[127:0]	DB key with ID=0 if KeyID=0 or with ID=1 if KeyID=1.
NMK 0/1	17 to 32	[127:0]	NMK with ID=0 if KeyID=0 or with ID=1 if KeyID=1, if security mode is 01. This field shall be skipped if security mode is 00. All of the fields describing NN keys shall be skipped if security mode is 01.
NN _{SA} key 0/1	33 to 48	[127:0]	Encryption key for supplicant-to-addressee direction with ID=0 if KeyID=0 or with ID=1 if KeyID=1
NN _{AS} key 0/1	49 to 64	[127:0]	Encryption key for addressee-to-supplicant direction with ID=0 if KeyID=0 or with ID=1 if KeyID=1
Number of addressees	65	[7:0]	Number of addressees N (1 for unicast transmission and up to 248 for multi-cast transmission) (Note 2).
Addressee name	66	[7:0]	First addressee unicast DEVICE_ID.
Addressee name	67	[7:0]	Second addressee unicast DEVICE_ID.
...
Addressee name	65 + N	[7:0]	N-th addressee unicast DEVICE_ID.
NOTE 1 – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			
NOTE 2 – If no addressee is authenticated, the list is empty and the field shall be set to zero.			

9.2.5.2.5 Format of SC_DMRes.req

The SC_DMRes.req message is a unicast management message sent by the SC to the domain master and is intended to inform the domain master that a particular node(s) has to be forced out of the domain due to authentication failure. This message is invalid if the SC and the domain master functions are performed by the same node. The MMPL of the SC_DMRes.req message shall be as presented in Table 9-17.

Table 9-17 – Format of the MMPL of the SC_DMRes.req message

Field	Octet	Bits	Description
Number entries	0	[7:0]	Indicates the number of nodes (n) in the following list, represented as an unsigned integer.
Entry 1	1	[7:0]	DEVICE_ID of the first node that is requested to be expelled from the domain.
...
Entry n	n	[7:0]	DEVICE_ID of the last node that is requested to be expelled from the domain.

9.2.5.2.6 Format of SC_DMRes.cnf

The SC_DMRes.cnf message is a unicast management message sent by the domain master to the SC to confirm the receipt of the SC_DMRes.req message from the SC. The MMPL of the SC_DMRes.cnf message shall be as presented in Table 9-18.

Table 9-18 – Format of the MMPL of the SC_DMRes.cnf message

Field	Octet	Bits	Description
Number entries	0	[7:0]	Indicates the number of nodes (n) in the following list, represented as an unsigned integer.
Entry 1	1	[7:0]	DEVICE_ID of the first node expelled from the domain.
...
Entry n	n	[7:0]	DEVICE_ID of the last node requested to be expelled from the domain.

9.2.5.2.7 Format of AKM_NewKey.ind

The AKM_NewKey.ind message is a unicast management message that shall only be used to inform the addressee that the supplicant received the NN key and communication using this new NN key is available. The MMPL of the AKM_NewKey.ind message shall be empty.

9.2.5.3 Key updating messages

9.2.5.3.1 Format of AKM_KeyUpdate.req

The AKM_KeyUpdate.req message is a unicast management message intended to be used for node re-authentication and update of the:

- NSC key, or
- NN keys or NMK, or
- DB key.

The format of the MMPL of the AKM_KeyUpdate.req message shall be as shown in Table 9-19.

Table 9-19 – Format of the MMPL of the AKM_KeyUpdate.req message

Field	Octet	Bits	Description
Supplicant	0	[7:0]	Device ID of the supplicant associated with this key. This field shall be set to FF ₁₆ if NSC, DB or NMK is updated.
Multicast Stream Identifier	1	[7:0]	Shall be set to the multicast stream identifier (MSID) for multicast keys. Otherwise it shall be set to 00 ₁₆
Type of the key	2	[1:0]	00 for NSC, 01 for NN or for NMK. 10 for DB, 11 is reserved by ITU-T.
KeyID		[2]	Set to 0 to request keys with ID = 0 or set to 1 to request keys with ID = 1
Request reason		[4:3]	00 for FN expiration, 01 for routine update, 10-11 are reserved by ITU-T
Reserved		[7:5]	Reserved by ITU-T (Note).
Authenticator	3	[7:0]	This field shall be set to the DEVICE_ID of the node requesting the key update.
Attempt number	4 and 5	[1:0]	Shall be set to 00 ₂ for the initial request and incremented for every next attempt.
Reserved		[3:2]	Reserved by ITU-T (Note).
Last update		[15:4]	Indicates time from the last successful update in minutes. Special value FFF ₁₆ indicates any period longer than 4095 minutes.
NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

9.2.5.3.2 Format of AKM_DomainKeyUpdate.req message

The AKM_DomainKeyUpdate.req message is a unicast management message the SC sends to the domain master to update the DB key or the NMK. The format of the MMPL of the AKM_DomainKeyUpdate.req message shall be as shown in Table 9-20.

Table 9-20 – Format of the MMPL of the AKM_DomainKeyUpdate.req message

Field	Octet	Bits	Description
Key Type	0	[0]	0 if DB Key is present 1 if NMK is present
Key ID		[1]	The key ID of the updated key.
Reserved		[7:2]	Reserved by ITU-T (Note)
Transaction_ID	1	[7:0]	Transaction identification for this key update. For each update of NMK or DB key. The Transaction_ID value is incremented by one until 255 and then wraparound to 0.
DB NMK_key	2 to 17	[127:0]	Encryption key of the DB or NMK according to the value set to Key_Type field.
NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

9.2.5.3.3 Format of AKM_DomainKeyUpdate.cnf message

Each node shall confirm reception of the AKM_DomainKeyUpdate.req by sending to the SC the AKM_DomainKeyUpdate.cnf message. The format of the MMPL of the AKM_DomainKeyUpdate.cnf message shall be as shown in Table 9-21.

Table 9-21 – AKM_DomainKeyUpdate.cnf message format

Field	Octet	Bits	Description
SC	0	[7:0]	Device ID of the SC node that sent the AKM_DomainKeyUpdate.req message.
NODE	1	[7:0]	Device ID of the node that sends this message.
Transaction_ID	2	[7:0]	The transaction identification that was specified in the confirmed AKM_DomainKeyUpdate.req message.

9.2.6 Re-establishment of the security controller

The operation of security controller shall be monitored by the domain master or other endpoint nodes using the criteria described in clause 9.2.6.1.

In case of failure of the SC based on the routing table provided by the DM:

- If the SC is assigned to a specific node and can only be changed by user/operator control, and no backup SC is assigned, the nodes may continue using the existing encryption keys for a maximum of 60 minutes provided they are still valid. If the SC does not recover within this time period, all endpoint nodes shall consider themselves as resigned when the timeout expires.
- If the SC is assigned to a specific node and can only be changed by user/operator control, and a backup SC is assigned, the DM shall re-establish the SC functionality using the procedure described in clause 9.2.6.3.
- If the SC is allowed to change autonomously (in this case the SC and DM are co-located see clause 9.2.1), the new DM shall re-establish the SC functionality by acting as the new SC using the procedure described in clause 9.2.6.2.

9.2.6.1 Security controller failure detection

If the DM and the SC are co-located, the node assigned as a backup of the domain master shall monitor the status of the domain master using the criteria described in clause 8.6.5.3.1. If no backup domain master is assigned or a node that is assigned to be a backup fails, a domain master-capable node shall detect the failure of the domain master using the criteria described in clauses 8.6.5.3.1 and 8.6.6.2. Whichever node detects the failure of the domain master, shall infer that the security controller has also been lost.

If the DM and the SC are not co-located, the DM shall monitor the operation of the SC through the periodic re-registration mechanism described in clause 8.6.1.1.2.

9.2.6.2 Security controller re-establishment procedure when the SC is allowed to change autonomously

If the SC is allowed to change autonomously (see clause 9.2.1), the backup domain master (if assigned) or a domain master-capable node (if no backup DM is assigned) shall take both the role of the DM and the SC, when it detects the failure of the node that plays the role of the DM and the SC.

The new domain master shall authenticate itself internally as soon as possible after it takes the role of the DM and SC successfully.

If an endpoint node detects the change of the DM and the SC by deriving the SA of the LCDU of the MAP or RMAP frames, it shall authenticate to the new SC as soon as possible. If, after 1 second, some endpoint node has still not requested authentication, the SC shall request that node to authenticate. If, after 200 ms, after the request from the SC, if the endpoint node still has not requested authentication, that node shall be resigned by the DM.

The new SC may then update the NMK key or DB key or both. The nodes shall continue using the existing encryption keys until the keys are updated by the new SC.

A node may distribute the current multicast key to a new joining node that has authenticated to the new SC using the procedure specified in clause 9.2.3.1.

9.2.6.3 Security controller re-establishment procedure when a backup SC is assigned

If a backup SC is assigned, the DM shall convey the information of the backup SC to a new joining node by including the `DEVICE_ID` and `REGID` of the backup SC in the `ADM_DmRegistrResponse.cnf` message.

If the DM and SC are co-located, the backup DM (if assigned) or a DM-capable node (if no backup DM is assigned) will become the new DM according to the DM recovery procedure in case of the failure of the SC (see clause 8.6.5.3.2 and clause 8.6.6.2). The new DM shall authenticate to the backup SC as soon as possible after it becomes the DM of the domain. After the successful authentication, the backup SC will take the role of the SC function, and the new DM shall send a `TM_DomainRoutingChange.ind` message, in which the `CurrentSC_DeviceID` field and `CurrentSC_RegID` field are set to the `DEVICE_ID` and `REGID` of the backup SC respectively, to indicate that the backup SC has become the new SC of this domain. Within 1 s after the backup SC has become the new SC of the domain, the endpoint nodes in this domain shall request to be authenticated by the new SC. If the new DM fails to authenticate to the backup SC, it shall resign all nodes from the domain and stop operation.

If the DM and SC are non-co-located, the DM can detect the failure of the SC (see clause 9.2.6.1). When the DM detects the failure of the SC, it shall authenticate to the backup SC as soon as possible. If the backup SC and the DM are co-located, the DM shall authenticate itself internally. After the successful authentication, the backup SC will take the role of the SC function, and the DM shall send a `TM_DomainRoutingChange.ind` message, in which the `CurrentSC_DeviceID` field and `CurrentSC_RegID` field are set to the `DEVICE_ID` and `REGID` of the backup SC respectively, to indicate that the backup SC has become the new SC of this domain. Within 1 s after the backup SC has become the new SC of the domain, the endpoint nodes in this domain shall request to be authenticated by the new SC. If the DM fails to authenticate to the backup SC, it shall resign all nodes from the domain and stop operation.

If, after 1 s, some endpoint node has still not requested authentication, the new SC shall request that node to authenticate. If, after 200 ms after the request from the new SC, the endpoint node still has not requested authentication, that node shall be resigned by the DM.

The new SC may then update the NMK key or DB key or both. The nodes shall continue using the existing encryption keys until the keys are updated by the new SC.

A node may distribute the current multicast key to a new joining node that has authenticated to the new SC using the procedure specified in clause 9.2.3.1.

If a backup SC is assigned, but it has not registered to the domain or has failed, the domain master shall consider no backup SC is assigned.

9.2.7 Handover of SC function

If the SC is allowed to change autonomously (in this case the SC and DM are co-located), when the DM hands the domain master's role to a more capable node (see clause 8.6.6.4), the SC function shall be also handed over to the same node.

When the new domain master starts transmitting MAP frames to take the role of the domain master, the nodes in its domain shall be able to detect the change of SC by deriving the SA of the LCDU of the MAP or RMAP frames and authenticate to the new SC as soon as possible.

The SC re-establishment procedure in this case is as described in clause 9.2.6.2.

Annex A

Application protocol convergence sublayer

(This annex forms an integral part of this Recommendation.)

Application protocol convergence (APC) specific sublayer maps the primitives of the application protocol used by the application entity (AE) into the native protocol of the data link layer. It is the responsibility of the APC to convert incoming data units of the particular application protocol used by the AE into APDUs, classify these APDUs into one or more traffic types (classes of services), and map them onto appropriate flows.

Each flow is associated with a particular service or traffic type with well-defined QoS requirements. Flows are established by the APC upon receipt of relevant data units from the AE, or during admission to the network, or by management requests coming across the A-interface, or upon demand from another node (by means of a flow establishment protocol message coming across the x1 reference point). The type of traffic for the flow and its other QoS related parameters are defined based on classification performed by APC.

By default, APC shall support Ethernet, while other protocols can also be supported.

The description of APC in this annex is partitioned into a data plane and a management plane. The data plane part specifies converging of the AE data units into APDUs and back. The functional model of the data plane is presented in Figures 8-1 and 8-2. The management plane part specifies APC primitives and protocols related to support different traffic classes, QoS related issues, and APC peer-to-peer management.

A.1 Ethernet APC (EAPC)

The EAPC is intended to operate with an Ethernet AE which supports IEEE bridging and switching protocols such as [IEEE 802.1D], [IEEE 802.1Q], [IEEE 802.1ad] (QinQ). Inter-domain bridging and bridging to alien domains implemented by the AE are beyond the scope of this Recommendation. The APC converts the standard set of primitives (at the MAC SAP, in terms of IEEE 802.3, and those defined as internal sublayer services in terms of IEEE 802.1) at the A-interface into an APDU, which is further communicated through the domain to the peer APC. The APC shall accommodate the differences in primitive sets of different versions of [IEEE 802.3] and IEEE 802.1 by substituting default values, as described in clause A.1.1.

A.1.1 Frame conversion

The incoming set of primitives (AIF_DATA.REQ) and the outgoing set of primitives (AIF_DATA.IND) at the A-interface of EAPC represent a sequence of Ethernet frames, each defined as a set of IEEE 802.1 primitives of M_UNITDATA.request and M_UNITDATA.indication, respectively, Table A.1.

Table A.1 – A-interface primitives description

AIF_DATA.REQ (AE → EAPC)	AIF_DATA.IND (EAPC → AE)
M_UNITDATA.request (frame_type, destination_address, source_address, mac_service_data_unit, user_priority, access_priority, frame_check_sequence)	M_UNITDATA.indication (frame_type, destination_address, source_address, mac_service_data_unit, user_priority, frame_check_sequence)

All unit-signal primitives specified in Table A.1 shall be interpreted in terms of clause 6.4 of [IEEE 802.1D]. Note that primitives frame_type, user_priority, access_priority, and frame_check_sequence, may not be provided by the AE, and primitives frame_type and user_priority may not be requested by the AE.

NOTE 1 – Clause 6.5.1 of [IEEE 802.1D] suggests that the "access priority" primitive be ignored and the frame_type primitive be set to user_data_type for 802.3 MAC frames.

NOTE 2 – The M_UNITDATA.request description in [IEEE 802.1Q] differs from that in [IEEE 802.1D] as it omits the frame_type and access_priority parameters. The frame_type is not required in [IEEE 802.1Q] as the receipt of a frame other than a user data frame does not cause a data indication, nor are such frames transmitted by the medium independent bridge functions. The mapping of M_UNITDATA.request to particular access methods specified in [IEEE 802.1Q] includes derivation of the access_priority parameter (for those media that require it) from the user_priority parameter.

NOTE 3 – The EM_UNITDATA.request and EM_UNITDATA.indication description in [IEEE 802.1ad] includes more QoS related primitives, such as drop_eligible and others. These primitives, similarly to those defined in clause 6.6.1 of [IEEE 802.1Q] should be accommodated in the corresponding tags fields of the APDU as described in Table A.1.

If the frame_check_sequence primitive is provided by the AE, the incoming M_UNITDATA.request primitives (described in Table A.1 for AIF_DATA.REQ) shall be verified to be error free by computing their FCS as defined in clause 6.5.1 of [IEEE 802.1D]. If the computed FCS does not match the received value of the frame_check_sequence, the incoming primitive shall be discarded. If the frame_check_sequence primitive is not provided by the AE, the APC shall compute the FCS of the incoming M_UNITDATA.request primitives as defined in clause 3.28 of [IEEE 802.3].

Error-free primitives described in Table A.1 for AIF_DATA.REQ shall be converted into the APDU format presented in Figure A.1. The same APDU format shall be used for in-band management messages sourced by the local DLL management entity for the remote AE.

6 octets	Destination address
6 octets	Source address
TG octets	VLAN TAGs
2 octets	MAC client length/EtherType
Application dependent	Service data unit (APDU payload)
4 octets	Frame check sequence (FCS)

Figure A.1 – APDU format (TX and RX)

All fields shall have the same content as the corresponding fields of the MAC frame defined in [IEEE 802.3], including various embedded tags mapped into the VLAN TAGs field. Mapping of the unit-data primitives, including embedded tags, into all these APDU fields shall comply with the [IEEE 802.3] or relevant IEEE bridging standard, such as [IEEE 802.1D], [IEEE 802.1Q], etc. The VLAN TAGs field shall only be present (i.e., $TG > 0$) for:

- Single-tagged MAC frames according to [IEEE 802.1Q] (8100_{16} , VLAN-tagged frames, $TG = 4$) or
- Single-tagged MAC frames according to [IEEE 802.1ad] ($88A8_{16}$, provider bridging, $TG = 4$) or
- Double-tagged MAC frames according to [IEEE 802.1ad] ($88A8_{16}$ for the 4-byte outer tag, followed by 8100_{16} for the 4-byte inner tag, $TG = 8$).

NOTE 4 – [IEEE 802.1ad] is an amendment to [IEEE 802.1Q].

NOTE 5 – Usage of tags 9100_{16} , 9200_{16} and 9300_{16} has been deprecated by IEEE.

Otherwise, TG shall be set to zero, and the 2 octets after the source address are considered as MAC client length/EtherType field. If AE provides neither `frame_type`, nor `access_priority` or `user_priority` primitives, the VLAN TAGs field of the APDU shall be zero octets long.

When sending a broadcast in-band management message (e.g., message addressed to all ITU-T G.hn nodes), the EtherType tag shall be set to $22E3_{16}$.

The unencrypted part of the APDU shall include all bytes starting from the first byte of the APDU and ending at the last byte of the "MAC client length/EtherType" field of the APDU. The length of the unencrypted part of the APDU depends on the length TG of the VLAN TAGs field of the APDU (see clause 9.1.2.2).

The FCS of APDU shall be used only if MIC is not used as a part of the encryption scheme (see clause 9.1.1); otherwise, the FCS shall be stripped off and not communicated through the domain.

NOTE 6 – Since the FCS is stripped off and reconstructed by the remote APC in the case MIC is included, verification of the incoming `M_UNITDATA.request` primitives to be error free is essential in order to avoid the creation and propagation of frames with undetectable errors.

Bits of APDU shall be transmitted starting from the first octet of the destination address. The least significant bit of each octet shall be transmitted first. The most significant octet of each field shall be transmitted first.

The order of outgoing APDUs at the x1 reference point associated with a particular destination and particular user priority shall be the same as the order of incoming unit-data of these same user priority and destination. No re-ordering inside the same user priority group for the same destination is allowed.

The M_UNITDATA.indication primitives shall be derived from the APDUs received from the LLC across the x1 reference point as defined in clause 6.4.1 of [IEEE 802.1D], with the following additional rules:

- The user_priority primitive shall be derived from the TAGs field for all embedded tags as defined in clauses 6.6.1 and 9 of [IEEE 802.1Q]; if TAGs field is of zero length, the user_priority primitive shall be set to zero.
- The frame_check_sequence primitive, if FCS is not a part of APDU, shall be computed as defined in clause 3.28 of [IEEE 802.3].
- The frame_check_sequence primitive, if FCS is a part of APDU, shall be verified as defined in clause 6.5.1 of [IEEE 802.1D]. APDUs that did not pass verification shall be discarded.

The same rules shall also be used to derive the M_UNITDATA.indication primitives for the in-band management messages sourced by the DLL management entity for the local AE.

The payload of in-band management data units generated by the DLL management entity shall follow the LCDU payload format defined in clause 8.1.3.4.

A.1.2 Classification

EAPC may perform classification of outgoing APDUs based on the following criteria:

- destination MAC address;
- source MAC address;
- VLAN priority (802.1Q and relevant amendments);
- VLAN ID (802.1Q and relevant amendments);
- IP ToS [b-IETF RFC 791] DSCP [b-IETF RFC 2474];
- IGMP/MLD [b-IETF RFC 3376].
- TCP [b-IETF RFC 793] /UDP [b-IETF RFC 768] Port Number.
- EtherType.
- Other classification parameters are for further study.

Notice that the presented criteria are just possible options; the rules of how to classify APDUs and to which flow or user priority a particular APDU has to be assigned, except those presented below, are beyond the scope of this Recommendation and left for implementers.

In case of classification using VLAN priority, the EAPC shall be capable of recognizing eight standard priority levels (priority tags) defined in [IEEE 802.1Q].

The EAPC shall identify all incoming in-band management data units addressed to the node (i.e., those for which the destination MAC address coincides with the MAC address of the node) and direct them to the DLL management entity. Those include in-band management messages arriving from the AE across the A-interface (local in-band management messages) and arriving from the LLC across the x1 reference point (remote in-band management messages). Other criteria for classification of the incoming APDU crossing the x1 reference point are beyond the scope of this Recommendation.

A.1.3 Flow control

Flow control at the A-interface is necessary to avoid packet loss in the case when the traffic generated by the source AE exceeds the throughput of the link between the source and the destination node. The flow control may be implemented by communicating an appropriate set of AIF_DATA.IND primitives from the APC to AE (e.g., corresponding to Ethernet PAUSE frame) or a set of AIF_DATA.CNF primitives, or by appropriate signalling at the management plane. The format of AIF_DATA.CNF and signalling used for flow control is vendor discretionary.

A.1.4 Management plane

The management plane of the EAPC is for further study.

A.2 Other types of APC

Other types of APC are for further study.

Annex B to Annex W

Annexes B to W have been intentionally left blank.

Annex X

Test vectors

(This annex forms an integral part of this Recommendation.)

This annex includes test vectors for core operations described in this Recommendation.

X.1 CCM encryption

This clause provides a set of test vectors for parameters involved in CCM encryption described in clause 9.1. Parameters are expressed in a hexadecimal form with the leftmost byte representing the lowest byte within a parameter (i.e., byte 0). Within a byte, the leftmost bit represents the MSB.

X.1.1 CCM test vector 1

X.1.1.1 Input parameters

This clause provides one set of examples for input parameters used in CCM encryption.

Data packet, APDU (75 bytes):

The APDU can be broken into the following parameters:

Destination MAC address, *DA* (6 bytes)

$DA = 00\ 00\ 5E\ 10\ 20\ 09_{16}$

Source MAC address, *SA* (6 bytes)

$SA = 00\ 00\ 5E\ 07\ 20\ 13_{16}$

MAC client length/type, *LT* (2 bytes)

$LT = 08\ 00_{16}$

APDU payload, *P* (57 bytes)

$P = 52\ 41\ 56\ 49\ 3B\ 45\ 52\ 45\ 5A\ 3B\ 41\ 56\ 4E\ 45\ 52\ 3B\ 4C\ 45\ 53\ 3B\ 52\ 4F\ 59$
 $3B\ 4D\ 41\ 52\ 43\ 4F\ 53\ 3B\ 41\ 47\ 55\ 53\ 54\ 49\ 4E\ 3B\ 4A\ 4F\ 48\ 4E\ 3B$
 $4A\ 42\ 3B\ 54\ 4F\ 4E\ 47\ 3B\ 56\ 5A\ 45\ 49\ 42_{16}$

Frame check sequence, *FCS* (4 bytes)

$FCS = A4\ 55\ 5A\ 26_{16}$

Note that VLAN TAG does not exist in this example (i.e., $TG = 0$), *Alen* is 21 bytes (see Table 9-5), and *Plen* is 57 bytes.

Parameters for the LLC frame header (LFH) are given as LLCFT = 2 (APDU), TSMPI = 0 (TSMPI field not present), CCMPI = 1, LPRI = 0, FLEN = 71, MCSTI = 0, OriginatingNode = 1, DestinationNode = 2, BRCTI = 0, and TTL = 0 (see clause 8.1.3.1.1).

Resulting LFH (6 bytes):

$LFH = 12\ 47\ 00\ 01\ 02\ 00_{16}$

Parameters for the CCMP header (CCMPH) are given as MIC size = 111_2 (16-byte MIC), NN or NMK, Key ID = 0, and FN = 1 (see clause 9.1.2.3).

Resulting CCMP header, CCMPH (6 bytes):

$CCMPH = 07\ 01\ 00\ 00\ 00\ 00_{16}$

Encryption Key K: 47 68 6F 43 65 72 74 66 32 30 31 33 47 68 6E 43₁₆.

X.1.1.2 Parameters generated

This clause provides one set of examples of parameters generated by the node based on given set of input parameters.

Nonce, N (13 bytes):

$N = 00\ 00\ 00\ 5E\ 07\ 20\ 13\ 00\ 00\ 00\ 00\ 00\ 01_{16}$

Nonce block, B_0 (16 bytes):

$B_0 = 79\ 00\ 00\ 00\ 5E\ 07\ 20\ 13\ 00\ 00\ 00\ 00\ 00\ 01\ 00\ 39_{16}$

Associated data block, B_1 (16 bytes):

$B_1 = 00\ 15\ 00\ 00\ 00\ 00\ 5E\ 10\ 20\ 09\ 00\ 00\ 5E\ 07\ 20\ 13_{16}$

Associated data block, B_2 (16 bytes):

$B_2 = 12\ 47\ 00\ 01\ 02\ 08\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00_{16}$

Payload block, B_3 (16 bytes):

$B_3 = 52\ 41\ 56\ 49\ 3B\ 45\ 52\ 45\ 5A\ 3B\ 41\ 56\ 4E\ 45\ 52\ 3B_{16}$

Payload block, B_4 (16 bytes):

$B_4 = 4C\ 45\ 53\ 3B\ 52\ 4F\ 59\ 3B\ 4D\ 41\ 52\ 43\ 4F\ 53\ 3B\ 41_{16}$

Payload block, B_5 (16 bytes):

$B_5 = 47\ 55\ 53\ 54\ 49\ 4E\ 3B\ 4A\ 4F\ 48\ 4E\ 3B\ 4A\ 42\ 3B\ 54_{16}$

Payload block, B_6 (16 bytes):

$B_6 = 4F\ 4E\ 47\ 3B\ 56\ 5A\ 45\ 49\ 42\ 00\ 00\ 00\ 00\ 00\ 00\ 00_{16}$

Counter block 0, Ctr_0 (16 bytes):

$Ctr_0 = 01\ 00\ 00\ 00\ 5E\ 07\ 20\ 13\ 00\ 00\ 00\ 00\ 00\ 01\ 00\ 00_{16}$

Counter block 1, Ctr_1 (16 bytes):

$Ctr_1 = 01\ 00\ 00\ 00\ 5E\ 07\ 20\ 13\ 00\ 00\ 00\ 00\ 00\ 01\ 00\ 01_{16}$

Counter block 2, Ctr_2 (16 bytes):

$Ctr_2 = 01\ 00\ 00\ 00\ 5E\ 07\ 20\ 13\ 00\ 00\ 00\ 00\ 00\ 01\ 00\ 02_{16}$

Counter block 3, Ctr_3 (16 bytes):

$Ctr_3 = 01\ 00\ 00\ 00\ 5E\ 07\ 20\ 13\ 00\ 00\ 00\ 00\ 00\ 01\ 00\ 03_{16}$

Counter block 4, Ctr_4 (16 bytes):

$Ctr_4 = 01\ 00\ 00\ 00\ 5E\ 07\ 20\ 13\ 00\ 00\ 00\ 00\ 00\ 01\ 00\ 04_{16}$

Cyphertext

$C = 04\ 4f\ 24\ 21\ 07\ fb\ 58\ 68\ ba\ 1a\ c7\ c3\ 1f\ 5c\ e7\ 20\ c1\ a2\ 09\ ed\ a0\ 29\ d5$
 $03\ b1\ e0\ 94\ 43\ ed\ 4f\ 28\ 24\ 62\ c8\ 28\ c5\ 53\ 50\ 95\ 74\ 86\ fc\ ea\ 0e$
 $92\ 4d\ 2c\ 4f\ 3b\ 25\ cf\ b5\ 3a\ 5e\ 1f\ 5e\ 3f_{16}$

Message Integrity code

$MIC = 8c\ df\ 6e\ 79\ 03\ 0f\ 7e\ 69\ cc\ 33\ b8\ 29\ ef\ e4\ 6d\ e2_{16}$

X.2 PAK test vectors

This clause provides a set of test vectors for parameters involved in the PAK protocol described in clause 9.2.2. Parameters are expressed in a hexadecimal form with the leftmost bit representing the MSB.

X.2.1 PAK test vector 1

X.2.1.1 Input parameters

This clause provides one set of examples for input parameters known to the supplicant and the authenticator before the initiation of key authentication process.

Node identifier of the supplicant, A (48 bits, see clause 9.2.2.2.1):

$A = 0019\ A717\ DD30_{16}$

Node identifier of the authenticator, B (48 bits, see clause 9.2.2.2.1):

$B = 0019\ A770\ 8A32_{16}$

Node password shared by the supplicant and the authenticator, PW (96 bits, see clause 9.2.2.2.2):

$PW = 5962\ A05A\ B89F\ C0AA\ FB14\ 0EF7_{16}$

X.2.1.2 Parameters generated or exchanged

This clause provides one set of examples of parameters generated and/or exchanged by the supplicant and the authenticator.

Secret exponent generated by the supplicant, RA (384 bits, see clause 9.2.2.2.5):

$RA = 89A1\ A7B4\ F433\ 9220\ 2C60\ 960D\ 172A\ 7C45\ 6B95\ C225\ 26B1\ 1C7A$
 $9E2E\ 7712\ C43C\ 9C77\ 48B6\ 3936\ A62B\ CF90\ 3C03\ A0E2\ 0E28\ D660_{16}$

Secret exponent generated by the authenticator, RB (384 bits, see clause 9.2.2.2.5):

$RB = F052\ 57CB\ 1840\ 6A91\ 173B\ 87E4\ 1F22\ 9289\ 7D3E\ 08A7\ BCA0\ 4EB9$
 $1A8A\ CFF3\ 940C\ AE00\ E15B\ 302D\ 7E67\ 2E81\ CCB4\ C103\ A241\ B133_{16}$

Concatenated input parameters, $P = A \parallel B \parallel PW$ (192 bits, see [\[ITU-T X.1035\]](#)):

$P = 0019\ A717\ DD30\ 0019\ A770\ 8A32\ 5962\ A05A\ B89F\ C0AA\ FB14\ 0EF7_{16}$

Intermediate result, $IR1 = H_I(P)$ (1152 bits, see clause 9.2.2.2.6 and [\[ITU-T X.1035\]](#)):

$IR1 = 11F9\ E6DD\ 6E7D\ 48EF\ 3672\ CA0F\ A2EC\ 2488\ 7678\ 34B9\ 506C\ FE86$
 $5BC0\ A847\ 3051\ F6FD\ 408D\ 0178\ 816D\ 80A7\ D8D3\ B75C\ 3176\ C8D3$
 $BD12\ 2AD1\ 2AE5\ C26C\ 29F8\ 3518\ BD91\ 1581\ 9483\ C303\ 68F3\ B137$
 $3A33\ A8E5\ 6193\ 83B8\ 34F1\ 59B4\ E1C3\ 8259\ B3DA\ D35F\ 7876\ A7FE$
 $3B0A\ 9E9A\ F594\ BEA6\ B126\ 77B0\ 50EC\ 672E\ 11F7\ 3A1E\ 231E\ 9ECE$
 $793A\ 34AE\ 154D\ 4EB0\ 82BB\ AC26\ 1F8E\ 0B50\ 735C\ 01FB\ C364\ 9081_{16}$

Intermediate result, $IR2 = g^{RA} \bmod p$ (1024 bits, see clause 9.2.2.2.3, clause 9.2.2.2.4, and [\[ITU-T X.1035\]](#)):

$IR2 = D678\ B9D6\ E866\ FB46\ 4865\ A430\ C2BA\ 0668\ 722D\ 236E\ 7BEA\ 1C51$
 $7E4A\ 4812\ 1CD4\ B42C\ 7803\ 2B8C\ F05F\ 497B\ 46EC\ F894\ CB5A\ 0678$
 $7104\ 7E99\ 448A\ D384\ 46A1\ 15AF\ 4640\ 7B9B\ F13C\ FFBD\ 2452\ FB69$
 $3D7C\ 6445\ DE1E\ 95AF\ DC13\ 7B33\ 01AE\ 6659\ 0839\ A05E\ 03A2\ 2169$
 $E10C\ C5F6\ D87B\ 62E5\ FF92\ B000\ 4DA9\ 8058\ 9F95\ 5F2E\ F66A\ 42D6$

CBC4 E70A A3CA D136₁₆

Parameter carried in AUT_NodeAuthentication.req, $X = IR1 \cdot IR2$ (2176 bits, see clause 9.2.5.1.1 and [\[ITU-T X.1035\]](#)):

X = 0F0F 612E 0137 3C14 AB36 88FB 07C9 98E6 EBA7 033C E635 4EDA
54D2 DA67 46D2 43AC FC19 3F9E 7E66 4B5F 1ED8 13D7 7763 0BFF
DE60 E3D5 397E 901A 1338 99CC 2E52 209E 441F 0DDE 9449 1CDA
8B36 B454 FF1B 1E9E 784A 07D4 5DF5 85C5 503E 65AD 7E34 EE82
2E92 99AC B766 EF21 0CEB 7D10 B620 AB10 BA09 7DF7 EEB0 25BE
E6AD 223B 3049 93F9 FCDB C996 EA09 8BFC 56C7 495E 2E17 BD88
E201 B2C2 40E9 9F79 B681 9963 3D8F 5F22 7BD8 5373 A868 902D
93FC 20CB 9F1D 369B 1C54 A143 E416 D7C5 2A59 EAC8 0B49 D013
575F C302 FA4D AD02 DDF7 BA96 71E9 9B56 DE44 9E57 9DFB 83AD
B1F1 1A43 0900 2F9C 8EFD A771 0A71 DAA0 176D E5ED C7A8 02F3
99D8 6E26 0458 3EF1 901F 7C1A 99E8 CBB7 5357 09DA 84F2 5393
9F2E 3706 79F9 CC36₁₆

Intermediate result, $IR3 = H_2(P)$ (1152 bits, see clause 9.2.2.2.6 and [\[ITU-T X.1035\]](#)):

IR3 = 2773 D699 51BB 3CC5 D595 F28E 3AAF CCBF C2A3 895D D429 A707
13EE C1D7 2E08 BCA9 D3C7 AE45 7317 5180 25AE 9B9D 6125 BED6
EA69 F440 FD1F D309 2404 0AD9 E3DB B2A4 8F1A 49DA 0F14 BD2E
15B7 2E9D E16E 9E95 EE26 6890 AA45 1ACE A1A7 394C 9BFB 55B8
54DE 5CFB 1385 028D 3A58 ED53 C8B1 639C 76D4 F4AF BB51 52D8
2E7F F099 4210 DA52 CDFD DF2B 973D EC89 DFEB A32C A4B7 4428₁₆

Intermediate result, $IR4 = g^{RB} \bmod p$ (1024 bits, see clause 9.2.2.2.3, clause 9.2.2.2.4, and [\[ITU-T X.1035\]](#)):

IR4 = B503 D0FE AAC7 D9D5 B2C1 ADAC ACB2 F4AC ED7D E0EA 65F2 D88F
39DA A98C CCA3 C197 40F4 B466 6DC4 310F 6969 482F 2B94 D5A2
BB64 4E8F 04A7 12D4 81FF 34E0 45F3 E351 E255 3A57 F32F E600
820A 7C9B 0407 F35C 588D 4C6A 0908 BD7C 9F76 A9BB A478 16BB
C6F8 73DC B9EF C0CD 54FC E949 F840 2EE6 DD0C D4B9 52F9 96BC
D529 9885 964C 394D₁₆

Parameter carried in AUT_Prompt.ind, $Y = IR3 \cdot IR4$ (2176 bits, see clause 9.2.5.1.2 and [\[ITU-T X.1035\]](#)):

Y = 1BE5 7D4B 4832 8ED0 90C6 5623 C4D1 1400 F58E FDEF A37E 2AAC
0EB6 9A5E 904E F71A 193D 46B2 6113 372A 0517 45CB 1FCB 5200
2FF9 A00C 9070 72C2 5946 E87D 630E 36A2 AFCE 5FB4 AA35 D2F7
74DA FED9 11A4 4EF4 698C 4582 9E47 8AA6 F74A 6714 09E0 8CA3
9654 1D65 9099 DC16 3C40 2E8D 6779 D9CD B182 4EF6 6A83 8A40
7537 64A9 ABDB 1619 33A7 44E6 8C9A 3D37 2D34 1C46 423F 4679
B03B 563D 0B02 D397 6171 776F 7FD4 31D6 6B26 4F6A 5AC5 BD89
434C C914 2698 36A0 DC88 2E31 D3FD B108 69FA 4F86 AFAB CDDC

7CE6 D753 F7C5 5286 6E12 C2C3 80E4 70A2 6F81 08CD D379 08F5
 AE54 4467 DA86 974D BE27 39A6 5058 E201 1387 AB08 6402 15A3
 E973 5002 8852 6DD8 302B 60F8 28AF 9806 4535 F825 425C 0652
 010F 763A 052C 6808₁₆

Intermediate result, $IR5 = P | g^{RA} \bmod p | g^{RB} \bmod p | g^{RA \cdot RB} \bmod p$ (3264 bits, see clause 9.2.2.2.3, clause 9.2.2.2.4, and [\[ITU-T X.1035\]](#)):

$IR5 =$ 0019 A717 DD30 0019 A770 8A32 5962 A05A B89F C0AA FB14 0EF7
 D678 B9D6 E866 FB46 4865 A430 C2BA 0668 722D 236E 7BEA 1C51
 7E4A 4812 1CD4 B42C 7803 2B8C F05F 497B 46EC F894 CB5A 0678
 7104 7E99 448A D384 46A1 15AF 4640 7B9B F13C FFBD 2452 FB69
 3D7C 6445 DE1E 95AF DC13 7B33 01AE 6659 0839 A05E 03A2 2169
 E10C C5F6 D87B 62E5 FF92 B000 4DA9 8058 9F95 5F2E F66A 42D6
 CBC4 E70A A3CA D136 B503 D0FE AAC7 D9D5 B2C1 ADAC ACB2 F4AC
 ED7D E0EA 65F2 D88F 39DA A98C CCA3 C197 40F4 B466 6DC4 310F
 6969 482F 2B94 D5A2 BB64 4E8F 04A7 12D4 81FF 34E0 45F3 E351
 E255 3A57 F32F E600 820A 7C9B 0407 F35C 588D 4C6A 0908 BD7C
 9F76 A9BB A478 16BB C6F8 73DC B9EF C0CD 54FC E949 F840 2EE6
 DD0C D4B9 52F9 96BC D529 9885 964C 394D 9768 81A5 5808 E976
 F569 319A 8764 8539 16E0 1496 6E1F 191A 482B 1838 0E4F 9A77
 99FA C4AF AE0B 9C74 7A57 630C DA71 DF19 5CB2 FE5F B951 52B7
 EADB C460 8B62 3464 944E 1011 8471 028C 8000 8F8E EC8E B6C7
 FC36 30DF 27DD 2D43 3277 2FB4 E1A8 FF9F CA61 6E4E E466 CDA4
 B6AD 9B02 F498 39BF 589B C793 2680 8C26 9AA6 B351 9418 EFEB₁₆

Parameter carried in AUT_Prompt.ind, $S_1 = H_3(IR5)$ (128 bits, see clause 9.2.2.2.6, clause 9.2.5.1.2, and [\[ITU-T X.1035\]](#)):

$S_1 =$ 3BB5 5C57 33CF 1E7F 0711 C525 CD89 3181₁₆

Parameter carried in AUT_Verification.res, $S_2 = H_4(IR5)$ (128 bits, see clause 9.2.2.2.6, clause 9.2.5.1.3, and [\[ITU-T X.1035\]](#)):

$S_2 =$ 3DB1 A72C 64B0 CAE6 57FF D4EA DC31 F676₁₆

NSC key generated, $K = H_5(IR5)$ (128 bits, see clause 9.2.2.2.6 and [\[ITU-T X.1035\]](#))

$K =$ ABA6 D8E8 BD2B 705B B4CC 34BD 1107 E00D₁₆

X.3 Huffman coding

Huffman coding may be used to compress management frame payloads as described in clause 8.24.

X.3.1 Test vector 1

X.3.1.1 Input (uncompressed) management message payload (MMPL)

MMPL (329 bytes)

Offset	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	--	0x8	0x9	0xa	0xb	0xc	0xd	0xe	0xf
0x000	0x56	0x69	0x63	0x74	0x6f	0x72	0x20	0x5a		0x65	0x69	0x62	0x20	0x42	0x6f	0x72	0x74
0x010	0x0a	0x4d	0x61	0x72	0x63	0x6f	0x73	0x20		0x4d	0x61	0x72	0x74	0x69	0x6e	0x65	0x7a
0x020	0x20	0x56	0x61	0x7a	0x71	0x75	0x65	0x7a		0x0a	0x21	0x5a	0x6e	0x68	0x2c	0x54	0x6e
0x030	0x75	0x67	0x70	0x73	0x69	0x1f	0x6c	0x6d		0x70	0x76	0x74	0x1f	0x75	0x67	0x66	0x1f
0x040	0x68	0x60	0x75	0x64	0x2f	0x1f	0x5a	0x6e		0x68	0x2c	0x54	0x6e	0x75	0x67	0x70	0x73
0x050	0x69	0x1f	0x6a	0x72	0x21	0x73	0x69	0x64		0x21	0x66	0x62	0x73	0x66	0x2d	0x21	0x58
0x060	0x70	0x66	0x2e	0x52	0x70	0x73	0x69	0x6e		0x75	0x67	0x21	0x68	0x74	0x1f	0x75	0x67
0x070	0x66	0x1f	0x6c	0x64	0x7a	0x1f	0x62	0x6d		0x65	0x1f	0x68	0x74	0x62	0x71	0x65	0x68
0x080	0x62	0x6d	0x21	0x6e	0x67	0x1f	0x75	0x67		0x66	0x1f	0x68	0x60	0x75	0x64	0x2f	0x1f
0x090	0x51	0x60	0x74	0x73	0x2d	0x1f	0x71	0x71		0x66	0x72	0x66	0x6d	0x75	0x2b	0x21	0x65
0x0a0	0x76	0x73	0x76	0x71	0x66	0x2b	0x21	0x60		0x6d	0x6b	0x21	0x60	0x73	0x64	0x21	0x6e
0x0b0	0x6f	0x64	0x21	0x68	0x6f	0x1f	0x5a	0x6e		0x68	0x2c	0x54	0x6e	0x75	0x67	0x70	0x73
0x0c0	0x69	0x2d	0x21	0x47	0x66	0x1f	0x6c	0x6d		0x70	0x76	0x74	0x1f	0x78	0x67	0x66	0x71
0x0d0	0x66	0x1f	0x75	0x67	0x66	0x1f	0x50	0x6b		0x65	0x1f	0x50	0x6d	0x66	0x72	0x21	0x61
0x0e0	0x73	0x6e	0x6c	0x64	0x21	0x73	0x69	0x71		0x70	0x74	0x68	0x67	0x21	0x6e	0x67	0x1f
0x0f0	0x70	0x6b	0x65	0x2b	0x21	0x60	0x6f	0x63		0x21	0x76	0x69	0x64	0x73	0x64	0x21	0x53
0x100	0x69	0x64	0x7a	0x1f	0x74	0x67	0x62	0x6b		0x6d	0x1f	0x63	0x71	0x66	0x60	0x6c	0x1f
0x110	0x75	0x67	0x73	0x6e	0x76	0x66	0x69	0x1f		0x62	0x66	0x62	0x68	0x6f	0x2d	0x23	0x09
0x120	0x21	0x1f	0x21	0xe1	0x81	0x92	0x49	0x2d		0x51	0x2d	0x21	0x4b	0x70	0x75	0x66	0x62
0x130	0x73	0x60	0x67	0x73	0x2d	0x1f	0x55	0x67		0x66	0x1f	0x45	0x74	0x6f	0x76	0x6a	0x62
0x140	0x69	0x1f	0x49	0x6e	0x73	0x71	0x70	0x71		0x0b							

X.3.1.2 Compressed management message payload

CMMPL (282 bytes)

Offset	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	--	0x8	0x9	0xa	0xb	0xc	0xd	0xe	0xf
0x000	0x00	0xa9	0xd4	0x95	0x8a	0xa9	0x45	0xa1		0x29	0x74	0x72	0xd5	0x56	0x30	0x37	0x87
0x010	0x87	0x24	0x58	0xec	0x22	0x17	0x09	0x4b		0xa4	0x76	0xb2	0xdb	0x65	0x14	0xda	0x4a
0x020	0xa3	0x50	0xb7	0x96	0x7b	0x99	0x60	0x5e		0xad	0x96	0x60	0xb8	0xd9	0x16	0x2b	0xa2
0x030	0x42	0xb8	0x5a	0x01	0x54	0x96	0x25	0x69		0x7c	0x2a	0xd2	0x5d	0xae	0x57	0x54	0x7d
0x040	0x6e	0x4b	0x55	0xa9	0x00	0x5c	0x89	0x33		0xeb	0xff	0xf1	0x73	0x2f	0xe0	0xa7	0xd6
0x050	0x78	0xe6	0x05	0xd4	0x43	0xaf	0xf3	0x02		0xeb	0x3c	0xf8	0xad	0x7f	0x0a	0x2b	0x50
0x060	0xd9	0x5a	0xc7	0xaf	0xde	0xbe	0x1c	0x1e		0xda	0x59	0x73	0xcd	0x92	0xda	0x4f	0xb6
0x070	0x8f	0xd7	0x66	0xe3	0x8d	0xdf	0xbd	0x7c		0x38	0x3d	0xb4	0xb2	0xe7	0x82	0x6b	0x4b
0x080	0x98	0xd3	0xc9	0x5f	0xea	0x02	0xb3	0x8d		0x00	0xd9	0x73	0xed	0xa5	0x5e	0x7d	0xb4
0x090	0x7e	0x6e	0x2d	0x74	0x88	0x5e	0xbc	0xe5		0x02	0xde	0x44	0xaf	0x27	0x6d	0x1f	0xae
0x0a0	0xcd	0xc7	0x1b	0x8d	0x32	0x6b	0xf7	0x42		0x0f	0xab	0x93	0x78	0xa8	0xba	0x2e	0x90
0x0b0	0x7c	0x54	0xc2	0x0f	0x4c	0x58	0xd7	0x8f		0x1a	0xb9	0xfb	0xf7	0xaf	0x87	0x07	0xb6
0x0c0	0x96	0x5c	0xfd	0x40	0xef	0xcd	0x92	0xda		0x4f	0x83	0x51	0xe0	0xfd	0xb4	0x7e	0x36
0x0d0	0x39	0x78	0xd9	0x1f	0x5a	0x15	0x7c	0xdc		0x69	0x73	0x0b	0x4e	0xe9	0x5e	0x4e	0xb0
0x0e0	0xe5	0xc5	0x4c	0x38	0x95	0xa6	0x62	0xb1		0xa0	0x0c	0xc5	0xae	0x9a	0x48	0x72	0x70
0x0f0	0x90	0x76	0x37	0xdb	0x45	0xf6	0x8f	0x9e		0x93	0xca	0xe7	0xf8	0x6c	0x66	0xba	0x83
0x100	0x83	0x03	0xcc	0xfc	0x6b	0xea	0x06	0xb6		0xde	0x4a	0xc4	0x5f	0xb8	0x0d	0x1f	0x81
0x110	0x26	0x7d	0x02	0x4b	0x3c	0x67	0xcb	0x0b		0x40	0x60						

Huffman Tree (549 bits = 68 bytes + 5bits) as present at CMMPL:

Offset	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	--	0x8	0x9	0xa	0xb	0xc	0xd	0xe	0xf
0x000	0x00	0xa9	0xd4	0x95	0x8a	0xa9	0x45	0xa1		0x29	0x74	0x72	0xd5	0x56	0x30	0x37	0x87
0x010	0x87	0x24	0x58	0xec	0x22	0x17	0x09	0x4b		0xa4	0x76	0xb2	0xdb	0x65	0x14	0xda	0x4a
0x020	0xa3	0x50	0xb7	0x96	0x7b	0x99	0x60	0x5e		0xad	0x96	0x60	0xb8	0xd9	0x16	0x2b	0xa2
0x030	0x42	0xb8	0x5a	0x01	0x54	0x96	0x25	0x69		0x7c	0x2a	0xd2	0x5d	0xae	0x57	0x54	0x7d
0x040	0x6e	0x4b	0x55	0xa9	0x00												

The figure shown in bold italic style in the table above indicates that its content is shared between the Huffman Tree and the Compressed payload (5 MSB belongs to Huffman Tree and 3 LSB belongs to Compressed payload).

This Huffman Tree, with its bits ordered from LSB to MSB, is the following:

```
00000000101010011101010010010101100010101010100101000101101000010010100101110100
01110010110101010101011000110000001101111000011110000111001001000101100011101100
00100010000101110000100101001011101001000111011010110010110110110110010100010100
11011010010010101010001101010000101101111001011001111011100110010110000001011110
10101101100101100110000010111000110110010001011000101011101000100100001010111000
010110100000000010101010010010110001001010110100101111100001010101101001001011101
10101110010101110101010001111101011011100100101101010101101010010000
```

Which has the following structure:

Code TracePath	Coded Char Bits	Coded Char Hex
00000000		
1	01010011	0x53
1	01010010	0x52
0		
1	01011000	0x58
1	01010101	0x55
00		
1	01000101	0x45
1	01000010	0x42
0		
1	01001011	0x4B
1	01000111	0x47
00		
1	01101010	0x6A
1	01010110	0x56
00		
1	10000001	0x81
1	01111000	0x78
0		
1	11100001	0xE1
1	10010010	0x92
00		
1	01100011	0x63
1	01100001	0x61
000		
1	00001011	0x0B
1	00001001	0x09
0		
1	00101110	0x2E
1	00100011	0x23
1	01101011	0x6B
00		
1	01101101	0x6D
1	01100101	0x65
000		
1	01001101	0x4D
1	01001001	0x49
0		
1	01010001	0x51
1	01010000	0x50
1	01101111	0x6F
00		
1	01100111	0x67
1	01110011	0x73
00		
1	01100000	0x60
0		
1	01111010	0x7A
1	01101100	0x6C
1	01100110	0x66
0000		
1	01110001	0x71
1	01100100	0x64
0		

Code TracePath	Coded Char Bits	Coded Char Hex
1	01100010	0x62
1	01110100	0x74
0		
1	00100001	0x21
0		
1	01110000	0x70
1	01101000	0x68
00000		
1	01010100	0x54
1	00101100	0x2C
0		
1	00101011	0x2B
0		
1	00101111	0x2F
1	00001010	0x0A
1	01101001	0x69
00		
1	01110110	0x76
1	01110010	0x72
1	01110101	0x75
0		
1	00011111	0x1F
0		
1	01101110	0x6E
0		
1	00101101	0x2D
0		
1	01011010	0x5A
1	00100000	0x20

And codes as shown in the following table:

Code	Symbol	Code Bits
0x00	0x53	8
0x01	0x52	8
0x02	0x58	8
0x03	0x55	8
0x04	0x45	8
0x04	0x6a	7
0x04	0x63	6
0x04	0x6d	5
0x04	0x67	4
0x05	0x42	8
0x05	0x56	7
0x05	0x61	6
0x05	0x65	5
0x05	0x73	4
0x06	0x4b	8
0x07	0x47	8
0x07	0x6b	6
0x07	0x6f	5
0x07	0x66	4
0x0a	0x21	4
0x0c	0x81	8
0x0c	0x60	5
0x0d	0x78	8
0x0e	0xe1	8
0x0e	0x1f	4
0x0f	0x92	8
0x10	0x71	5
0x11	0x64	5
0x12	0x62	5
0x13	0x74	5
0x16	0x70	5
0x17	0x68	5
0x18	0x0b	8
0x18	0x4d	7
0x19	0x09	8
0x19	0x49	7
0x19	0x69	5
0x1a	0x2e	8
0x1a	0x51	7
0x1a	0x7a	6
0x1b	0x23	8
0x1b	0x50	7
0x1b	0x6c	6
0x1b	0x75	5
0x1e	0x6e	5
0x34	0x76	6
0x35	0x72	6
0x3e	0x2d	6
0x60	0x54	7
0x61	0x2c	7
0x62	0x2b	7
0x7e	0x5a	7
0x7f	0x20	7
0xc6	0x2f	8

X.3.1.3 Uncompressed received management message payload

Using an intermediate buffer of 726 bytes for uncompressing, it should be as follows:

Offset	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	--	0x8	0x9	0xa	0xb	0xc	0xd	0xe	0xf
0x000	0x56	0x69	0x63	0x74	0x6f	0x72	0x20	0x5a		0x65	0x69	0x62	0x20	0x42	0x6f	0x72	0x74
0x010	0x0a	0x4d	0x61	0x72	0x63	0x6f	0x73	0x20		0x4d	0x61	0x72	0x74	0x69	0x6e	0x65	0x7a
0x020	0x20	0x56	0x61	0x7a	0x71	0x75	0x65	0x7a		0x0a	0x21	0x5a	0x6e	0x68	0x2c	0x54	0x6e
0x030	0x75	0x67	0x70	0x73	0x69	0x1f	0x6c	0x6d		0x70	0x76	0x74	0x1f	0x75	0x67	0x66	0x1f
0x040	0x68	0x60	0x75	0x64	0x2f	0x1f	0x5a	0x6e		0x68	0x2c	0x54	0x6e	0x75	0x67	0x70	0x73
0x050	0x69	0x1f	0x6a	0x72	0x21	0x73	0x69	0x64		0x21	0x66	0x62	0x73	0x66	0x2d	0x21	0x58
0x060	0x70	0x66	0x2e	0x52	0x70	0x73	0x69	0x6e		0x75	0x67	0x21	0x68	0x74	0x1f	0x75	0x67
0x070	0x66	0x1f	0x6c	0x64	0x7a	0x1f	0x62	0x6d		0x65	0x1f	0x68	0x74	0x62	0x71	0x65	0x68
0x080	0x62	0x6d	0x21	0x6e	0x67	0x1f	0x75	0x67		0x66	0x1f	0x68	0x60	0x75	0x64	0x2f	0x1f
0x090	0x51	0x60	0x74	0x73	0x2d	0x1f	0x71	0x71		0x66	0x72	0x66	0x6d	0x75	0x2b	0x21	0x65
0x0a0	0x76	0x73	0x76	0x71	0x66	0x2b	0x21	0x60		0x6d	0x6b	0x21	0x60	0x73	0x64	0x21	0x6e
0x0b0	0x6f	0x64	0x21	0x68	0x6f	0x1f	0x5a	0x6e		0x68	0x2c	0x54	0x6e	0x75	0x67	0x70	0x73
0x0c0	0x69	0x2d	0x21	0x47	0x66	0x1f	0x6c	0x6d		0x70	0x76	0x74	0x1f	0x78	0x67	0x66	0x71
0x0d0	0x66	0x1f	0x75	0x67	0x66	0x1f	0x50	0x6b		0x65	0x1f	0x50	0x6d	0x66	0x72	0x21	0x61
0x0e0	0x73	0x6e	0x6c	0x64	0x21	0x73	0x69	0x71		0x70	0x74	0x68	0x67	0x21	0x6e	0x67	0x1f
0x0f0	0x70	0x6b	0x65	0x2b	0x21	0x60	0x6f	0x63		0x21	0x76	0x69	0x64	0x73	0x64	0x21	0x53
0x100	0x69	0x64	0x7a	0x1f	0x74	0x67	0x62	0x6b		0x6d	0x1f	0x63	0x71	0x66	0x60	0x6c	0x1f
0x110	0x75	0x67	0x73	0x6e	0x76	0x66	0x69	0x1f		0x62	0x66	0x62	0x68	0x6f	0x2d	0x23	0x09
0x120	0x21	0x1f	0x21	0xe1	0x81	0x92	0x49	0x2d		0x51	0x2d	0x21	0x4b	0x70	0x75	0x66	0x62
0x130	0x73	0x60	0x67	0x73	0x2d	0x1f	0x55	0x67		0x66	0x1f	0x45	0x74	0x6f	0x76	0x6a	0x62
0x140	0x69	0x1f	0x49	0x6e	0x73	0x71	0x70	0x71		0x0b	0x53	0x53	0x53	0x53	0x53	0x53	0x53
0x150	0x53	0x53	0x53	0x53	0x53	0x53	0x53	0x53		0x53	0x53	0x53	0x53	0x53	0x53	0x53	0x53
0x160	0x53	0x53	0x53	0x53	0x53	0x53	0x53	0x53		0x53	0x53	0x53	0x53	0x53	0x53	0x53	0x53
0x170	0x53	0x53	0x53	0x53	0x53	0x53	0x53	0x53		0x53	0x53	0x53	0x53	0x53	0x53	0x53	0x53
0x180	0x53	0x53	0x53	0x53	0x53	0x53	0x53	0x53		0x53	0x53	0x53	0x53	0x53	0x53	0x53	0x53
0x190	0x53	0x53	0x53	0x53	0x53	0x53	0x53	0x53		0x53	0x53	0x53	0x53	0x53	0x53	0x53	0x53
0x1a0	0x53	0x53	0x53	0x53	0x53	0x53	0x53	0x53		0x53	0x53	0x53	0x53	0x53	0x53	0x53	0x53
0x1b0	0x53	0x53	0x53	0x53	0x53	0x53	0x53	0x53		0x53	0x53	0x53	0x53	0x53	0x53	0x53	0x53
0x1c0	0x53	0x53	0x53	0x53	0x53	0x53	0x53	0x53		0x53	0x53	0x53	0x53	0x53	0x53	0x53	0x53
0x1d0	0x53	0x53	0x53	0x53	0x53	0x53	0x53	0x53		0x53	0x53	0x53	0x53	0x53	0x53	0x53	0x53
0x1e0	0x53	0x53	0x53	0x53	0x53	0x53	0x53	0x53		0x53	0x53	0x53	0x53	0x53	0x53	0x53	0x53
0x1f0	0x53	0x53	0x53	0x53	0x53	0x53	0x53	0x53		0x53	0x53	0x53	0x53	0x53	0x53	0x53	0x53
0x200	0x53	0x53	0x53	0x53	0x53	0x53	0x53	0x53		0x53	0x53	0x53	0x53	0x53	0x53	0x53	0x53
0x210	0x53	0x53	0x53	0x53	0x53	0x53	0x53	0x53		0x53	0x53	0x53	0x53	0x53	0x53	0x53	0x53
0x220	0x53	0x53	0x53	0x53	0x53	0x53	0x53	0x53		0x53	0x53	0x53	0x53	0x53	0x53	0x53	0x53
0x230	0x53	0x53	0x53	0x53	0x53	0x53	0x53	0x53		0x53	0x53	0x53	0x53	0x53	0x53	0x53	0x53
0x240	0x53	0x53	0x53	0x53	0x53	0x53	0x53	0x53		0x53	0x53	0x53	0x53	0x53	0x53	0x53	0x53
0x250	0x53	0x53	0x53	0x53	0x53	0x53	0x53	0x53		0x53	0x53	0x53	0x53	0x53	0x53	0x53	0x53
0x260	0x53	0x53	0x53	0x53	0x53	0x53	0x53	0x53		0x53	0x53	0x53	0x53	0x53	0x53	0x53	0x53
0x270	0x53	0x53	0x53	0x53	0x53	0x53	0x53	0x53		0x53	0x53	0x53	0x53	0x53	0x53	0x53	0x53
0x280	0x53	0x53	0x53	0x53	0x53	0x53	0x53	0x53		0x53	0x53	0x53	0x53	0x53	0x53	0x53	0x53
0x290	0x53	0x53	0x53	0x53	0x53	0x53	0x53	0x53		0x53	0x53	0x53	0x53	0x53	0x53	0x53	0x53
0x2a0	0x53	0x53	0x53	0x53	0x53	0x53	0x53	0x53		0x53	0x53	0x53	0x53	0x53	0x53	0x53	0x53
0x2b0	0x53	0x53	0x53	0x53	0x53	0x53	0x53	0x53		0x53	0x53	0x53	0x53	0x53	0x53	0x53	0x53

0x2c0	0x53	0x53	0x53	0x53	0x53	0x53	0x53	0x53		0x53	0x53	0x53	0x53	0x53	0x53	0x53	0x53
0x2d0	0x53	0x53	0x53	0x53	0x53	0x53											

X.3.1.4 Output (uncompressed) management message payload (MMPL)

MMPL (329 bytes)

Offset	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	--	0x8	0x9	0xa	0xb	0xc	0xd	0xe	0xf
0x000	0x56	0x69	0x63	0x74	0x6f	0x72	0x20	0x5a		0x65	0x69	0x62	0x20	0x42	0x6f	0x72	0x74
0x010	0x0a	0x4d	0x61	0x72	0x63	0x6f	0x73	0x20		0x4d	0x61	0x72	0x74	0x69	0x6e	0x65	0x7a
0x020	0x20	0x56	0x61	0x7a	0x71	0x75	0x65	0x7a		0x0a	0x21	0x5a	0x6e	0x68	0x2c	0x54	0x6e
0x030	0x75	0x67	0x70	0x73	0x69	0x1f	0x6c	0x6d		0x70	0x76	0x74	0x1f	0x75	0x67	0x66	0x1f
0x040	0x68	0x60	0x75	0x64	0x2f	0x1f	0x5a	0x6e		0x68	0x2c	0x54	0x6e	0x75	0x67	0x70	0x73
0x050	0x69	0x1f	0x6a	0x72	0x21	0x73	0x69	0x64		0x21	0x66	0x62	0x73	0x66	0x2d	0x21	0x58
0x060	0x70	0x66	0x2e	0x52	0x70	0x73	0x69	0x6e		0x75	0x67	0x21	0x68	0x74	0x1f	0x75	0x67
0x070	0x66	0x1f	0x6c	0x64	0x7a	0x1f	0x62	0x6d		0x65	0x1f	0x68	0x74	0x62	0x71	0x65	0x68
0x080	0x62	0x6d	0x21	0x6e	0x67	0x1f	0x75	0x67		0x66	0x1f	0x68	0x60	0x75	0x64	0x2f	0x1f
0x090	0x51	0x60	0x74	0x73	0x2d	0x1f	0x71	0x71		0x66	0x72	0x66	0x6d	0x75	0x2b	0x21	0x65
0x0a0	0x76	0x73	0x76	0x71	0x66	0x2b	0x21	0x60		0x6d	0x6b	0x21	0x60	0x73	0x64	0x21	0x6e
0x0b0	0x6f	0x64	0x21	0x68	0x6f	0x1f	0x5a	0x6e		0x68	0x2c	0x54	0x6e	0x75	0x67	0x70	0x73
0x0c0	0x69	0x2d	0x21	0x47	0x66	0x1f	0x6c	0x6d		0x70	0x76	0x74	0x1f	0x78	0x67	0x66	0x71
0x0d0	0x66	0x1f	0x75	0x67	0x66	0x1f	0x50	0x6b		0x65	0x1f	0x50	0x6d	0x66	0x72	0x21	0x61
0x0e0	0x73	0x6e	0x6c	0x64	0x21	0x73	0x69	0x71		0x70	0x74	0x68	0x67	0x21	0x6e	0x67	0x1f
0x0f0	0x70	0x6b	0x65	0x2b	0x21	0x60	0x6f	0x63		0x21	0x76	0x69	0x64	0x73	0x64	0x21	0x53
0x100	0x69	0x64	0x7a	0x1f	0x74	0x67	0x62	0x6b		0x6d	0x1f	0x63	0x71	0x66	0x60	0x6c	0x1f
0x110	0x75	0x67	0x73	0x6e	0x76	0x66	0x69	0x1f		0x62	0x66	0x62	0x68	0x6f	0x2d	0x23	0x09
0x120	0x21	0x1f	0x21	0xe1	0x81	0x92	0x49	0x2d		0x51	0x2d	0x21	0x4b	0x70	0x75	0x66	0x62
0x130	0x73	0x60	0x67	0x73	0x2d	0x1f	0x55	0x67		0x66	0x1f	0x45	0x74	0x6f	0x76	0x6a	0x62
0x140	0x69	0x1f	0x49	0x6e	0x73	0x71	0x70	0x71		0x0b							

Bibliography

- [b-IETF RFC 768] IETF RFC 768 (1980), *User Datagram Protocol*.
- [b-IETF RFC 791] IETF RFC 791 (1981), *Internet Protocol*.
- [b-IETF RFC 793] IETF RFC 793 (1981), *Transmission Control Protocol*.
- [b-IETF RFC 2474] IETF RFC 2474 (1998), *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*.
- [b-IETF RFC 3376] IETF RFC 3376 (2002), *Internet Group Management Protocol, Version 3*.
- [b-BCL] Basic Compression Library. Available at <http://bcl.comli.eu/home-en.html>
-

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	Tariff and accounting principles and international telecommunication/ICT economic and policy issues
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Environment and ICTs, climate change, e-waste, energy efficiency; construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling, and associated measurements and tests
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities
Series Z	Languages and general software aspects for telecommunication systems