International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

G.9961

(06/2010)

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

Access networks – In premises networks

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

Recommendation ITU-T G.9961

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
In premises networks	**G.9950–G.9999**

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

Recommendation ITU-T G.9961

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

Summary

Recommendation ITU-T G.9961 specifies the data link layer (DLL) for wire-line based home networking transceivers capable of operating over premises wiring including inside telephone wiring, coaxial cable, and power-line wiring. It complements the system architecture and physical layer (PHY) specification in Recommendation ITU-T G.9960.

History

Edition	Recommendation	Approval	Study Group
1.0	ITU-T G.9961	2010-06-11	15

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/.

Table of Contents

Introduction

Recommendation ITU-T G.9961 specifies the data link layer (DLL) of home networking transceivers capable of operating over premises wiring including inside telephone wiring, coaxial cable, power-line wiring, and combinations of these. Transceivers defined by this Recommendation provide the data rate and quality of service necessary for triple-play residential services as well as business-type services delivered over xDSL, PON, or other access technologies. The physical layer for transceivers associated with this Recommendation is specified in Recommendation ITU-T G.9960. The transceivers use orthogonal frequency division multiplexing (OFDM) type modulation and are designed to provide electromagnetic compatibility (EMC) and spectral compatibility between home networking transmission and VDSL2 or other types of digital subscriber line (DSL) used to access the home.

Recommendation ITU-T G.9961

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

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;
- 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.

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 (2010), *Unified high-speed wire-line based home networking transceivers – System architecture and physical layer specification*.

[ITU-T G.9972] Recommendation ITU-T G.9972 (2010), *Coexistence mechanism for wireline home networking transceivers*.

[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.*
[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.*
[NIST FIPS 197]	FIPS-PUB-197 (2001), *Advanced Encryption Standard (AES)*, National Institute of Standards and Technology. http://csrc.nist.gov/publications/
[NIST 800-38C]	Special Publication 800-38C (2004), *Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality*, National Institute of Standards and Technology. http://csrc.nist.gov/publications/nistpubs/800-38C/SP800-38C_updated-July20_2007.pdf.

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 two nodes uniquely identified by the following parameters:

- Source DEVICE_ID,
- Destination DEVICE_ID,
- value of management queue flag (MQF) field in the associated LPDUs, indicating whether the flow is for delivering data LLC frame blocks or management LLC frame blocks, and
- value of CONNECTION_ID field, indicating the flow identifier, if the flow corresponds to parameterized QoS or the priority queue if the flow corresponds to prioritized QoS.

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 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 according to the actual consequences.

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

This Recommendation uses the following abbreviations:

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
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
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
CTMG	Control and Management frame
CTS	Clear To Send
CURRTS	Current Time Slot
CYCSTART	MAC Cycle Start
CW	Contention Window
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
EFD	Enhanced Frame Detection
FACK	Frame ACKnowledgement
FCS	Frame Check Sequence
FLEN	Frame Length
FN	Frame Number
FSLT	Future Schedule Life Time
HCS	Header Check Sequence
HOIP	Handover In Progress
IDB	Inter-Domain Bridge

IFG	Inter-Frame Gap
Imm-ACK	Immediate ACKnowledgement
ITS	Idle Time Slot
LCDU	Link Control Data Unit
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
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
NMK	Network Membership Key
NSC	Node to Security Controller
NTR	Network Time Reference
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
RCBTS	Registration Contention-Based Time Slot
REGID	Registration Identifier
RMAP	Relayed Medium Access Plan
RPRQ	Reply Required
RTS	Request To Send
SC	Security Controller
SM	Sub-carrier Mask
SSN	Segment Sequence Number
STXOP	Shared Transmission Opportunity
TDMA	Time Division Multiple Access
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.

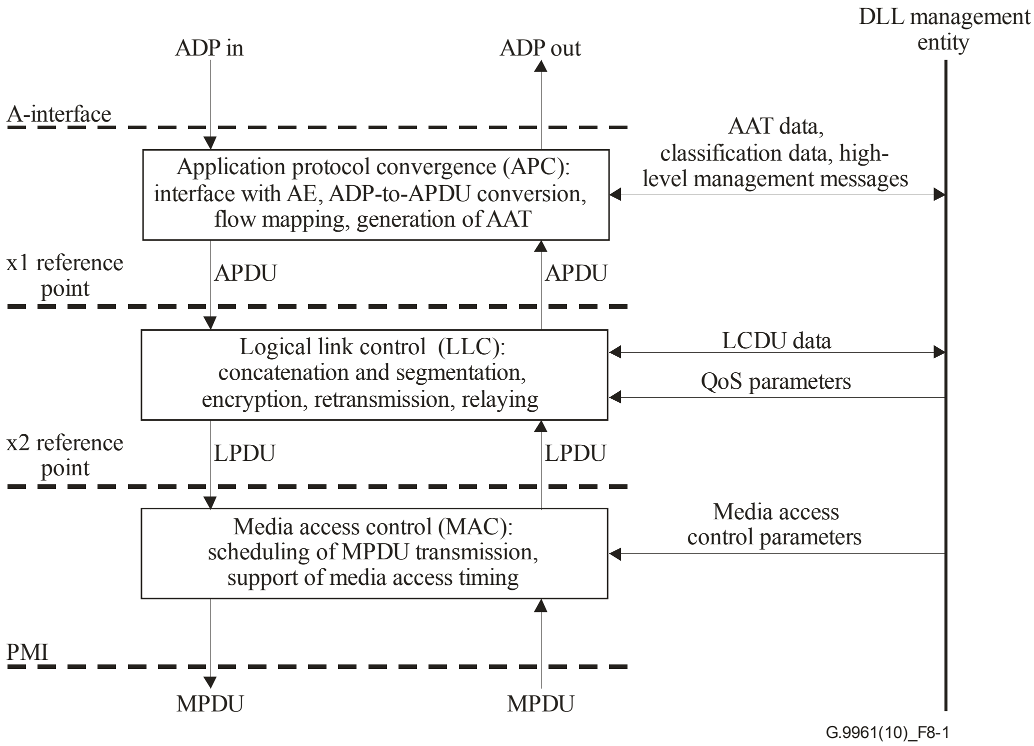


Figure 8-1 – Functional model of the DLL

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].

NOTE 1 – The L2 bridging function between the ITU-T G.9960/1 node and associated clients is considered to be part of the AE and is beyond the scope of this Recommendation.

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 2 – 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.

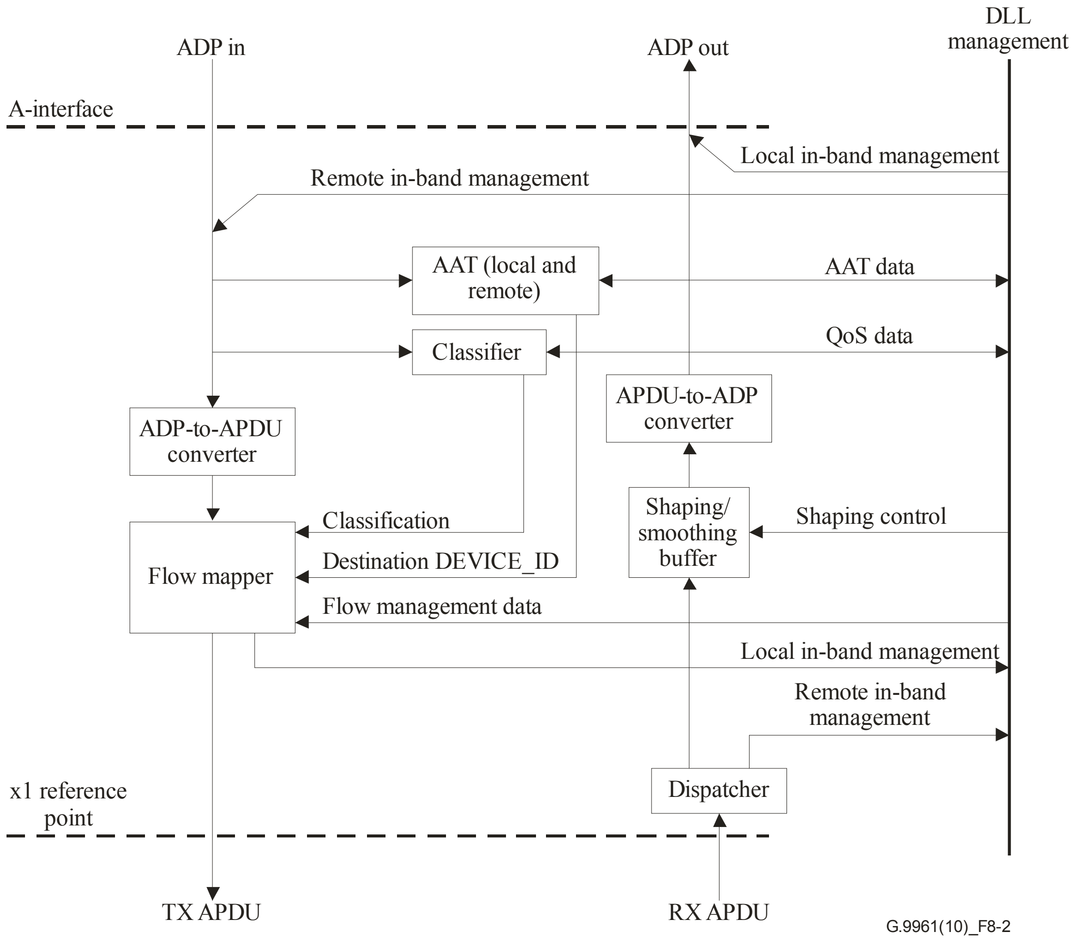


Figure 8-2 – Functional model of APC

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, 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 address association table (AAT) contains identification data (MAC addresses) of the clients associated with the node. This data is collected from the incoming ADP data units and stored in the "local" part of the AAT; the local AAT data is passed to the DLL management entity for network management purposes (see clause 8.5.3). In its "remote" part, AAT stores MAC addresses of the clients associated with other nodes that were advertised on the network.

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.

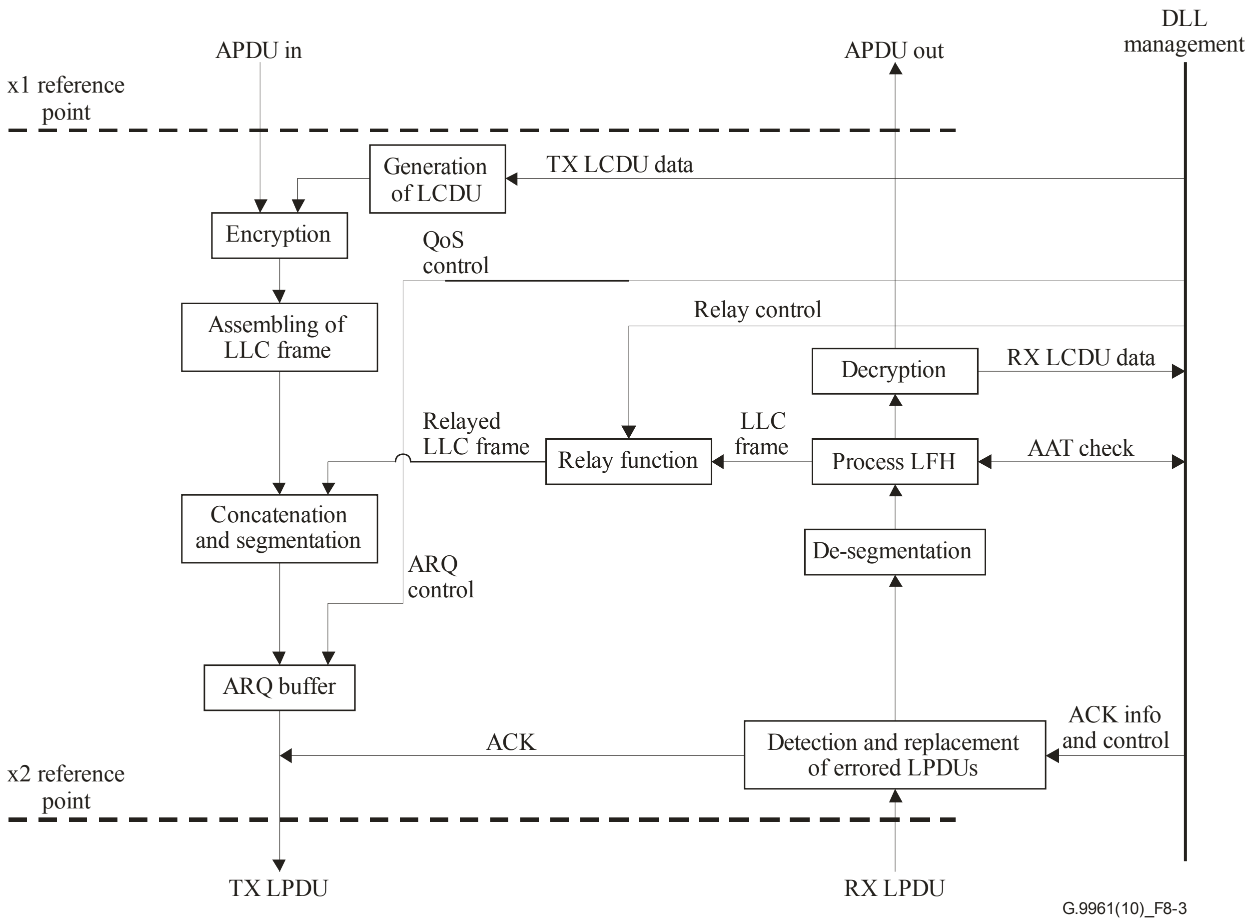


Figure 8-3 – Functional model of LLC

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.

The LLC management data to be conveyed is assembled into an LCDU. The format of LCDU is universal for all types of media and described in clause 8.1.3.4. The LCDU is further mapped into an MPDU as described in clause 8.1.4.1.

LPDUs that are subject to ARQ (need to be retransmitted) are extracted from the ARQ buffer and passed to the MAC to be assembled into the outgoing MPDU. To assist retransmission, the receive part of the LLC generates ACKs, which are also passed to the MAC (see clause 8.9). The number of LLC frames to be concatenated, the size of the segment, and other MPDU formatting parameters are controlled by the LLC. The LPDUs are passed to the MAC across the x2 reference point.

In the receive direction, the incoming MPDU is disassembled into LPDUs in the MAC and passed over the x2 reference point. The LLC verifies the LPDUs, requests replacements for any errored ones if so instructed, and recovers LLC frames from the LPDUs. The recovered LLC frames are decrypted and passed to APC as APDUs. Recovered LCDUs are passed to the DLL management entity.

The relay function extracts LLC frames that are subject to relaying and passes them to the transmit side, which concatenates them into the traffic to the next hop. DLL management controls flow and priority settings for the relayed LLC frames. Relayed LLC frames shall not be decrypted.

8.1.3.1 LLC frame format

The LLC frame is formed from either an APDU, with format as described in Annex A, or an LCDU, with format as described in clause 8.1.3.4 and Figure 8-4.

If encryption is required, the incoming APDU or LCDU shall be encrypted using CCMP, as described in clause 9.1.2. A CCMP header, and a message integrity code (MIC) are added as described in Figure 8-4 (case = encrypted); their content shall be as specified in clause 9.1.2 and the LLC frame header shall indicate the presence of a CCMP header. The length of the PAD used for encryption (clause 9.1.1.1), which is necessary for decryption of the APDU or LCDU and MIC verification at the receive side, can be derived from the frame length (FLEN) field communicated in the LFH (clause 8.1.3.1.1.5). The length of the MIC and other parameters required for decryption are indicated in CCMP header (see clause 9.1.2).

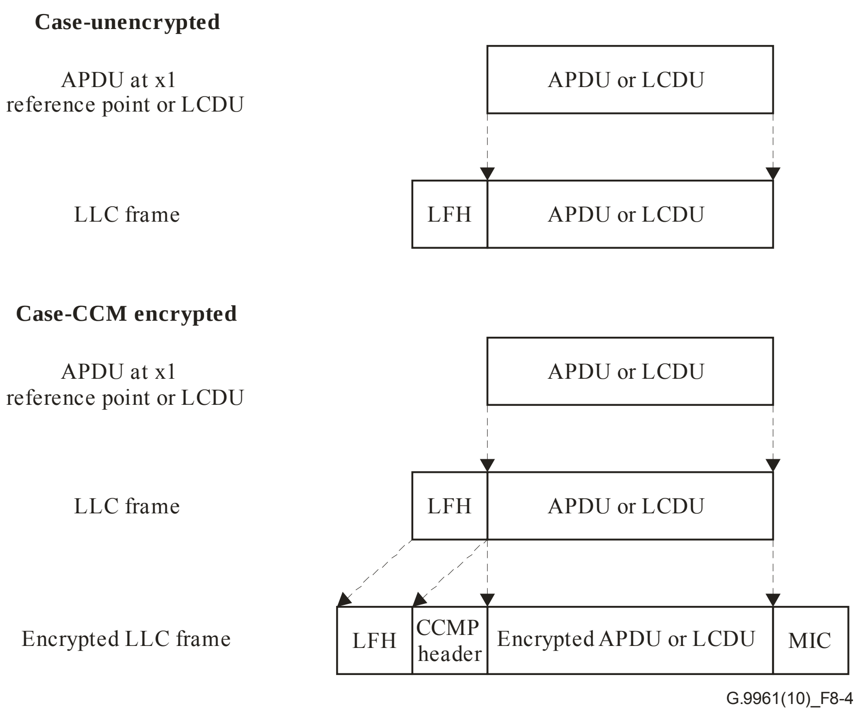


Figure 8-4 – LLC frame format (unencrypted and encrypted)

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 5 octets long if no time stamp is present, and 9 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
Reserved		[15:14]	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
BRCTI	4	[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	5 to 8	[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 5 octets).

8.1.3.1.1.3 CCMP header present indication (CCMPI)

CCMPI is a 1-bit field that is used to indicate whether a CCMP header is present following the LLC frame header or not. If set to one, the LLC frame shall be encrypted and the CCMP header shall follow the LFH. If set to zero, the LLC frame shall not be encrypted and shall not include CCMP header and MIC.

8.1.3.1.1.4 LLC frame priority (LPRI)

The LPRI field is a 3-bit unsigned integer field with valid values from 0 to 7. For LLC frames carrying APDUs, this field shall be set to the user priority assigned by the classifier (see clause 8.1.2) of the node that originated the LLC frame; otherwise, it shall be set to 0. For LLC frames carrying LCDUs, this field shall be set to 7 (LCDUs are always considered to be of the highest user priority).

User priority 0 is considered higher than 1 and 2 but lower than the rest of user priorities (see Table III.1 of [ITU-T G.9960]). This criterion shall be applied in the rest of this Recommendation when different user priorities are compared.

The content of this field shall not be changed when an LLC frame is relayed.

8.1.3.1.1.5 Frame length (FLEN)

The FLEN field indicates the length in bytes of the frame contained within the LLC frame. This is the actual length of the LLC frame excluding the LFH. In case of encryption, it also includes the CCMP header and MIC. It is formatted as a 14-bit unsigned integer.

8.1.3.1.1.6 OriginatingNode

The OriginatingNode field carries the DEVICE_ID of the node that originated the LLC frame. The content of this field shall not be changed when an LLC frame is relayed by another node.

8.1.3.1.1.7 Broadcast indicator (BRCTI)

This bit shall be set to one if the APDU or LCDU contained in the LLC frame shall be broadcasted following the BRT and zero otherwise.

8.1.3.1.1.8 Time to live (TTL)

The TTL field indicates the number of times the LLC frame is allowed to be relayed. It is formatted as a 6-bit unsigned integer. If a node receives an LLC frame to be relayed with TTL field not zero, it shall decrement it by one in the relayed LLC frame. If a node receives a frame with TTL field equal to zero, such LLC frame shall not be relayed.

The initial value of the TTL field is set by the node originating the frame and should be higher than the number of times that the LLC frame is expected to be relayed before reaching its destination.

8.1.3.1.1.9 Timestamp (TSMP)

The TSMP field indicates the arrival time of each ADP at the A-interface of the transmitting node. The TSMP shall carry the value of the node clock timer at the instant the first byte of the ADP crosses the A-interface, represented as a 32-bit unsigned integer with resolution of 10 ns per unit (see clauses 7.1.2.3.2.1.2 and 7.1.2.3.2.1.3 of [ITU-T G.9960]).

The value of the TSMP for management messages (LCDU and in-band management messages) is for further study.

NOTE 1 – The TSMP may be used to perform monitoring on the QoS requirements of a flow, on its latency and on its jitter.

NOTE 2 – The timestamps may be used by a transmitting node in order to restore relative frame arrival timing at the receiver as it was at the transmitter.

8.1.3.2 Generation of LPDUs

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

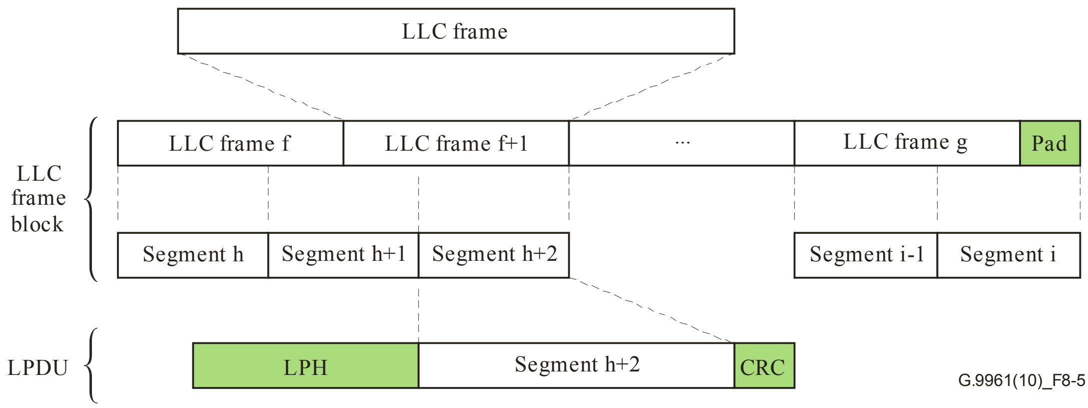


Figure 8-5 – Generation of LPDUs from LLC frames

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 block is treated as a data LLC frame block.

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 size of the FEC block 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.

For decoding purposes, the FEC block size is specified in the PHY-frame header (PFH) (see clause 7.1.2.3 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.

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
Reserved		[15:13]	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.				

8.1.3.2.1.1 Segment sequence number (SSN)

This field identifies the relative location of the segment within the stream of segments corresponding to a connection. Segment sequence number (SSN) is a 16-bit field indicating the order of segments that are associated with a connection. The SSN shall be initialized to zero for the first valid segment that belongs to a new connection and shall be incremented by one for each new valid segment that is associated with this connection. The SSN shall be expressed as a 16-bit unsigned integer and shall wrap around (goes back to value 0000_{16} after $FFFF_{16}$).

NOTE – A receiver might receive segments in an "out-of-order" manner when lost segments are retransmitted or LPDUs from the management LLC frame block are members of the same MPDU.

8.1.3.2.1.2 LLC frame boundary offset (LFBO)

This field indicates the location of the start of the first LLC frame within the segment. This enables the receiver to recover when one or more segments are lost (e.g., when the transmitter drops a segment due to timeout). The LFBO is a 10-bit field that carries the offset in octets of the first octet of the first new LLC frame relative to the start of the segment of the LPDU (in case the LLC frame starts at the start of the segment, the LFBO = 0).

The value of LFBO shall be coded as an unsigned integer as shown in Table 8-4.

Table 8-4 – Format of LFBO

LFBO Value	Description
000_{16} to 213_{16}	The LLC frame boundary offset in bytes
214_{16} to $3FE_{16}$	Reserved by ITU-T
$3FF_{16}$	No LLC frame boundary exists in the LPDU

8.1.3.2.1.3 Valid segment flag (VSF)

This field indicates whether the LPDU contains a valid or an invalid segment. The VSF shall be set to one to indicate that the LPDU contains a valid segment. VSF shall be set to zero to indicate that the LPDU contains an invalid segment. In case the segment is indicated as invalid, the remaining fields of this LPDU header shall be ignored by the receiver.

The transmitter shall set the VSF = 0 (invalid segments) in LPDUs that pad the MPDU (see clause 8.1.3.2).

8.1.3.2.1.4 Management queue flag (MQF)

This field indicates whether a segment belongs to a management LLC frame block or to a data LLC frame block that is associated with the destination indicated in the PHY-frame header. The MQF shall 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)

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.

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.

	LSB MSB
6 octets	Destination (MAC address)
6 octets	Source (MAC address)
2 octets	EtherType ($22E3_{16}$)
6 to 1500 octets	LCDU payload
	PAD
4 octets	FCS

G.9961(10)_F8-6

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 $22E3_{16}$.

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.

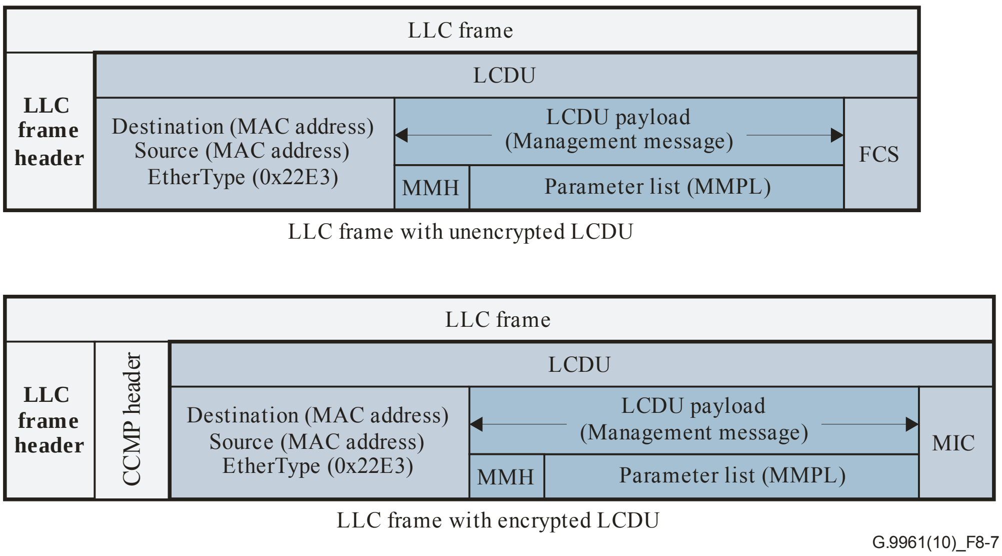


Figure 8-7 – Encapsulation of an LCDU into an LLC frame

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.

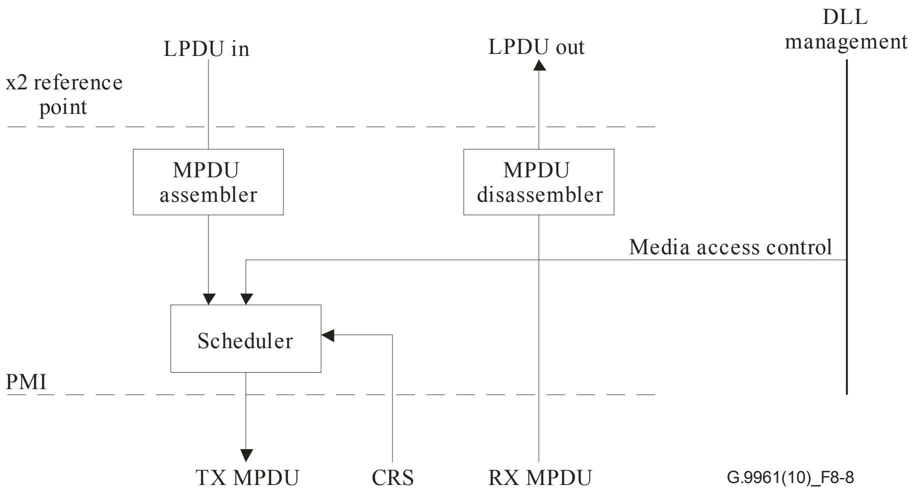


Figure 8-8 – Functional model of MAC

In the transmit direction, MPDUs are assembled from LPDUs passed over the x2 reference point. Then the MPDUs are scheduled for transmission using one of the medium access procedures described in clauses 8.2 and 8.3. For scheduling, one or more transmission queues can be established. The carrier sense (CRS) primitive indicates whether the medium is busy or not. After being scheduled for transmission, the MPDU is passed to the PHY across the PMI. The octet 0 of the LPH of the LPDU#1 of the MPDU (see Figure 8-9) shall be passed to the PHY first.

When the MPDU transmission requires usage of RTS/CTS protocol, the DLL management shall instruct the MAC to schedule an RTS prior to passing the MPDU to the PHY. The scheduled MPDU will be passed to the PHY only if a correct CTS PHY frame was received (see clause 8.3.3.4.4).

The MAC also schedules transmission of priority resolution (PR) and INUSE signals to support media access protocols described in clause 8.3 if these signals are required.

The MAC is also responsible for scheduling an ACK frame transmission by the PHY if ACK is required.

In the receive direction, the incoming MPDU is disassembled and the resulting LPDUs are passed to the LLC over the x2-reference point.

8.1.4.1 Assembling of an MPDU from LPDUs

The process of assembling an MPDU from one or more LPDUs is presented in Figure 8-9.

To form the MPDU, LPDUs are concatenated by the MAC in the same order as received across the x2 reference point.

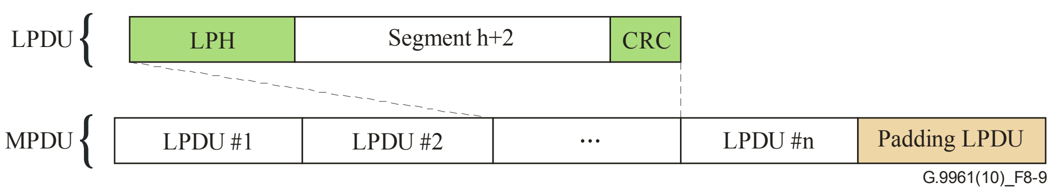


Figure 8-9 – Assembling of an MPDU from LPDUs

In the MAC, LPDUs, which the LLC generated from LLC frame blocks, shall be grouped into one or more MPDUs. The relative order of LPDUs in the MPDU shall be the same as in the sourcing LLC frame block. The MPDU can start with any LPDU of an LLC frame block.

The MPDU may also include LPDUs from other LLC frame blocks, LPDUs intended for retransmission that are associated with the same data connection, LPDUs belonging to a management LLC frame block that are associated with the same DID, and dummy LPDUs intended for padding. Padding LPDUs shall only be added when the number of bits to be loaded on the unused sub-carriers in the last OFDM symbol of the PHY frame carrying the MPDU before padding is bigger than the number of bits in the LPDU. Retransmitted LPDUs shall be the same size as the originally transmitted LPDUs.

LPDUs shall be ordered in groups inside the MPDU. A group of LPDUs is defined as a set of LPDUs from the same connection transmitted contiguously in the MPDU. The segment sequence numbers (SSNs) of the LPDUs in group *i* shall follow the condition $SSN_1^i \leq SSN_2^i \leq \ldots \leq SSN_{K_i}^i$, where K_i is the number of LPDUs in the group and SSN_1^i is the SSN of the first LPDU of the group passed to the MAC across the x2 reference point (it should be taken into account that SSNs wrap around; see clause 8.1.3.2.1.1).

An MPDU shall not contain more than one group of data LPDUs and shall not contain more than one group of management LPDUs. The group of management LPDUs shall appear in the MPDU before the group of data LPDUs, if both groups are present.

Padding LPDUs, if present, shall be placed at the end of the MPDU, see Figure 8-9, and shall be indicated as "Invalid" in the LPH (see clause 8.1.3.2.1.3). The content of segments of padding LPDUs is vendor discretionary.

The resulting priority of the MPDU (MPDU priority) shall be calculated as follows:

If an MPDU contains only management LPDUs, the MPDU priority shall be 7. Otherwise, the MPDU priority is determined by the data LPDUs in the MPDU.

If the MPDU contains one or more data LPDUs that are transmitted for the first time, the MPDU priority shall be the lowest LPRI (see Table 8-1) of the LLC frames (or fragments of LLC frames) from which those LPDUs are generated. Otherwise (i.e., all the data LPDUs within an MPDU are retransmitted LPDUs), the MPDU priority shall be the lowest LPRI of the LLC frames (or fragments of LLC frames) from which those LPDUs are generated.

8.2 MAP controlled medium access

Medium access shall be scheduled by MAC cycles continuously following one another, as shown in Figure 8-10.

Each MAC cycle is divided into two or more time intervals; one or more time intervals of which are for domain management purposes, while other time intervals are assigned as transmission opportunities (TXOPs) for different nodes or groups of nodes. At least one of the time intervals allotted for domain management purposes shall be assigned to the domain master for transmission of the medium access plan (MAP) (as described in Figure 8-10). The domain management information transmitted by the domain master in the MAP frame identifies the boundaries of the MAC cycle and includes the list of assigned TXOPs (content of the cycle) for one or more of the following MAC cycles (e.g., the MAP transmitted in cycle N can describe the timing boundaries and TXOPs of cycle N+1).

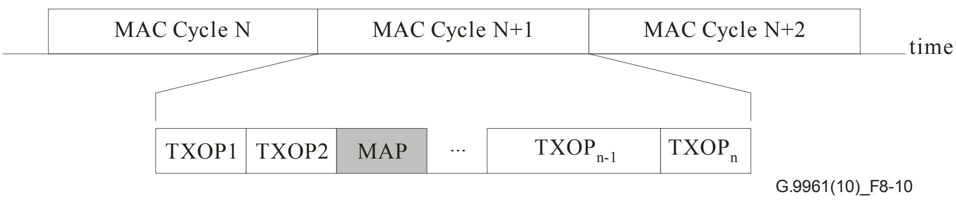


Figure 8-10 – Illustration of MAP controlled medium access

8.2.1 The MAC cycle

A MAC cycle shall start at the time published by a previous MAP frame sent by the domain master and ends at the end of the last TXOP scheduled for this MAC cycle as described in the MAP. The content of the MAC cycle is determined by the domain master based on the available communication resources inside the domain and communication resources and parameters required by different nodes for communications. The domain master may modify the content and the duration of MAC cycle from one cycle to another in order to accommodate changes in medium characteristics (channel/noise parameters), in services, or in the number of nodes operating in the domain. The position of the MAP in the MAC cycle is not fixed; it may change from one MAC cycle to another.

Nodes shall synchronize with the MAC cycle by detecting the presence of a MAP message and shall access the medium according to the TXOPs described in the MAP. The MAP describes the allocation of each TXOP by its start-time, duration, and by the assignment of the node or nodes that may transmit within the TXOPs. Timing references within the MAP shall be specified relative to the start of the MAC cycle.

Frames transmitted inside a TXOP shall be separated by inter-frame gaps (IFG). During IFG the medium shall be idle. Each TXOP shall also include an idle time period at its end as described in Figure 8-11. The duration of this idle period, $T_{IDLE,}$ is measured from the end of the last symbol of the last frame transmitted in the TXOP and shall be greater than or equal to T_{IFG_MIN}, where the value of T_{IFG_MIN} is medium dependent and is described in clause 8.4. The period of T_{IDLE} at the end of the last TXOP separates two subsequent MAC cycles.

The actual duration of IFG is measured from the last sample of the window of the last transmitted symbol of a PHY frame (see Figure 7-22 of [ITU-T G.9960]) to the first sample of the window of the first symbol of the preamble (see Figure 7-23 of [ITU-T G.9960]) of the subsequent PHY frame transmitted in the same or in the next TXOP.

The durations of the IFGs between frames of the same frame sequence depend on the type of the frame sequence. Values of IFGs for specific frame sequences are medium-dependent and are defined in clause 8.4.

The duration of the IFG between two subsequent frame sequences inside a TXOP shall be greater than or equal to T_{IFG_MIN}.

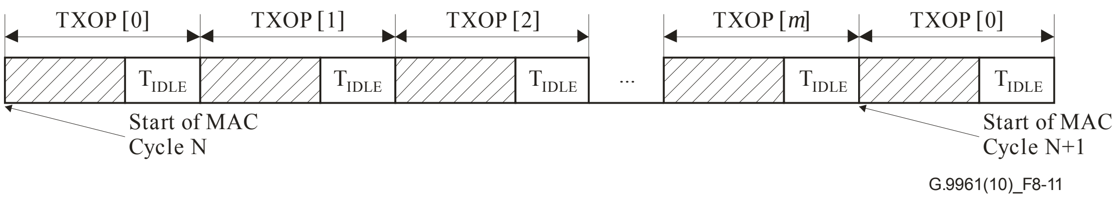


Figure 8-11 – IFG placement in the MAC cycle

A TXOP can be assigned to a single node or shared between several nodes (shared TXOP). Different types of TXOPs are described in clause 8.3. Inside a shared TXOP, medium access methods described in clause 8.3.3 shall be used.

8.2.2 Duration of the MAC cycle

The duration of the MAC cycle shall be in the range between CYCLE_MIN to CYCLE_MAX, as described in clause 8.4. The actual duration of the MAC cycle may change from one MAC cycle to another, although typically MAC cycles are of the same duration.

In some environments, the MAC cycle may be synchronized to an external clock source. In such cases, the duration of the MAC cycle is constrained to an integer number of external clock periods. Synchronization to an external source is described in clause 8.6.3.

The duration of the MAC cycle is explicitly indicated in the MAP header as described in clause 8.8.3. The duration of MAC cycle N+1 (published in the MAP transmitted in MAC cycle N) is the time period from the start time of MAC cycle N+1, indicated by the CYCSTART of that MAC cycle, till the end time of the MAC cycle N+1, indicated by the CYCSTART marking the start time of MAC cycle N+2 (This CYCSTART is transmitted in the PHY frame header of MAP transmitted in MAC cycle N+1).

NOTE – The latest end time of TXOPs, described by the MAPs in the MAC cycle, may be smaller than or equal to the start time of the next MAC cycle.

8.2.3 TXOP timing

The start time of a TXOP can be specified (or inferred) in the following two ways:

1) Implicitly, using the start time and duration of the previous TXOP, as specified in clause 8.8.4.1.1.
2) Explicitly, using TXOP absolute timing, as specified in clause 8.8.4.1.2.

By default, the start-time of TXOP[n] is defined implicitly and is equal to the start-time of TXOP[n–1] plus the duration of TXOP[n–1]. The implicit start-time of the first TXOP is 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.

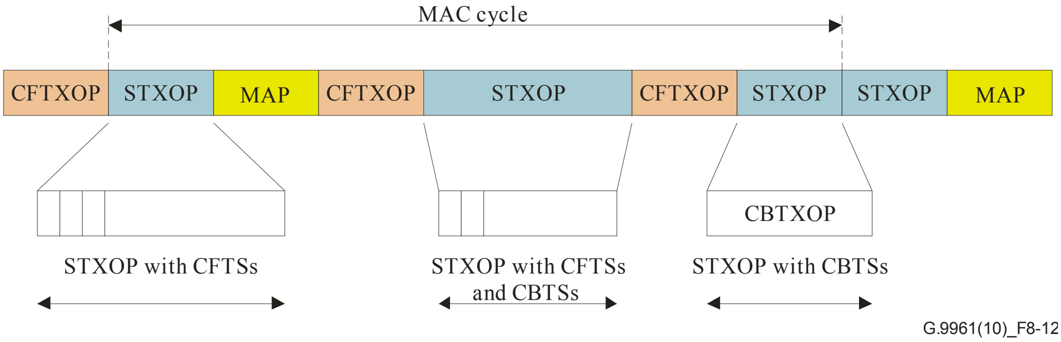


Figure 8-12 – Example of a MAC cycle structure

At least one MAP shall be sent each MAC cycle in a dedicated CFTXOP assigned by the domain master.

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.

The MAP message can define a mix of persistent and non-persistent TXOPs over the same MAC cycle, although the total duration of non-persistent TXOPs in all MAC cycles during the time of persistency shall be constant.

NOTE – The use of persistent and non-persistent TXOPs is a scheduling decision and is beyond the scope of this Recommendation.

8.3.1.2 Persistent access

When a node receives a MAP message that carries a persistent schedule, it shall use this information during the subsequent MAC cycles. The duration of the persistent schedule and the way it can be changed is described in clause 8.8.6. If a node received no MAP during the period which is longer than the previously announced period of MAP persistency, it shall halt all transmissions and search for the MAP to synchronize with the MAC cycle.

When a MAC cycle includes persistent TXOPs, the MAC cycle duration shall not change within the persistent schedule (all MAC cycles within a persistent schedule have the same MAC cycle). The duration may change when changing the persistent schedule.

When a MAP is lost, nodes shall maintain synchronization with the MAC cycle by inferring the start time of the next MAC cycle based on the persistency of the MAC cycle duration and reference to the start time of the previous MAC cycle. Until a MAP is correctly received again, nodes may transmit only in the persistent TXOPs allocated for them, in accordance with medium access rules for these TXOPs (for example, in a CFTXOP allocated for a connection or in an appropriate TS of a persistent STXOP).

The details of description of a persistent TXOP in the MAP are in clause 8.8.3.

8.3.2 TXOP and TS attributes

The TXOPs and TSs are described in the MAP by TXOP descriptors (see clause 8.8.4). A TXOP descriptor shall contain at least the duration of the TXOP in TIME_UNITS, the type of the TXOP (indicating whether the TXOP is a CFTXOP or STXOP or CBTXOP) and an association between the TXOP and the DEVICE_IDs of the nodes allowed to transmit within the TXOP. The TSs inside a STXOP are described using descriptors, specified in clause 8.8.4.

8.3.3 Medium access in STXOPs

An STXOP may contain zero, one or more than one TSs of CFTS or CBTS type.

Each TS inside an STXOP is identified by its order within the STXOP. Each CFTS 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).

A CBTS is assigned to a group of nodes as described in clause 8.8.4.1.5 and a minimum user priority as described in clause 8.8.4.2. The order in which TSs appear, and how nodes, connections or user priorities are assigned to them, is based on a scheduling policy used by the domain master. Those scheduling policies are beyond the scope of this Recommendation.

An STXOP is divided into a grid of TSs providing transmission opportunities to the nodes sharing the STXOP. The grid starts at the beginning of an STXOP and the grid timing is reset after each transmission as described in clause 8.3.3.1 below.

Nodes that share an STXOP shall track the passage of TSs on the line using carrier sensing and transmit only within their assigned TS.

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.

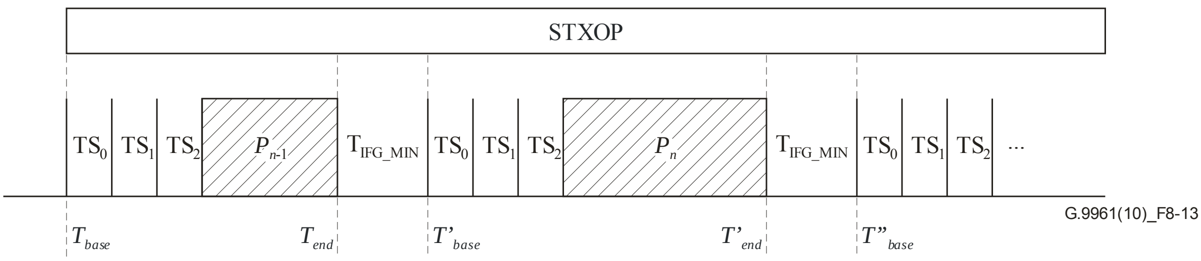


Figure 8-13 – Time slot timing

The duration of an unused time slot (e.g., TS_0 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.

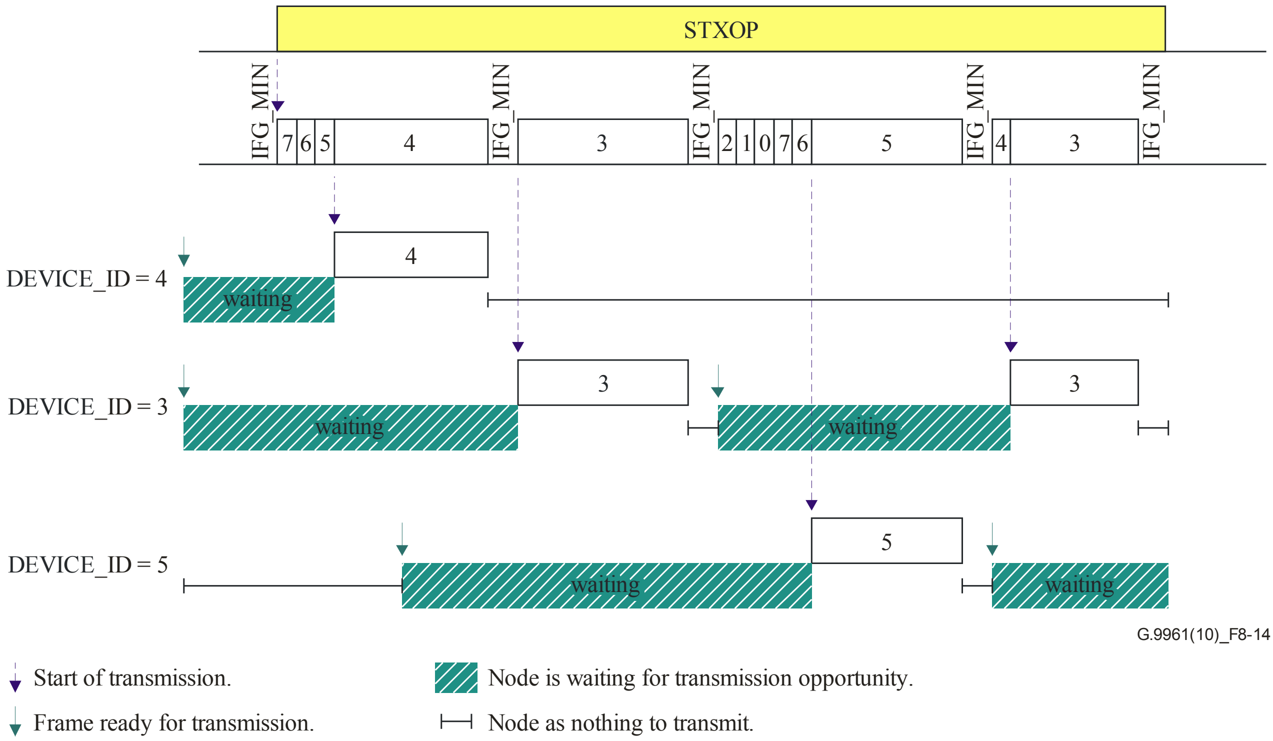


Figure 8-14 – Sequential TS assignment rule example

In Figure 8-14, each TS is marked with a number that is associated with the node to which 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 255 groups due to the limitation in field size of the group information extension (see clause 8.8.4.1.3). 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.

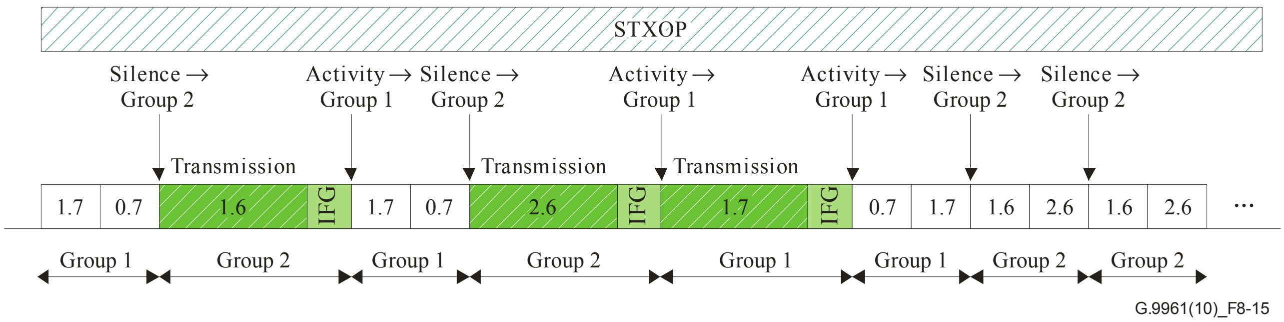
The next TS of a group is the TS that follows the current TS in that group as described in the MAP. The next TS of that STXOP shall be the next TS of the group of TSs to which control is passed following the current TS. The group to which control is passed following the current TS is defined by the sequential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

When the current group is assigned a line activity dependent assignment rule via the group information extension as described in clause 8.8.4.1.3, the next TS depends on the actual usage of the current TS as described below:

– If the current TS was not used, and all the TSs of the current group have provided opportunities to the nodes assigned to these TSs to access the medium in the current appearance of the group in the STXOP:
  - then the next TS of that STXOP shall be the next TS of the group that control is to be passed to upon silence (GroupOnSilence);
  - otherwise, control shall be passed to the next TS in the current group of that STXOP according to the sequential TS assignment rule.

– If the current TS was used, the next TS of that STXOP shall be the next TS of the group of TSs that control is to be passed to upon activity (GroupOnActivity).

Figure 8-15 describes an example of an STXOP that contains two groups. The first group consists of one CFTS (node 1) and one CBTS (node 0) both with user priority 7. The second group consists of two CFTSs (nodes 1 and 2), both with user priority 6. The line activity dependent TS assignment was used to specify that control shall be passed from either group to the first group on activity, and to the second group on silence.



NOTE – x.y denotes - node x and priority y.

Figure 8-15 – An example of line activity dependent TS assignment rule

In the example in Figure 8-15, whenever any TS of the two groups is used, control is passed to the first group. Whenever there is no activity in all the TSs of either of the two groups, control is passed to the second group. The assignment of the TSs of that STXOP within each group remains sequential, as described in clause 8.3.3.2.1.

8.3.3.3 Transmission in CFTS

A node shall transmit at the beginning of the CFTS, within a time window of TX_ON microseconds after the start of the CFTS. The values of TX_ON are described in clause 8.4. The start of the CFTS shall be computed as described in clause 8.3.3.1.

If the CFTS 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 CFTS.

If the CFTS 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 using this CFTS.

If the CFTS 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 CFTS.

8.3.3.4 Transmission in CBTS

A node assigned to the CBTS may contend for the medium in a CBTS only with MPDUs of equal or higher MPDU priority than the user priority assigned to the CBTS. Transmission in CBTS is illustrated in Figure 8-16.

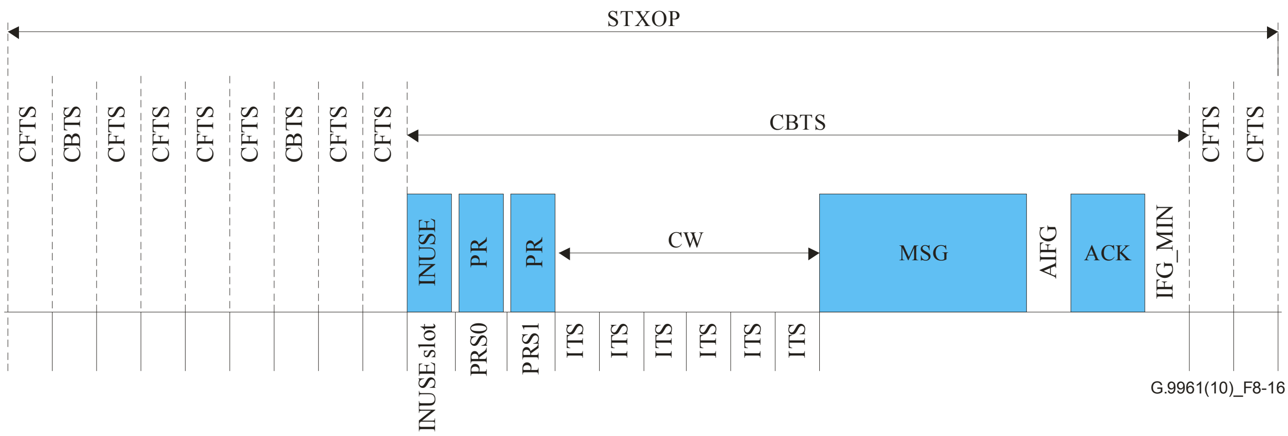


Figure 8-16 – An example of transmission in a CBTS

In general, a CBTS used for transmission is combined from an INUSE slot, followed by two priority resolution slots (PRS), followed by a contention time window (CW) , which consists of idle time slots (ITS), and the frame sequence transmission time. The INUSE and priority resolution slots are present only if transmission of INUSE signal and PR signals, respectively, is required by the MAP; otherwise these slots shall not be a part of the CBTS, as described in clauses 8.3.3.4.5 and 8.3.3.4.6.

The overall medium access process includes the following three steps:

1) The contending nodes indicate their participation in contention by INUSE signal (if required by the MAP).
2) The contending nodes perform priority resolution by transmitting and monitoring priority resolution (PR) signals during the two PRS (if required by the MAP) as defined in clause 8.3.3.4.1.
3) The nodes that won the priority resolution contend for transmission during the CW, based on back-off rules defined in clause 8.3.3.4.3.
4) The node that won the contention transmits a frame or a frame sequence.

In case the attribute of a CBTS in the MAP requires using an INUSE signal prior to contending (see clause 8.8.4), the contending node shall transmit an INUSE signal at the beginning of the CBTS, within a time window of TX_ON (see clause 8.4) microseconds after the start of the CBTS.

Nodes sharing an STXOP shall consider a CBTS as not used if an INUSE signal is required but was not detected. In this case nodes shall advance to the next TS according to the TS assignments rules described in the MAP.

8.3.3.4.1 Priority resolution

Priority resolution between nodes contending in a CBTS shall be done using PR signals that shall be transmitted in two PRS: PRS0 and PRS1. The PRS shall follow the INUSE signal, if INUSE signal is used, or start at the beginning of the CBTS, if INUSE signal is not used. The transmission of PR signal shall occur within a time window of TX_ON microseconds after the start of the corresponding PRS.

A node contending for the medium within a CBTS shall advertise the medium access priority of its planned transmission by signalling within the PRS slots according to the mapping specified in Table 8-5. A node shall use the medium access priority according to Table 8-6.

Table 8-5 describes mapping of medium access priorities to the PR signal combinations, where medium access priority 0 (least important) is denoted as MA0 and medium access priority 3 (most important) is denoted as MA3.

Table 8-5 – Mapping of medium access priorities to PR signals

Medium Access Priority	PR signal transmitted in PRS0	PR signal transmitted in PRS1
MA3	Yes	Yes
MA2	Yes	No
MA1	No	Yes
MA0	No	No

If a node participates in priority resolution and detects a PR signal in any PRS slot in which it did not transmit, the node shall not transmit in the remaining PRS slot and shall not compete during the contention time, unless it receives an MPDU for transmission with MPDU priority that is the same or higher than the MA priority that won the priority resolution in the CBTS.

If a node did not participate in priority resolution during PRS0, it may participate in priority resolution during PRS1 if the MPDU to be transmitted corresponds to MA2 or MA3 using the procedure described in this clause.

To compete for transmission of its frame, the node shall use the back-off procedure defined in clause 8.3.3.4.3.

In case the attribute of the CBTS in the MAP does not require priority resolution (see in clause 8.8.4), all nodes that are allowed to contend for transmission in a CBTS may compete for medium access during the CW using a back-off procedure that is defined in clause 8.3.3.4.3. In this case the CW shall follow the INUSE signal slot, if INUSE signal is required, or start at the beginning of the CBTS, if INUSE signal is not required.

NOTE – Nodes sharing a STXOP should track the priority resolution if they intended to contend for transmission with a data frame that arrives after the priority resolution is complete.

8.3.3.4.2 Mapping of MPDU priorities to medium access priorities

The medium access priorities MA0-MA3 shall be associated with the MPDU priority of the MPDU carried in the PHY frame contending for transmission in a CBTS, as defined in Table 8-6.

If a node supports less than four MA priorities, the mapping shall be as shown in Table 8-4.

Table 8-6 – Mapping of MPDU priorities to medium access priorities

Number of supported MA priorities	**MPDU priority**							
	lowest							**highest**
	1	2	0	3	4	5	6	7
4	MA0		MA1		MA2		MA3	
3	MA1				MA2		MA3	
2	MA1				MA3			
1 (Note)	MA1							
NOTE – A node supporting only 1 MA priority shall still use MA3 for MAP transmission.								

8.3.3.4.3 CBTS back-off rules

All nodes contending in a CBTS shall use the back-off rules described in this clause in the CW. In the general case, CW immediately follows the PRS, as shown in Figure 8-16. The size of CW is expressed in the number of ITS. The valid values for the maximum range of the CW are defined in Table 8-7, the value of ITS is defined in clause 8.4. If PR signals are not required, the CW shall start right after the INUSE signal slot, as described in clause 8.3.3.4.6, or at the beginning of the CBTS, if INUSE is not used.

Each node shall maintain the following back-off parameters for each MA priority of the frame that node intends to transmit:

- back-off-counter (BC);
- defer counter (DC); and
- back-off stage counter (BSC).

The BC determines the number of ITS the node has to wait before it begins the transmission. The DC keeps track of the number of consecutive times a node can lose contention before changing the back-off parameters. The BSC keeps track of the back-off stage to enable the selection of BC and DC when the back-off stage changes.

Nodes that are allowed to compete in the CW shall use their back-off parameters for that MA priority, and act according to the following rules before starting a transmission in a CBTS:

1) If the BC is zero, the node shall start transmitting its frame within a time window of TX_ON microseconds after the start of the first ITS of the CW.
2) If the BC is not zero, the node shall decrement its BC upon completion of each ITS in which it detects no transmission.
3) If, upon completion of certain ITS, the value of BC is zero, the node shall start transmitting its frame within a time window of TX_ON microseconds after the end of the ITS.
4) If a node detects a transmission during an ITS, it shall not transmit in this CBTS and shall do the following:
   - The node shall decrement the DC.
   - If the DC is zero and BSC is less than BSC_{max}, the node shall increment the BSC. If the DC is zero and BSC is equal to BSC_{max}, the node shall maintain the current BSC. It shall then set DC to $DC_{max}(BSC)$ and BC to a random value in the range of $(0, NCW_{max}(BSC) – 1)$.
   - If the DC is greater than zero, the node shall decrement the BC.

Nodes that have inferred a collision (see clause 8.3.3.4.9) shall increment the BSC if BSC is less than BSC_{max}. It then sets DC to $DC_{max}(BSC)$ and BC to a random value in the range of $(0, NCW_{max}(BSC) – 1)$.

After initialization and upon successful transmission, nodes shall initialize BSC to 1, DC to $DC_{max}(1)$ and BC to a random value in the range $(0, NCW_{max}(1) – 1)$.

Table 8-7 shows the valid values of $DC_{max}(BSC)$ and $NCW_{max}(BSC)$. These valid values are used for all MA priorities. BSC_{max} shall be 4.

Table 8-7 – Valid $DC_{max}(BSC)$ and $NCW_{max}(BSC)$ values

BSC	$DC_{max}(BSC)$	$NCW_{max}(BSC)$
1	1	8
2	2	16
3	4	32
4	16	64
NOTE – Other values of BSC, DC_{max} and NCW_{max} are for further study.		

If a node that is allowed to contend in a CBTS has an MPDU ready to transmit after the start of the CW, it is still allowed to contend with this MPDU using the back-off procedure defined in this clause only if the MPDU's MA priority is equal to or higher than the MA priority that won the priority resolution. The node shall pick the BC random value for the ITS in the CW in the same way as nodes that had the frame ready to transmit prior to the start of the CW, and shall start decrementing the BC from the ITS where the frame was ready for transmission. The BC, DC, BSC values that shall be used are of the frame's MA priority value.

8.3.3.4.4 Use of RTS/CTS signalling

If the attributes of the CBTS defined in the MAP indicate usage of an RTS/CTS protocol (see clause 8.8.4.1.1), a node that gets a right for transmission in a CBTS shall transmit an RTS frame prior to the transmission of the MSG frame. The node whose DEVICE_ID is indicated in the DID field of the PHY-frame header of an RTS frame shall transmit a CTS frame, except in cases described below, to the node that sourced the RTS frame T_{RCIFG} after it receives the RTS frame. The node that sourced the RTS shall transmit its MSG frame only if it has received the corresponding CTS frame after the RTS frame it has sent. The MSG frame shall be sent T_{CCIFG} after the CTS frame is received. Transmission of a frame sequence including an MSG frame with Imm-ACK using RTS/CTS signalling is presented in Figure 8-17.

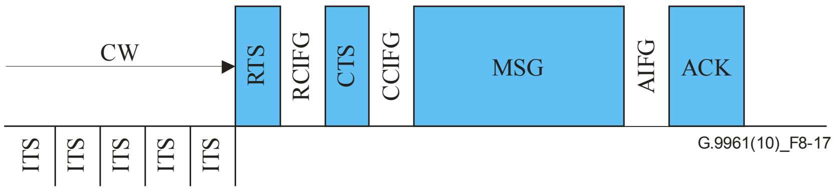


Figure 8-17 – Use of RTS/CTS signalling in a CBTS

In case of unicast transmission, the destination address of the RTS frame shall be the same destination address as in the following MSG frame. In case of multicast (broadcast) transmission the destination address of the RTS frame shall be the destination address as in the following MSG frame (which is a multicast address) and the CTS proxy ID (CID) shall be set to the destination address of one of the nodes that are members of the multicast group.

A node that received an RTS frame shall not transmit a CTS frame to the source of the RTS frame in the following cases:

– if the node is not prepared or capable of receiving the following MSG frame;

– if the node detected that the medium is busy or is expected it to be busy at the time of the CTS frame or the expected following MSG frame transmission (i.e., the node detected that another frame or frame sequence, e.g., another RTS or CTS, is transmitted).

The duration field of the RTS and CTS frames shall indicate the duration of the frame sequence, as defined in clauses 7.1.2.3.2.4.1 and 7.1.2.3.2.5.1 of [ITU-T G.9960]. All nodes detecting RTS or CTS, or both RTS and CTS, shall consider the CBTS as already used for transmission and refrain transmission until the closure of the CBTS, as described in clause 8.3.3.4.5.

Nodes that detected no CTS frame during the time period of $T_{CTS\text{-}MAX}$ microsecond after the RTS frame was transmitted shall declare the status of the CTS frame as "not received". The value of $T_{CTS\text{-}MAX}$ shall be equal to:

$$T_{CTS\text{-}MAX} = T_{RTS} + T_{RCIFG} + T_{CTS} + T_{CCIFG}$$

where T_{RTS} and T_{CTS} are the durations of the RTS and CTS frames, respectively, and T_{RCIFG} and T_{CCIFG} are the durations of the RCIFG and CCIFG gaps, respectively (see clause 8.4).

Figure 8-18 describes an example of a CBTS in which RTS/CTS protocol is used and CTS frame is not detected.

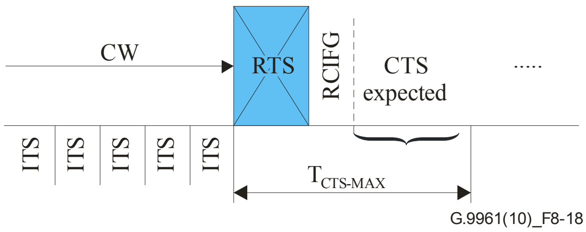


Figure 8-18 – Example of CTS not detected when RTS/CTS is used

Use of RTS/CTS signalling is not allowed for STXOP and CBTXOP in which INUSE signal is required (see clause 8.3.3.4.7).

8.3.3.4.5 Closing of CBTS

8.3.3.4.5.1 STXOP containing CBTS and CFTS

Nodes sharing a STXOP shall close a CBTS that was used for transmission according to its closure mode as defined in the MAP (see clause 8.8.4).

A CBTS closure mode can be one of the following:

1) Duration-based.

2) Timeout-based from frame sequence start.

3) Timeout-based from CBTS start.

8.3.3.4.5.1.1 Duration-based CBTS closure

When the CBTS closure mode is duration-based, all nodes sharing a STXOP shall close a CBTS that was used for transmitting IFG_MIN after the transmission sequence ends using the DURATION 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.

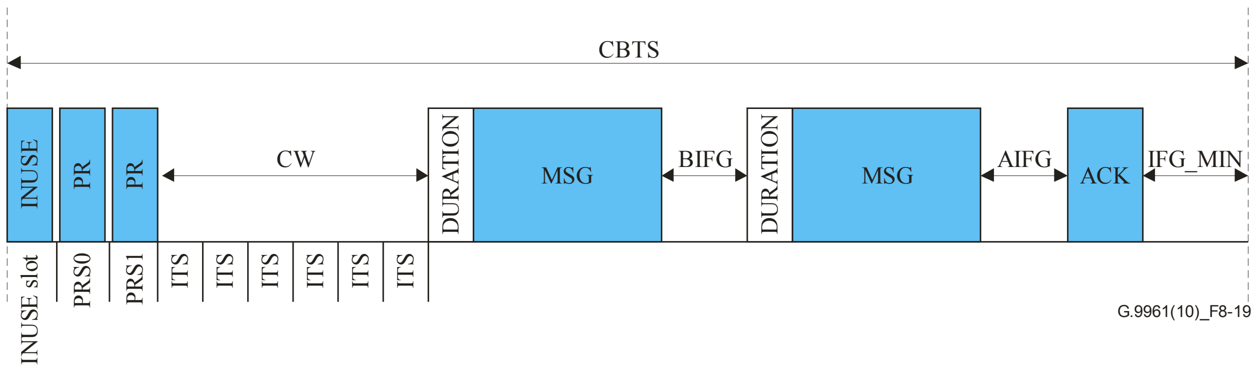


Figure 8-19 – Example of duration-based CBTS closure

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_TBFFSS_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.

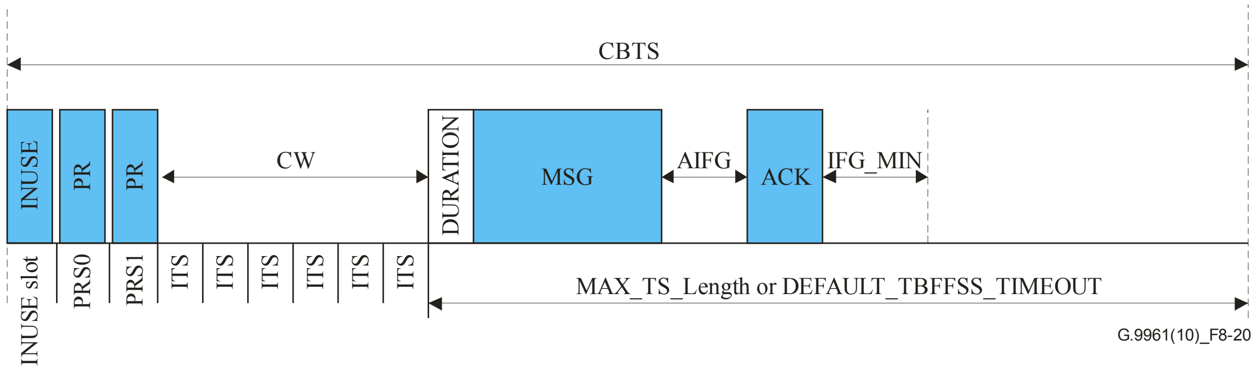


Figure 8-20 – Example of timeout-based from frame sequence start CBTS closure

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.

8.3.3.4.5.1.3 Timeout-based from CBTS start CBTS closure

When the CBTS closure mode is timeout-based from CBTS 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 CBTS, 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_TBFCS_TIMEOUT (see clause 8.4) shall be used instead of MAX_TS_Length.

Figure 8-21 describes a timeout-based from CBTS start CBTS closure.

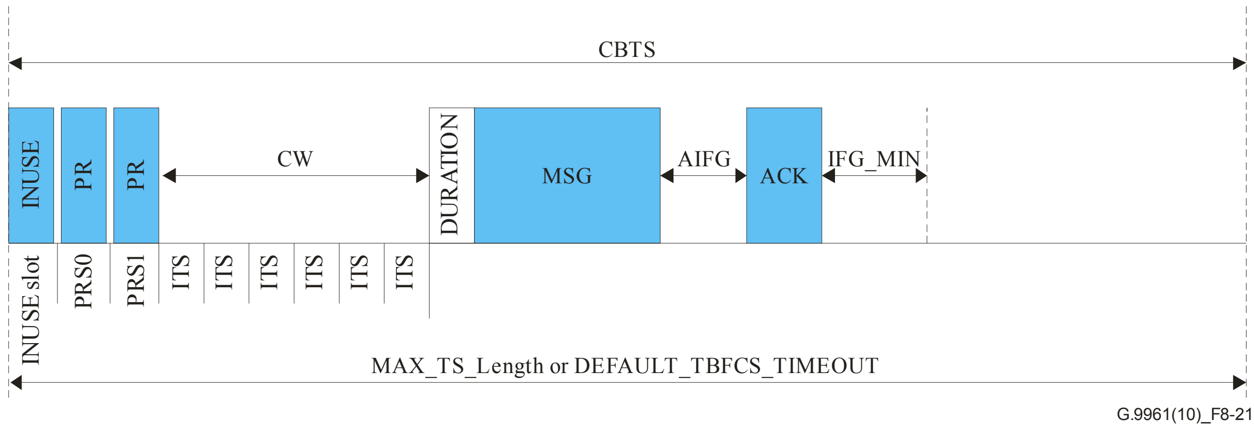


Figure 8-21 – Example of timeout-based from CBTS start CBTS closure

8.3.3.4.5.2 CBTXOP containing CBTS with different attributes

The rules described in this clause relate to CBTXOPs containing CBTSs whose MAP attributes define different user priorities, or different set of contending nodes, or different use of PR signals. For this CBTXOP the same rules as for STXOP containing CBTS and CFTS (see clause 8.3.3.4.5.1) shall be applied.

8.3.3.4.5.3 CBTXOP containing CBTS with same attributes

A receiving node sharing a CBTXOP where its CBTSs are without INUSE signal shall close a CBTS using duration-based closure (see clause 8.3.3.4.5.1.1) when a valid PHY-frame header is received.

A transmitting node sharing a CBTXOP where its CBTSs are without INUSE signal shall close a CBTS using duration-based closure (see clause 8.3.3.4.5.1.1) when it transmits a frame sequence and did not infer a collision.

When RTS/CTS signalling is used (see clause 8.3.3.4.4) CBTS closure shall be as follows:

1) When the transmitter of the RTS did not receive any CTS (as described in clause 8.3.3.4.4) it shall close the CBTS $T_{RTSCTS\text{-}TIMEOUT}$ microseconds (see clause 8.4) from the start of the RTS frame. The value of $T_{RTSCTS\text{-}TIMEOUT}$ shall be equal to:

$$T_{RTSCTS\text{-}TIMEOUT} = T_{RTS} + T_{RCIFG} + T_{CTS} + T_{CCIFG} + T_{Preamble\text{-}first\text{-}section}$$

where T_{RTS} and T_{CTS} are the durations of the RTS and CTS frames, respectively, T_{RCIFG} and T_{CCIFG} are the durations of the RCIFG and CCIFG gaps, respectively (see clause 8.4), and $T_{Preamble\text{-}first\text{-}section}$ is the duration of the first section of the preamble.

2) When the transmitter of the RTS receives a CTS intended for another node, it shall consider the media as busy and shall close the CBTS according to the duration of this CTS, i.e., using duration-based closure.

3) When a node receives an RTS frame but did not detect a preamble for $T_{RTSCTS\_TIMEOUT}$ microsecond from the start of the RTS frame, it shall close the CBTS $T_{RTSCTS\text{-}TIMEOUT}$ microsecond from the start of the RTS frame.

4) In all other cases the CBTS shall be closed using duration-based closure.

In the following error conditions, the node shall refrain from transmission for the specified timeout and shall close the CBTS using duration-based closure if a valid PHY-frame header is received during the timeout or shall close the CBTS when the timeout has expired if no preamble is detected. If a subsequent PHY-frame header error occurs during the timeout period, the node shall act according to case 1 below.

Table 8-8 – Timeout setting for error conditions in CBTS without INUSE

Case	Error condition	Timeout setting
1	PHY-frame header error	DEFAULT_TBFFSS_TIMEOUT (see clause 8.4) from frame start
2	Collision is inferred (see clause 8.3.3.4.9)	DEFAULT_ERR_CWOI_TIMEOUT (see clause 8.4) from expected ACK start
3	PR signal detected, but no frame detection occurred	DEFAULT_ERR_CWOI_TIMEOUT from CBTS start

When there is no PR signal and no frame is detected, the node shall infer that the medium is idle and shall not close the CBTS until one of the following occurs:

1) A frame is detected – the node shall close the CBTS according to the rules specified above.

2) The node starts transmitting a frame – the node shall close the CBTS according to the rules specified above.

3) The CBTXOP has ended – the node shall close the CBTS.

In the case that a node closes a CBTS due to a timeout ($T_{RTSCTS\text{-}TIMEOUT}$, DEFAULT_TBFFSS_TIMEOUT, or DEFAULT_ERR_CWOI_TIMEOUT), the node shall not transmit Priority Resolution signals during the priority resolution period in the following CBTS and shall assume MA0 priority won the priority resolution.

8.3.3.4.6 Use of CBTS with no PR signals

If PR signals are not required by the MAP, the CW shall start right after the INUSE slot, if INUSE signal is required, as described in Figure 8-22, or at the beginning of the CBTS if INUSE signal is not required.

When PR signals are not required, all nodes that are allowed to contend in the CBTS may compete for transmission in the CW.

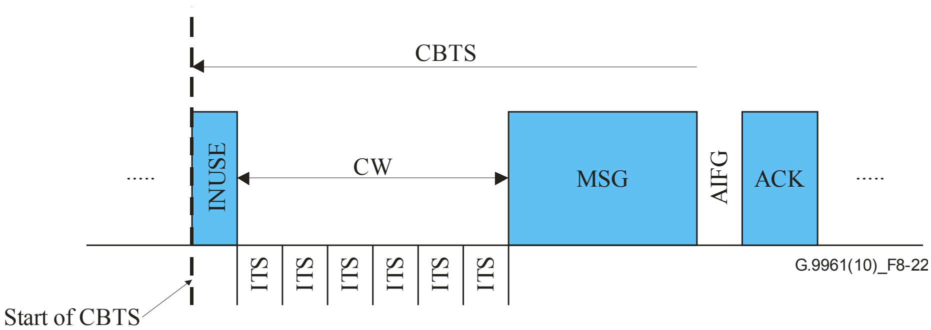


Figure 8-22 – Example of transmission in a CBTS with no PR signals (INUSE signal is used, RTS/CTS is not used)

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 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×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 a robust STXOP
No frame detection after PRS was detected	Duration-based CBTS without INUSE
CURRTS (see clause 7.1.2.3.2.2 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 shall resynchronize with the TS grid identity using the CURRTS field of the received PHY-frame header.

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 payload symbols, the gap between the frame and the following Imm-ACK shall be $T_{AIFG\text{-}D}$ (see clause 8.4), otherwise the gap shall be $T_{AIFG.}$ The parameter MIN_SYM_VAR_AIFG is defined in clause 8.4, for each media. The transmitter indicates usage of either T_{AIFG} or $T_{AIFG\text{-}D}$ by using the AIFG_IND bit in the PHY-frame header (see clause 7.1.2.3.2.2.16 of [ITU-T G.9960]).

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

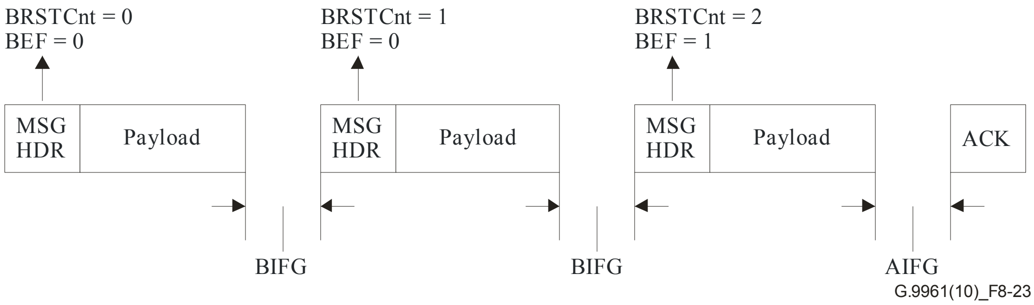


Figure 8-23 – Example of PHY frame transmission with bursting and Imm-ACK

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 and 7.1.2.3.2.2.14 of [ITU-T G.9960], respectively.

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

When 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 CFTXOP 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-bursted 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-bursted 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-bursted 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-bursted 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-bursted 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-bursted 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 consecutive time that a node remains in active state is denoted as active period. The consecutive 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.

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) and idle mode (L3).

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. In this case the specified inactive period followed by the active period will repeat until it is cancelled or changed by the domain master.

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).

All nodes shall track the inactivity scheduling using the domain master transmit clock which is distributed via the MAP message.

A node that is scheduled to enter inactive state shall be able to receive frames transmitted 100 μs before the beginning of the inactive state and shall finish the current frame sequence exchange before entering inactive state.

The node that requested an inactivity schedule 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.

During the inactive period, a node is not required to decode the MAP. After the inactive period ends, the node shall transition back into the active state. The duration of any inactive period shall be larger than or equal to a MAC cycle and shall not exceed the re-registration period except for the case of idle mode (L3). After the current schedule expires, a node may request another inactivity schedule.

The domain master, if the request is rejected, shall indicate the denial of the request for inactivity with reason code by sending the IAS_LongInactivity.cnf message. The node that received an inactivity denial shall act based on the reason code.

A node can request to change the current inactivity schedule by sending another inactivity schedule, or cancel the current inactivity schedule by sending IAS_LongInactivity.req with LIS_TYPE = 2 while it is in active period. 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_LongInactivity.req and IAS_LongInactivity.cnf messages shall be as shown in Table 8-10 and Table 8-11, respectively.

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

Field	Octet	Bits	Description
LIS_TYPE	0	[2:0]	Proposed type of long inactivity scheduling 0: inactivity schedule is valid only once. In this case LIS_ACT_DUR shall be set to zero. 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 LIST_INACT_DUR and LIST_ACT_DUR shall be set to zero. Other values are reserved by ITU-T.
Reserved		[7:3]	Reserved by ITU-T (Note)
LIS_INACT_DUR	1 and 2	[15:0]	Requested duration of the inactive period, expressed in 5 ms units, 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.
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 5 ms units, 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
Reason code	0	[2:0]	3-bit reason code of inactivity denial 000 = no reason specified (Note 1) 001 = proposed inactivity period is too long 002 = proposed inactivity period is too short
Reserved		[7:3]	Reserved by ITU-T (Note 2)
NOTE 1 – Definition of other reason codes 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.			

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 efficient-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.

The domain master, if the request is accepted, shall announce the inactivity scheduling as proposed by the receiver. 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 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 the denial of the request for inactivity with reason code by sending the IAS_ShortInactivity.cnf message. The node that received an inactivity denial shall act based on the reason code.

Nodes in short inactivity scheduling shall be able to decode the MAP at every MAC cycle. The domain master shall not schedule inactive portions of the MAC cycle when it is scheduled to transmit a MAP.

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 and IAS_ShortInactivity.cnf messages shall be as shown in Table 8-12 and Table 8-13, 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	[7:0]	Requested indication of one or more inactive periods within a MAC cycle represented, as an 8-bit unsigned integer. 8-bit map is used to represent inactive periods. The bit0 (LSB) and bit7 (MSB) correspond to the first and last 1/8-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
Reason code	0	[2:0]	3-bit reason code of inactivity denial 000 = no reason specified 001 = proposed inactivity period is too long 002 = proposed inactivity period is too short (Note 1)
Reserved		[7:3]	Reserved by ITU-T (Note 2)
NOTE 1 – Definition of other reason codes 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.			

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 7-43 and 7-44 of [ITU-T G.9960], and the BACK PHY frames shall use the format described in Tables 7-45 and 7-46 of [ITU-T G.9960], in which the PHY frame header contains 2×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]). If using unacknowledged bidirectional transmission, the BMSG and BACK PHY frames shall use the format described in Tables 7-43 and 7-45 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 set to zero).

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 only immediate acknowledgement is allowed (the valid values of RPRQ field are 00 and 01 only).

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.

If a source node requested by a destination node to initiate bidirectional transmission accepts the request, it shall indicate that the request is granted and shall initiate bidirectional transmission by transmitting a BMSG frame with the BTXGL field set to a non-zero value. A source node requested to initiate bidirectional transmission may decline the request. In this case it indicates that the bidirectional transmission request is declined by continuing to send MSG frames to the requesting node, instead of BMSG frames.

A source node may initiate bidirectional transmission autonomously, without a request from the destination node by transmitting a BMSG frame with the BTXGL field set to a non-zero value.

The acknowledgement information in a BMSG frame that initiates a bidirectional transmission shall be disabled by setting the FACK field to 111 if the last MSG frame received from the destination node was already acknowledged or no acknowledgement is required; otherwise it shall include acknowledgement information on the recent MSG frames received from the destination node.

A source node may at any time terminate bidirectional transmission and re-start it again. The destination node may indicate to the source node when the bidirectional transmission may be stopped, while the decision is up to the source node.

Once bidirectional transmission is initiated by the sourcing node, the following procedure shall be used for bidirectional transmission:

1) A destination node responds to the BMSG frame that initiates bidirectional transmission by transmitting a BACK frame that contains data in the payload intended for the source node. If the source node requested acknowledgement the BACK frame additionally contains acknowledgement information for data previously transmitted by the source node. In the BTXRL field of the frame header the destination node indicates the requested duration of the next BACK frame it expects to transmit.
2) The source node, in response to the received BACK frame, transmits a BMSG frame indicating the granted maximum duration of the next BACK frame in the BTXGL field of the PHY-frame header.
3) The destination node, in response to the BMSG frame, transmits a BACK frame, continuing the exchange between the communicating nodes. The duration (see clause 7.1.2.3.2.10.1 of [ITU-T G.9960]) of the BACK frame shall not exceed the granted duration.
4) The source node may terminate the bidirectional transmission by one of the following methods:
   a) By setting BTXGL = 0 in any of the BMSG frames. In case BTXGL = 0 in the received BMSG frame and the RPRQ field indicates request for immediate acknowledgement, the destination node shall respond by an Imm-ACK frame.
   b) By setting BTXEF = 1 and BTXGL ≠ 0 in any of the BMSG frames. In this case, as BTXGL ≠ 0, the destination node may send a BACK frame prior to the termination of bidirectional transmission.
   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.

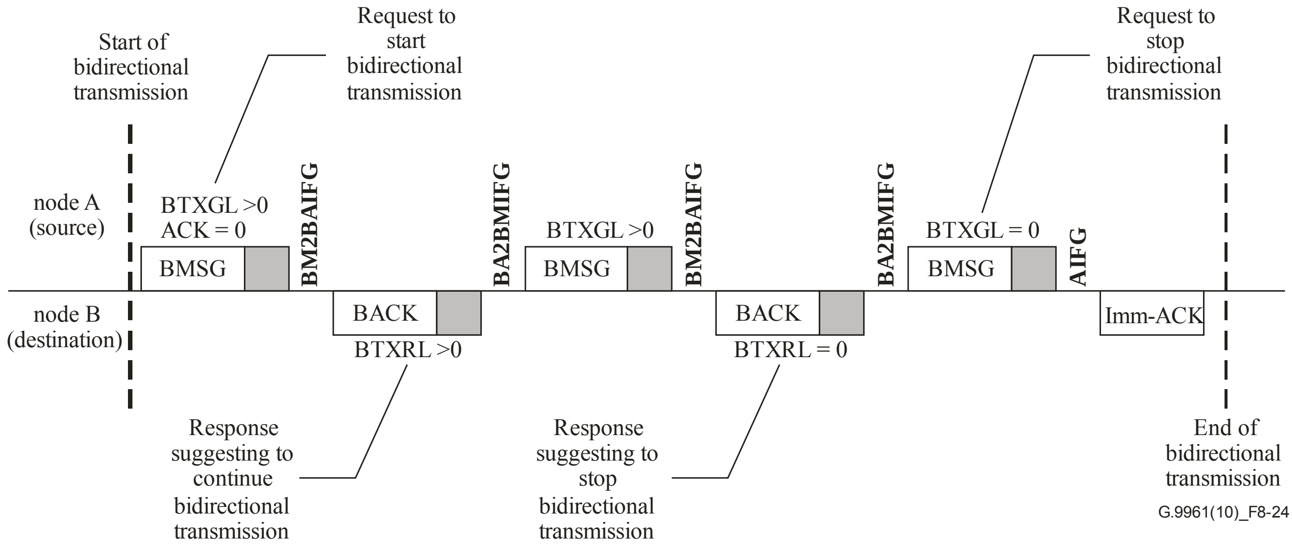


Figure 8-24 – Example of bidirectional transmission (invited by the originating node)

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 payload symbols, the gap between this frame and the following frame shall be $T_{AIFG\text{-}D}$ (see clause 8.4), otherwise the gap shall be $T_{AIFG.}$ The parameter MIN_SYM_VAR_AIFG is defined in clause 8.4, for each media. The transmitter indicates usage of either T_{AIFG} or $T_{AIFG\text{-}D}$ by using the AIFG_IND bit in the PHY-frame header (see clause 7.1.2.3.2.2.16 of [ITU-T G.9960]).

Bidirectional transmission can be used in CFTXOP, STXOP, and CBTXOP. The source node shall ensure that the total duration of the bidirectional frame sequence does not violate the boundaries of the TXOP or the maximum allowed duration of the TS. Particularly:

– if bidirectional transmission is established in a CFTXOP, the last frame in the sequence shall end at least T_{IFG_MIN} before the end of the CFTXOP;

– if bidirectional transmission is established in a CFTS or in a CBTS, the last frame in the sequence shall end at least T_{IFG_MIN} before the end of the Max_TS_Length assigned in the MAP for the TS and at least T_{IFG_MIN} before the end of the TXOP where this TS is defined.

Both the BMSG frame and the BACK frame may be sent as bursts of frames. The format of burst transmission and associated rules shall be as defined in clause 8.3.5 (all frames in a burst shall be of BMSG type or of BACK type). In case of acknowledged transmission, the acknowledgement information in the BACK and BMSG frame header shall use the format described in clause 8.3.5. All BMSG (or BACK) frames of the same burst shall carry the same acknowledgement information.

Both BMSG and BACK frames indicate their duration in the Duration field of the PHY-frame header as defined in clause 7.1.2.1 of [ITU-T G.9960]. For virtual carrier sense, the end of the bidirectional transmission frame sequence shall be calculated based on the duration of the last BMSG frame sent by the source node and the values of BTXEF, BTXGL and RPRQ depending on how the bidirectional transmission is terminated. When the bidirectional transmission is terminated with a BMSG frame with BTXEF = 1 and BTXGL ≠ 0, the total duration of the frame sequence shall include this BTXGL value, regardless of the actual duration of the last BACK frame.

Nodes detecting a bidirectional transmission shall stay silent until the end of the bidirectional transmission sequence or until the expiration of the Max_TS_Length of the corresponding TS, whichever comes first.

Bidirectional transmission is not allowed when RTS/CTS is used.

8.4 Control parameters for APC, LLC, and MAC

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

Parameter	Description	Medium			
		Power-line baseband (Note 2)	Coax BB	Coax RF	Phoneline
T_{IFG_MIN}	Duration of inter frame gap	90 µs	29 µs	29 µs	55 µs
CYCLE_MIN	Minimum duration of MAC cycle	2 AC cycles (Note 1)	5 ms	5 ms	5 ms
CYCLE_MAX	Maximum duration of MAC cycle	2 AC cycles (Note 1)	100 ms	100 ms	100 ms
TX_ON	A time window after the start of TS during which a transmission can start	1 µs	1 µs	1 µs	1 µs
TS_DURATION	Duration of time slot	35.84 µs	16.64 µs	16.64 µs	23.04 µs
T_{ITS}	Duration of idle time slot (ITS) composing the contention window (CW) in CBTS	35.84 µs	16.64 µs	16.64 µs	23.04 µs
$T_{AIFG\text{-}D}$	Default value of inter frame gap before Imm-ACK	122.88 µs	39.68 µs	39.68 µs	74.24 µs
T_{AIFG}	Range of values for inter frame gap before immediate acknowledgment (Note 4)	20.48 to 122.88 µs	5.12 to 39.68 µs	5.12 to 39.68 µs	20.48 to 74.24 µs

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

Parameter	Description	Medium			
		Power-line baseband (Note 2)	Coax BB	Coax RF	Phoneline
MIN_SYM_VAR_AIFG	The minimum number of payload symbols required in a frame to use receiver specific T_{AIFG}, instead of $T_{AIFG\text{-}D}$, as the A_{IFG} gap, between the frame and the following immediate acknowledgment.	2	5	5	2
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 minimal 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_CWOI_TIMEOUT	Default timeout for error conditions in "CBTS without INUSE" (see clause 8.3.3.4.5.3)	4.9 ms	3.61 ms	3.61 ms	4.04 ms
REG_TIMEOUT	Time out for domain master to respond to registration request (see clause 8.6.1.1.1)	200 ms	200 ms	200 ms	200 ms
REG_RETRY_TIMEOUT	Timeout for node to retry registration (see clause 8.6.1.1.1)	1 s	1 s	1 s	1 s
MAX_REG_ATTEMPTS	Max registration attempts	4	4	4	4
RES_TIMEOUT	Timeout for resigning node to wait for response from the domain master (see clause 8.6.1.1.3.1)	200 ms	200 ms	200 ms	200 ms
MAX_RES_ATTEMPTS	Max number of resignation attempts	4	4	4	4

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

Parameter	Description	Medium			
		Power-line baseband (Note 2)	**Coax BB**	**Coax RF**	**Phoneline**
CNM_TIMEOUT	Timeout associated with release of connections	200 ms	200 ms	200 ms	200 ms
T_{MCST}	The maximal time the transmitter waits for MC_GrpInfoUpdate.cnf from the multicast group receivers, before it may re-transmit the MC_GrpInfoUpdate.ind message	100 ms	100 ms	100 ms	100 ms
N_{MCST}	Maximal number of retransmissions of the MC_GrpInfoUpdate.ind message	2	2	2	2
T_{DM_UPDATE}	The domain master broadcasts the updated topology information, within this time duration, after receiving topology updates.	40 ms	40 ms	40 ms	40 ms
T_{N_RSP}	A node replies to the request for topology information from the domain master within this time duration, after receiving the request.	40 ms	40 ms	40 ms	40 ms
T_{UPDATE_MIN}	The minimum time a node waits after receiving message TM_DomainRoutingChange.ind from the domain master, before it can send a TM_ReturnDomainTopology.req message to the domain master.	40 ms	40 ms	40 ms	40 ms
INTER_MAP_RMAP_GAP	The minimum gap between the end of a MAP or RMAP frame and the beginning of a subsequent relay of this MAP or RMAP frame.	1 ms	1 ms	1 ms	1 ms
NOTE 1 – For power lines, the duration of the MAC cycle is 2 AC cycles (see clause 8.6.3.1). NOTE 2 – Specification of power-line passband is for further study. NOTE 3 – Use of PHY frame bursting for coax is for further study. NOTE 4 – A receiving node shall choose a value in this range.					

8.5 Functions of the endpoint node

The following paragraphs list the functions of an endpoint node.

8.5.1 MAC cycle synchronization and synchronized transmissions

An endpoint node is not allowed to transmit until it detects the domain master of the domain it is intended to operate and synchronizes with the MAC cycle indicated by the domain master in the MAP or RMAP. Selection of the domain master shall follow the procedures described in clause 8.6.6.

After synchronization with the MAC cycle (see clause 7.1.6.2 of [ITU-T G.9960]), the endpoint node may register with the domain master using the admission procedure described in clause 8.6.1.1.

After registration, the endpoint node shall operate according to the medium access rules described in clause 8.2 and clause 8.3, and shall strictly follow the TXOP and TS assignments in each MAC cycle advertised by the domain master via the MAP message.

The endpoint node indicates its capabilities and topology information as described in clause 8.6.4.

Resignation of the endpoint node from the domain shall be as defined in clause 8.6.1.1.3.

8.5.2 Bandwidth reservation

In order to support bandwidth reservation for flows and to manage flows that require QoS, endpoint nodes shall support the flow signalling protocol described in clause 8.6.2. The flow signalling protocol is used to establish flows with particular QoS parameters, modify them, or terminate them.

The endpoint node shall inform the domain master using the FL_ModifyFlowParameters.ind message (see clause 8.6.2.3.15) and the bandwidth reservation request field (BRURQ) in its PHY-frame header (see clause 7.1.2.3.2.2.19 of [ITU-T G.9960]) on changes in the service flow data rate and in the line transmission data rate for flows that have bandwidth reservations. The domain master shall be able to extend or shrink the resource allocation reserved for the flow accordingly.

8.5.3 Routing of ADPs

Each node shall inform the domain about the nodes of its domain it has detected as defined in clause 8.6.4.3.

Each node can have one or more applications associated with its AE (above its A-interface). Each application is identified by a unique 6-octet MAC address. Each node shall maintain the full list of addresses associated with applications above its A-interface. This list is referred to as local address association table (AAT). Each node shall also maintain the list of addresses associated with the AEs of other nodes in the domain. This list is referred to as a remote AAT. Each node provides its local AAT to the domain master and other nodes of the domain using topology management messages as described in clause 8.6.4.3.

Whenever a node receives an ADP from the A-interface, it uses its AAT to determine if the ADP is intended for the node itself (local in-band management message, see Annex A) or for an AE associated with another node.

- If the ADP is intended for a remote AE or is an in-band management message addressed to a different node, the node shall determine the destination DEVICE_ID of the node in its domain through which the remote AE can be reached and send the corresponding ADP directly or via relay nodes to this node. This destination DEVICE_ID is provided to the flow mapper (see Figure 8-2) and is further reached either directly or via relays.
- If the ADP is intended for a group MAC address belonging to the AEs of different nodes of the domain, the corresponding APDU may be associated with a destination MULTICAST_ID and be sent using multicast transmission. The association between the group of MAC addresses and the MULTICAST_IDs is provided by the DLL management entity. The mechanism of this association is vendor discretionary and may be based on various multicast protocols, such as IGMP. That ADP may also be sent to the appropriate nodes using unicast transmissions.

- If the ADP has a standard broadcast destination address ($FFFFFFFFFFFF_{16}$) or is not intended for any of the applications for which MAC addresses are listed in the AAT and not intended to the node itself, then the BRCTI bit in the LFH of the LLC frame carrying the corresponding APDU shall be set to one, so that APDU will be broadcast to all nodes in the domain using the procedure described in clause 8.5.4. If the EtherType of the ADP equals $22E3_{16}$ and the DA of that LCDU is the standard broadcast address, the corresponding APDU shall also be forwarded to the local DLL management entity.
- If the destination MAC address of a received ADP is found in the local AAT, the ADP shall be dropped without notification.

8.5.4 Broadcast of LLC frames

To facilitate broadcast of an LLC frame, every node shall obtain the broadcast routing table (BRT), as defined in clause 8.6.4.1.1.2. The BRT of a particular node contains a list of destination nodes (list of DEVICE_IDs), to which this particular node shall relay the broadcasted APDU or LCDU. This list depends on the source from which the broadcasted APDU or LCDU was received (see clause 8.6.4.1.1.2). It is up to the node to create multicast groups (see clause 8.16) or use PHY unicast transmissions or PHY broadcast transmissions to reach the destination nodes indicated in the BRT (the DID of the PHY frame could be a DEVICE_ID, or a MULTICAST_ID, or a BROADCAST_ID (FF_{16})). The use of the BRT for multicast distribution is for further study.

To broadcast an LLC frame, the node that originates the broadcast shall set the BRCTI bit in the LFH of the transmitted APDU or LCDU to one. The DID of the PHY frame shall be set based on the BRT. The DA of the broadcasted frame may be any address, including the standard broadcast address ($FFFFFFFFFFFF_{16}$).

If a node receives a broadcast LLC frame (APDU or LCDU, BRCTI = 1) from the medium, it shall first perform the filtering procedure described in clause 8.5.4.1. If the node does not drop the LLC frame as a result of that filtering procedure, the node shall perform the actions described in the rest of this clause.

A node that receives a broadcast LLC frame from the medium (APDU or LCDU, BRCTI = 1) shall forward this frame to the nodes indicated in the BRT (except in the case described below when the DA of the broadcast LLC frame is in the local AAT) without modifying the value of BRCTI.

NOTE – Nodes that are leaf nodes of the tree will have an empty branch path in its BRT (see clause 8.6.4.1.1.2), while non-leaf nodes of the tree will have one or more destination entries in its branch path. Non-leaf nodes are supposed to have relay capabilities in this description.

If a node (leaf or non-leaf) received from the medium a broadcast LLC frame that contains an APDU, it shall recover the ADP and pass it to the AE via the A-interface. If the EtherType of the received APDU equals $22E3_{16}$ and the DA of that APDU is the standard broadcast address or is the address of the DLL management, this APDU shall be also passed to the local DLL management entity.

If a node (leaf or non-leaf) received from the medium a broadcast LLC frame that contains an LCDU, it shall recover this LCDU and pass it to the local DLL management entity if the DA of that LCDU is the standard broadcast address or is the address of the DLL management.

In addition, if a node (leaf or non-leaf) received a broadcast LLC frame from the medium that contains an APDU with a DA that belongs to the local AAT of the node, different than the standard broadcast address, the node shall pass it to the A-interface and not relay the broadcast frame. If that broadcast LLC frame includes an LCDU with a DA equal to the address of the node, the node shall pass it to the DLL management entity and not relay the broadcast frame.

If a node (leaf or non-leaf) receives from the medium a non-broadcast APDU (BRCTI equal to zero) with a DA that does not belong to its AAT (both local and remote parts), it shall recover the

ADP and pass it to the AE. If the node is a non-leaf node, it shall also set the BRCTI bit in the LFH of the LLC frame to one and forward the frame based on the BRT.

If a non-leaf node receives from the medium a non-broadcast LCDU (BRCTI equal to zero) with a DA that is not its own address and does not belong to its remote AAT, the relay node shall set the BRCTI bit in the LFH of the LCDU to one and forward the frame based on BRT.

If a leaf node receives from the medium a non-broadcast LCDU with a DA that is not its own address, it may either drop it or pass it to its DLL management entity.

If a node (leaf or non-leaf) receives from the medium an LCDU with DA equal to the reserved MAC address 01-19-A7-52-76-96, the node shall pass that LCDU to its DLL management entity. The action taken by the DLL management entity depends on the contents of the LCDU and the role of the node in the domain.

The nodes relaying a broadcast message shall associate this message with the same priority as assigned by the sourcing node (communicated in the LPRI field of LFH).

8.5.4.1 Filtering of broadcast LLC frames

If a node receives a broadcast LLC frame (APDU or LCDU, BRCTI = 1) from the medium, it shall perform the following filtering operation:

If the SID of the PHY-frame containing this broadcast LLC frame is not in the root path from the OriginatingNode of that frame (see clause 8.6.4.1.1.2), the LLC frame shall be discarded without relaying it and without passing it to either the A-interface or DLL management entity.

8.5.5 Reporting of detected neighbouring domains

Each node shall send information to the domain master and other nodes of the domain about all detected neighbouring domains using the TM_NodeTopologyChange.ind message, as defined in clause 8.6.4.3.

8.5.6 MAP relaying

In some media types there is a chance that a node may not receive transmissions from the domain master (i.e., is hidden from the domain master). In order for such a node to be able to synchronize with the MAC cycle, another endpoint node shall relay the MAP upon the domain master (see Table 8-70) request, i.e., generate and transmit a relayed MAP (RMAP) frame.

All endpoint nodes indicate their capability to relay the MAP in ADM_NodeRegistrRequest.req and in TM_NodeTopologyChange.ind message (see clause 8.6.1.1.4.1).

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 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 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 which is assigned via the MAP to transmit an RMAP-D frame shall generate a reduced MAP-D frame according to the most updated information it currently has which is needed to build a reduced MAP-D frame. The reduced MAP-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. The reduced MAP-D shall include as well a TXOP descriptor to enable the hidden registering node to transmit the registration request message frame. The node that generates the MAP-D frame shall set the TXOPs descriptors to be consistent with those in the MAP of the same MAC cycle containing the full schedule. The TXOPs descriptors included in a reduced MAP-D frame shall be described using the absolute timing extension (see clause 8.8.4.1.1) when their start time is not equal to the end time of the previous described TXOP.

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 the MAP or an RMAP 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 in every MAC cycle. A node is designated if the domain master allocates a TXOP or TS to it to transmit an RMAP (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 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-leaf nodes as MAP/RMAP 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/RMAP 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 (see clause 8.6.4).

8.5.8 Retransmissions and acknowledgement

Every node shall be able to acknowledge transmissions as specified in the retransmission and acknowledgement protocol (see clause 8.9).

8.5.9 Bidirectional flows

Some transmission sessions, like TCP traffic, are actually bidirectional in nature, that is, both the transmitter and the receiver send data that is part of the same transmission session. Bidirectional transmission sessions can be served by using bidirectional service flows as defined in clause 8.5.9.1 or by bidirectional prioritized data connections as defined in clause 8.5.9.2.

8.5.9.1 Bidirectional service flows

In order to use bidirectional transmission in the context of service flows, the originator node shall establish a bidirectional service flow using the FL_OriginateFlow.req message (clause 8.6.2.3.6) with the bidirectional indication set to one in order to establish two service flows. It shall also include the TSpec and classifiers for the flow in the reverse direction. The bidirectional service flow has two FLOW_IDs: one from the originating node to the endpoint node (i.e., forward flow) assigned by the originating node and another in the reverse direction, from the endpoint node to the originating node, assigned by the endpoint node (i.e., reverse flow). In addition to its established service flow, the originating node shall inform the domain master about the identity of the endpoint node FLOW_ID as well as the bandwidth requirements specified by the TSpec of that flow. As a response to a 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.

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 transmit periodically 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 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 Name 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 device has no user interface, the manufacturer shall provide the device with a 6-byte registration code, which is also supplied explicitly to the user. Such a device 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 device 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.

The list of target domains (configured by the user) may include more than one entry. If a node fails to register to one domain from the list, it may try to register to another one (if more than one domain was detected), until the node is either successfully admitted to one of the target domains or runs out of the list.

If no MAP frame meeting the target DNI is found, the 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 of the registering node shall be set to zero. After registration 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.

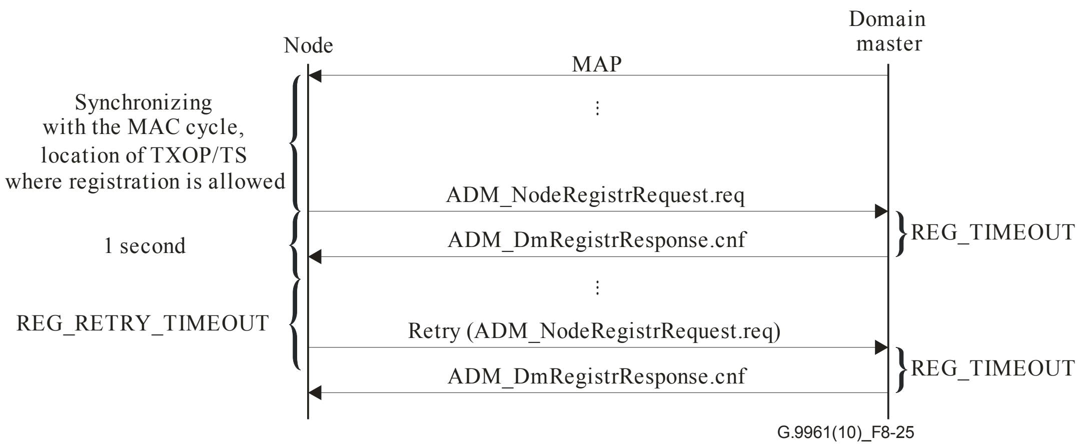


Figure 8-25 – Protocol diagram describing node registration

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).

To start the registration, the node shall send a registration request (ADM_NodeRegistrRequest.req) message to the domain master, which includes the REGID and other registration related parameters, as described in clause 8.6.1.1.4.1. The registering node is allowed to send ADM_NodeRegistrRequest.req during the RCBTS or during any other CBTS for which registration is allowed, using medium access rules for CBTS described in clause 8.3.3.4 with the MA priority associated with MPDU priority = 7.

The domain master shall process the registration request and shall reply within REG_TIMEOUT to the node with a registration response (ADM_DmRegistrResponse.cnf) message, which includes a status flag that indicates whether the domain master admitted the node to the domain or not. If the node is admitted, the ADM_DmRegistrResponse.cnf message shall contain a non-zero DEVICE_ID for the registering node assigned by the domain master and relevant configuration data. If the domain master rejects the admission, the ADM_DmRegistrResponse.cnf message shall contain a rejection code, describing the reason of rejection (see Table 8-15) and assigned DEVICE_ID = 0. The details of the ADM_DmRegistrResponse.cnf message are described in clause 8.6.1.1.4.2. The DID in the header of the PHY frame containing the ADM_DmRegistrResponse.cnf message shall be set to zero.

Registering nodes shall identify the ADM_DmRegistrResponse.cnf message based on its REGID field. The ADM_DmRegistrResponse.cnf message may be sent during the dedicated TSs or TXOPs, if assigned by the domain master, or during any CBTS, using medium access rules for CBTS described in clause 8.3.3.4 with the MA priority associated with MPDU priority = 7. If the registering node does not receive an ADM_DmRegistrResponse.cnf message from the domain master within one second, the node shall retry registration within REG_RETRY_TIMEOUT. If the registering node does not receive a response after MAX_REG_ATTEMPTS registration attempts, the node shall not continue registration attempts. If the registering node was rejected by the domain master, depending on the rejection code, the node may either retry registration during REG_RETRY_TIMEOUT or shall stop registration attempts. Valid admission rejection codes are presented in Table 8-15.

Table 8-15 – Admission rejection codes

Rejection code (Note 1)	Reason	Retry allowed
000	Unspecified	Yes
001	Insufficient bandwidth resources	Yes
010	Invalid set of registration parameters	No
011	Invalid REGID	No
100	Admission limit expired	Note 2
NOTE 1 – Other values reserved by ITU-T. NOTE 2 – Retry procedure in case of admission limit expired is for further study.		

Rejection codes associated with "Retry not allowed" requires re-configuration of the node, which includes modification of at least one of registration related parameters. After re-configuration, the node can attempt a new registration.

The domain master shall decide on the admission of the registering node based on the information supplied in ADM_NodeRegistrRequest.req message and the current status of the domain, evaluated by the domain master. The evaluation rules are vendor discretionary. The domain master may then assign resources to the registered node.

8.6.1.1.2 Periodic re-registrations

A node that is not in idle mode (L3) shall re-register with the domain master within the time period indicated in the MAP message (see Table 8-82) after registration (receiving the last ADM_DmRegistrResponse.cnf message) or re-registration. Re-registration shall use the same message exchange protocol as for registration with ADM_NodeRegistrRequest.req and ADM_DmRegistrResponse.cnf message format as described in clause 8.6.1.1.4.

For re-registration, the node shall transmit ADM_NodeRegistrRequest.req message, with format as described in clause 8.6.1.1.4.1, during any of its available TXOP or TS, but not during RCBTS. The domain master recognizes a re-registering node by its REGID. The domain master shall reply to the node by sending a ADM_DmRegistrResponse.cnf message during the dedicated TSs or TXOPs, if assigned by the domain master, or during any CBTS in which the node is allowed to transmit using medium access rules for CBTS described in clause 8.3.3.4 with the MA priority associated with MPDU priority = 7.

The domain master may force resignation from the domain of all nodes that failed periodic re-registration for two consecutive times using the procedure described in clause 8.6.1.1.3.2. The domain master shall cancel all bandwidth resources associated with the resigned nodes.

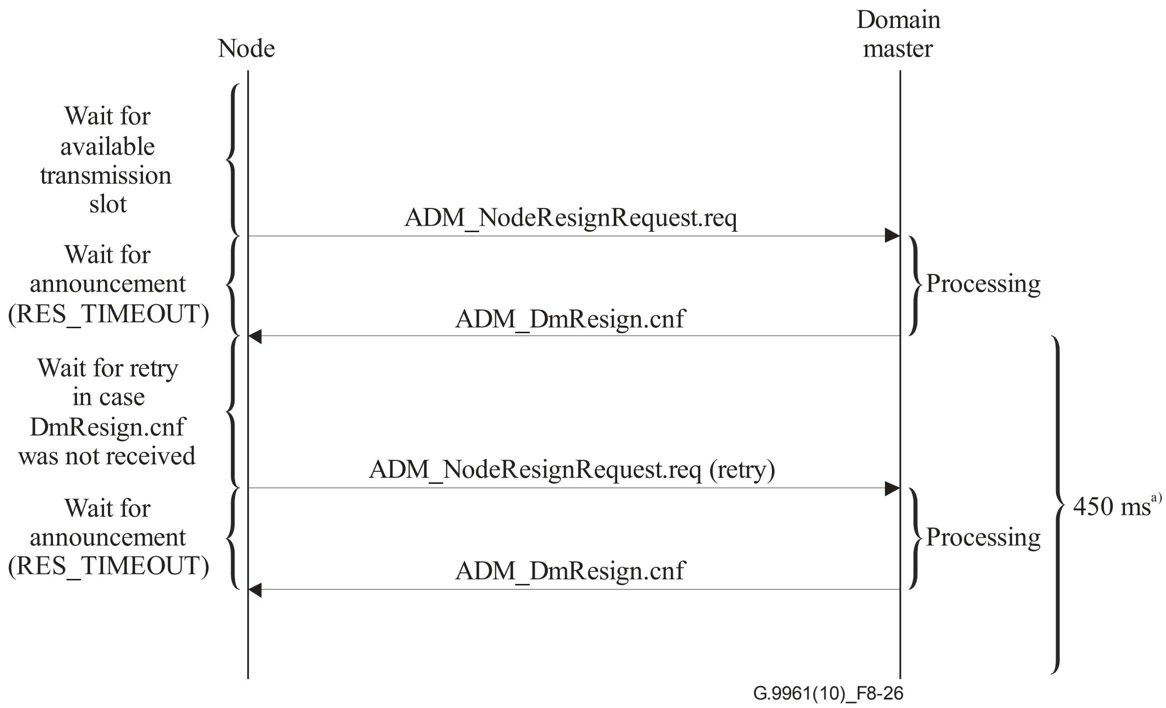
The resigned node may register again using the standard registration procedure, starting from the first available CBTS for which registration is allowed.

Re-registration of nodes in idle mode (L3) is for further study.

8.6.1.1.3 Resignation from the domain

8.6.1.1.3.1 Self-resignation

A node may resign itself from the domain (e.g., if there is no foreseen activity from the clients associated with the node). Self-resignation shall be performed using the protocol presented in Figure 8-26.



[a)] If no NodeResign.req is received, then the node is resigned.

Figure 8-26 – Protocol diagram describing node resignation

To resign, the node shall send to the domain master a resignation request (ADM_NodeResignRequest.req) message with the format defined in clause 8.6.1.1.4.3. The ADM_NodeResignRequest.req may be sent in any of the TXOP/TS available for the node.

The domain master shall process the resignation request and shall transmit the resignation confirmation (ADM_DmResign.cnf) message, as described in clause 8.6.1.1.4.4, which confirms the resignation request. The domain master shall also cancel all bandwidth resources associated with the resigned nodes. After receiving an ADM_DmResign.cnf message, the resigning node shall halt transmission until it decides to register back. If the resigning node does not receive a ADM_DmResign.cnf within RES_TIMEOUT, it shall re-send the ADM_NodeResignRequest.req message at the first opportunity. Not more than MAX_RES_ATTEMPTS attempts are allowed. After the last attempt the node shall considered itself as resigned. If no resend of ADM_NodeResignRequest.req message is detected within 450 ms after transmission of ADM_DmResign.cnf, the domain master shall consider the node as resigned.

NOTE 1 – If the domain master failed to receive any of ADM_NodeResignRequest.req messages, it will anyway resign the node at some point because of failed re-registration.

The DEVICE_ID of the resigned node shall be released at the first opportunity after the node resignation; domain master may assign released DEVICE_ID to new registered nodes.

NOTE 2 – If the domain master detects more than one node with the same DEVICE_ID, it can force all of them out, to register back later with different DEVICE_ID.

8.6.1.1.3.2 Forced resignation

Any node may be forced by the domain master to resign from the domain. To force resignation, the domain master shall send to the node a forced resignation request (ADM_DmForcedResign.req) message with the format presented in clause 8.6.1.1.4.5. Upon reception of ADM_DmForcedResign.req, the node shall reply with an ADM_NodeResignRequest.req message initiating a self-resignation procedure described in clause 8.6.1.1.3.1.

If the domain master does not receive the reply in 200 ms, it shall repeat the request and wait for the ADM_NodeResignRequest.req message again. If the reply is again not received in 200 ms, the domain master shall broadcast the ADM_DmResign.cnf and cut off all available bandwidth assignment for the node forced to resign.

8.6.1.1.4 Registration and resignation messages

8.6.1.1.4.1 Registration request message (ADM_NodeRegistrRequest.req)

The ADM_NodeRegistrRequest.req message is a unicast management message sent by a registering node to the domain master, and is intended to be used for registration and periodical re-registration requests only. The format of the MMPL of the ADM_NodeRegistrRequest.req message shall be as shown in Table 8-16.

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.
Reserved		[7:3]	Reserved by ITU-T (Note 1)

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

Field	Octet	Bits	Description
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 that the node shall use after registration represented as described in clause 7.1.2.3.2.2 of [ITU-T G.9960] (BNDPL/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 $n \times 1.28$ μs; 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.
NodeVersion	4	[7:0]	0 – Node supports version 0 of ITU-T G.9960 and ITU-T G.9961. All other values of this field are reserved by ITU-T for indicating support for future versions of the Recommendation. (Note 3)
Parameters	5	[0]	Set to one if node is capable of calculating routing tables, zero otherwise.
		[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. 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. NOTE 3 – A node indicating support for a certain version of this Recommendation shall also support all earlier versions of this Recommendation.			

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, 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
Bandplan		[3:1]	Bandplan used by new registering node represented as described in clause 7.1.2.3.2.2 of [ITU-T G.9960] (BNDPL/GRP_ID field)
Rejection code		[6:4]	As described in Table 8-15

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

Field	Octet	Bits	Description
Security mode		[7]	Set to zero for insecure 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
Reserved		[7:2]	Reserved by ITU-T (Note)
Security	9 to 15	[55:0]	If Security mode is set to zero, this field shall be set to zero. 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 represent the REGID of the security controller
NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

8.6.1.1.4.3 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
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.			

8.6.1.1.4.4 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 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

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
Reserved	0	[7:0]	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.			

8.6.1.2 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.

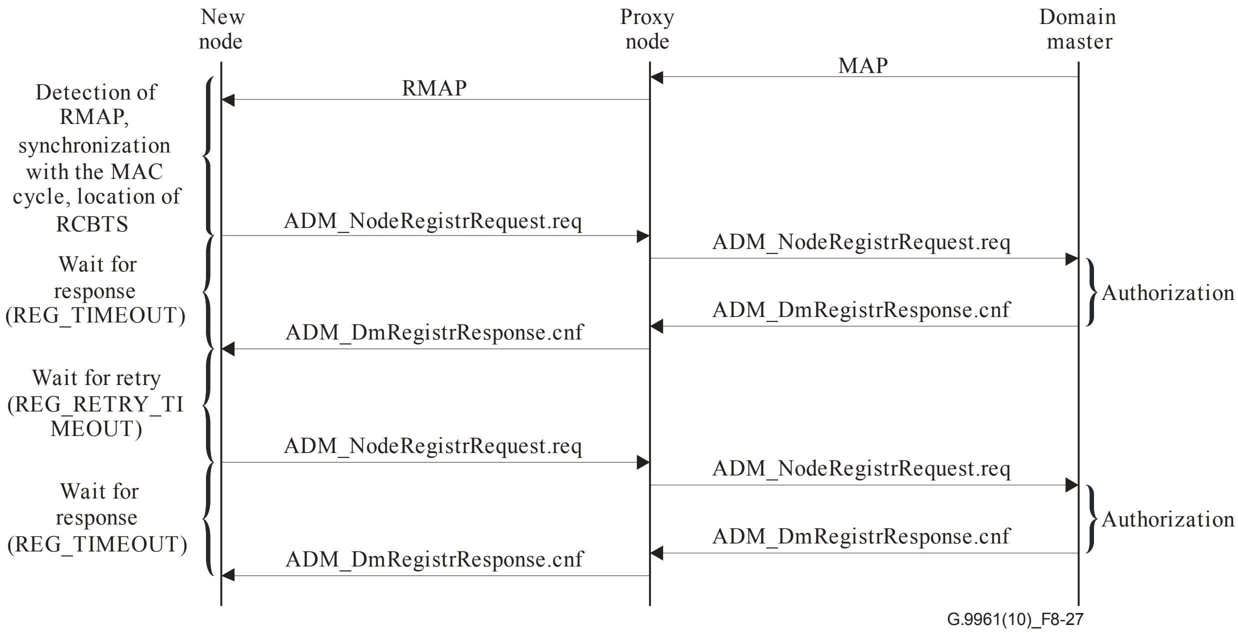


Figure 8-27 – Protocol diagram describing node registration via proxy

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 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 set the destination address of the LCDU carrying the ADM_NodeRegistrRequest.req message to the REGID of the domain master, and the DID of the PHY frame carrying ADM_NodeRegistrRequest.req shall be the DEVICE_ID of the registration proxy. 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 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). The destination address of the LCDU carrying ADM_DmRegistrResponse.cnf shall be the REGID of the proxy. The registration proxy node shall then unicast the received ADM_DmRegistrResponse.cnf message to the new node. The DID in the header of the PHY frame containing the ADM_DmRegistrResponse.cnf message shall be set to zero. The destination address of the LCDU carrying ADM_DmRegistrResponse.cnf from the proxy node to the registering node shall be the REGID of the registering node.

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.

8.6.2 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 for it. 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 entry of a packet at the APC of the node and the forwarding of the APDU to its destined 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 $t_i = t_0 + i \times$ interval. The actual CFTXOP time, t_i' shall be in the range t_i – jitter/2 $\leq t_i' \leq t_i$ + 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 $t_i = t_{(i-1)}$ + 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.

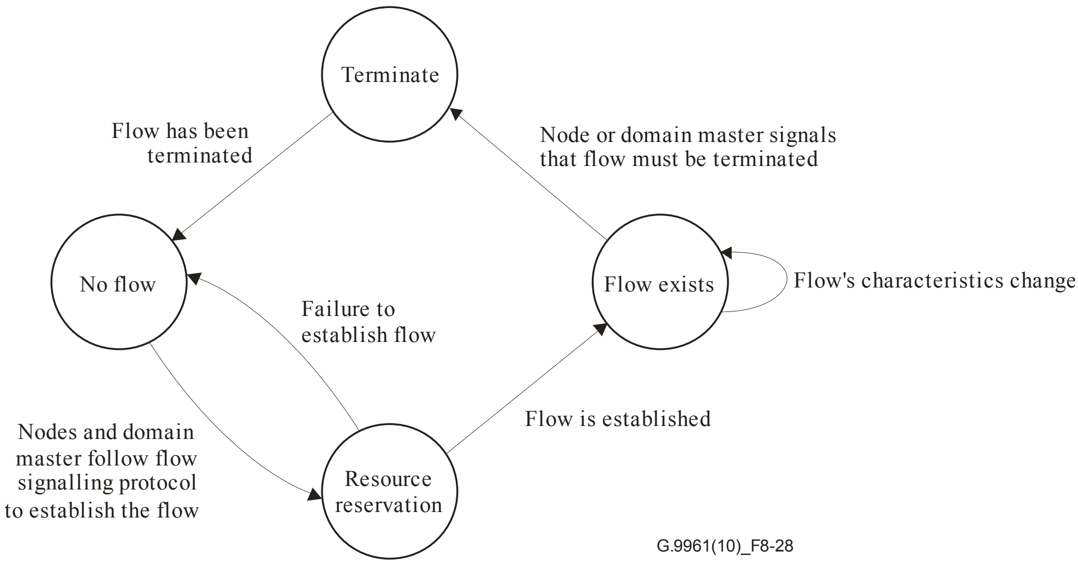


Figure 8-28 – Lifecycle of a flow

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 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.

- 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).

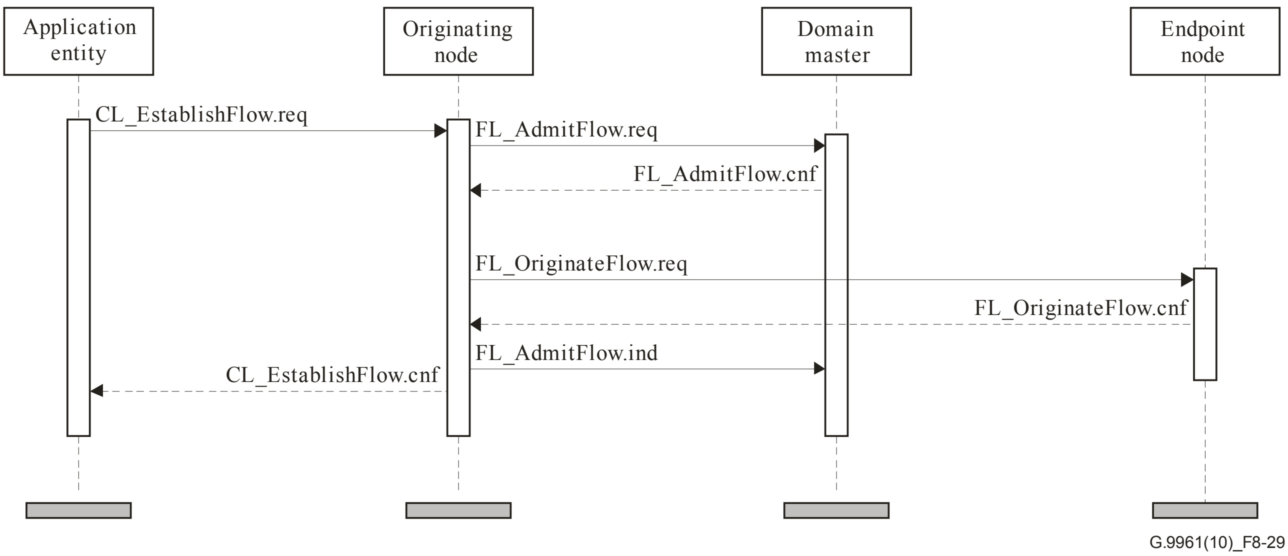


Figure 8-29 – Successful flow establishment

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).

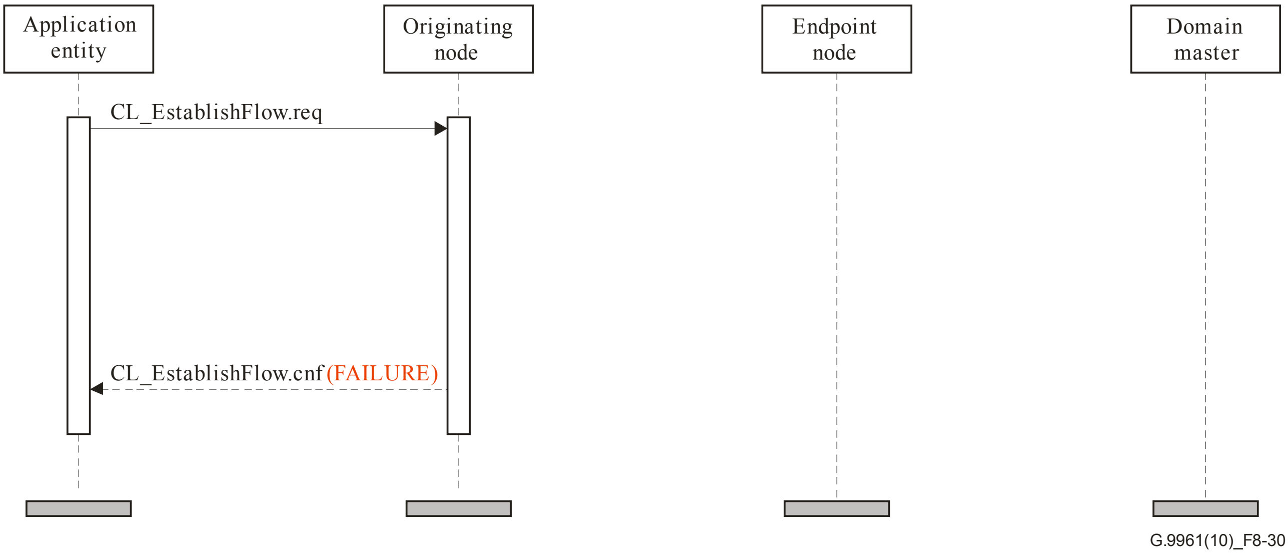


Figure 8-30 – Failure in flow establishment due to rejection by the originating node rejection

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

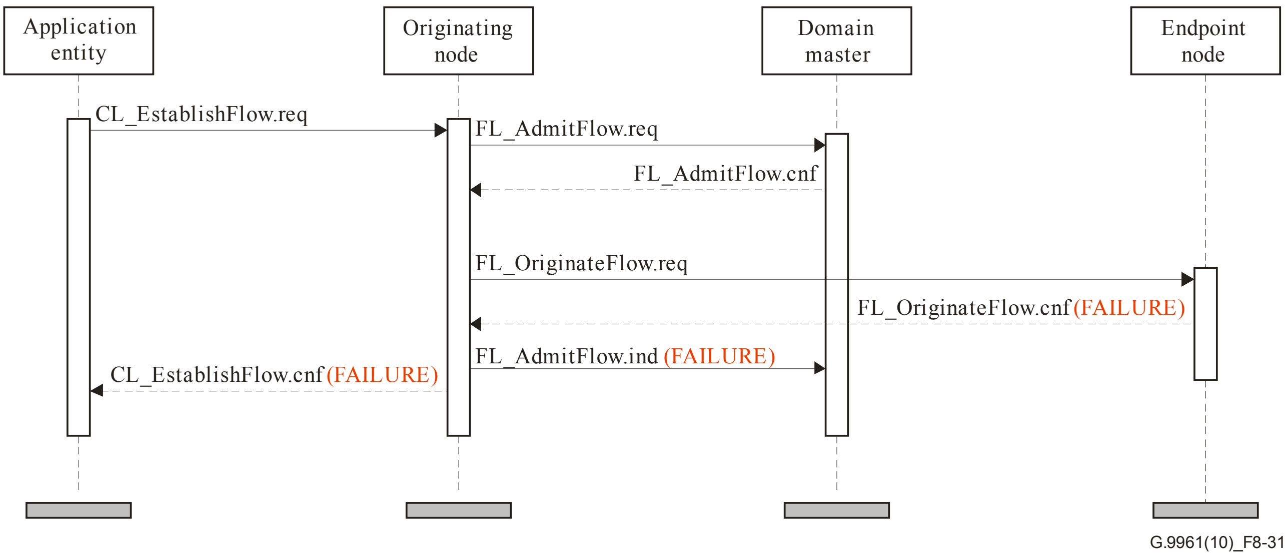


Figure 8-31 – Failure in flow establishment due to rejection by endpoint node

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

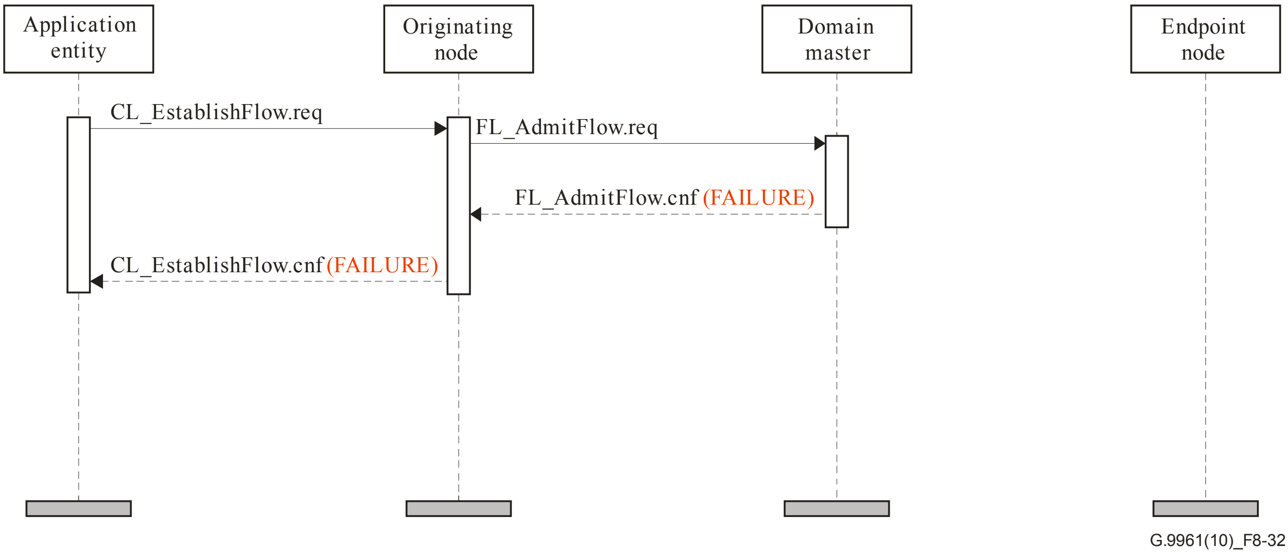


Figure 8-32 – Failure in flow establishment due to rejection by the domain master

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. The originating node shall send to the higher application entity the CL_EstablishFlow.cnf message with success code.

j) After the domain master receives the FL_AdmitFlow.ind it shall 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.

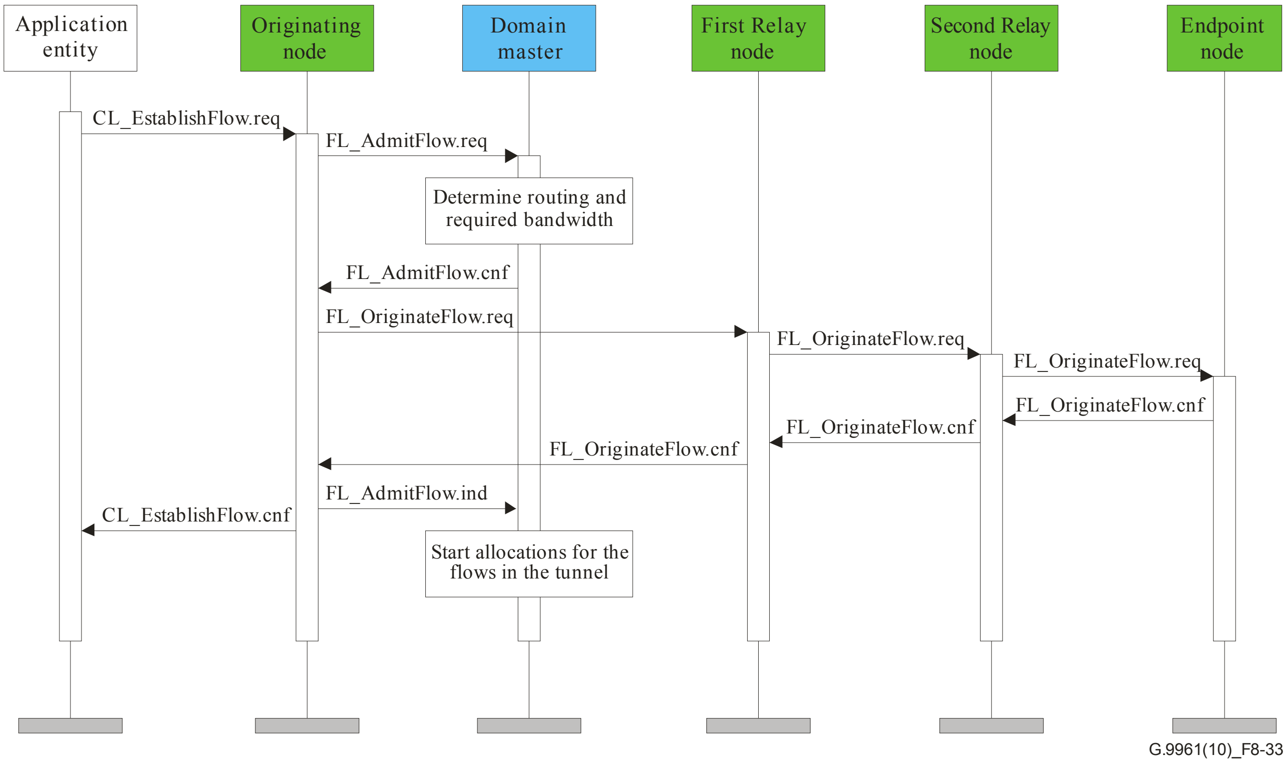


Figure 8-33 – Example of a successful tunnel establishment with two relay nodes

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.

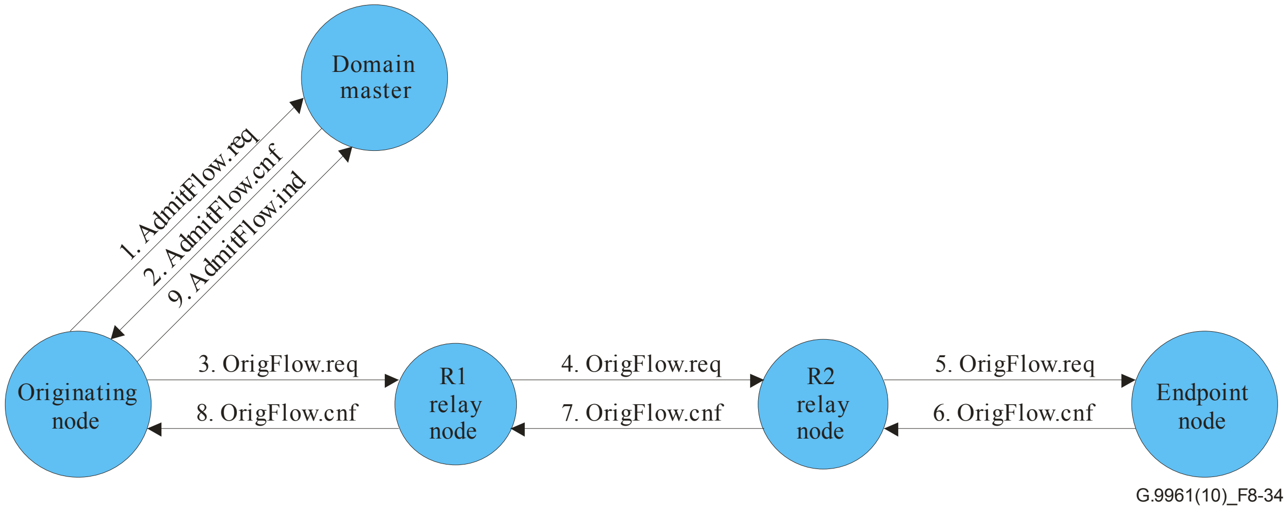


Figure 8-34 – Example of a successful tunnel establishment via two relay nodes

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.

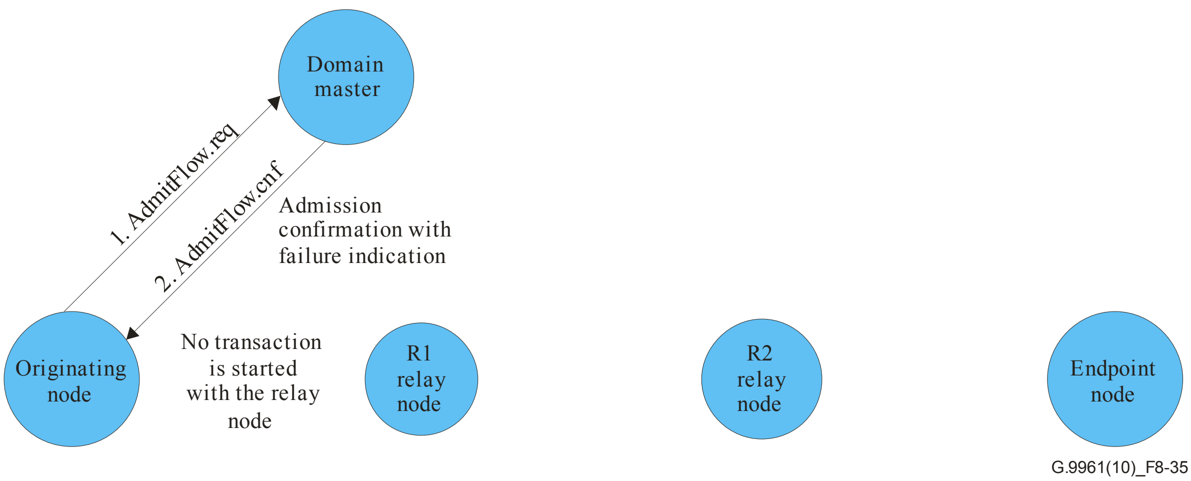


Figure 8-35 – Example of a tunnel establishment failure scenario due to the domain master rejecting the admission request

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.

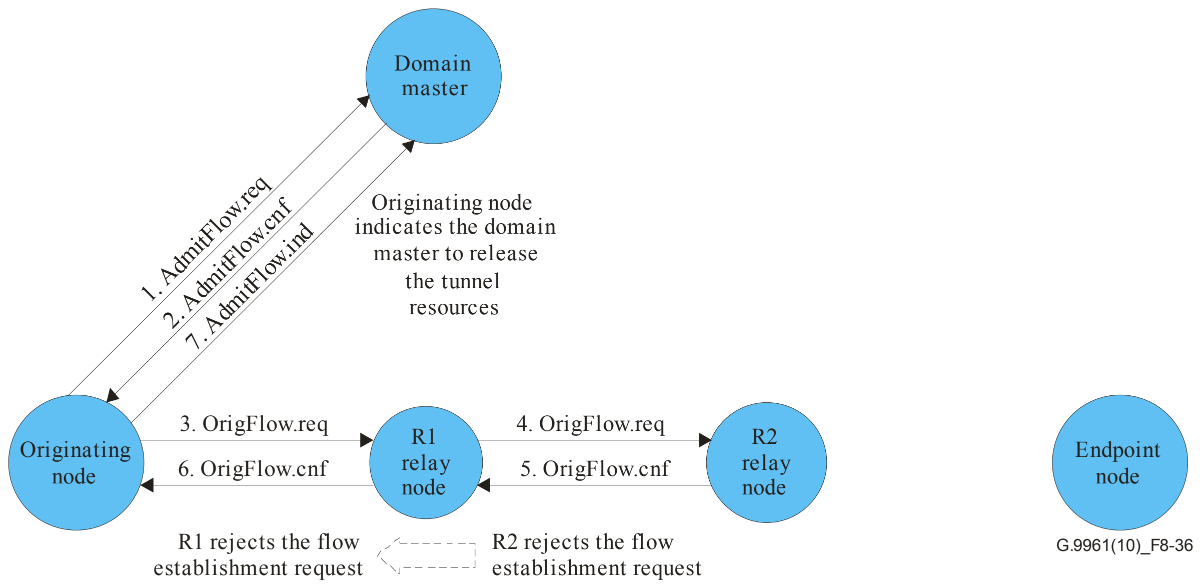


Figure 8-36 – Example of a tunnel establishment failure in the case where the second relay node, R2, rejects the flow establishment request

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 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 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.cnf 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 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

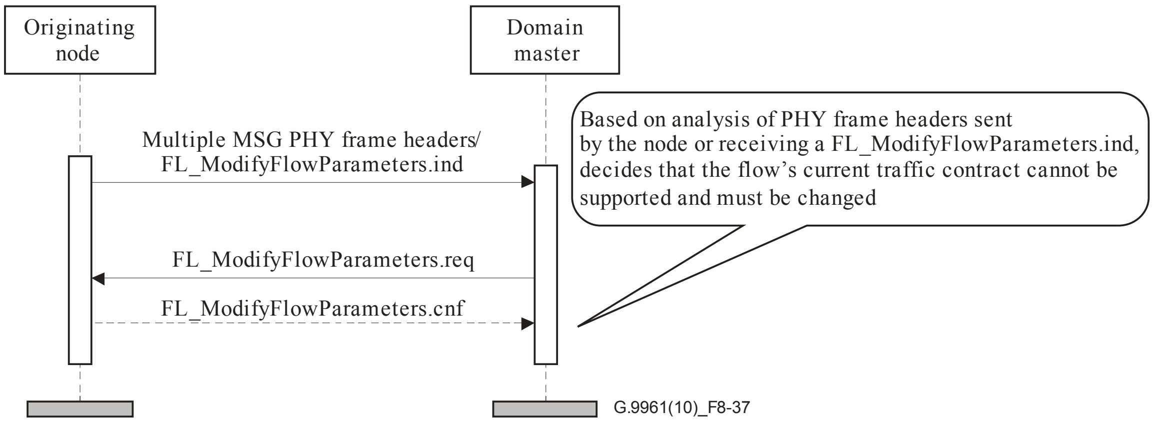


Figure 8-37 – Flow parameters modification example

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 periodical 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.

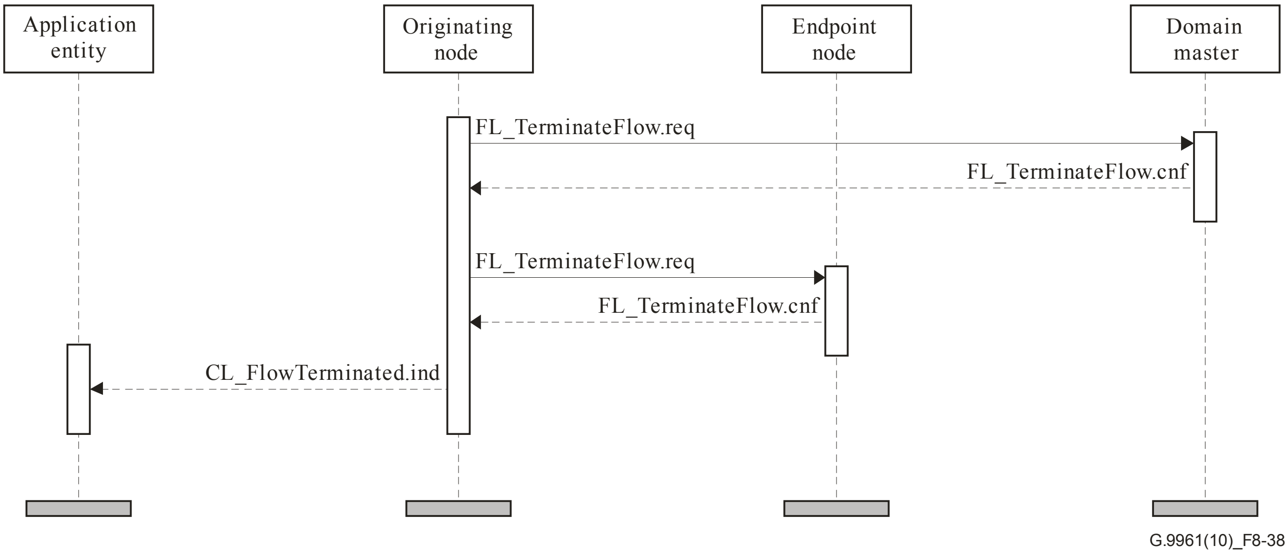


Figure 8-38 – MSC showing an example of how a flow can be terminated

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.

8.6.2.3 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	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 – Structure of the list of classifiers

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.
…	…	…	…
Classifier[k–1]	Variable	See Table 8-23	Last classifier in the list.

Table 8-23 – 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 Address 1: TOS 2: VLAN priority 3: VLAN TAG 4: Destination Port 5: Source port 6: IP address + destination port 7: IP address + source port 8: IP address + TOS 9: Generic classifier: offset, length, value 10-255: Reserved by ITU-T
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 it is a function of the Classifier_typ.

Table 8-24 – Format for the TSpec field

<table>
<tr><th>Field</th><th>Octet</th><th>Bits</th><th>Description</th></tr>
<tr><td>Length</td><td>0</td><td>[7:0]</td><td>The length of the TSpec sub-fields following this field expressed in number of octets in the range between 2 and 255.</td></tr>
<tr><td>TSpecBitMask</td><td>1 and 2</td><td>[15:0]</td><td>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:
<table>
<tr><th>Bit</th><th>TSpec attribute</th></tr>
<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>
</table>
</td></tr>
</table>

Table 8-24 – Format for the TSpec field

Field	Octet	Bits	Description
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 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.
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 an additional tuple (DeviceID, FlowID), where DeviceID corresponds 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 shall be as 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. • 05_{16} = Failure – error in TSpec passed in CL_EstablishFlow.req for the flow in the reverse direction • 06_{16} = Failure – insufficient capacity to start the flow in the reverse direction • 07_{16} – FF_{16} = Reserved (Note 1).
TSpecReject	3 and 4	[15:0]	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. 0: If bit 0 is set to one then traffic priority was rejected 1: If bit 1 is set to one then maximum information rate was rejected. 2: If bit 2 is set to one then maximum traffic burst was rejected. 3: If bit 3 is set to one then committed information rate was rejected. 4: If bit 4 is set to one then tolerated jitter was rejected. 5: If bit 5 is set to one then maximum latency was rejected. 6: If bit 6 is set to one then grant interval was rejected. 7: If bit 7 is set to one then polling interval was rejected. 8: If bit 8 is set to one then APDU size was rejected. 9-15: Reserved by ITU-T. (Note 1)
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).

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

Field	Octet	Bits	Description
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).
NOTE 1 – If StatusCode is lower than 2_{16}, then the TSpecReject field shall be ignored. NOTE 2 – If Bidirectional field is set to zero, these fields shall not appear in the message.			

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 reason code 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: • 00_{16} = Normal termination in response to CL_TerminateFlow.req. • 01_{16}-FF_{16} = 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 reason code 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: • 00_{16} = Normal termination (originating node has inferred that the flow has ended). • 01_{16} = Terminated at request of the domain master. • 02_{16}-FF_{16} = 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: • 00_{16} = Success. • 01_{16} = Failure – Maximum number of flows already started by the endpoint node. • 02_{16} = Failure – Error in TSpec passed in FL_OriginateFlow.req. • 03_{16} = Failure – Insufficient resources • 04_{16} = Failure – Failed to establish flow in reverse direction because maximum number of flows already started by the endpoint node. • 05_{16} = Failure – Error in TSpec passed in CL_EstablishFlow.req for the bidirectional flow other direction • 06_{16} = Failure – Insufficient capacity to start the flow in the reverse direction • 07_{16} – FF_{16} = 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.

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

Field	Octet	Bits	Description
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 01_{16} 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 symbol for each channel estimation window, based on the bit loading per symbol, the symbol time, the FEC rate, the number of repetitions, and the overhead according to the block size. 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 transmission 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 BitsPerSymbol. The list shall not exceed n=32 items.
EstimOverhead		[7:5]	Estimated DLL overhead in percentage represented as an unsigned integer minus 1 (Note). 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.
BitsPerSymbol	3 and 4	[15:0]	Number of data bits per symbol for the first channel estimation window.
⋮	⋮	⋮	⋮
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.
BitsPerSymbol	4n–1 to 4n	[15:0]	Number of data bits per symbol for the last channel estimation window.
NOTE – 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.			

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 01_{16}, the FL_AdmitFlow.cnf shall contain this field set to 01_{16} 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: • 00_{16} = Success. • 01_{16} = Failure – Maximum number of flows already started by originating node. • 02_{16} = Failure – Error in TSpec passed in FL_AdmitFlow.req. • 03_{16} = Failure – Insufficient capacity to admit the flow given the TSpec passed in FL_AdmitFlow.req. • 04_{16} = Failure – Failed to establish flow in reverse direction because maximum number of flows already started by the endpoint node. • 05_{16} = Failure – Error in TSpec passed in FL_AdmitFlow.req for the flow in the reverse direction. • 06_{16} = Failure – Insufficient capacity to support the flow for the reverse direction. • 07_{16} – FF_{16} = Reserved.
Bidirectional	3	[7:0]	Set to 01_{16} in case the admitted flow is a bidirectional flow.
DeviceID_B	4	[7:0]	DEVICE_ID of the originating node (Note 1)
TSpecReject	5 and 6	[15:0]	This field has applicable information only if StatusCode is set to a value that is greater than 2. This field specifies the TSpec attributes that have been rejected by the domain master. This field relates to the forward flow, or to the reverse flow in case the reverse flow was rejected. The field format is as specified in Table 8-25.
Tunnel	Variable	[7:0]	00_{16} – Direct flow admission request. 01_{16} – Tunnel flow admission request.
EndPoint	Variable	[7:0]	DEVICE_ID of the endpoint node.
RouteList	Variable	See Table 8-30	Routing list toward the destination endpoint (Note 2).
NOTE 1 – This field appear only if Bidirectional field is set to one. NOTE 2 – This field is included in the message only if the Tunnel field contains the value 1.			

8.6.2.3.10 Format of FL_AdmitFlow.ind

This message is sent by the originating node to the domain master to inform the domain master that the flow establishment has been completed. The message shall contain the established flow from the originating node towards the endpoint node and the reverse flow in case of bidirectional flow. In case a tunnel has been established, the message shall contain the list of established flows from the originating node towards the endpoint node. In case of bidirectional flow, the message shall also contain the list of established flows from the endpoint node towards the originating node.

The format of the MMPL of the FL_AdmitFlow.ind message shall be as shown in Table 8-35.

Table 8-35 – Format of the MMPL of the FL_AdmitFlow.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.
StatusCode	2	[7:0]	Status of the request to establish a flow: • 00_{16} = Success. • 01_{16} = Failure – Maximum number of flows exceeded. • 02_{16} = Failure – Insufficient capacity to support the flow • 03_{16} - FF_{16} = Reserved.
TSpecReject	3 and 4	[15:0]	This field has applicable information only if StatusCode is set to 2. This field specifies the TSpec attributes that have been rejected by the domain master. This field relates to one of the relay nodes in the route towards the endpoint node. The field format is as specified in Table 8-25.
Bidirectional	5	[7:0]	Set to 01_{16} in case the admitted flow is a bidirectional flow.
FlowID_B	6	[7:0]	FLOW_ID assigned by the endpoint node for the reverse flow (Note 1).
Tunnel	Variable	[7:0]	00_{16} – Direct flow admission request. 01_{16} – Tunnel flow admission request.
EndPoint	Variable	[7:0]	DEVICE_ID of the endpoint node
RouteList	Variable	(Note 2)	Routing list toward the destination endpoint (Note 3).
NOTE 1 – These fields appear only if Bidirectional field is set to one. NOTE 2 – If the Bidirectional field is set to 00_{16}, the RouteList is as defined in Table 8-36. If the Bidirectional field is set to 01_{16}, the RouteList is as defined in Table 8-37. NOTE 3 – This field appears only if the Tunnel field has a value of one.			

Table 8-36 – Format of RouteList for unidirectional flow

Field	Octet	Bits	Description
NumRelays	0	[7:0]	Number of relay nodes (n) in the RouteList.
FlowID	1	[7:0]	FLOW_ID between the originating node and the first relay node.
…	…	…	…
FlowID	n	[7:0]	FLOW_ID between the last relay node and the endpoint node.

Table 8-37 – Format of RouteList for bidirectional flow

Field	Octet	Bits	Description
NumRelays	0	[7:0]	Number of relay nodes (n) in the RouteList.
FlowID	1	[7:0]	FLOW_ID between the originating node and the first relay node.
FlowID_B	2	[7:0]	FLOW_ID between the first relay node and the originating node as the reverse bidirectional flow.
….			
FlowID	2n–1	[7:0]	FLOW_ID between the last relay node and the endpoint node.
FlowID_B	2n	[7:0]	FLOW_ID between the endpoint node and the last relay node as the reverse bidirectional flow.

8.6.2.3.11 Format of FL_ModifyFlowParameters.req

This message is sent by the domain master to the originating node. This message allows the domain master to alter the traffic contract if necessary; for example, if channel conditions warrant a larger or smaller allocation for a CFTXOP. The message contains the flow's identity and the domain master's proposed TSpec for the flow. In case of bidirectional flow the modification can refer only to one direction of the flow or for both directions. In case of bidirectional flow this message shall be sent to both of the nodes.

The format of the MMPL of the FL_ModifyFlowParameters.req message shall be as shown in Table 8-38.

Table 8-38 – Format of the MMPL of the FL_ModifyFlowParameters.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.
ProposedTSpec	2 to (N+1)	[8*N–1:0]	Revised traffic specification for this flow proposed by the domain master (see Table 8-24), based on the original TSpec.
Bidirectional	N+3	[7:0]	When this field contains one, it specifies that the flow is a bidirectional flow. When it is set to zero, it specifies a unidirectional flow.
FlowID_B	N+4	[7:0]	FLOW_ID assigned by the endpoint node for the reverse direction (Note).
ProposedTSpec_B	N+5 to N+5+M	[8*M–1]	Revised traffic specification for this flow proposed by the domain master based on the original TSpec for the flow in the reverse direction (Note).
NOTE – These fields exist only if Bidirectional field is set to 01_{16}.			

8.6.2.3.12 Format of FL_ModifyFlowParameters.cnf

This message is sent by the originating node and by the endpoint node (in case of a bidirectional flow) to the domain master in response to FL_ModifyFlowParameters.req. The message contains the flow's identity and the status returned by the node for the previous request.

The format of the MMPL of the FL_ModifyFlowParameters.cnf message shall be as shown in Table 8-39.

Table 8-39 – Format of the MMPL of the FL_ModifyFlowParameters.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 modify a flow: • 00_{16} = Success. • 01_{16} = Failure – Originating node does not support this service. • 02_{16} = Failure – Error in new TSpec supplied in FL_ModifyFlowParameters.req. • 03_{16} = Failure due to other reason. • 04_{16} - FF_{16} = Reserved.

8.6.2.3.13 Format of FL_TerminateFlow.req

The message is sent by the originating node to the endpoint and to the domain master when the originating node has determined that the flow should be ended. In case the ended flow is a bidirectional flow, the endpoint node shall also terminate its own flow. This message may also be sent by the domain master to the originating node if the flow must be terminated (e.g., this can occur if the channel becomes over-subscribed). This message contains a reason code for the request plus the flow's identity. In case of a bidirectional flow, when the endpoint node has determined that the flow should be ended, it shall send this message to the originating node and to the domain master. In this case, the DeviceID and the FlowID shall be the DEVICE_ID and FLOW_ID of the endpoint node. The originating node, after receiving this message from the endpoint node, shall also terminate its bidirectional flow.

The format of the MMPL of the FL_TerminateFlow.req message shall be as shown in Table 8-40.

Table 8-40 – Format of the MMPL of the FL_TerminateFlow.req message

Field	Octet	Bits	Description
DeviceID	0	[7:0]	DEVICE_ID of the originating node or of the endpoint node in case of bidirectional flow.
FlowID	1	[7:0]	FLOW_ID assigned by the originating node or of the endpoint node in case of bidirectional flow.
ReasonCode	2	[7:0]	Reason for which the flow is being terminated: • 00_{16} = Normal termination by originating node or by endpoint node in case of bidirectional flow. • 01_{16} = Termination by domain master due to over-subscription of the channel. • 03_{16} - FF_{16} = Reserved.

8.6.2.3.14 Format of FL_TerminateFlow.cnf

This message is sent by either the domain master or the endpoint node in response to FL_TerminateFlow.req. This message contains the flow's identity. In case of bidirectional flow, this message shall be sent by the originating node in the case where the endpoint node ended the bidirectional flow.

The format of the MMPL of the FL_TerminateFlow.cnf message shall be as shown in Table 8-41.

Table 8-41 – Format of the MMPL of the FL_TerminateFlow.cnf message

Field	Octet	Bits	Description
DeviceID	0	[7:0]	DEVICE_ID of the originating node or of the endpoint node in case of bidirectional flow.
FlowID	1	[7:0]	FLOW_ID assigned by the originating node or of the endpoint node in case of bidirectional flow.

8.6.2.3.15 Format of FL_ModifyFlowParameters.ind

This message is sent by the originating node of a flow, or by the endpoint node in case of bidirectional flow, to the domain master when the originating node or the endpoint node needs to update the domain master on changes in the bandwidth requirements of the flow. Nodes that are hidden from the domain master shall use this message in addition to the BRURQ field in the PHY-frame header of the flow's frames.

The format of the MMPL of the FL_ModifyFlowParameters.ind message shall be as shown in Table 8-42.

Table 8-42 – Format of the MMPL of the FL_ModifyFlowParameters.ind message

Field	Octet	Bits	Description
DeviceID	0	[7:0]	DEVICE_ID of the originating node or of the endpoint node in case of bidirectional flow.
FlowID	1	[7:0]	FLOW_ID assigned by the originating node or of the endpoint node in case of bidirectional flow.
BRURQ	2 and 3	[15:0]	See clause 7.1.2.3.2.2.19 of [ITU-T G.9960]

8.6.2.3.16 Format of FL_ModifyFlowAllocations.req

This message is sent by the node that originated the flow, or by the endpoint node in case of bidirectional flow, to the domain master, to request timing adjustment of the CFTXOP allocations. The format of the MMPL of the FL_ModifyFlowAllocations.req message shall be as shown in Table 8-43.

Table 8-43 – Format of the MMPL of the FL_ModifyFlowAllocations.req message

Field	Octet	Bits	Description
DeviceID	0	[7:0]	DEVICE_ID of the originating node or of the endpoint node in case of bidirectional flow.
FLOW_ID	1	[7:0]	The flow identifier to which the domain master allocates CFTXOP allocations.
CFTXOP allocation_Adjustment	2 and 3	[15:0]	The requested allocation time adjustment represented as a signed integer (using 2's complement coding) in microseconds relative to the last CFTXOP allocation.

8.6.2.3.17 Format of FL_ModifyFlowAllocations.cnf

This message is sent by the domain master to the node requested adjustment of CFTXOP allocation in response to FL_ModifyFlowAllocations.req. This message confirms that the request was received by the domain master. The format of the MMPL of the FL_ModifyFlowAllocations.cnf message shall be as shown in Table 8-44.

Table 8-44 – Format of the MMPL of the FL_ModifyFlowAllocations.cnf message

Field	Octet	Bits	Description
DeviceID	0	[7:0]	DEVICE_ID of the originating node.
FLOW_ID	1	[7:0]	The flow identifier to which the domain master allocates CFTXOP allocations.
Status	2	[7:0]	00_{16}: The request is received. 01_{16} - FF_{16}: reserved by ITU-T.

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 ± 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 ± 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.

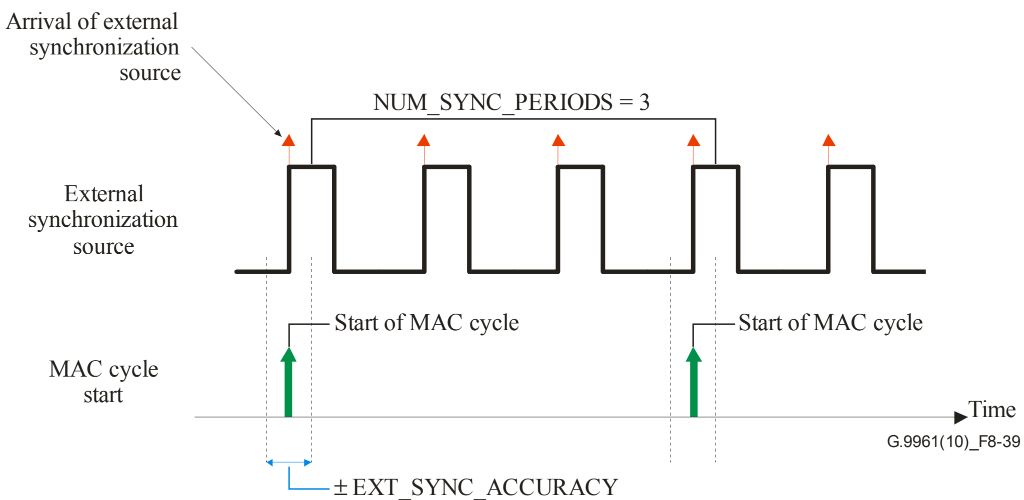


Figure 8-39 – Example of synchronization of MAC cycle to a periodic external source

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, vendor discretionary, angular offset Δ (which may be zero) from the AC cycle zero-crossing point with a maximum tolerance of EXT_SYNC_ACCURACY = 100 µs.

NOTE – Variances at power generators and through the power distribution system cause the actual power-line frequency supplied to a node to jitter, compared to the nominal 50 or 60 Hz line frequency. 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).

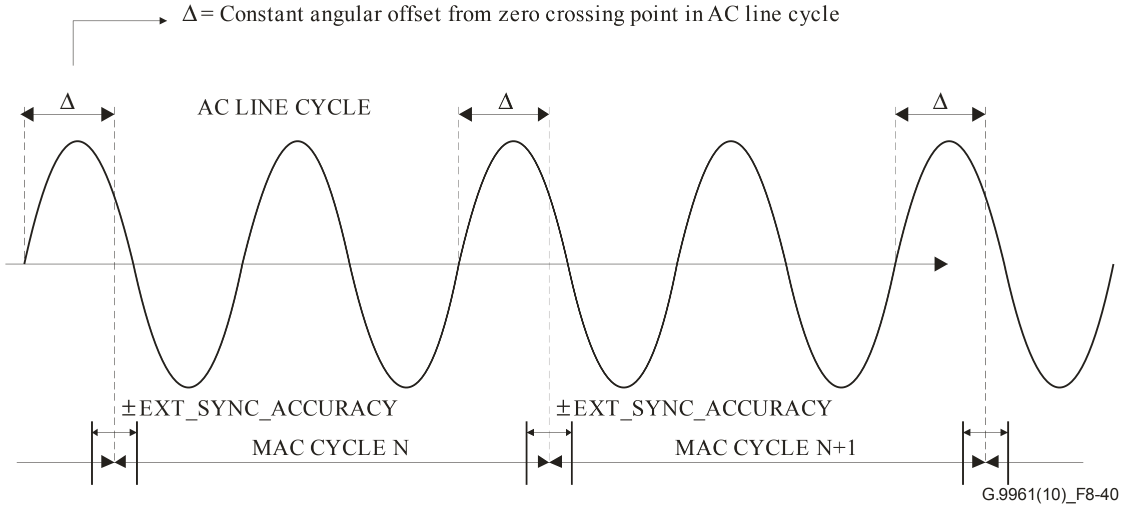


Figure 8-40 – Relationship between the AC line cycle and the MAC cycle

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 Table 8-82) 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 TM_DMBackup.ind messages associated with a loss of one or more nodes from the visibility list of any endpoint node. 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.

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 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.

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, and the sequence numbers of those messages that resulted in the updating of the routing table (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.

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.

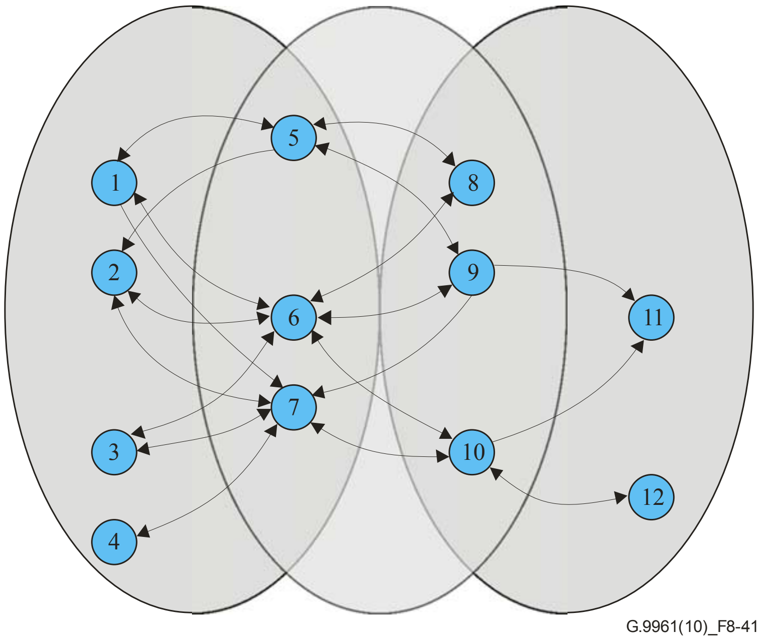


Figure 8-41 – Example 12-node network showing topology coverage ellipses and optimal paths

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

From (i) \ To (j)	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 relaying the topology and routing change messages to all the nodes in the domain. A multipoint relay distribution tree (MRDT) is a spanning tree consisting of all the MPRs selected by the domain master. The routing tables shall be distributed by the domain master using the MRDT. The domain master shall send TM_DomainRoutingChange.ind to all nodes in its visibility list that are in the updated MRDT (i.e., those with hop count = 0). Each MPR that receives this message, shall update its MRDT using the content of the message and further relay the message to all the nodes of the updated MRDT in its 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 and TM_DomainRoutingChange.ind messages shall be sent using connections with acknowledgements. If a node that generates a TM_NodeTopologyChange.ind does not receive a TM_DomainRoutingChange.ind message that includes an indication that this message was already received by the domain master, it shall retry the procedure described in clause 8.6.4.2.1.1, after incrementing the sequence number of the message. After several attempts, if the node cannot get the indication on this message being received by the domain master, it shall stop the procedure and indicate failure to its user interface. The number of attempts and time interval between attempts are vendor discretionary.

A node relaying a TM_DomainRoutingChange.ind message as described in this clause, shall first check if it has already relayed that message to all its destinations, using the sequence 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 NxN 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 whenever any of the following events occurs:

- after the node successfully joined the domain (in an insecure domain, after node successfully registered by the domain master, in a secure domain, after node successfully authenticated by the SC);
- in accordance with the topology update interval (see Table 8-82);
- 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 information about the AE 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.

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

The time interval before the next transmission of the TM_NodeTopologyChange.ind message shall be selected by a node uniformly within the interval ($T_{MIN\_DISCOVER}$, $T_{MAX\_DISCOVER}$) where $T_{MIN\_DISCOVER}$ and $T_{MAX\_DISCOVER}$ are determined by the MAP (see Tables 8-82 and 8-83).

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 (bandplan, capability to serve as a domain master, as a security controller, or as a relay);
- list of the detected neighbouring domain DODs. This component shall be included in the message if there was a change in the list of detected neighbouring domains;
- 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 message TM_ReturnDomainRouting.req to the domain master. If a node has received a message TM_DomainRoutingChange.ind 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 for an update of the routing information by sending the 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 that are in its visibility list using the reserved MAC address 01-19-A7-52-76-96 as the DA. If there are no 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 clause to relay the message further, as if the message were 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 (use domain master's DA), and transmit it using unicast. 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 shall 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 to all its destinations, using the sequence 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 02_{16}.

All parameters provided by a report with a particular sequence number shall be associated with the same collection time or time interval. Parameters related to different times or time intervals shall be sent as separate reports, with different sequence numbers.

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).

In CRTM mode, when the message is sent directly to the DM, the destination MAC address of TM_NodeTopologyChange.ind shall be the REGID of the DM. When the message is sent via the MPRs, the destination MAC address shall be the group address, 01-19-A7-52-76-96.

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	[7:0]	Shall be set to: – 00_{16} if report includes all the topology parameters available to the node (full report) – 01_{16} if the report includes any fraction of topology parameters available to the node – 02_{16} if the report includes only parameters that changed relatively to the last report Other values are for further study.
Reserved	2	[7:0]	Reserved by ITU-T (Note 1)
Reserved	3 and 4	[15:0]	Reserved by ITU-T (Note 1)
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]	Size of the first neighbouring domain Info field in bytes represented as an unsigned integer. 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]	Size of the last neighbouring domain Info field in bytes represented as an unsigned integer. 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).

Other node-related topology parameters are for further study.

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 x 1.28 μs; the value of n is an unsigned integer in the range between 4 and 96 (Note 1).
NodeVersion	4	[7:0]	0 – Node supports version 0 of ITU-T G.9960 and ITU-T G.9961. All other values of this field are reserved by ITU-T for indicating support for future versions of the Recommendation (Note 2).
AAT_Size	5 and 6	[15:0]	Number (k) of local AAT entries associated with the reporting node (Note 3).
AAT [0]	7 to 13	[47:0]	The first entry in the AAT. It contains the first local MAC address (Note 4).
…	…	…	….
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 5).
RemAAT [0]	Variable	[47:0]	First entry in the RemAAT. It contains the first MAC address that has been removed from the node AAT.
…	…	…	….
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 6).
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 7)	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.

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

Field	Octet	Bits	Description
Visibility_List	Variable	[31:0]	List of M fields, 4 octets each, describing a single detected node, formatted as described in Table 8-48
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 – A node indicating support for a certain version of this Recommendation shall also support all earlier versions of this Recommendation. NOTE 3 – 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 4 – The first MAC address shall be the REGID of the reporting node. NOTE 5 – 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 6 – 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 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 7 – 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 used by the node as described in Table 7-11 of [ITU-T G.9960] (BNDPL/GRP_ID field).

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

Field	Octet	Bits	Description
	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
DEVICE_ID	0	[7:0]	DEVICE_ID of a node that the reporting node detected.
BitRate	1 to 3	[23:0]	Bits [11:0] indicate the PHY data rate from the reporting node to the detected node; Bits [23:12] indicate the PHY data rate from the detected node to the reporting node. Both data rates shall be represented as 12-bit unsigned integers, in steps of 0.5 Mbit/s (Note).
NOTE – If the data rate with the particular detected node is not available, the value of this parameter shall be set to FFF_{16}; the value shall be set to zero if the detected data rate is less than 0.5 Mbit/s.			

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]	Bit 2 – Set to one if visibility information is required Bit 3 – Set to one if AAT information is required, set to zero if AAT information is not required (Note). 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. Other bits are reserved by ITU-T and shall be set to zero.
NOTE – Setting of both bits 2 and 3 to zero is not allowed.			

8.6.4.3.3 Format of TM_NodeTopologyChange.cnf

Message TM_NodeTopologyChange.cnf is a management message that shall be sent by a node in response to the message TM_NodeTopologyChange.req from the domain master. The MMPL of the TM_NodeTopologyChange.cnf message shall be the same as the MMPL of the TM_NodeTopologyChange.ind message (Table 8-46).

8.6.4.3.4 Format of TM_DMBackup.ind

Message TM_DMBackup.ind is a management message that shall be sent by the domain master to the backup domain master to inform about change in a node's topology information. The MMPL of the TM_DMBackup.ind message shall be the same as the MMPL of the TM_NodeTopologyChange.ind message (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 nodes in the domain.

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.

Records in the TM_DomainRoutingChange.ind message shall only contain information for non-zero elements from the routing table. The message includes the number of node records in the message and a list of node records. Each node record includes the local AAT of the node and a list of tuples which represents the routing to other nodes in the domain (see clause 8.6.4.1.1.1, Transmission of routing table). In addition to the node records, the message includes a list of nodes that have resigned from the domain since the last update (the last transmitted TM_DomainRoutingChange.ind message).

The TM_DomainRoutingChange.ind message shall also contain the list of nodes that have been assigned to transmit RMAPs in subsequent MAC cycles.

The format of the MMPL of the TM_DomainRoutingChange.ind message shall be as shown in Table 8-50. Both full and fractional reports shall use the format defined in Table 8-50.

Table 8-50 – Format of the MMPL of the TM_DomainRoutingChange.ind message

Field	Octet	Bits	Description
NumTmInd	0	[7:0]	This value indicates the number (m) of node topology change messages received by the node after the previous transmission of domain routing change message.
DEVICE_ID[0]	1	[7:0]	DEVICE_ID of the first node whose topology information was used in generating this routing change message.
SeqNumber[0]	2 and 3	[15:0]	Sequence number of the first node that sent the topology change message that was used in generating this routing change message.
…	…	…	
DEVICE_ID[m–1]	1+3*(m–1)	[7:0]	DEVICE_ID of the m-th node whose topology information was used in generating this routing change message.
SeqNumber[m–1]	2+3*(m–1) and 3+3*(m–1)	[15:0]	Sequence number of the m-th node that sent the topology change message that was used in generating this routing change.
NumNodesRecs	Variable	[7:0]	Number of source node records (n) in the message (Note 1).
NodeRec[0]_ID	Variable	[7:0]	DEVICE_ID of the first source node in the list.
NodeRec[0]_Size	Variable	[15:0]	Size of the first record in bytes represented as an unsigned integer (Note 3).
NodeRec[0]_Info	Variable	See Table 8-51	First record information field, with a format as defined in Table 8-51.
…	…	…	
NodeRec[n–1]_ID	Variable	[7:0]	DEVICE_ID of the last source node in the list.
NodeRec[n–1]_Size	Variable	[7:0]	Size of the last record in bytes represented as an unsigned integer.
NodeRec[n–1]_Info	Variable	See Table 8-51	Last record information field, with a format as defined in Table 8-51.
NumResignNodes	Variable	[7:0]	Number of resigned nodes (m) in the resigned node list (Note 2).
ResignedNodes[0]	Variable	[7:0]	DEVICE_ID of the first resigned node in the list.
…	…	…	…
ResignedNodes[m–1]	Variable	[7:0]	DEVICE_ID of the last resigned node in the list.
NOTE 1 – The number of node records in the list includes the domain master. NOTE 2 – If there are no nodes that resigned from the domain since the last update, this field shall be set to zero and the list of resigned nodes shall have no entries. NOTE 3 – All NodeRec[i]_Size fields shall be > 0.			

Table 8-51 – Format of NodeRec[i]_Info

Field	Octet	Bits	Description
NumDestNodes	0	[7:0]	Number of destination hidden node pairs (n) of the unicast routing table. Each pair contains the DEVICE_ID of the destination hidden node and the DEVICE_ID of the relay node toward the specified destination hidden node.
DestNodeID[0]	1	[7:0]	DEVICE_ID of the first destination hidden node.
RelNodeID[0]	2	[7:0]	DEVICE_ID of the relay node toward the first specified destination hidden node.
…	…	…	…
DestNodeID[n–1]	2×(n–1) + 1	[7:0]	DEVICE_ID of the last destination hidden node.
RelNodeID[n–1]	2×(n–1) + 2	[7:0]	DEVICE_ID of the relay node toward the last specified destination hidden node.
NumBRTEntries	Variable	[7:0]	Number of entries (b) of the BRT of the node
BRTEntry[0]	Variable	Table 8-52	Content of the first entry of the BRT as described in Table 8-52.
…	…	…	…
BRTEntry[b–1]	Variable		Content of the last entry of the BRT as described in Table 8-52.
NodeAIFG	Variable	[7:0]	The value of T_{AIFG} supported by the node, represented as n × 1.28 µs; the value of n is an unsigned integer in the range between 4 and 96.
IsMpr	Variable	[0]	Set to one if node is an MPR, otherwise set to zero.
HopCount	Variable	[7:1]	Set to the (number of hops – 1) that the node is from the domain master. It is set to zero, if the node has a direct link to the domain master.
AAT_Size	Variable	[15:0]	Number (k) of local AAT entries associated with the reporting node (Note 1).
AAT [0]	Variable	[47:0]	The first entry in the AAT. It contains the first local MAC address.
…	…	…	….
AAT [k–1]	Variable	[47:0]	The last entry in the AAT. 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 2).
RemAAT [0]	Variable	[47:0]	First entry in the RemAAT. It contains the first MAC address that has been removed from the node AAT.
…	…	…	….
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.

Table 8-51 – Format of NodeRec[i]_Info

Field	Octet	Bits	Description
NewAAT_Size	Variable	[15:0]	Number of AAT entries (q) that were added to the node AAT (Note 3).
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.
NOTE 1 – 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. NOTE 2 – 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 3 – 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 new entries to the local AAT since the last transmitted report for that node. This field shall be set to zero if AAT_Size field is non-zero.			

Table 8-52 – Format of BRTEntry[i]

Field	Octet	Bits	Description
NumOrigNodes	0	[7:0]	Number of DEVICE_IDs (m) included in the list of originating nodes (see clause 8.6.4.1.1.3). This field shall be set to FF_{16} to indicate that the list of originating nodes contains all nodes in the domain, except the node itself. In this case, the OrigNode fields shall not be present (Note).
OrigNode[0]	1	[7:0]	DEVICE_ID of the first node in the list of originating nodes.
…	…	…	…
OrigNode[m–1]	Variable	[7:0]	DEVICE_ID of the last node in the list of originating nodes.
RootPath	Variable	[7:0]	DEVICE_ID of the root path of this node for LLC frames originated by all nodes in the list of originating nodes (see clause 8.6.4.1.1.3) (Note).
NumBranchNodes	Variable	[7:0]	Number of DEVICE_IDs (p) included in the branch path of this node for LLC frames originated by all nodes in the list of originating nodes (see clause 8.6.4.1.1.3). This field shall be set to FF_{16} to indicate that the branch path of this node includes the rest of nodes in the domain. In this case, the BranchNode fields shall not be present (Note).

Table 8-52 – Format of BRTEntry[i]

Field	Octet	Bits	Description
BranchNode[0]	Variable	[7:0]	DEVICE_ID of the first node in the branch path.
…			
BranchNode[p–1]	Variable	[7:0]	DEVICE_ID of the last node in the branch path.
NOTE – NumOrigNodes shall be set to 00_{16} to describe the entry of the BRT where this node is the originating node of broadcast LLC frames. When NumOrigNodes is equal to 00_{16}, the RootPath field shall be set to the DEVICE_ID of this node and the branch path shall contain the DEVICE_IDs to the nodes to which this node shall send broadcast LLC frames generated by itself.			

8.6.4.3.6 Format of TM_ReturnDomainRouting.req

TM_ReturnDomainRouting.req is a management message that shall be unicast by a node to the domain master when the node needs to refresh its copy of the domain's routing information. The MMPL of the TM_ReturnDomainRouting.req message shall be as shown in Table 8-53. By appropriate settings in the ReqRep field, a node may request only node records, or parts of those, or only information on neighbouring domains.

Table 8-53 – Format of the MMPL of the TM_ReturnDomainRouting.req message

Field	Octet	Bits	Description
ReqRep	0	[7:0]	Bit 0 – Set to one if node AAT is required. Bit 1 – Set to one if routing information is required. Bit 2 – Set to one if complete AAT is required. If set to zero, only changes since last report are required. Bit 3 – Set to one if information of specific nodes is required. Bit 4 – Set to one if unicast routing table is requested. Bit 5 – Set to one if broadcast routing table is requested. Other bits are reserved by ITU-T and shall be set to zero.
NumReqNodes	1	[7:0]	Number (n) of nodes for which information is requested. This field exists in the message only when Bit 3 in ReqRep field is set to one.
ReqNodes[0]	2	[7:0]	DEVICE_ID of the first node in the required nodes list for which information is required. This entry is present only if Bit 3 in ReqRep field is set to one.
….	..	…	…
ReqNodes[n–1]	n+1	[7:0]	DEVICE_ID of the last node in the required nodes list for which information is required. This entry is present only if Bit 3 in ReqRep is set to one.

8.6.4.3.7 Format of TM_ReturnDomainRouting.cnf

Message TM_ReturnDomainRouting.cnf is a management message that shall be sent by the domain master to a node in response to a message of type TM_ReturnDomainRouting.req. The format of the MMPL of the TM_ReturnDomainRouting.cnf message shall be as shown in Table 8-50, the same as the MMPL of the TM_DomainRoutingChange.ind message with the information included according to the specified settings in the TM_ReturnDomainRouting.req message that has been sent.

8.6.5 Backup domain master

Each domain master-capable node shall be able to act as a backup domain master. The role of a backup domain master shall be assigned to a domain master-capable node by the acting domain master. Only a single node acting as a backup domain master or none may be assigned at each time.

Assignment of backup domain master is optional and the considerations for assignment are up to the implementer.

The backup domain master shall take the role of the domain master only in case a failure of the domain master was detected as described in clause 8.6.5.3.

If the backup domain master option is used, the assignment and the release of a backup domain master shall comply with the rules as specified in clause 8.6.5.1.

8.6.5.1 Backup domain master assignment and release

The domain master shall select a node of its domain to be a backup using the following criteria:

– the node is domain master capable;

– the node has the highest in the domain rank to become a domain master.

To assign the selected node to be a backup, the domain master shall send to this node a DM_BackupAssign.req message which indicates the type of request (voluntary or forced) and includes the domain information necessary for the backup node to continue management of the domain after the failure of the domain master with minimum interruption.

The selected node shall respond to the domain master within 100 ms by the DM_BackupAssign.cnf message, indicating that the node either confirms the role of the backup or rejects it with indication of the rejection code. If a node rejects the role per a voluntary request, the domain master may either select another node or force the backup operation by sending to the node a DM_BackupAssign.req message with "Forced backup" flag set.

NOTE 1 – If for some reason the domain master does not receive the response from the node within 300 ms after the DM_BackupAssign.req message was sent, it may decide to repeat the request or select another node as a candidate to be its backup.

After a backup node is assigned, the domain master shall announce the ID of the assigned backup in the Auxiliary Info field of the first MAP after the assignment of the backup (see clause 8.8.5.7) and further repeat the announcement on a periodical basis with the period determined by the domain master.

The domain master shall periodically update the relevant domain management information communicated to the backup node upon its assignment by sending to the backup a DM_BackupData.ind message; the frequency of the updates is left to the discretion of the domain master.

The domain master may release the node from the role of a backup domain master by sending a DM_BackupRelease.req message. The node shall acknowledge the request within 100 ms by sending a DM_BackupRelease.cnf message. If the domain master does not receive a DM_BackupRelease.cnf message within 300 ms after it has sent the DM_BackupRelease.req message, it may repeat the request. The node shall terminate its role as a backup if it does not receive another DM_BackupRelease.req message within 1 s after the last DM_BackupRelease.cnf message has been sent. The domain master is not allowed to assign a new backup until the existing backup is released.

NOTE 2 – Robustness and efficiency of the described protocol will increase if the domain master requests Imm-ACK on the DM_BackupAssign.req, DM_BackupData.ind, and DM_BackupRelease.req messages.

8.6.5.2 Message formats for assignment and release of the backup

The format of the MMPL of the DM_BackupAssign.req, DM_BackupAssign.cnf and DM_BackupData.ind messages used for the domain master backup assignment and release procedures described in clause 8.6.5.1 shall be as shown in Tables 8-54, 8-55 and 8-56, respectively.

Table 8-54 – Format of the MMPL of the DM_BackupAssign.req message

Field	Octet	Bits	Description
Forced backup	0	[0]	Set to one for forced assignment, set to zero for voluntary assignment.
Reserved		[7:1]	Reserved by ITU-T (Note).
Backup data	0 to (N–1)	[(8*N)–1:0]	Data related to domain management provided by the acting domain master as defined in Table 8-60.
NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

Table 8-55 – Format of the MMPL of the DM_BackupAssign.cnf message

Field	Octet	Bits	Description
Rejection code	0	[1:0]	If Accept flag is set to one or if the request is forced, this field shall be set to 00_2. Otherwise it shall indicate one of three possible rejection codes: 01 – Busy. 10, 11 – Reserved by ITU-T.
Reserved		[6:2]	Reserved by ITU-T (Note).
Accept flag		7	Shall be set to one if assignment is accepted (including forced assignment), and zero if rejected (zero is allowed only for voluntary assignment).
NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

Table 8-56 – Format of the MMPL of the DM_BackupData.ind message

Field	Octet	Bits	Description
Backup Data	0 to (N–1)	[(8*N)–1:0]	Data related to domain management provided by the acting domain master as defined in Table 8-60.

The sequence number of the DM_BackupData.ind message (see Table 8-87) shall be set to zero for the first transmission related to the backup assignment and shall be incremented by one after every transmission.

The MMPL of both DM_BackupRelease.req and DM_BackupRelease.cnf messages shall be empty.

8.6.5.3 Recovery of the domain master failure by backup

The node assigned as a domain master backup shall monitor the operation of the domain master and replace it in case of failure. The failure of the domain master shall be detected using the criteria described in clause 8.6.5.3.1. After replacement, the node assigned to be the backup shall act as domain master.

If the failed domain master recovers or joins again the domain, it shall act as a regular node, and may request a handover to get back the domain master role by using the domain master handover procedure described in clause 8.6.6.4.

Besides the domain information provided by the acting domain master (see clause 8.6.5.2), the node assigned as a backup domain master shall collect and maintain the full topological map (see clause 8.6.4) of the domain (the same as the acting domain master), and track all persistent schedules and bandwidth reservation requests in the domain.

8.6.5.3.1 Domain master failure detection

If the node being assigned as a backup domain master observes the following conditions in the domain, it shall detect a failure of the acting domain master:

During NUM_CYCLE_DM_FAIL_DET consecutive MAC cycles:

– no MAP or RMAP frame detected (no MAP or a non-MAP frame at the TXOP when MAP is expected); and

– no transmissions from other nodes of the domain detected, except those that are on persistent schedule; and

– no indication from other nodes (those that are still allowed to transmit) that they detect a domain master (e.g., MDET – see clause 7.1.2.3.2.2.7 of [ITU-T G.9960]).

The value of NUM_CYCLE_DM_FAIL_DET shall be three. Other criteria are for further study.

8.6.5.3.2 Domain master recovery procedure

The node assigned as a backup of the domain master shall track the MAC cycle timing and the position of the CFTXOP assigned for the MAP transmission (clause 8.3) in the MAC cycle. When the node detects a failure of the domain master, it shall immediately take the role of the domain master and start transmitting the MAP, respecting the following conditions:

- the first MAP frame shall be sent on the time position derived with an assumption that the former domain master used a persistent schedule for CFTXOP assigned for MAP transmission and had a constant MAC cycle duration;
- the first generated MAC cycle shall be a continuation of MAC cycles generated by the former domain master, derived by repeating the MAC cycle sequence generated by the former domain master before the failure assuming constant MAC cycle duration during the failure detection period;
- the MAP sequence number of the first MAP frame shall account for the lost MAC cycles assuming constant MAC cycle duration during the failure detection period;
- the new domain master shall keep the frequency of its transmit clock to the same value it has when detecting the failure of the former domain master. That is, it shall maintain the NTR value of the former domain master;
- all persistent schedules for nodes of the domain assigned by the former domain master shall be respected.

Bandwidth reservations and other scheduling-related decisions shall be taken based on the last relevant backup data updates from the former domain master and new bandwidth reservation requests. After the MAP relay nodes receive the MAP frame from the new domain master, they may relay the MAP in the same MAC cycle on the time position derived with an assumption that the former domain master used a persistent schedule for CFTXOPs assigned for RMAP transmissions and had a constant MAC cycle duration.

8.6.6 Domain master selection

At any time, only one node in the domain shall take the role of a domain master in order to coordinate and schedule transmissions in the domain. Among nodes that are capable of operating as a domain master, any can potentially attain the right conditions to take the role of domain master.

NOTE – A domain network that contains more than one node that is capable of becoming the domain master allows for quick recovery from domain master failure and is inherently more fault/failure tolerant (see clause 8.6.5).

A domain master selection protocol defined in this clause shall be used to dynamically select a single domain master in the presence of multiple nodes capable of operating as domain master.

This clause refers only to nodes that are capable of operating as domain master. Nodes that are incapable of acting as domain master shall operate as endpoint nodes and are not subject to the domain master selection procedures described in this clause and its subclauses.

8.6.6.1 Domain master selection at initialization

Following its initialization, a node shall not transmit and shall try to detect MAP frames or RMAP frames associated with one of the domains the node targets to join during a time interval up to t_0. The values of t_0 are specified in clause 8.4.

NOTE 1 – The node identifies a domain it intends to join by comparing the domain name in the detected MAP or RMAP messages with the parameter "Target Domain Name" in its information database (see clause 8.6.1).

NOTE 2 – The value of t_0 is selected taking into account that with relayed admission the time period between two RMAP frames may be up to 200 MAC cycles (see clause 8.5.6)

If MAP or RMAP frames of the target domain are detected within t_0, the node shall start the admission procedure to join the domain, as defined in clause 8.6.1. If no MAP or RMAP frames of the target domain have been detected within t_0 time, the node shall infer that there is no active domain master present in the domain and, after the t_0 interval expires, shall act using the following rules:

- It may start a new domain by becoming its domain master and shall start transmitting MAP frames within duration of one MAC cycle after a t_1 time interval following the expiration of t_0.

  The value of t_1 shall be randomly generated by the node and shall be the range between 0 and 1 seconds. The method of generation of t_1 values is left to the discretion of the implementer.

- If either a MAP or an RMAP frame of the target domain is detected during the t_1 time interval, the node shall not transmit the MAP frame and shall try to synchronize with the detected MAP or RMAP frames and register to the domain using the procedure specified in clause 8.6.1 (as an endpoint node).

8.6.6.2 Domain master recovery in case of no backup ready

If no backup domain master is assigned or a node assigned to be a backup fails, a domain master-capable node shall take the role of the domain master. If a domain master-capable node detects the failure of the domain master, the node shall infer that no active domain master is present in the domain (the backup, if assigned, is not operable) and shall act according to the following rules:

- If the node decides to become a domain master, it shall start transmitting MAP frames after a t_1 time interval following the inference of no presence of the domain master. Otherwise, the node shall stay silent until it detects a new domain master.

  The value of t_1 shall be randomly generated by the node and shall be between 0 and 1 seconds. The method of generation of t_1 values is left to the discretion of the implementer.
- If either a MAP or an RMAP frame of the target domain is detected during the t_1 interval, the node shall not transmit the MAP frame and continue to operate as an endpoint node.
- The duration of the first MAC cycle, after the node takes over the role of the domain master, shall be the same as the one for the MAC cycle before the failure of the domain master.

The same criteria described in clause 8.6.5.3.1 shall be used for the detection of domain master failure except for the value of NUM_CYCLE_DM_FAIL_DET. The value of NUM_CYCLE_DM_FAIL_DET shall be three MAC cycles if no backup is assigned and six MAC cycles if a backup is assigned. Once the node takes the role of the domain master and starts transmitting the MAP, it shall respect all the conditions described in clause 8.6.5.3.2, except for the start time of the first MAP frame and the start time of the subsequent MAC cycle (first two bullets in clause 8.6.5.3.2).

8.6.6.3 Ranking of domain master capabilities

Each node capable of operating as a domain master shall rank its domain master capabilities based on the criteria specified in this clause. If a node that has been admitted to the domain has a higher ranking to be domain master than the existing domain master, the acting domain master may pass the role of domain master to this node. This will ensure that a node acting as domain master is the most suitable one for this role.

Nodes shall rank their domain master capabilities using the following criteria, with highest priority listed first:

1) By configuration setting. All nodes that were preferentially selected (designated by the user or remote management system) to operate as domain master have the highest ranking. In the case when more than one node is configured in this way, the next lowest ranking priority shall be taken into account.
2) By profile number. The node advertising its compliance to the higher profile number shall be ranked higher.
3) By the visibility rate, computed as a ratio between the number of visible nodes to the total number of nodes in the domain. A node with higher rate shall be ranked higher.
4) By capability to operate as a security controller.

Additional criteria are for further study.

Nodes capable of operating as domain master shall advertise their ranking parameters (configuration settings, profile, etc.) in their topology report, as described in clause 8.6.4.

8.6.6.4 Handing domain master's role to a more capable node

The domain master, in support of a possible handover, shall examine the rank of the capability to operate as a domain master for all nodes in the domain using the ranking parameters defined in clause 8.6.6.3 to determine if any node is more capable of operating as a domain master than itself.

If the domain master determines that it should hand over its role to the particular node, the domain master shall send to this node a DM_Handover.req message.

The node shall reply to the domain master in 100 ms with a DM_Handover.cnf message accompanied by a status code that indicates whether the node accepts or denies the handover request.

NOTE 1 – When the node receives a DM_Handover.req message, it may determine that it has insufficient resources to manage the domain.

If the node denies the request, the handover attempt to the node has ended and no further action shall be taken by either the node or the domain master.

Otherwise, the domain master shall take the following actions to facilitate the handover:

- reject any incoming registration requests, to avoid changes to the set of allocated DEVICE_IDs and other domain information;
- reject any requests to establish new or modify existing flows;
- extract the MAC cycle countdown (MCCD) value from the DM_Handover.cnf message. This value sets the number of MAC cycles requested by the node for preparation to assume the role of domain master.

If the domain master does not receive the DM_Handover.cnf message within 300 ms after DM_Handover.req message was sent, it shall repeat the request once more. If, after the second request, the domain master still does not receive the DM_Handover.cnf message within a reasonable time, it shall take no further action (handover failed).

The domain master responds to the DM_Handover.cnf message within 200 ms with a DM_Handover.ind message containing the domain's current state information, which is necessary for continuation of the domain management. The node shall reply to the domain master within 100 ms with DM_Handover.rsp message.

After receiving the acknowledgement (DM_Handover.rsp message) from the node on reception of a DM_Handover.ind message, for the next MCCD successive MAC cycles, the domain master shall transmit each MAP with the following settings:

- Set the HOIP bit in each MAP frame header to one. When detecting that this bit is set in MAP frames, nodes shall refrain from actions that would change the topology of the domain i.e., re-registration, re-authentication, resignation, and acting as proxy to register new nodes. Endpoint nodes shall also suspend their topology update reports.
- Indicate the DEVICE_ID and the REGID (MAC address) of the node that will become the domain master after the handover is complete in the auxiliary information field of the MAP frame. Registered nodes shall read and store this MAC address. This value becomes the permanent MAC address of the new domain master after handover is complete.

If the domain master does not receive the DM_Handover.rsp message within 300 ms after the DM_Handover.ind message was sent, it shall repeat the request once more. If, after the second request, the domain master still does not receive the DM_Handover.rsp message within a reasonable time, it shall take no further action (handover fails).

During these MCCD MAC cycles, the domain master shall broadcast at least once a TM_DomainRoutingChange.ind message (to refresh the topology table of the nodes in the domain).

After the domain master has transmitted MCCD successive MAP frames that include the handover information mentioned above, it shall stop transmitting MAP, while the node that receives the handover shall start transmitting MAP frames taking the role of the domain master. The first transmitted MAP frame shall have the same schedule as the last MAP transmitted by the former domain master and shall be sent in the CFTXOP of the MAP described by the last MAP transmitted by the former domain master. The HOIP bit shall be cleared. This completes the handover protocol.

The new domain master shall respect all persistent schedules assigned by the former domain master, including long inactivity.

NOTE 2 – Robustness and efficiency of the described protocol will increase if the domain master requests Imm-ACK of the DM_Handover.req and DM_Handover.ind messages.

8.6.6.5 Message formats to support handover

8.6.6.5.1 DM_Handover.req message

This message is sent by the domain master to a node selected by the domain master for handover, to determine whether or not the node will accept the role of domain master.

The format of the MMPL of the DM_Handover.req message shall be as shown in Table 8-57.

Table 8-57 – Format of the MMPL of the DM_Handover.req message

Field	Octet	Bits	Description
NumNodes	0	[7:0]	Number of nodes that are registered with the domain master.
NumFlows	1 and 2	[15:0]	Number of service flows that have been established in the domain (Note 1).
NOTE 1 – NumFlows shall not exceed 255. Larger values are for further study. NOTE 2 – Other parameters to evaluate the capability of a node to accept handover are also for further study.			

8.6.6.5.2 DM_Handover.cnf message

This message is sent by a node to the domain master in response to DM_Handover.req, to indicate whether or not the node accepts the handover.

The format of the MMPL of the DM_Handover.cnf message shall be as shown in Table 8-58.

Table 8-58 – Format of the MMPL of the DM_Handover.cnf message

Field	Octet	Bits	Description
StatusCode	0	[7:0]	Value that indicates whether or not the node will accept the role of domain master: 00_{16} = Success (the node will assume the role). 01_{16} = Failure (insufficient resources to register nodes). 02_{16} = Failure (insufficient resources to support flows). 03_{16}-FF_{16} = Reserved by ITU-T.
MAC cycle countdown (MCCD)	1	[7:0]	The number of MAC cycles that shall pass before the node will accept the role of domain master, represented as an unsigned integer. The minimum value of MCCD is 8, the maximum value shall limit the time to accept the role of the domain master to one second.

8.6.6.5.3 DM_Handover.ind message

This message is sent by the domain master to a node to pass the domain's current state information.

The format of the MMPL of the DM_Handover.ind message shall be as shown in Table 8-59.

Table 8-59 – Format of the MMPL of the DM_Handover.ind message

Field	Octet	Bits	Description
Backup Data	0 to (N–1)	[(8*N)–1:0]	Data related to domain management provided by the acting domain master as presented in Table 8-60.

Table 8-60 – Format of the Backup Data field

Field	Octet	Bits	Description
Size of the record	0 and 1	[15:0]	The size of the record in bytes represented as an unsigned integer.
Backup Data	2 to (N–3)	[8*(N–2)–1:0]	Data related to domain management provided by the acting domain master. This (N–2)-octet field is left for further study.

8.6.6.5.4 DM_Handover.rsp message

This message is sent by a node to acknowledge the reception of DM_Handover.ind. The MMPL of this message is empty.

8.6.7 Selection of PHY-frame header segmentation

The domain master shall use $D = 2$ (as defined in clause 7.1.3.5.2 of [ITU-T G.9960]) for the MAP-D frame transmission regardless of its medium type. Any node relaying the MAP-D frame shall also use $D = 2$.

The value of D used in a specific TXOP shall be indicated in TXOP attributes extension (clause 8.8.4.1.1) and field HSEG of the PHY-frame header (see Table 7-1 of [ITU-T G.9960]). Selection of D for a given TXOP is vendor discretionary.

8.6.8 Selection of the DNI and the DOD

As the node starts its role as a domain master (either at initialization or as a result of a handover or a backup), it shall set the values of DOD and DNI and use them starting from the first transmitted MAP.

8.6.8.1 Selection and maintenance of the DOD

The domain master, prior to sending its first MAP frame, shall monitor the DODs of all visible neighbouring domains and pick the DOD randomly between one and 15, excluding values already used by neighbouring domains. The value DOD=0 is reserved by ITU-T.

If during domain operation, a neighbouring domain with the same DOD is discovered by the domain master or reported by a node of the domain (in its topology report), the domain master shall perform a procedure for DOD change containing the following steps:

1) Reject all active admission procedures.
2) Select a new DOD value between one and 15, excluding values already used by neighbouring domains.

3) Modify the DOD in the domain using the following procedure:
   - pick a random time period between one and four seconds with a granularity of 200 ms;
   - at the expiration of the selected period, check whether the same DOD is still in use by the neighbouring domain. If not, abandon the DOD modification procedure;
   - if the neighbouring domain still uses the same DOD, the domain master shall set the value of the AUX_VALID counter to four and indicate the new DOD value in the "DOD Update" sub-field of the Auxiliary information field, see clause 8.8.5;
   - start transmitting MAPs with the new value of DOD in the MAC cycle described in the MAP in which the AUX_VALID counter reaches zero.

When a node detects a MAP indicating a change of the DOD Update sub-field, it shall use the new DOD value starting from the MAC cycle in which the AUX_VALID counter reaches 0. When a node turns into active state after being inactive for a time period longer than one MAC cycle, it shall verify the DNI of the MAP frame and modify the DOD if necessary.

The node indicates neighbouring domain with the same DOD in its topology report.

8.6.8.2 Selection and maintenance of the DNI

The domain master, prior to sending its first MAP frame, shall compute the DNI using the default value of the hash key and using the procedure defined in clause 8.6.8.2.1. Further, the domain master shall monitor the DNI of all visible neighbouring domains and re-compute the DNI using one of the alternative hash keys if the same value of DNI is discovered.

If during domain operation, a neighbouring domain with the same DNI is discovered by the domain master or reported by a node of the domain (in its topology report), the domain master shall change the DNI by applying one of the alternative hash keys. Further, the domain master shall transmit MAPs with a new DNI.

Nodes that join the domain shall verify the DNI using the hash key indicated in the DNI_KeyID field of the MAP header.

8.6.8.2.1 Generation of the DNI

The 16-bit value of the DNI shall be generated by the domain master from the 32-byte domain name using the following steps:

1) The bytes of the domain name shall be examined sequentially from the first one to the last one: all bytes whose hexadecimal value is less than 20_{16} shall be removed.
2) The rest of the bytes shall be concatenated into a single codeword in the order they are in the domain name. Prior to concatenation, the MSB of each byte shall be removed. The valid length of this codeword is from seven to 224 bits. The first bit of the codeword is the LSB of the domain name byte of the lowest order that is a member of the codeword.
3) The obtained codeword is hashed by picking each m-th bit of the codeword, starting from the first bit (b0) and going through the codeword. The process shall stop when the number of collected bits is 16. If less than 16 bits are collected when the end of the codeword is reached, the pointer shall wrap around to the beginning of the codeword. The value of *m* is a hash key and shall have valid values from two to nine. The default value of the hash key is four.

The process is illustrated in Figure 8-42.

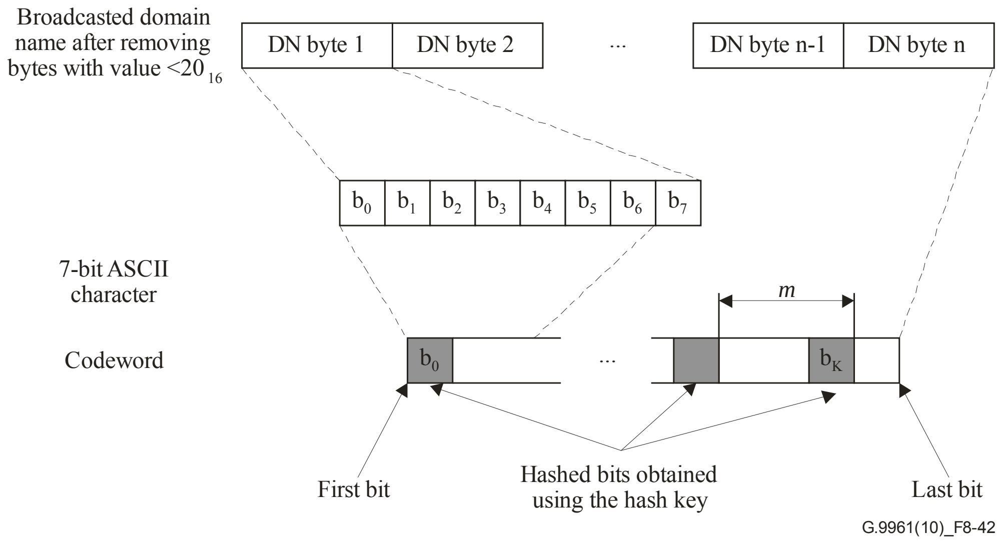


Figure 8-42 – Generation of a DNI from the domain name

8.7 Addressing scheme

8.7.1 Node identifier

The following three node identification parameters shall be used:

– DEVICE_ID;

– MULTICAST_ID;

– BROADCAST_ID.

The same node can be identified by its unique DEVICE_ID, by several MULTICAST_IDs and by the BROADCAST_ID, which is the same for all nodes.

The node identifier shall be used to identify the assignment of TXOPs and TSs within STXOPs to the nodes in the MAP and to identify the source and destination of a PHY frame (SID and DID, see clause 7.1.2.3 of [ITU-T G.9960]). Definition and valid values for the node identifier are summarized in Table 8-61.

NOTE – The SID is always a DEVICE_ID, while the same node can be addressed by multiple DIDs: by its unique DEVICE_ID, by several Multicast_IDs, and by the predefined Broadcast_ID (see clause 7.1.2.3 of [ITU-T G.9960]).

8.7.1.1 DEVICE_ID

After a domain master has successfully registered the node to the domain, as described in clause 8.6.1, it shall assign a unique DEVICE_ID for that registered node and communicate the assigned DEVICE_ID to the registered node. The assigned DEVICE_ID shall be used until the node is resigned (explicitly or implicitly) from the domain. After the node resigns from the domain, the domain master shall terminate its DEVICE_ID. The same DEVICE_ID value can be further assigned to any new registered node.

The domain master shall communicate the assigned DEVICE_ID to the registered node and terminate the DEVICE_ID after the node resigns from the domain using the registration protocol described in clause 8.6.1.1.1.

The DEVICE_ID shall be represented by an 8-bit unsigned integer with valid values in the range from zero to 250 as presented in Table 8-61. Value zero shall be used as the default DEVICE_ID of a node attempting to join the network.

8.7.1.2 MULTICAST_ID and BROADCAST_ID

MULTICAST_IDs shall be generated autonomously by the nodes creating multicast groups for the multicast transmission. MULTICAST_IDs only apply to multicast transmissions among nodes communicating directly (i.e., not via a relay node). A node shall generate a unique MULTICAST_ID for each multicast group that it creates. The node creating the multicast group shall communicate the MULTICAST_ID to all multicast destination nodes using the protocol described in clause 8.16 before starting the multicast transmission. As the multicast session is complete, nodes shall terminate the MULTICAST_ID according to an explicit notification from the node creating the multicast group (see clause 8.16).

A node may be addressed by multiple MULTICAST_IDs generated by different nodes for a multicast transmission to this node. The same MULTICAST_ID can be used as DID by several nodes for different multicast groups. The differentiation in the receiver is by the SID of the node creating the multicast group.

The MULTICAST_ID shall be represented by an 8-bit unsigned integer with valid values in the range from 1 to 254. It is distinguished from a DEVICE_ID in the PHY-frame header and in the MAP by the Multicast Indicator described in clause 7.1.2.3.1.5 of [ITU-T G.9960].

A BROADCAST_ID is a MULTICAST_ID with a fixed value of 255 and shall be used for broadcast transmission only.

Table 8-61 – Definition and valid values of node identification parameters

Parameter	Valid values	Description
DEVICE_ID	0	The ID used by a new node joining the network before it is assigned a unique DEVICE_ID by the domain master. The domain master shall not assign the DEVICE_ID = 0 to any node admitted to the network.
	1 to 250	IDs reserved for assignment by the domain master to nodes admitted to the network.
	251 to 255	Reserved by ITU-T
MULTICAST_ID	0	Reserved by ITU-T
	1 to 254	IDs reserved for assignment for multicast traffic
BROADCAST_ID	255	A special value of MULTICAST_ID reserved for broadcast traffic

8.7.2 Flow identifier (FLOW_ID)

A node may source multiple flows where each flow is identified by a FLOW_ID. The FLOW_ID is uniquely assigned by the node originating the flow (sourcing the flow) as the flow and its associated data connection are established, and is released by the same node as the flow is terminated.

A FLOW_ID shall be represented by an 8-bit unsigned integer. Valid values of FLOW_ID are in the range of 8 to 254. A flow is uniquely identified in the domain by the tuple (SID, FLOW_ID). Different nodes may assign the same values of FLOW_ID to the flows they originate.

The FLOW_ID with a value of 255 has a special meaning when used within a TXOP descriptor in the MAP frame (see clause 8.8.4.2).

8.8 Medium access plan (MAP) frame

The MAP frame describes when a subsequent MAC cycle shall start and describes the TXOPs it shall include, and includes information and parameters of the domain operation. Each MAC cycle is identified by a sequence number (see clause 8.8.3) contained in the MAP frame describing it.

By default, the medium access plan described by a MAP frame is for a single MAC cycle, but it may carry persistency information related to one or more subsequent MAC cycles.

A MAP frame shall have a format of a regular PHY frame (see clause 7.1.2.1 of [ITU-T G.9960]) identified by the frame type = MAP in the PHY-frame header (see clause 7.1.2.3.1.1 of [ITU-T G.9960]) with indication of the type of the MAP (MAP-A or MAP-D – see clause 7.1.2.3.2.1.10 of [ITU-T G.9960]). Other fields of the MAP PHY-frame header shall be as described in clause 7.1.2.3 of [ITU-T G.9960].

The structure of the MAP message shall be as described in Figure 8-43.

MAP header	TXOP Desc [0]	...	TXOP Desc [N]	Auxiliary information

G.9961(10)_F8-43

Figure 8-43 – MAP message structure

The MAP message contains a MAP header (clause 8.8.3) followed by a number of TXOP descriptors (clause 8.8.4), followed by an Auxiliary Information field (clause 8.8.5). The number of TXOP descriptors in the MAP as well as control and sequence information is encoded in the MAP header.

The MAP message, as presented in Figure 8-43 shall be transmitted in a single LCDU and shall be communicated as an LLC frame with LLCFT = 1. The MAP message shall be the payload of the LCDU, and it shall be ≤ 1500 bytes (see clause 8.1.3.4).

The LLC frame containing the MAP message shall be the only data unit contained within the MAP PHY frame. The SA of the LCDU shall have the value of REGID of the domain master both for MAP and RMAP (RMAP-A, RMAP-D). The DA of the LCDU shall be set to $01\text{-}19\text{-}A7\text{-}52\text{-}76\text{-}96_{16}$.

The MAP frame shall not be subject to ARQ (no ACKs shall be sent). The SSN field in the first LPDU in a MAP PHY frame shall be initialized to zero and shall be incremented by one for each subsequent LPDU in the same MAP PHY frame.

The MQF flag in the LPH shall be set to one.

The MAP frame shall be sent unencrypted and shall not carry the timestamp: the TSMPI and the CCMPI bits of the LFH of the MAP LCDU shall be set to zero.

8.8.1 MAP generation and distribution

The domain master shall generate and manage distribution of a MAP each MAC cycle. The MAP may vary from one MAC cycle to another.

The domain master shall transmit at least one MAP frame each MAC cycle and may transmit additional MAP frames each MAC cycle. However, the MAP transmitted by the domain master shall not change within a MAC cycle, except the sub-fields of the Auxiliary Information field that are not related to scheduling and persistence information.

In addition, the domain master may designate one or more nodes as MAP relays. Designated nodes shall transmit RMAP frames containing the MAP, as described in clauses 8.5.6, 8.6.1 and in Table 8-70.

The domain master may distribute the MAP by transmitting default MAP frames (MAP-D) or both default and active MAP frames (MAP-A). The payload bits of the MAP-D frame (and RMAP-D frame) shall be mapped to sub-carriers using Pre-defined BAT Type 1, while the payload bits of the MAP-A frame (and RMAP-A frame) shall be mapped to sub-carriers using Pre-defined BAT Type 2. The type of the MAP frame (MAP-D or MAP-A) is indicated by the MAP_TYPE field of the MAP frame header (see 7.1.2.3.2.1.10 of [ITU-T G.9960]) and in the TXOP allocated for the MAP frame transmission (see MAP Type field in Table 8-63).

The decision to transmit a MAP-D or a MAP-A frame or both in a particular MAC cycle is left to the discretion of the domain master. If the domain master transmits both MAP-D(s) and MAP-A(s) in the same MAC cycle, the scheduling information and persistence information defined in the MAP messages of those MAP frames shall not conflict.

The MAP-D frame is transmitted to facilitate admission of new nodes to the domain. Therefore, the MAP-D frame may include only the relevant information needed by the registering nodes to synchronize with the MAC cycle, to learn the regional transmission parameters, and to locate the TXOP to transmit the registration request message. A MAP-D message that only contains this reduced information is referred to as a reduced MAP-D. It is recommended that the MAP-D frame be transmitted in the reduced format to save bandwidth. The TXOPs descriptors included in a reduced MAP-D frame shall be described using the absolute timing extension (see clause 8.8.4.1.1) when it is needed to skip over TXOPs that are specified in the complete (not reduced) MAP of the same MAC cycle.

NOTE – The MAP-D frame is normally transmitted to facilitate admission of new nodes to the domain; rare transmission of a MAP-D may result in unacceptably long admission time and failure of the admission procedure.

8.8.2 MAP frame transmission

During each MAC cycle, the domain master shall allocate at least one CFTXOP assigned for MAP transmission. The domain master may allocate additional CFTXOPs and/or CFTSs in STXOPs assigned for MAP transmission.

The domain master shall transmit only one MAP frame in each allocated CFTXOP assigned for MAP transmission. The domain master may transmit additional MAP frames in CFTSs in STXOPs assigned for MAP transmission. The first transmitted MAP frame in a MAC cycle shall be a complete MAP. The domain master may transmit MAP frames in CBTS. MAP frames transmitted in CBTS shall use a medium access priority of MA3.

At least one MAP frame shall be transmitted during each MAC cycle that describes the complete schedule of the immediately following MAC cycle except for cases where the part of the MAP corresponding to Persistent TXOPs might not contain the scheduling information for the immediately following MAC cycle (see clause 8.8.6). Once the MAP for a particular MAC cycle is announced, the scheduling for that MAC cycle shall not be changed by any subsequent transmissions of MAP/RMAP frames. Transmission of MAP or RMAP frames shall be completed at least MAP_TX_SETUP_TIME before the start of the MAC cycle that it describes. The value of MAP_TX_SETUP_TIME is defined in clause 8.4. The scheduler shall ensure a gap of INTER_MAP_RMAP_GAP (see clause 8.8.6) between the end of the transmission of a MAP or RMAP frame and the start of the RMAP that has to be derived from that MAP or RMAP. The destination identifier (MI and DID fields) in transmitted MAP frames shall indicate the broadcast address.

NOTE – Nodes already registered to the domain are familiar with the domain specific parameters, such as the regional PSD masks. For these nodes, decoding MAP-As is likely to result in improved performance compared to decoding MAP-Ds. It is therefore recommended that the MAP carrying the complete schedule be a MAP-A.

To enable potential hidden nodes to join the domain, the domain master shall schedule the transmission of RMAP-D frames by the MAP relay capable nodes. For each MAP relay capable node the domain master shall schedule RMAP-D transmission in three consecutive MAC cycles. The domain master shall schedule, in a round-robin manner and with a maximum interval of 1 s, a different "MAP relay capable" node each time it schedules an RMAP-D transmission.

If the domain master intends to change some of the sub-fields of the auxiliary information field, it shall use the mechanism of the auxiliary information validity counter (AUX_VALID) described in clause 8.8.5.

8.8.3 MAP header

The MAP header is of variable size and shall include at least the fields listed in Table 8-62. The size of the MAP header shall be a multiple of four octets. The format of the MAP header allows its extension for future versions by adding fields after the last "Reserved" field in Table 8-62.

Table 8-62 – MAP header format

Field	Octet	Bits	Description
Sequence Number	0 and 1	[15:0]	A MAP sequence number. The sequence number shall be incremented by one for each MAC cycle (modulo 2^{16}). An RMAP shall keep the sequence number of the original MAP (shall not increment the sequence number).
MAP Header Length	2	[7:0]	The length of the MAP header expressed in a number of 32 bit words.
Number of entries	3 and 4	[15:0]	Number of TXOP descriptors, including TXOP extensions, in the MAP.
TICK_Factor	5	[2:0]	A time shift factor that shall be used to determine the resolution of a TXOP TIME_UNIT (see clause 8.2.3). The resolution of a TIME_UNIT is determined as follows: $TIME\_UNIT = TICK * 2^{TICK\_Factor}$ The values of TICK are defined in clause 8.4.
Reserved		[3]	Reserved by ITU-T (Note 1).
RoutingAuthorization		[4]	Relevant to CRTM mode in case of broken link (see clause 8.6.4.2.1). 0 – nodes are not authorized to calculate routing 1 – nodes are authorized to calculate routing temporarily until routing information arrived from the domain master.
Topology Mode		[5]	0 – CRTM mode. 1 – DRTM mode.
Handover In Progress (HOIP)		[6]	When set to one, indicates that the present domain master is handing over its role to a newly registered node. At other times, is set to zero.
MAP Modified		[7]	The domain master shall set this bit when the MAP is modified compared with the previous MAP and reset otherwise.
Future Schedule Life Time (FSLT)	6	[3:0]	If FSLT is non-zero, the MAP frame carries the future TXOP schedule which shall take effect when CSLT reaches zero and shall remain valid for FSLT plus one consecutive MAC cycles after it takes effect.
Current Schedule Life Time (CSLT)		[7:4]	CSLT + 1 is the number of consecutive MAC cycles in which the TXOP schedule described in the MAP shall remain valid. The value of CSLT shall be reduced by one after each MAC cycle with FSLT > 0.
Domain Name Identifier (DNI)	7 and 8	[15:0]	The generation of DNI value and its format are defined in clause 8.6.8.2.

Table 8-62 – MAP header format

Field	Octet	Bits	Description
RoutingSequenceNumber	9 and 10	[15:0]	The sequence number of the last transmitted routing message.
Reserved	11	[4:0]	Reserved by ITU-T.
Routing Algorithm		[6:5]	Contains a specified standard algorithm (Note 2).
PrvRouting Algorithm		[7]	Bit 7 – If set to one it means that the domain master uses a vendor-specific algorithm. If it is zero, then bits 6:5 contains a specified standard algorithm.
MAC cycle duration	12 to 14	[23:0]	The duration of the MAC cycle in TICK units. There are two cases: If the MAP includes a future persistent schedule, then the duration is of this future MAC cycle. In all other cases the duration is of the next MAC cycle. This duration covers the time period between two consecutive CYCSTARTs (see clause 7.1.2.3.2.1.3 of [ITU-T G.9960]). The minimal and maximal durations of the MAC cycle are defined in clause 8.4.
DNI_KeyID	15	[2:0]	A value of DNI key (m) encoded as an unsigned integer minus 2; this key shall be used to compute the DNI as defined in clause 8.6.8.2.1.
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 – These bits shall be set to zero. Specification of standard routing algorithms is for further study.			

8.8.4 TXOP descriptor

Each TXOP is described by at least one TXOP descriptor. A TXOP descriptor is composed of a basic TXOP descriptor that may be extended by one or more additional TXOP descriptor extensions (see clauses 8.8.4.1.1 to 8.8.4.1.3). TXOP descriptor extensions supply additional information like scheduling information, timing information and TXOP attributes.

Basic TXOP descriptors and TXOP descriptor extensions are each four octets in length.

A TXOP descriptor represents the right of a certain node or a set of nodes to transmit within a certain TXOP. A CFTXOP shall be described using a single TXOP descriptor. A CBTXOP shall be described by either a single TXOP descriptor (see clause 8.3.3.4.5.3) or by multiple TXOP descriptors (see clause 8.3.3.4.5.2). A STXOP shall be described using several TXOP descriptors representing the TSs within the STXOP.

The domain master shall not assign more than 127 TXOP descriptors in the MAP, describing TSs, within a single STXOP (including CBTXOP).

The differentiation between different TXOPs shall be done via the TXOP attributes extension, which shall be appended to the last TXOP descriptor of a TXOP. The TXOP attributes extension supplies the timing information for the TXOP (see clause 8.8.4.1.1). A node associated with a TXOP or a TS is uniquely identified in a TXOP descriptor by the *SID* field, which shall be set to the DEVICE_ID of the node as was assigned by the domain master.

A flow associated with a TXOP or a TS is uniquely identified in a TXOP descriptor by the combination (SID, FLOW_ID). A FLOW_ID is a unique identifier of a flow associated with the SID.

A user priority associated with a TXOP or TS is uniquely identified in a TXOP descriptor by the tuple (SID, PRI). The PRI value shall represent the lowest MPDU priority that may be sent in the TXOP or TS.

Table 8-63 describes the basic TXOP descriptor. When the extension bit is set, the TXOP descriptor shall have an extension, as described in clause 8.8.4.1. Different types of TXOP descriptor extensions are distinguished by extension type.

Table 8-63 – Basic TXOP descriptor format

Field	Octet	Bits	Description
SID	0	[7:0]	SID = 1-250 identifies the DEVICE_ID of the node assigned to the TXOP. SID = 0, 255 indicates special values for the TXOP descriptor (see clause 8.8.4.2).
DID	1	[7:0]	DID = 0 indicates that the DID of the destination node of the flow is not known to the domain master. DID > 0 indicates the destination node for the flow. DID shall be set to the DEVICE_ID as described in Table 8-61.
Multicast Indication/MAP type	2 and 3	[0]	If this field is a special TXOP descriptor of a MAP (see clause 8.8.4.2) it indicates the type of MAP that shall be transmitted: 0 indicates MAP-A, 1 indicates MAP-D. If this field is not a special TXOP descriptor of a MAP this field contains the multicast indication: 1 indicates multicast/broadcast DID, 0 otherwise.
PR signal required		[1]	This bit instructs nodes contending for transmission in a CBTS whether to use the PR signal: 0 – PR signal shall not be used. 1 – PR signal is required.
CBTS Closure Mode		[3:2]	This field instructs nodes where to close a CBTS that was used for transmission (see in clause 8.3.3.4.5): 00 – Duration-based. 01 – Timeout-based from frame sequence start. 10 – Timeout-based from CBTS start. 11 – Reserved by ITU-T.
Reduced MAP		[4]	0 – TXOP for a complete MAP. 1 – TXOP for a reduced MAP. See clause 8.8.1.
Reserved		[5]	Reserved by ITU-T (Note).
FLOW_ID/PRI		[13:6]	Identifies the flow or the user priority associated with the TXOP/TS. Valid values for user priority assignments are 0-7 Valid values for FLOW_ID assignments are 8-254 Value 255 indicates special values for the TXOP descriptor (see clause 8.8.4.2).
Last_in_Group		[14]	1 indicates the last TS of a group of TSs in STXOP, 0 otherwise. Shall be set to zero for CFTXOP.

Table 8-63 – Basic TXOP descriptor format

Field	Octet	Bits	Description
Extension		[15]	0 – No extension is present. 1 – This TXOP descriptor contains an extension.
NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

Several TSs within the same STXOP can be grouped together to specify common attributes for these TSs via a group information extension (see clause 8.8.4.1.3). Grouping of several TSs shall be done by setting the Last_in_Group indication in the TXOP descriptor of the last TS of the group. Groups are implicitly numbered according to their appearance in the MAP. The first group shall be identified as group number one and so on. If a group contains only one TS, the descriptor of this TS shall have its Last_in_Group bit set to one.

8.8.4.1 TXOP descriptor extension

All TXOP descriptor extensions shall be of 4 octets and shall have the format as described in Table 8-64. A TXOP descriptor may have more than one extension.

Table 8-64 – TXOP descriptor extension format

Field	Octet	Bits	Description
Extension data	0 to 3	[26:0]	Extension data (see clauses 8.8.4.1.1 to 8.8.4.1.5).
Extension Type		[30:27]	TXOP descriptor extension type: 0 – TXOP attributes (see clause 8.8.4.1.1). 1 – TXOP absolute timing (see clause 8.8.4.1.2). 2 – Group information (see clause 8.8.4.1.3). 3 – Maximum transmission limitation (see clause 8.8.4.1.4). 4 – CBTS nodes information (see clause 8.8.4.1.5). 5-15 – Reserved by ITU-T.
Extension		[31]	0 – No more extensions present. 1 – This TXOP descriptor contains more extensions.

8.8.4.1.1 TXOP attributes extension data

A TXOP attributes extension shall be identified by extension type 0 and shall be used to specify the TXOP duration and restrictions on the type of traffic that can be sent within the TXOP.

The TXOP start time is defined as the sum of the durations of all preceding TXOPs, unless the start time is marked as the same as the previous TXOP start time (by setting the 'Start_Time_Type' bit to one) and unless the TXOP absolute timing extension is used. The Start_Time_Type bit in the extension shall be ignored if the TXOP absolute timing extension is present. The end time of each TXOP in the MAP shall be equal to or smaller than the end of the MAC cycle.

The format of the TXOP attributes extension is described in Table 8-65.

Table 8-65 – TXOP attributes extension data format

Field	Octet	Bits	Description
Length	0 to 2	[17:0]	Duration allocated to the TXOP in TIME_UNIT units where the size of a TIME_UNIT is equal to the base TICK size (the values of TICK are defined in clause 8.4) multiplied by a constant factor defined in the MAP header (see TICK_Factor in clause 8.8.3).
Traffic Limitation		[19:18]	Restrictions on the type of traffic that can be sent in the TXOP: 0 – No restriction (default). 1 – Channel estimation only. 2-3 – Reserved by ITU-T.
Non-Persistent/Persistent		[20]	0 – Non-persistent TXOP (Default). 1 – Persistent TXOP.
Start_Time_Type		[21]	0 – TXOP start time is at the end of the previous TXOP and shall be computed as the sum of the durations of all preceding TXOPs (default). 1 – TXOP start time is the same as the start time of the previous TXOP (e.g., spatial reuse). This field shall be ignored if the TXOP absolute timing extension is appended to the TXOP descriptor.
Header segmentation		[22]	0 – PHY-frame header is segmented into one symbol (D = 1). 1 – PHY-frame header is segmented into two symbols (D = 2). (see clause 7.1.3.5.2 of [ITU-T G.9960])
Enhanced frame detection (EFD) STXOP Indicator		[23]	0 – Indicates a non-EFD STXOP (see clause 8.3.3). 1 – Indicates an EFD STXOP (see clause 8.3.3.5).
TS_Grid_Resync	3	[0]	0 – A node that inferred loss of synchronization with the TS grid of this STXOP shall attempt to resynchronize with the TS grid (as described in clause 8.3.3.6) (Default). 1 – A node that inferred loss of synchronization with the TS grid of this STXOP shall refrain from transmission until the end of the STXOP (as described in clause 8.3.3.6) (Note).
INUSE signal required		[1]	This bit instructs nodes contending for transmission in a CBTS in this TXOP whether to use INUSE signal: 0 – INUSE signal shall not be used. 1 – INUSE signal is required.
RTS/CTS required		[2]	This bit instructs the transmitter to use RTS/CTS prior to the data: 0 – RTS/CTS shall not be used. 1 – RTS/CTS is required.
Extension Type and Extension		[7:3]	See Table 8-64.
NOTE – This bit does not apply to CBTXOP without INUSE.			

8.8.4.1.2 TXOP absolute timing extension data

A TXOP absolute timing extension shall be identified by extension type 1 and shall be used to specify absolute start time of a TXOP within the MAC cycle. When this extension is not present, a TXOP shall start as defined in the TXOP attribute extension.

The TXOP absolute timing extension is described in Table 8-66.

Table 8-66 – TXOP absolute timing extension data format

Field	Octet	Bits	Description
Start_Time	0 to 2	[17:0]	Start time of the TXOP 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.
Reserved		[23:18]	Reserved by ITU-T (Note).
Reserved	3	[2:0]	Reserved by ITU-T (Note).
Extension Type and Extension		[7:3]	See Table 8-64.
NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

8.8.4.1.3 Group information extension data

By default the order of appearance of TSs in the MAP describes the relative scheduling order of the TSs within a STXOP. The TSs shall appear on the line sequentially one after the other, as described in the MAP regardless of each TS usage. After the last TS, the first TS shall reappear, and so on until the end of the STXOP.

The group information extension shall be identified by extension type 2 and shall be used to specify the order of TS in STXOP different from the default sequential behaviour depending on TS usage. Several TSs can be grouped together in order to apply common behaviour to all TSs within the group. When a certain TS requires specific rules, it shall be defined as a separate group to which the group information extension shall be applicable.

The group information extension shall be used to indicate whether the control of the line shall be passed to a different group. Control can be passed back to the current group as well in order to create repetition.

The group information extension is composed as described in Table 8-67, and shall always extend the TXOP descriptor of the last TS of a group.

Table 8-67 – Group information extension data format

Field	Octet	Bits	Description
GroupOnActivity	0	[7:0]	Group number (see clause 8.8.4) of the next group to which control is passed when activity is detected in any of the TSs of the current group. A value of zero indicates default sequential behaviour, described in clause 8.3.3.2.2.

Table 8-67 – Group information extension data format

Field	Octet	Bits	Description
GroupOnSilence	1	[15:8]	Group number of the next group to which control is passed when no activity is detected in all of the TSs of the current group. A value of zero indicates default sequential behaviour, described in clause 8.3.3.2.2.
Reserved	2 and 3	[10:0]	Reserved by ITU-T (Note).
Extension Type and Extension		[15:11]	See Table 8-64.
NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

8.8.4.1.4 Maximum transmission limitation extension data

A maximum transmission limitation extension shall be identified by extension type 3 and shall be used to specify a maximum allowed transmission length in all the TSs of a STXOP.

The maximum transmission limitation extension is described in Table 8-68.

Table 8-68 – TS timing extension data format

Field	Octet	Bits	Description
Max_TS_Length	0 to 3	[17:0]	Maximum transmission length in all the TSs included in the STXOP 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.
Reserved		[26:18]	Reserved by ITU-T (Note).
Extension Type and Extension		[31:27]	See Table 8-64.
NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

8.8.4.1.5 CBTS nodes information extension data

A CBTS node information extension shall be identified by extension type 4 and shall be used to specify the specific list of nodes that are allowed to contend in a particular CBTS as specified via a TXOP descriptor (see clause 8.8.4.2). The list of nodes shall be described by indicating the DEVICE_IDs. Several CBTS nodes information extensions may be used for describing a list.

The CBTS nodes information extension is described in Table 8-69.

Table 8-69 – CBTS nodes information extension data format

Field	Octet	Bits	Description
Include_Exclude	0	[0]	0 – All nodes indicated in the following entries can contend in this CBTS. 1 – All nodes indicated in the following entries cannot contend in this CBTS.

Table 8-69 – CBTS nodes information extension data format

Field	Octet	Bits	Description
Entry format		[1]	0 – byte map format. 1 – bit map format.
Reserved		[7:2]	Reserved by ITU-T (Note).
Byte map format			
Entry number 1	1	[7:0]	0 = New nodes joining network. 1-250 – Identifies the DEVICE_ID of a registered node. 251-254 – Reserved by ITU-T. 255 – This entry shall be ignored.
Entry number 2	2	[7:0]	0 = New nodes joining network. 1-250 – Identifies the DEVICE_ID of a registered node. 251-254 – Reserved by ITU-T. 255 – This entry shall be ignored.
Entry number 3	3	[7:0]	0 = New nodes joining network. 1-250 – Identifies the DEVICE_ID of a registered node. 251-254 – Reserved by ITU-T. 255 – This entry shall be ignored.
Reserved	4	[2:0]	Reserved by ITU-T (Note).
Extension Type and Extension		[7:3]	See Table 8-64.
Bit map format			
Entry number 1	1	[7:0]	0 – New nodes joining network. 1-250 – Identifies the DEVICE_ID of a registered node. 251-255 – Reserved by ITU-T.
Entry number 2	2 and 3	[0]	Identifies status for DEVICE_ID = Entry number 1+1. 0 – Node included in the list. 1 – Node not included in the list.
Entry number 3		[1]	Identifies status for DEVICE_ID = Entry number 1+2. 0 – Node included in the list. 1 – Node not included in the list.
…		…	…
Entry number 15		[13]	Identifies status for DEVICE_ID = Entry number 1+14. 0 – Node included in the list. 1 – Node not included in the list.
Entry number 16		[14]	Identifies status for DEVICE_ID = Entry number 1+15. 0 – Node included in the list. 1 – Node not included in the list.
Reserved		[15]	Reserved by ITU-T (Note).
Reserved	4	[2:0]	Reserved by ITU-T (Note).
Extension Type and Extension		[7:3]	See Table 8-64.
NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

8.8.4.2 Special values for the TXOP descriptor

Several values for the TXOP descriptor are defined to serve as special MAP control directives as described in Table 8-70.

Table 8-70 – Special values for the TXOP descriptor

Descriptor	SID	FLOW_ID	Semantics
CBTS	0	0-7	A TXOP descriptor that specifies a CBTS associated with a user priority as specified by FLOW_ID/PRI field.
RCBTS	0	255	A TXOP descriptor that identifies RCBTS (TS for registration use only).
MAP	1 to 250	255	Identifies a TXOP descriptor allocated for transmission of a MAP frame or an RMAP frame. If the SID field contains the DEVICE_ID of the domain master then the TXOP descriptor allocation is for the domain master to transmit a MAP frame, otherwise the allocation is for a non-domain master node to transmit a RMAP frame.
Silent TXOP or TS	255	N/A	A TXOP descriptor that specifies a TXOP or TS in which transmissions are prohibited.

8.8.5 Auxiliary information field

The format of the auxiliary information field shall be as shown in Table 8-71. The auxiliary information field consists of three components – the auxiliary information validity counter (AUX_VALID), the length of the field (AUX_LEN) and the aggregated auxiliary information field (AUX_INFO).

The auxiliary information validity counter (AUX_VALID) is to indicate the MAC cycle in which changes in the sub-fields of the auxiliary information field shall take effect. This field, in conjunction with the ModificationFlag defined in each of the auxiliary information sub-fields, shall be used by the domain master to publish changes in the auxiliary information ahead of time (see clause 8.8.2), for some of the auxiliary information sub-fields. The ModificationFlag is used to signal whether the AUX_VALID counter is applicable for the specific sub-field. If the domain master intends to change some of the sub-fields of the auxiliary information field, it shall start transmitting these new sub-fields in the MAP, setting the AUX_VALID counter to a value of N in this MAP, and setting the ModificationFlag of these sub-fields to one. The value of N is vendor discretionary in the range between three and seven. The domain master shall decrement the AUX_VALID counter by one in each one of the following MAC cycles, until the counter reaches zero. Nodes shall update the auxiliary information in the sub-fields marked by a ModificationFlag set to one in the MAC cycle described by the MAP containing the AUX_VALID counter with a value of zero. The parameters intended for modification (marked with ModificationFlag set) shall be transmitted during all N MAC cycles and their values during these N MAC cycles shall not change. Auxiliary information sub-fields having their ModificationFlag set to zero are not using the validity counter mechanism aforementioned. Nodes shall update the auxiliary information in such sub-fields in the MAC cycle described by the MAP, without considering the value of the AUX_VALID counter. Some of the auxiliary information sub-fields have their ModificationFlag always set to one, while others can be set to either zero or one by the domain master.

AUX_INFO includes an integer number of octets and consists of one or more concatenated auxiliary information sub-fields of different type and length. If there is no auxiliary information to send, AUX_LEN shall be set to zero. Otherwise, AUX_LEN shall be set to the size of AUX_INFO in octets.

Table 8-71 – Format of auxiliary information field

Field	Octet	Bits	Description
Validity counter (AUX_VALID)	0 and 1	[2:0]	This counter indicates the number of MAC cycles after which the changes in the sub-fields of the auxiliary information field shall take effect.
Length (AUX_LEN)		[13:3]	Length of the aggregated auxiliary information field (AUX_INFO) in octets.
Reserved		[15:14]	Reserved by ITU-T (Note).
Aggregated auxiliary information (AUX_INFO)	≥ 2		Aggregated auxiliary information, containing one or more concatenated sub-fields, each of which can have fixed or variable length. The size of AUX_INFO shall be limited by the maximum MAP length.
NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

The format of auxiliary information sub-fields shall be as shown in Table 8-72.

Table 8-72 – Format of an auxiliary information sub-field

Field	Octet	Bits	Description
Type	0	[6:0]	Type of the sub-field expressed as a hexadecimal integer.
ModificationFlag		[7]	This bit shall be set to zero if the parameters are subject to immediate modification regardless of the value of AUX_VALID. This bit shall be set to one if the parameters in this field are subject to validity counter-based modification.
Length	1	[7:0]	Length of the sub-field data in octets, expressed as a decimal integer in the range between 1 and 255.
Sub-field data	≥ 2 (up to 254)		Sub-field data, as presented in the sub-clauses of clause 8.8.5.

The types of auxiliary information sub-fields are specified in Table 8-73.

Table 8-73 – Types of auxiliary information sub-fields

Type	Value	Description
Reserved	00_{16}	Reserved by ITU-T.
Domain name	01_{16}	A sub-field indicating domain name represented in ASCII characters, as described in clause 8.8.5.2.
Long inactivity schedule	02_{16}	A sub-field indicating long inactivity schedules, as described in clause 8.8.5.3.
Short inactivity schedule	03_{16}	A sub-field indicating short inactivity schedules, as described in clause 8.8.5.4.
PSD-related domain Info	04_{16}	A sub-field carrying PSD-related domain information, as described in clause 8.8.5.5.

Table 8-73 – Types of auxiliary information sub-fields

Type	Value	Description
New domain master ID	05_{16}	A sub-field carrying the DEVICE_ID and the REGID of the node that will take the role of the domain master after the handover is complete, as described in clause 8.8.5.6.
Backup domain master ID	06_{16}	A sub-field carrying the DEVICE_ID and the REGID of the node assigned as a backup domain master for the domain, as described in clause 8.8.5.7.
Timer-related domain info	07_{16}	A sub-field carrying timer-related domain information, as described in clause 8.8.5.8.
Reserved	08_{16}	Reserved by ITU-T.
Registration code	09_{16}	A sub-field indicating registration code to register nodes to which domain name cannot be provided by the user, as described in clause 8.8.5.9.
DOD update	$0A_{16}$	The new value of DOD.
Reserved	$0B_{16}$ to $7F_{16}$	Reserved by ITU-T.

8.8.5.1 Reserved field

This field is reserved for future use by ITU-T.

8.8.5.2 Domain name sub-field

The format of the domain name sub-field shall be as presented in Table 8-74. The length of the sub-field data is 32 octets.

Table 8-74 – Format of domain name sub-field

Field	Octet	Bits	Description
Type	0	[6:0]	Set to 01_{16}.
ModificationFlag		[7]	This flag shall be set to zero.
Length	1	[7:0]	Set to 32_{10}.
Domain name	2 to 33	[255:0]	32-octet domain name represented in ASCII characters (Note).
NOTE – The ASCII characters shall be mapped onto the bytes of the domain name in the following way: – the LSB of the 7-bit ASCII character is mapped onto bit b0 of the corresponding byte of the domain name; – the MSB of all bytes shall be set to zero; – the first ASCII character of the domain name shall be mapped on the least significant byte of the domain name (e.g., if the domain name is "Network", the first ASCII character is letter "N" that shall be mapped at byte 0 of the domain name); – if the number of provided ASCII characters is less than 32, the rest of the domain name field bytes shall be set to 00_{16}.			

8.8.5.3 Long inactivity schedule sub-field

The format of the long inactivity schedule sub-field shall be as shown in Table 8-75. It can announce an inactivity schedule over multiple MAC cycles for up to M nodes (see clause 8.3.6.1.1). The length of the sub-field data is $6M$ octets. The value of M shall not exceed 42.

Table 8-75 – Format of long inactivity schedule sub-field

Field	Octet	Bits	Description
Type	0	[6:0]	Set to 02_{16}.
ModificationFlag		[7]	This flag may be set to either zero or one.
Length	1	[7:0]	Length of the sub-field data in octets (range from 6 to 6*M*).
LIS_ID of the first node	2	[7:0]	DEVICE_ID of the node scheduled for long inactivity.
LIS_TYPE of the first node	3	[1:0]	Type of long inactivity schedule for the first node: 00_2: Schedule is valid once. 01_2: Schedule is valid until it is changed. 10_2: Schedule is cancelled. If LIS_TYPE is set to 10_2, LIS_INACT_DUR and LIS_ACT_DUR shall be set to 0 indicating the node shall be active immediately.
Reserved		[7:2]	Reserved by ITU-T (Note).
LIS_INACT_DUR of the first node	4 and 5	[15:0]	Duration of the inactive period, expressed in 5 ms units, represented as a 16-bit unsigned integer. This value shall be larger than or equal to the length of one MAC cycle.
LIS_ACT_DUR of the first node	6 and 7	[15:0]	Duration of the active period that follows LIS_INACT_DUR, expressed in 5 ms units, represented as a 16-bit unsigned integer. This value shall be larger than or equal to the length of one MAC cycle. This field shall be set to 0, and ignored by the receiver if LIS_TYPE = 00_2.
…	…	…	…
LIS_ID of *M*-th node	6*M*–4	[7:0]	…
LIST_TYPE of *M*-th node	6*M*–3	[1:0]	…
Reserved		[7:2]	…
LIS_INACT_DUR of *M*-th node	(6*M*–2) and (6*M*–1)	[15:0]	…
LIS_ACT_DUR of *M*-th node	6*M* and (6*M*+1)	[15:0]	…
NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

8.8.5.4 Short inactivity schedule sub-field

The format of the short inactivity schedule sub-field shall be as presented in Table 8-76. It can advertise an inactivity schedule within a MAC cycle for up to *M* nodes across the 32 TXOPs in the MAC cycle (see clause 8.3.6.2). The values of *M* shall not exceed 51.

Table 8-76 – Format of short inactivity schedule sub-field

Field	Octet	Bits	Description
Type	0	[6:0]	Set to 03_{16}.
ModificationFlag		[7]	This flag may be set to either zero or one.
Length	1	[7:0]	Length of sub-field data in octets (range from 3 to 3*M*).
SIS_ID	2	[7:0]	DEVICE_ID of the node schedule for short inactivity.
SIS_TYPE of the first node	3	[2:0]	Type of short inactivity schedule for the first node: 0: Schedule is valid once. 1: Schedule is valid until it is changed. 2: Schedule is cancelled. If SIS_TYPE is set to 2, SIS_BMAP shall be set to FF_{16} indicating the node shall be active immediately.
Reserved		[7:3]	Reserved by ITU-T (Note).
SIS_BMAP	4	[7:0]	Bitmap of inactive status for eight equal portions of the MAC cycle. bit0 (LSB) corresponds to the first portion of the MAC cycle, and bit7 (MSB) corresponds to the last portion of the MAC cycle. A bit corresponding to a portion of the MAC cycle shall be set to one if the node is active in that portion, and set to zero otherwise. If a TXOP includes a change in node state according to the SIS_BMAP, the node shall be active for the whole TXOP.
…	…	…	…
SIS_ID of *M*-th node	3*M*–1	[7:0]	…
SIS_TYPE of *M*-th node	3*M*	[2:0]	…
Reserved		[7:3]	…
SIS_BMAP of *M*-th node	3*M*+1	[7:0]	…
NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

8.8.5.5 PSD-related domain info sub-field

The format of the PSD-related domain info sub-field shall be as presented in Table 8-77. The length of the sub-field data is variable.

Table 8-77 – Format of PSD-related domain info sub-field

Field	Octet	Bits	Description
Type	0	[6:0]	Set to 03_{16}.
ModificationFlag		[7]	This flag shall be set to one.
Length	1	[7:0]	Length of the field in octets (range 3-199).
DmVersion	2	[7:0]	0 – Domain master supports version 0 of ITU-T G.9960 and ITU-T G.9961. All other values of this field are reserved by ITU-T for indicating support for future versions of the Recommendation (Note 1).
Regional PSD shaping mask	3	[0]	0, when PSD shaping is not used. 1, when PSD shaping is used.
Regional SM		[1]	0, when sub-carrier masking is not used. 1, when sub-carrier masking is used.
Regional TX power limit		[2]	0, when standard transmit power limit is used (see clause 7.2.6 of [ITU-T G.9960]). 1, when TX power limit is used.
Regional Amateur radio bands		[3]	0, when all international Amateur radio bands are masked. 1, when one or more bands are not masked.
Reserved		[7:4]	Reserved by ITU-T (Note 2).
Amateur radio band descriptor	4 and 5	[9:0]	Zero octet if bit 3 of octet 3 is set to zero, otherwise represents a bit map representing usage of international amateur bands (0 = masked, 1 = unmasked). The LSB represents the lowest band (1.8-2.0 MHz), the second LSB represents the second lowest band (3.5-4.0 MHz), etc. Masked amateur bands are part of RMSC (see clause 7.1.4.2.1 of [ITU-T G.9960]).
Reserved		[15:10]	Reserved by ITU-T (Note 2).
TX power limit	6	[7:0]	Zero octet if bit 2 of octet 3 is set to zero, otherwise represents the value of maximum transmit power in dBm, represented as 0.1 dBm per unit.
PSD shaping descriptor	7 to (6+L)	[(8*L) – 1:0]	Zero octet if bit 0 of octet 3 is set to zero, otherwise see Table 8-78 (Note 3).
SM descriptor	(7+L) to (6+L+M)	[(8*M) – 1:0]	Zero octet if bit 1 of octet 3 is set to zero, otherwise see Table 8-79. Masked bands are part of RMSC (see clause 7.1.4.2.1 of [ITU-T G.9960]) (Note 4).

NOTE 1 – A domain master indicating support for a certain version of this Recommendation shall mean that it also supports all the earlier versions of the Recommendation.

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 value of L equals to the value of the first octet of the PSD shaping descriptor multiplied by 3 plus 1. The value of M equals to the value of the first octet of the SM descriptor multiplied by 3 plus 1.

NOTE 4 – The SM is intended to incorporate masked sub-carriers defined by the regional Annex to comply with local regulations and masked sub-carriers defined by the user or service provider to facilitate local deployment practices.

Table 8-78 – PSD shaping descriptor

Octet	Bits	Description
0	[4:0]	Number of breakpoints (B_p). The valid range of this field is 0 (B_p=1) to 31 (B_p=32).
	[7:5]	Reserved by ITU-T (Note 1).
1 to 3	[11:0]	Sub-carrier index of first breakpoint being described (Notes 2 and 4).
	[23:12]	PSD level on this sub-carrier in steps of 0.1dB with an offset of –140 dBm/Hz (Notes 3 and 4).
…	…	…
3*B_p–2 to 3*B_p	[11:0]	Sub-carrier index of last breakpoint being described (Notes 2 and 4).
	[23:12]	PSD level on this sub-carrier in steps of 0.1 dB with an offset of –140 dBm/Hz (Notes 3 and 4).
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 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. NOTE 3 – The dynamic range of the PSD level specified in this descriptor shall be 30 dB (see clause 7.1.5.3 of [ITU-T G.9960]). NOTE 4 – Example: A 3-octet field value of 320400_{16} represents a breakpoint with PSD of $320_{16}\times0.1 - 140 = -60$ dBm/Hz on a sub-carrier with index $400_{16} = 1024$.		

Table 8-79 – SM descriptor

Octet	Bits	Description
0	[4:0]	Number of bands to be masked (B_s). The valid range of this field is 0 (B_s=1) to 31 (B_s=32).
	[7:5]	Reserved by ITU-T (Note 1).
1 to 3	[11:0]	Index of the lowest frequency sub-carrier in the first band to be masked (Note 2).
	[23:12]	Index of the highest frequency sub-carrier in the first band to be masked (Note 2).
…	…	…
3*B_s–2 to 3*B_s	[11:0]	Index of the lowest frequency sub-carrier in the last band to be masked.
	[23:12]	Index of the highest frequency sub-carrier in the last band to be masked.
NOTE 1 – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver. NOTE 2 – Example: A 3-octet field value 400200_{16} describes a masked band started from sub-carrier $200_{16} = 512$ and ended by sub-carrier $400_{16} = 1024$.		

8.8.5.6 New domain master ID sub-field

The format of the domain master ID shall be as presented in Table 8-80. The length of the sub-field data is 7 octets. This field shall be only used during the handover (see clause 8.6.6.4).

Table 8-80 – Format of new domain master ID sub-field

Field	Octet	Bits	Description
Type	0	[6:0]	Set to 05_{16}.
ModificationFlag		[7]	This flag may be set to either zero or one.
Length	1	[7:0]	Shall be set to 7_{16}.
DM_DEVICE_ID	2	[7:0]	An 8-bit DEVICE_ID of a node that will take the role of the domain master after the handover is complete.
DM_REGID	3 to 8	[47:0]	A 48-bit REGID of the node that will take the role of the domain master after the handover is complete.

8.8.5.7 Backup domain master ID sub-field

The format of the domain master ID shall be as presented in Table 8-81. The length of the sub-field data is 7 octets. This field shall be transmitted as described in clause 8.6.5.

Table 8-81 – Format of backup domain master ID sub-field

Field	Octet	Bits	Description
Type	0	[6:0]	Set to 06_{16}
ModificationFlag		[7]	This flag may be set to either zero or one.
Length	1	[7:0]	Shall be set to 07_{16}.
DM_DEVICE_ID	2	[7:0]	The 8-bit DEVICE_ID of a node that is assigned as a backup of the acting domain master.
DM_REGID	3 to 8	[47:0]	The 48-bit REGID of a node that is assigned as a backup of the acting domain master.

8.8.5.8 Timer-related domain info sub-field

This sub-field indicates values for timers to be adopted by all nodes in the domain, as described in clause 8.6.4. The format of the sub-field shall be as presented in Table 8-82. The length of the sub-field data is 2 octets. This sub-field shall be sent in every MAP-D. In MAP-A, the sub-field shall be sent periodically, with a period determined by the domain master.

Table 8-82 – Format of timer-related domain info sub-field

Field	Octet	Bits	Description
Type	0	[6:0]	Set to 07_{16}.
ModificationFlag		[7]	This flag shall be set to one.
Length	1	[7:0]	Shall be set to 03_{16}.
Topology update interval	2 and 3	[15:0]	Indicates the topology update interval that shall be accommodated by all nodes in the domain (see clause 8.8.5.8.1).
Re-registration time period	4	[5:0]	Time period for re-registration (see clause 8.6.1.1.2) in seconds represented as an unsigned integer with a step size of 2 seconds. The valid range is from 5 (10 s) to 63 (126 s).
Reserved		[7:6]	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.			

8.8.5.8.1 Topology update interval sub-field

This sub-field indicates the topology update interval to be adopted by all nodes in the domain, as described in clause 8.6.4. The format of the sub-field shall be as shown in Table 8-83. The length of the sub-field is 2 octets.

Table 8-83 – Format of topology update interval sub-field

Field	Octet	Bits	Description
Topology update interval	0	[7:0]	Minimum interval, $T_{MIN\_DISCOVER}$, in seconds represented as an unsigned integer.
	1	[7:0]	Maximum interval, $T_{MAX\_DISCOVER}$, in seconds represented as an unsigned integer.

8.8.5.9 Registration code sub-field

The format of the registration code sub-field shall be as presented in Table 8-84. The length of the sub-field data is 6 octets.

Table 8-84 – Format of registration code sub-field

Field	Octet	Bits	Description
Type	0	[6:0]	Set to 09_{16}.
ModificationFlag		[7]	This flag may be set to either zero or one.
Length	1	[7:0]	Set to 06_{16}.
Domain name	2 to 7	[47:0]	6-octet registration code.

8.8.5.10 DOD update sub-field

The format of the DOD update sub-field shall be as presented in Table 8-85. The length of the sub-field data is 1 octet.

Table 8-85 – Format of DOD update sub-field

Field	Octet	Bits	Description
Type	0	[6:0]	Set to $0A_{16}$.
ModificationFlag		[7]	This flag may be set to either zero or one.
Length	1	[7:0]	Set to 01_{16}.
NewDOD	2	[3:0]	New value of DOD.
Reserved		[7:4]	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.			

8.8.6 MAP schedule persistence publication

Schedule persistence shall be indicated in the MAP using the following counters:

– CSLT (current schedule life time): The validity of the currently applied schedule shall be CSLT+1 MAC cycles.

– FSLT (future schedule life time): The validity of the future schedule, to be applied right after the current schedule persistence period ends, shall be FSLT+1 MAC cycles.

The CSLT and FSLT counters shall only apply to persistent TXOPs (see clause 8.3.1.2). A single set of CSLT and FSLT counters shall be used for all the persistent TXOPs and is indicated in the MAP header (see MAP frame format in clause 8.8.3, Table 8-62).

To apply a persistent schedule, the domain master shall use the CSLT counter. CSLT is set to the desired duration of the persistence period in MAC cycles minus one. Once CSLT is set to a non-zero value, it shall not be decreased by more than one in every successive MAP.

If the domain master intends to continue with the current persistent schedule, it may keep or increase the validity of the currently applied persistent schedule by maintaining or increasing the value of the CSLT counter in subsequent MAP messages. FSLT shall be set to zero in this case. If the domain master intends to change the persistent schedule, it shall set the FSLT counter to a non-zero value. The CSLT counter shall then be decremented by one each MAC cycle and the current persistent schedule shall only be valid while the CSLT counter is greater than or equal to zero. Once FSLT is set to a non-zero value, the future schedule is published.

The life time of the future scheduling shall be indicated in the MAP via the FSLT counter. The future schedule and the FSLT value shall not be changed once they are published.

If FSLT is set to zero, the current persistent schedule is transmitted in the MAP, otherwise the future persistent schedule is transmitted in the MAP.

NOTE – A node that just receives its first MAP, and the MAP contains a non-zero FSLT value, is not aware of the current persistent schedule. When the FSLT counter is greater than zero and the CSLT counter is zero in the MAP transmitted in MAC cycle N–1, the CSLT counter in the MAP transmitted in MAC cycle N shall be set to the value of the FSLT counter that was transmitted in the MAP for MAC cycle N–1. The FSLT value in the MAP transmitted in MAC cycle N shall be zero. The future schedule shall take effect and become the current schedule in MAC cycle N+1 as illustrated in Figure 8-44. The change between the currently applied schedule and future schedule shall be with no interruption.

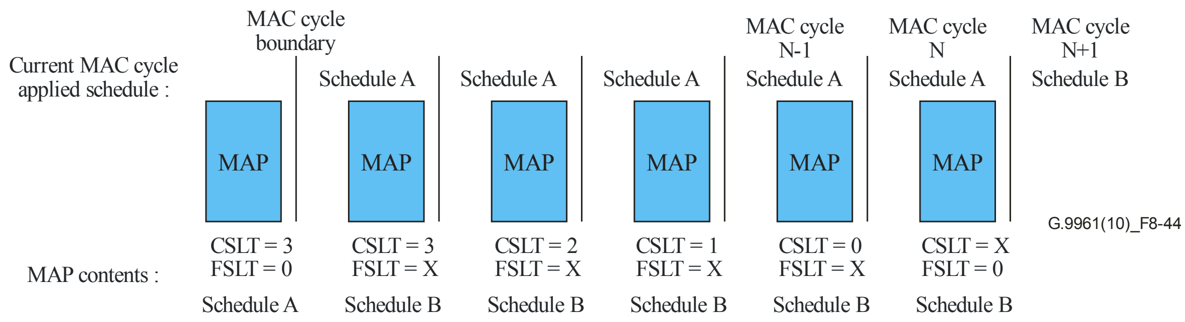


Figure 8-44 – Example of a persistent schedule switch

The domain master shall not change a persistent schedule until the persistence period expires.

When the domain master broadcasts a new future schedule, it shall set the FSLT counter with the intended duration of the persistence period minus one. From that point, the MAP shall include the new future schedule in addition to the current non-persistent schedule.

8.9 Retransmission and acknowledgement protocol

The retransmission and acknowledgment protocol specifies acknowledgment (either immediate or delayed) by the receiver of the reception of a frame.

The transmitter shall indicate in the header of the transmitted frame, via the RPRQ field (see clause 7.1.2.3.2.2.13 of [ITU-T G.9960]), whether Imm-ACK, delayed-ACK, or no acknowledgement is required.

8.9.1 Acknowledgment for a unicast PHY frame

8.9.1.1 Immediate acknowledgment

When Imm-Ack is required for a unicast frame (MI set to zero), the acknowledging node shall follow the reception of a frame with an Imm-ACK frame as specified in clause 7.1.2.3.8.2 of [ITU-T G.9960], T_{AIFG} or $T_{AIFG\text{-}D}$ (see clause 8.4) after the end of the frame that has requested the Imm-ACK. The transmitter indicates usage of either T_{AIFG} or $T_{AIFG\text{-}D}$ by using the AIFG_IND bit in the PHY-frame header (see clause 7.1.2.3.2.2.16 of [ITU-T G.9960]).

A gap of $T_{AIFG\text{-}D}$ shall only be used by the transmitter, 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 transmitted frame includes less than MIN_SYM_VAR_AIFG payload symbols. In all other cases the gap shall be $T_{AIFG.}$ The parameter MIN_SYM_VAR_AIFG is defined in clause 8.4, for each media.

All nodes in the domain shall refrain from transmission when Imm-ACK is expected and within T_{IFG_MIN} following it.

The sender shall plan its transmission so that the Imm-ACK that follows a frame is contained within the TXOP or TS assigned for the transmission.

8.9.1.2 Delayed acknowledgment

If delayed-ACK is required, the transmission of the acknowledgement shall be deferred to a TXOP or TS assigned to the receiver unless an Imm-ACK request is received prior to transmission of the delayed-ACK. If an Imm-ACK request is received prior to transmission of the delayed-ACK, the deferred acknowledgement shall be sent in the requested Imm-ACK.

If the delayed-ACK is sent in a TXOP or TS assigned to the receiver:

- If the delayed-ACK refers to a prioritized data connection, the ACK PHY frame shall be considered as having an MPDU priority equal to the highest user priority mapped to the prioritized data connection it refers.
- If the delayed-ACK refers to a data connection associated with a service flow or to the management connection, the ACK PHY frame shall be considered as having an MPDU priority equal to 7.

8.9.2 Acknowledgment for multicast PHY frames

8.9.2.1 Multicast acknowledgement overview

With multicast acknowledgement, a frame addressed to a group of nodes is acknowledged by one or more nodes of the group using acknowledgment frames that are transmitted in predefined time slots that immediately follow the frame requesting acknowledgement response. Each Mc-ACK frame slot is uniquely assigned to a single destination node from the multicast group that acknowledges the multicast frame. In addition, a NACK signalling time slot may follow Mc-ACK slots, if requested by the sender, and in this case all destination nodes of the multicast group that are not assigned a Mc-ACK slot in which to respond, shall indicate reception failure by transmitting a NACK signal in the NACK signalling slot.

The reception failure shall be declared if one or more errors were detected in those LPDUs of the received frame that were not received correctly in the previous transmissions; if errors are detected only in the LPDUs that were already received correctly in previous transmissions, the frame should be considered as received correctly.

All destination nodes of the multicast group that received at least one Mc-ACK frame but did not receive the original multicast frame corresponding to this Mc-ACK frame, shall send NACK signal in the NACK signalling slot.

The assignment of the node(s) that shall transmit acknowledgement and corresponding Mc-ACK frame slot(s) is communicated to all the nodes of the multicast group through the multicast binding protocol (clause 8.16). The NUM_MCACK_SLOTS field in the PHY-frame header indicates the number of Mc-ACK slots that follow the transmitted frame.

NOTE – The NUM_MCACK_SLOTS field is useful for virtual carrier sensing.

The source of the multicast transmission determines the number of acknowledging nodes and assigns Mc-ACK slot(s), and determines whether NACK signalling shall be used or not. The method for determining these selections is outside the scope of this Recommendation.

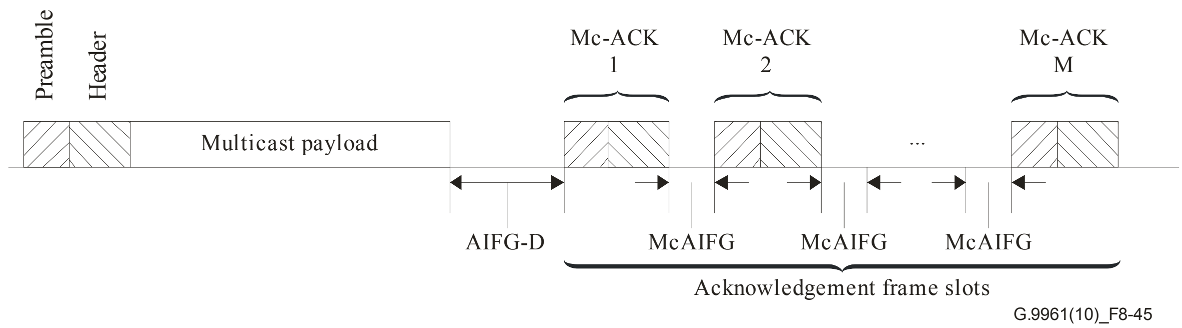


Figure 8-45 – Multicast acknowledgment without NACK signalling

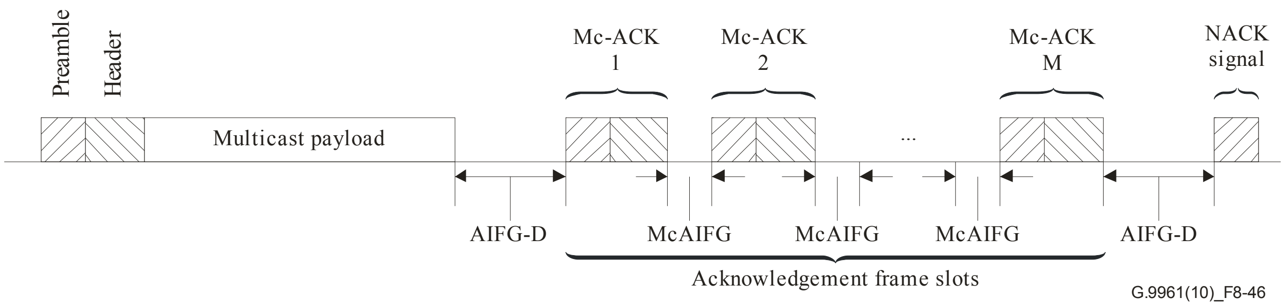


Figure 8-46 – Multicast acknowledgment with NACK signalling

8.9.2.2 Multicast acknowledgement procedure

Nodes sending multicast frames intended for acknowledgement shall set the multicast indication (MI) field to one, and set the reply required (RPRQ) field to indicate the specific type of acknowledgement to be used (see Table 8-86). The type of acknowledgement depends on whether NACK signalling is to be used or not (see the RPRQ field of the PHY-frame header in clause 7.1.2.3.2.2.13 of [ITU-T G.9960]). The number of acknowledgement slots shall not exceed MAX_ARQ_SLOTS (see clause 8.4).

The duration of each Mc-ACK slot, $T_{Mc\text{-}ACK}$, is the time required to transmit one Mc-ACK frame, and it shall be the same as the duration of an ACK frame. The duration of the NACK signalling slot, T_{NACK}, is the time required to transmit a NACK signal (see clause 7.1.4.5.3.1.2 of [ITU-T G.9960]). There shall be an IFG with a duration of $T_{AIFG\text{-}D}$ assigned between the multicast frame being acknowledged and the first Mc-ACK slot, an IFG with a duration of T_{McAIFG} assigned between adjacent Mc-ACK slots, and an IFG with a duration of $T_{AIFG\text{-}D}$ between the last Mc-ACK slot and the NACK slot. The acknowledging nodes shall transmit Mc-ACK frames and NACK signals within TX_ON microseconds from the start of the Mc-ACK and NACK slots, respectively. The duration of the McAIFG, T_{McAIFG}, the duration of AIFG used in this case, $T_{AIFG\text{-}D}$, and the value of TX_ON are medium-dependent and are defined in clause 8.4.

Nodes that correctly received the PHY-frame header of a multicast frame requesting acknowledgement shall defer from transmitting, except for transmitting a Mc-ACK frame or a

NACK signal, as described in this clause, for the duration of the whole multicast frame sequence, which equals:

$T_{sequence} = T_{frame} + T_{AIFG\text{-}D} + M \times T_{Mc\text{-}ACK} + (M-1) \times T_{McAIFG}$ (if NACK signalling is not used) (see Figure 8-45),

$T_{sequence} = T_{frame} + T_{AIFG\text{-}D} + M \times T_{Mc\text{-}ACK} + (M-1) \times T_{McAIFG} + T_{AIFG\text{-}D} + T_{NACK}$, if NACK signalling is used (see Figure 8-46),

where T_{frame} is the duration of the multicast frame and M is the number of nodes assigned for Mc-ACK.

Table 8-86 summarizes the types of Mc-ACK depending on RPRQ settings (see also Table 7-11 of [ITU-T G.9960]).

Table 8-86 – Types of multicast acknowledgement

RPRQ value	NUM_MCACK_SLOTS value	ARQ mechanism
01	Number of Mc-ACK slots	Acknowledgement with a slot assignment using the multicast group binding mechanism; no NACK signalling. This mode shall only be used if each receiving node in the multicast group is assigned a Mc-ACK slot.
11	Number of Mc-ACK slots	Acknowledgement with a slot assignment using the multicast group binding mechanism. All receivers in the multicast group not assigned an acknowledgement slot that fail to receive the transmission by criteria described in clause 8.9.2.1 shall transmit a NACK in the NACK signalling slot.

The Mc-ACK frame shall use the following assignments in the PHY-frame header:

– The MI field shall be set to one and DID field shall be set to the value of multicast ID of the multicast group.

– The MCACK_D field shall be set as defined in clause 7.1.2.3.2.3.9.2 of [ITU-T G.9960], indicating the number of Mc-ACK slots assigned after this Mc-ACK slot, and whether NACK has to be sent or not by nodes that did not receive the original multicast frame.

– All other fields of the Mc-ACK frame shall be set the same as in the ACK frame for unicast acknowledgement.

8.9.3 Request for ACK retransmission

The transmitter may request the receiver to retransmit an ACK for a certain connection or for the management connection or both according to the RX_WIN_TYPE field (see clause 7.1.2.3.2.8.1 of [ITU-T G.9960]).

The destination node shall transmit an ACK frame informing the current status of the requested receiver window according to the RX_WIN_TYPE field (see clause 7.1.2.3.2.8.1 of [ITU-T G.9960]) of either the connection specified in the CONNECTION_ID field or of the management connection, or both from that sender (identified by SID).

The duration of the AIFG between an ACKRQ frame and the following ACK frame shall be $T_{AIFG\text{-}D}$.

8.9.4 Acknowledgement protocol parameters

8.9.4.1 General parameters

ACK_MAX_WINDOW_SIZE represents the maximum possible size of the transmission and reception windows (see clauses 8.9.4.2 and 8.9.4.3). The value of ACK_MAX_WINDOW_SIZE shall be 376 for data connections with LPDUs of size 540 bytes and 564 for data connections with LPDUs of size 120 bytes (see Table 7-21 of [ITU-T G.9960]), 16 for management connections with LPDUs of size 540 bytes, and 32 for management connections with LPDUs or size 120 bytes.

8.9.4.2 Transmitter variables and control flags

The transmission window is formed by the segments that are eligible for transmission; each segment is identified by its SSN.

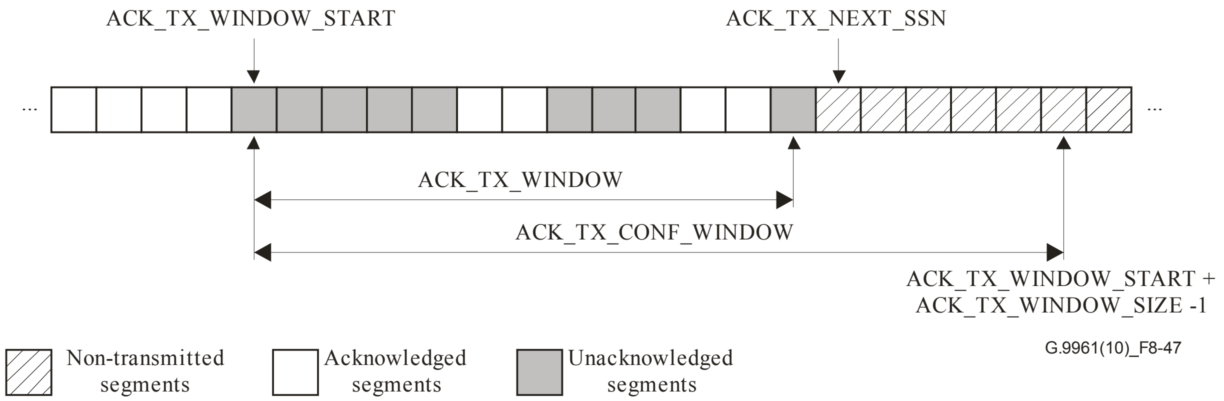


Figure 8-47 – Transmission window

ACK_TX_WINDOW_START is the SSN of the oldest unacknowledged segment: all segments with SSNs up to (ACK_TX_WINDOW_START – 1) have been acknowledged. A segment is called unacknowledged if it has been transmitted but no positive acknowledgement has been received.

ACK_TX_CONF_WINDOW is the maximum range of SSNs corresponding to segments the transmitter is permitted to send. This range is defined by ACK_TX_WINDOW_START and ACK_TX_CONF_WINDOW_SIZE as shown in Figure 8-47. ACK_TX_CONF_WINDOW_SIZE is a parameter that depends on the connection and shall be initialized as described in clause 8.12. ACK_TX_CONF_WINDOW_SIZE shall not exceed ACK_MAX_WINDOW_SIZE.

ACK_TX_WINDOW is the range of SSNs between the oldest unacknowledged segment and the newest unacknowledged segment, inclusive. This range is defined by ACK_TX_WINDOW_START and ACK_TX_NEXT_SSN, as shown in Figure 8-47, and may contain acknowledged and unacknowledged segments. The run-time size of the ACK_TX_WINDOW is ACK_TX_NEXT_SSN – ACK_TX_WINDOW_START.

ACK_TX_NEXT_SSN is the SSN of the next segment to send. This value shall belong to the interval ACK_TX_WINDOW_START to (ACK_TX_WINDOW_START + ACK_TX_CONF_WINDOW_SIZE), inclusive.

ACK_BLOCK_LIFETIME is the maximum time interval a segment shall be kept in the ACK_TX_WINDOW after this segment was transmitted the first time. If the segment is not acknowledged by the receiver within ACK_BLOCK_LIFETIME, the segment shall be discarded. Multiple retransmissions are allowed during this time.

NOTE – The value of ACK_BLOCK_LIFETIME may affect the latency and jitter of a flow. When selecting a value for it, implementers should take into account the delay and delivery (effect of losing LPDUs) requirements of the flow associated with the connection.

ACK_TX_RESET is the transmission window reset flag. When set to one, the transmitter state machine is in TX_RESET state and no segments shall be transmitted. When set to zero, the transmitter state machine is not in TX_RESET state and segments may be transmitted.

8.9.4.3 Receiver variables and control flags

The reception window is formed by the segments that can be accepted in the receiver to wait for retransmission of missing segments.

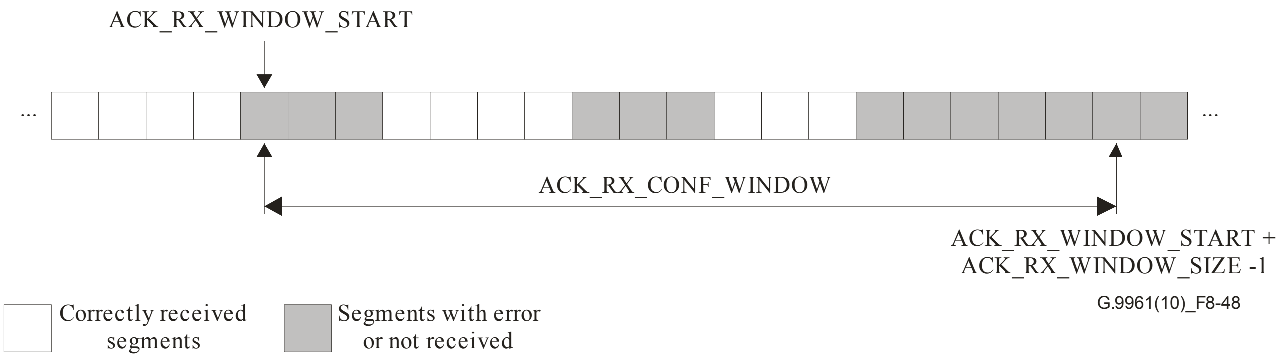


Figure 8-48 – Reception window

ACK_RX_WINDOW_START is the SSN of the oldest segment received in error or not received: All segments with SSNs up to (ACK_RX_WINDOW_START – 1) have been received correctly or have been discarded by the transmitter.

ACK_RX_CONF_WINDOW is the maximum range of SSNs corresponding to segments that the receiver is expecting to receive and accept. This range is defined by ACK_RX_WINDOW_START and ACK_RX_CONF_WINDOW_SIZE as shown in Figure 8-48. ACK_RX_CONF_WINDOW_SIZE shall be greater than or equal to the number of segments that the receiver can buffer for a connection as described in clause 8.12. ACK_RX_CONF_WINDOW_SIZE shall not exceed ACK_MAX_WINDOW_SIZE.

ACK_RX_RESET is the reception window reset flag. When set to one, the receiver state machine is not in RX_WIN_SYNC state and received segments shall be discarded. When set to zero, the received segments may be accepted.

8.9.5 Acknowledgement protocol operation

8.9.5.1 SSN comparison

Acknowledgement protocol operation include comparing SSN and taking actions based on which is larger or smaller. The sequence of SSNs presents a circular behaviour. In the following clauses, it is assumed that the SSNs are normalized prior to their comparison with respect to the appropriate state machine reference value, using the following equation:

$$SSN' = \begin{cases} SSN - SSN_{ref} & , \left(SSN - SSN_{ref}\right) \geq 0 \\ SSN - SSN_{ref} + ACK\_SSN\_MODULUS, & \left(SSN - SSN_{ref}\right) < 0 \end{cases}$$

The reference values for the transmission and reception window operation (SSN_{ref}) are ACK_TX_WINDOW_START and ACK_RX_WINDOW_START, respectively.

Besides, the acknowledgement protocol operation includes comparisons of the transmitter variables (clause 8.9.4.2) with acknowledgement information sent by the receiver that corresponds to the *N* LSB bits of ACK_RX_WINDOW_START; i.e., LSSN (see clause 7.1.2.3.2.3.9.1.6 of [ITU-T G.9960]) and MNMT_LSSN (clause 7.1.2.3.2.3.9.1.3.1 of [ITU-T G.9960]). *N* is equal to 12 for data connections and to six for management connections. In the following clauses, it is assumed that all the magnitudes included in those comparisons are normalized, prior to their comparison, with respect to the appropriate reference value by using the following equation:

$$X' = \begin{cases} [X]_N - [SSN_{ref}]_N \quad , & ([X]_N - [SSN_{ref}]_N) \geq 0 \\ [X]_N - [SSN_{ref}]_N + 2^N, & ([X]_N - [SSN_{ref}]_N) < 0 \end{cases}$$

NOTE 1 – $[x]_N$ represents the operation of taking the N LSB bits of x. The above equation is equivalent to $X' = (X - SSN_{ref}) \bmod 2^N$; where "mod" represents the modulus operation.

NOTE 2 – SSN' or X' corresponding to the SSN equal to ACK_TX_WINDOW_START or ACK_RX_WINDOW_START is always equal to 0.

8.9.5.2 Segment lifecycle

Figure 8-49 shows the segment lifecycle. A segment may be in one of the following five states: not-sent, waiting-for-ack, waiting-for-retransmission, discarded and done. Initially, all segments are in a not-sent state. Once a segment is sent, it transitions to waiting-for-ack state. The segment state changes to waiting-for-retransmission upon reception of a negative acknowledgement (retransmission request). If the transmitter infers that the acknowledgement is lost or not sent, it may request the retransmission of the acknowledgement, by sending an ACKRQ frame (see clause 8.9.3) or requesting an Imm-ACK (when using delayed-ACK, see clause 8.9.1.2), or it may also retransmit a segment without having received a negative acknowledgement. The criterion for this decision is vendor discretionary. After being unacknowledged during the time longer than ACK_BLOCK_LIFETIME, the segment shall be discarded. Acknowledged segments are marked as done.

NOTE – Inferring that the acknowledgement is lost or not sent may be based on the opportunities that the receiver had to send it (in the requested Imm-ACK or based on the TXOP allocations for the receiver in case of delayed-ACK).

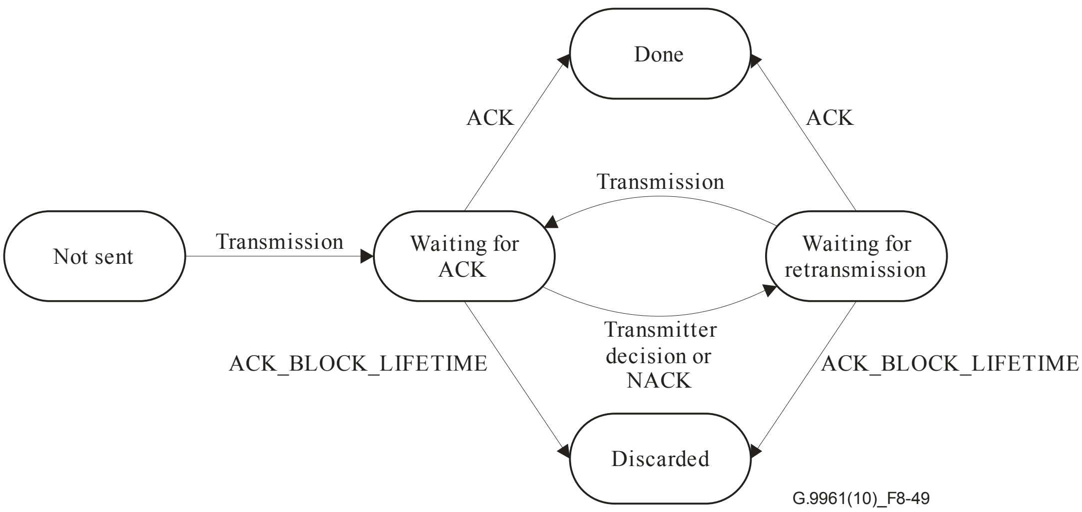


Figure 8-49 – Segment lifecycle

Segments in waiting-for-ack and in discarded or done states shall not be transmitted. When segments of the same connection are retransmitted, the segment with the lowest SSN shall be retransmitted first.

As each not-sent segment is mapped into an LPDU, it shall be assigned the current value of ACK_TX_NEXT_SSN, which shall be then incremented by 1.

8.9.5.3 Acknowledgement protocol state machine for unicast transmission

The protocol to be used between nodes to facilitate unicast transmission with acknowledgements is initialized as presented in Figure 8-50 (which shows the case where no transmissions have been lost) and Figure 8-51 (which shows an example of a case where some transmissions have been lost). The procedure includes the establishment of the connection as defined in clause 8.12. The initialization is based on the exchange of ACK_TX_RESET and ACK_RX_RESET flags. ACK_TX_RESET is sent in the PHY-frame header of the MSG frame (see clause 7.1.2.3.2.2.18 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 and 7.1.2.3.2.3.6 of [ITU-T G.9960]) according to clause 8.9.1.1 or clause 8.9.1.2.

A transmitting node may be in any one of the following states: TX_RESET, TX_WAIT_SYNC or TX_WIN_SYNC. A receiving node may be in any one of the following states: RX_RESET, RX_WAIT_SYNC or RX_WIN_SYNC.

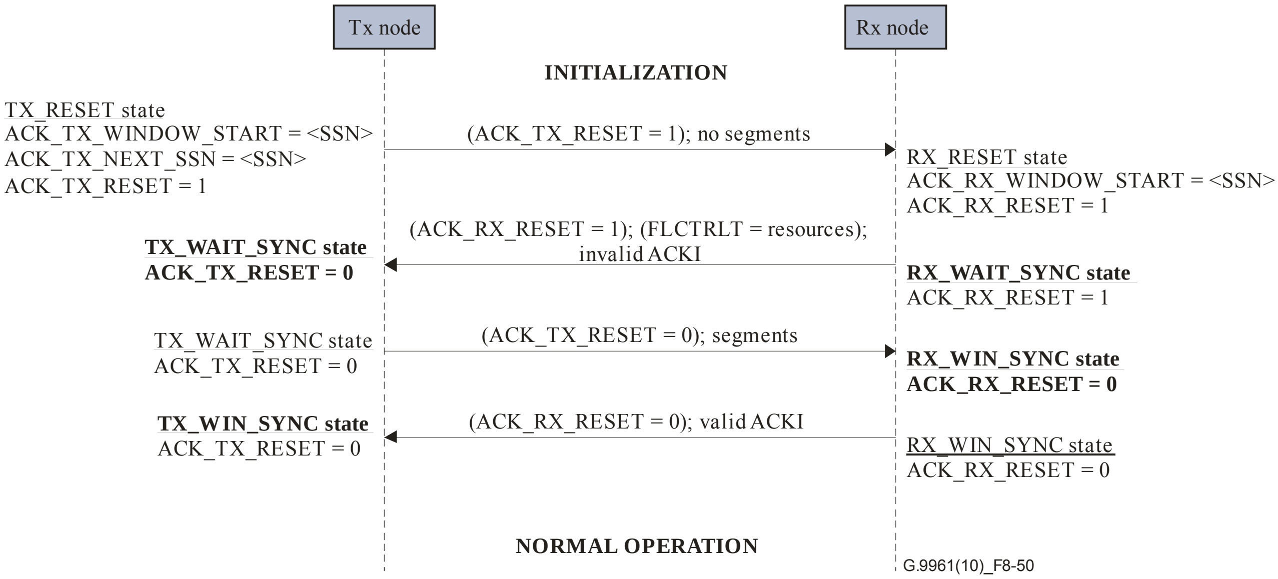


Figure 8-50 – Initialization of the acknowledgement protocol

First, the transmitting and receiving nodes state machines shall be in reset state (TX_RESET and RX_RESET). The flag ACK_TX_RESET = 1 shall be transmitted prior to the PHY frame carrying the first data segment of the established connection. This flag indicates that the transmitting node is in TX_RESET state. In TX_RESET state, ACK_TX_WINDOW_START and ACK_TX_NEXT_SSN shall be an arbitrary <SSN> value set by the transmitter.

Upon reception of ACK_TX_RESET = 1 in any state, the receiver shall reset its ARQ state machine and reply with the flag ACK_RX_RESET = 1 and shall indicate the availability of resources (see clause 8.12). This flag indicates that the receiver is in RX_RESET state. In RX_RESET state, ACK_RX_WINDOW_START shall be set to the <SSN> value specified by the transmitter in the START_SSN field. After sending the flag, if a status report was indicated in the flow control information (see clause 8.12) the receiving node shall transition to RX_WAIT_ SYNC state. Otherwise, the receiver shall remain in RX_RESET state.

Segments of the established connection shall not be sent while the transmitting node is in TX_RESET state.

Once in TX_RESET state, if the receiver indicated the availability of resources (see clause 8.12) and after receiving the flag ACK_RX_RESET = 1, the transmitter shall set the flag ACK_TX_RESET to zero and transition into the TX_WAIT_SYNC state. Segments of the established connection may be sent in TX_WAIT_SYNC state.

If in TX_RESET state the transmitter does not receive the ACK_RX_RESET = 1 in the requested Imm-ACK, the transmitter shall resend ACK_TX_RESET = 1. If the receiver signalled a hold time, the transmitter shall wait that time before resending the PHY frame with ACK_TX_RESET = 1. If the receiver indicated the unavailability of resources (see clause 8.12), the transmitter shall remain in the TX_RESET state keeping ACK_TX_RESET = 1. Then, the initialization of the acknowledgement protocol for that connection cannot be completed.

If in TX_RESET state the transmitting node receives an ACK frame with ACK_RX_RESET = 0, the transmitter shall ignore this ACK frame and resend ACK_TX_RESET = 1.

After resending two times ACK_TX_RESET = 1 in TX_RESET state, the segments of the established connection shall be discarded and the initialization of the acknowledgement protocol for the connection cannot be completed.

If the receiving node receives ACK_TX_RESET = 0 while being in RX_WAIT_SYNC state, it shall set the flag ACK_RX_RESET to zero, process the segments included in the PHY frame as described in clause 8.9.5.3.2, transition into RX_WIN_SYNC state and send ACK_RX_RESET = 0 to the transmitter.

The transmitting node shall transition from TX_WAIT_SYNC state into TX_WIN_SYNC state after the reception of an ACK frame with ACK_RX_RESET = 0. The transmitter shall process the ACK information as described in clause 8.9.5.3.1.

If in TX_WAIT_SYNC state the transmitting node does not receive the ACK_RX_RESET = 0 in the requested Imm-ACK or after inferring that the acknowledgement is lost or not sent (see clause 8.9.5.2), the transmitter shall resend ACK_TX_RESET = 0.

If in TX_WAIT_SYNC state the transmitting node receives ACK_RX_RESET = 1 with a status report in the flow control information, it shall resend the PHY frame with ACK_TX_RESET = 0. If the flow control information contains a valid hold time (see clause 8.12), the transmitter shall wait that time before resending the PHY frame with ACK_TX_RESET = 0.

After resending two times ACK_TX_RESET = 0 in TX_WAIT_SYNC state, the segments of the established connection shall be discarded and the initialization of the acknowledgement protocol for the connection cannot be completed.

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.

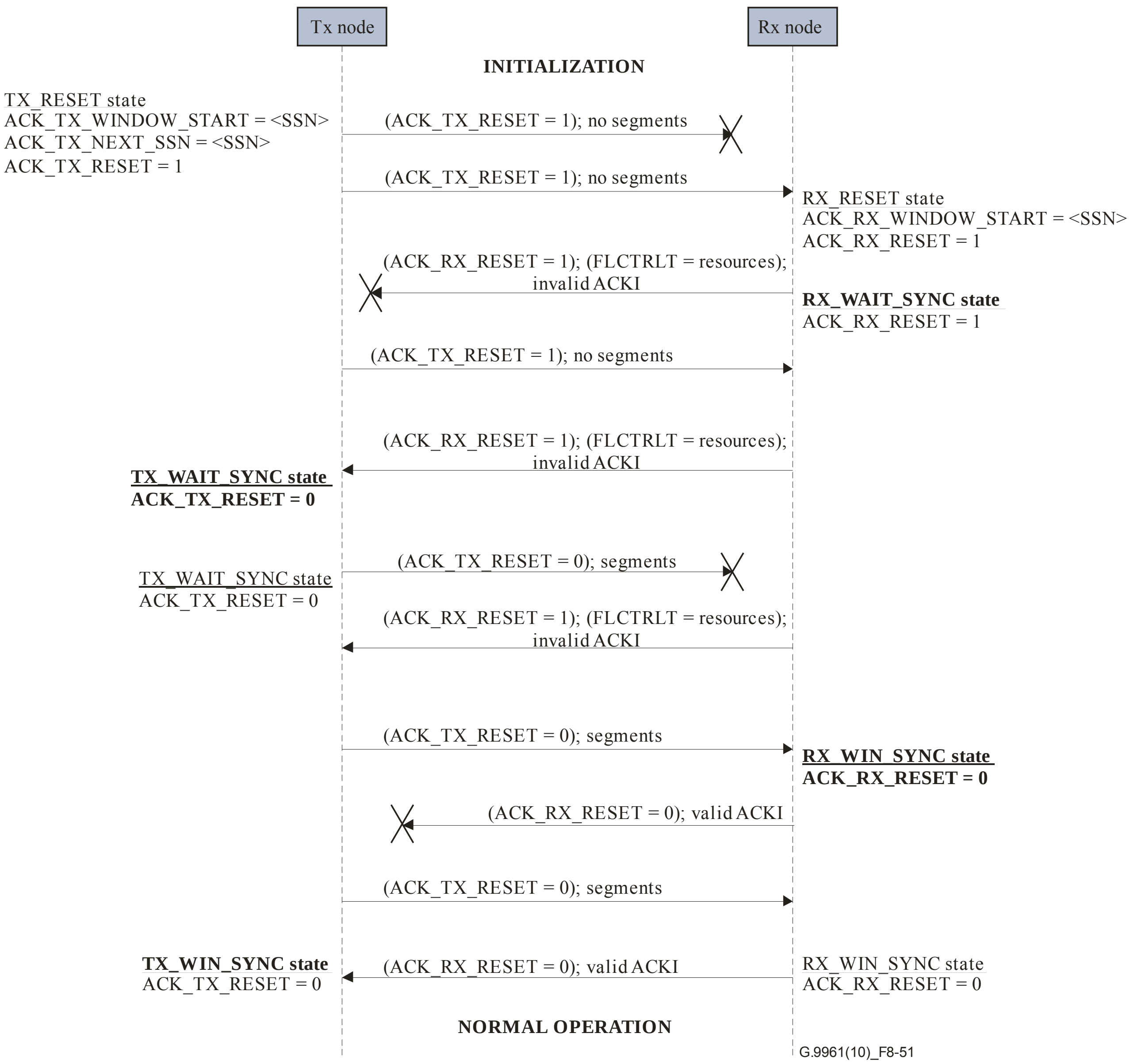


Figure 8-51 – Detailed initialization of the Acknowledgement protocol

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 coveys 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 of [ITU-T G.9960]) and MNMT_LSSN (clause 7.1.2.3.2.3.9.1.3.1 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 ≤ 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 to zero, see 7.1.2.3.2.3.9.1.7 of [ITU-T G.9960]).

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 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 ≤ 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 ≤ 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.

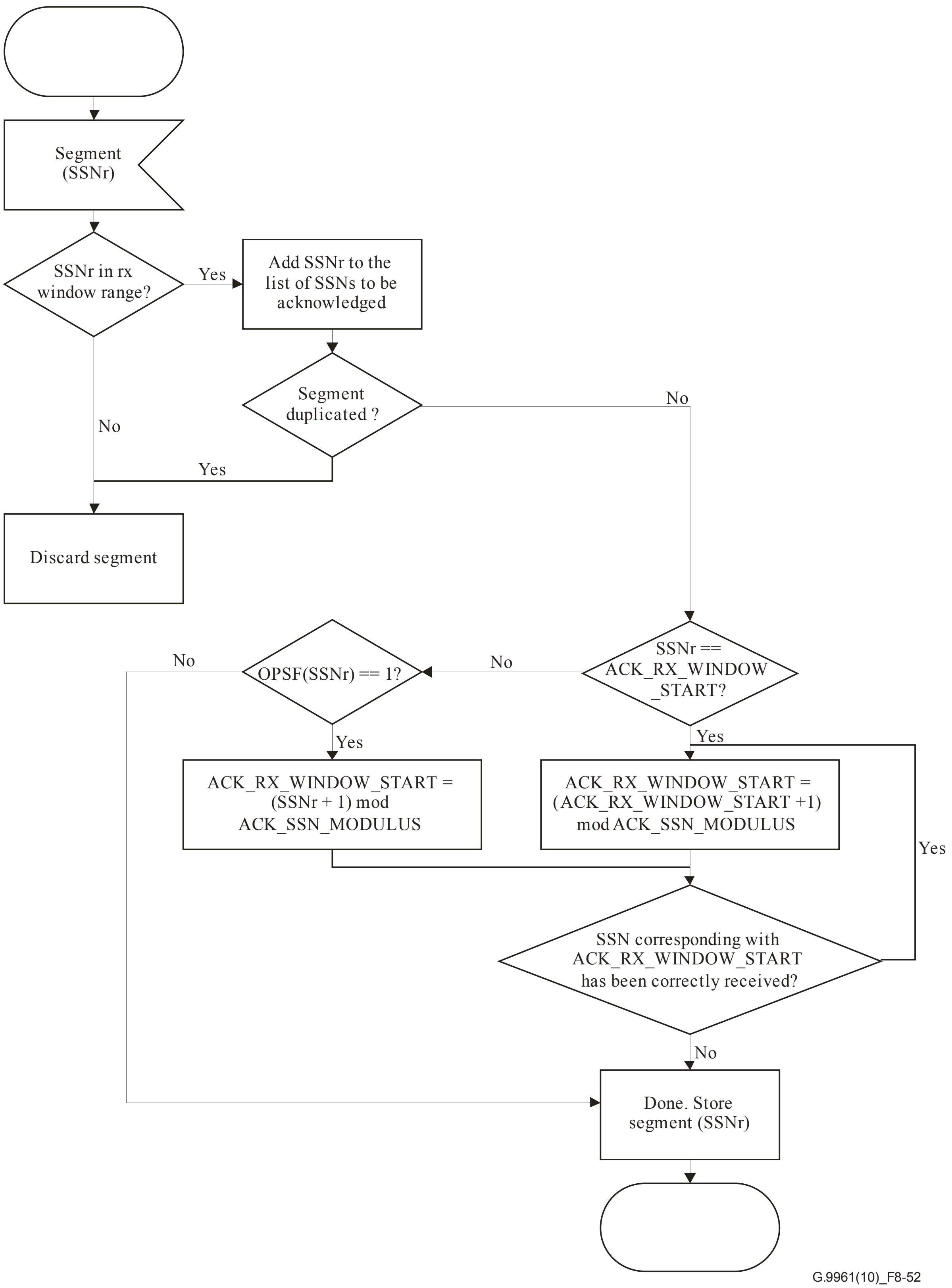


Figure 8-52 – Receiver SSN processing

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 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 and 7.1.2.3.2.3.6 of [ITU-T G.9960]) according to clause 8.9.2.

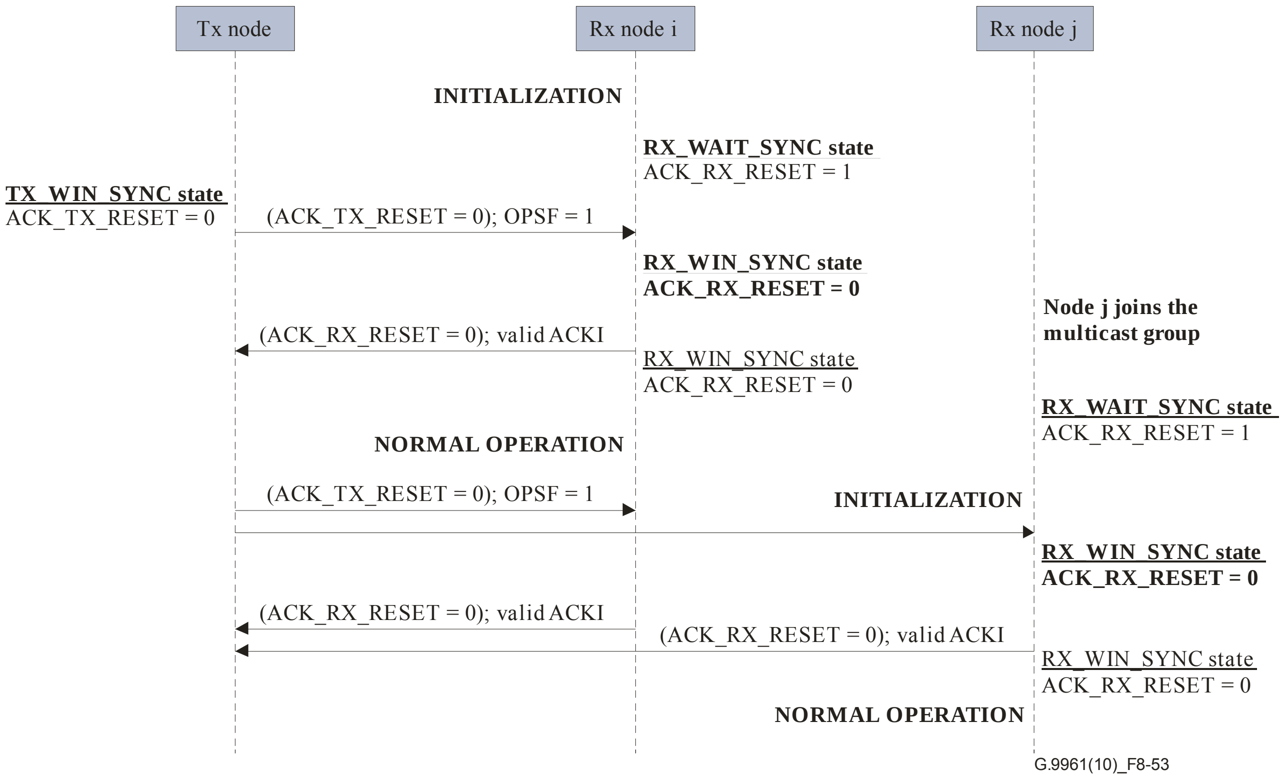


Figure 8-53 – Initialization of the acknowledgement protocol for multicast transmission

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_MCACK_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) may 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). The first byte (octet 0) of the MMH shall be the first byte of the LCDU payload, as described in clause 8.1.3.3. Encapsulation of the management message into an LLC frame is shown in Figure 8-7.

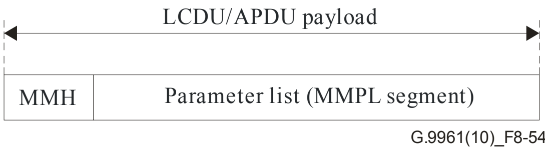


Figure 8-54 – Format of a management message

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 includes a list of management message parameters, depending on the management function. The format of any management message, except the MAP message, shall be as shown in Table 8-87. The format of the MAP message shall be as described in clause 8.8.

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 version of the Recommendation that they support 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).
	Reserved	3	[7:0]	Reserved by ITU-T (Note 2).
	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.
	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.
	Sequence number	5 and 6	[15:0]	Sequence number of the segmented message in a format of 16-bit unsigned integer (Note 3).
	Reserved	7	[7:0]	Reserved by ITU-T (Note 2).
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.
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 meaning of the sequence number depends on the OPCODE. See clause 8.10.1.2.

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
AKM (02X)	AUT_NodeRequest.req	020	Request for authentication	Clause 9.2.5.1.1
	AUT_Promp.ind	021	Delivers authentication prompt	Clause 9.2.5.1.2
	AUT_Verification.res	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
	AKM_KeyUpdate.req	027	Request for re-authentication and update the keys	Clause 9.2.5.3.1
	AKM_KeyAck.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_KeyAddRequest.req	02B	Request to join a node to a multicast group	Clause 9.2.5.2.1.1
Topology maintenance (03X)	TM_NodeTopologyChange.ind	030	Topology report from a node	Clause 8.6.4.2.1
	TM_DomainRoutingChange.ind	031	Optimal routing update from the domain master	Clause 8.6.4.3.5
	TM_ReturnDomainRouting.req	032	Request for routing update from the node to the domain master	Clause 8.6.4.3.6
	TM_ReturnDomainRouting.cnf	033	Reply on routing request by the Domain master	Clause 8.6.4.3.7
	TM_DMBackup.ind	034	Topology report from a node sent by backup domain master to a node	Clause 8.6.4.3.4

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]			
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
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
	DM_BackupRelease.cnf	068	Backup domain master release confirmation	Clause 8.6.5.2
Channel estimation (07X)	CE_ProbeSlotRequest.ind	070	Channel estimation bandwidth request	Clause 8.11.7.1
	CE_ProbeSlotRelease.ind	071	Channel estimation bandwidth release	Clause 8.11.7.2
	CE_ParamUpdate.ind	072	Channel estimation parameters update	Clause 8.11.7.3
	CE_ParamUpdateRequest.ind	073	Channel estimation parameter request	Clause 8.11.7.4
	CE_PartialBatUpdate.ind	074	BAT update indication	Clause 8.11.7.5
	CE_ACESymbols.ind	075	Request for an ACE symbol	Clause 8.11.7.6
Neighbouring networks coordination (08X)	For further study	For further study	For further study	For further study

Table 8-88 – OPCODEs of management messages

Category	Message name	OPCODE (hex)	Description	MMPL Reference
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
Flow establishment (0AX)	CL_EstablishFlow.req	0A0	Flow establishment request	Clause 8.6.2.3.1
	CL_EstablishFlow.cnf	0A1	Flow establishment confirmation	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_OriginateFlow.req	0A4	Flow origination request	Clause 8.6.2.3.6
	FL_OriginateFlow.cnf	0A5	Flow origination confirmation	Clause 8.6.2.3.7
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
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		Clause 8.6.2.3.13
	FL_TerminateFlow.cnf	0C4		Clause 8.6.2.3.14
Reserved	Reserved	0A0-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 management of the sequence numbers for a given OPCODE depends on the protocol associated with such OPCODE.

However, the following segmentation rules apply to any segmented LCDU:

- 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 LCDU is retransmitted, unless a new sequence number is generated.

Segmentation shall only be done for LCDUs with payload 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 it shall be incremented by one for each new message. It 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. 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.

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 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 and 7.1.2.3.3.2 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.

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)	[(8*V*–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 version of the Recommendation that they support 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 presented in Table 8-90. OPCODEs are categorized (typically by their associated protocol or procedure) according to the value of their six MSBs.

Table 8-90 – OPCODEs of control messages

Category	Message name	OPCODE hex	Description	CMPL reference
Channel estimation (01X)	CM_CE_Request.ind	010	Channel estimation request	Clause 8.11.8.1
	CM_CE_Initiation.req	011	Channel estimation initiation	Clause 8.11.8.2
	CM_CE_Initiation.cnf	012	Channel estimation confirmation	Clause 8.11.8.3
	CM_CE_ProbeRequest.ind	013	PROBE frame request	Clause 8.11.8.4
	CM_CE_Cancellation.ind	014	Channel estimation cancellation	Clause 8.11.8.5
	CM_BatIdMaintain.ind	015	BAT ID maintenance	Clause 8.11.8.6

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 the receiver. The core part of the channel estimation protocol 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 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.

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., probe 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 protocol provides numerous options to expedite the channel estimation process for faster channel adaptation.

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.

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 CM_CE_Initiation.req message (channel estimation initiation, see clause 8.11.1.1).
2) Upon reception of the channel estimation initiation, 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 CE_ProbeSlotRequest.ind 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 CM_CE_Initiation.cnf message (channel estimation confirmation, see clause 8.11.1.3).
4) Upon reception of the channel estimation confirmation, the receiver may request the transmitter to send additional PROBE frames by sending CM_CE_ProbeRequest.ind message (PROBE frame request, see clause 8.11.1.4).
5) Upon reception of the PROBE frame request, the transmitter shall transmit the PROBE frame as the receiver requested (PROBE frame transmission, see clause 8.11.1.5).

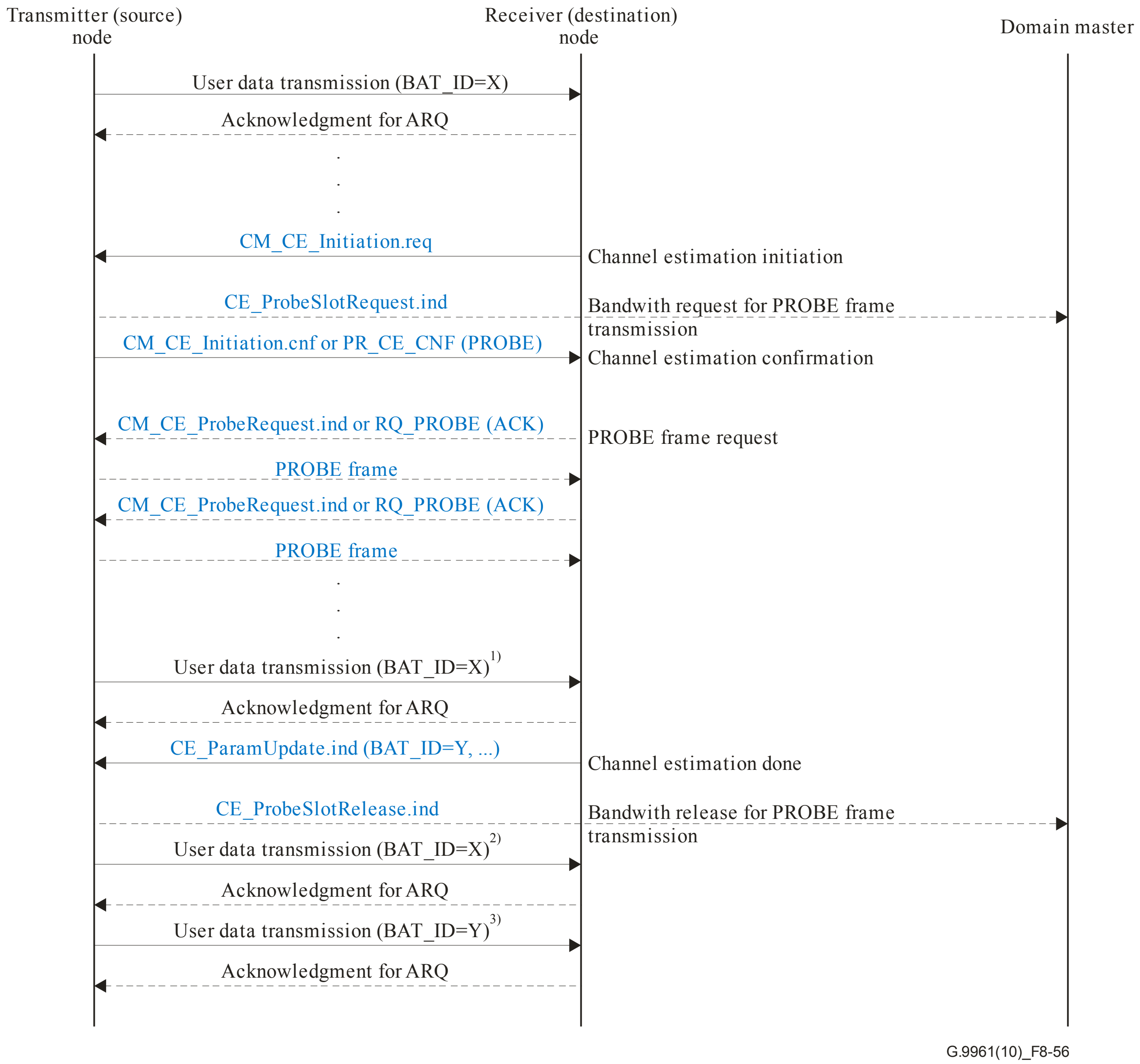
6) Steps 4 and 5 can repeat until the receiver sends the transmitter the final outcome of channel estimation using the CE_ParamUpdate.ind message (channel estimation completion, see clause 8.11.1.6). Steps 4 and 5 may be skipped altogether if the receiver does not need additional PROBE frames.

7) The receiver may cancel the channel estimation process after it receives the channel estimation confirmation by sending CM_CE_Cancellation.ind message or by setting a flag in an ACK frame (channel estimation cancellation, see clause 8.11.1.7).

8) Upon reception of CE_ParamUpdate.ind message, the transmitter may send CE_ProbeSlotRelease.ind message to the domain master to release the bandwidth used for PROBE frame transmission (bandwidth release, see clause 8.11.1.8).

The transmitter can send an MSG frame with the existing settings (e.g., old runtime BAT or predefined BAT) at any time during this process.

The receiver-initiated channel estimation process is illustrated in Figure 8-56.



[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 CM_CE_Initiation.req message. This message shall be carried using a CTMG frame.

The receiver shall select CE_GRP_MIN (G_{min}), which indicates the minimum 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 PROBE frame.

The receiver may resend the CM_CE_Initiation.req message if it does not receive a CM_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_ProbeSlotRequest.ind message.

The transmitter shall provide the domain master the channel estimation identifier (i.e., CE_BAT_ID), channel estimation window (CE_STIME and CE_ETIME), and PROBE frame parameters (CE_PRB_PARM) as provided by CM_CE_Initiation.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.

8.11.1.3 Channel estimation confirmation

The transmitter confirms the channel estimation process by sending the receiver CM_CE_Initiation.cnf message.

The transmitter shall indicate whether it grants or rejects the channel estimation initiation request by setting CE_CNF_TYPE and CE_CNF_CODE. The transmitter shall set CE_BAT_ID to the value selected by the receiver via channel estimation initiation. The transmitter shall finalize CE_GRP_MIN, which shall be larger than or equal to the one indicated by the receiver. The transmitter may use any value of G (sub-carrier grouping, see clause 7.1.4.2.4 of [ITU-T G.9960]) that satisfies the following conditions: $G(t_i) \geq G_{min}$, and $G(t_{i+1}) \geq G(t_i)$, where $G(t_i)$ denotes the value of G at arbitrary time t_i, and $t_i < t_{i+1}$. If the transmitter uses $G > G_{min}$, the new BAT (B') shall be formed by decimating the old BAT (B) by taking the minimum BAT entry from the original group of sub-carriers. That is, the new bit allocation entry for sub-carrier i, $B'_i = \min\{B_i\}$, where $i = G_j, G_{j+1}, \ldots, G_{j+G-1}$, and $j = 0, 1, \ldots, (N/G)-1$.

If the receiver has requested one or more PROBE frames in CM_CE_Initiation.req message, then the transmitter shall send a CM_CE_Initiation.cnf message over the first PROBE frame (i.e., CMPL of CM_CE_Initiation.cnf message is carried in PRB_CE_CNF field of PROBE frame as described in clause 7.1.2.3.2.7.6 of [ITU-T G.9960]). This PROBE frame shall contain the PROBE symbols as requested in CM_CE_Initiation.req message. If the receiver has not requested PROBE frames, the transmitter shall send CM_CE_Initiation.cnf message using a CTMG frame. The transmitter shall send CM_CE_Initiation.cnf message within 100 ms after it receives CM_CE_Initiation.req message.

8.11.1.4 PROBE frame request

Once channel estimation initiation is confirmed, the receiver may request the transmitter to send additional PROBE frames by sending CM_CE_ProbeRequest.ind message. This message shall be carried using a CTMG frame.

The receiver can request specific parameters of the PROBE frame via PROBE request parameter fields of CM_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 7.1.2.3.2.3.8 of [ITU-T G.9960]).

The receiver may not request PROBE frames at all if it uses MSG frames to estimate the channel.

8.11.1.5 PROBE frame transmission

Upon reception of the PROBE frame request, the transmitter shall transmit PROBE frames as the receiver requested.

If the receiver requests a PROBE frame via ACK_CE_CTRL (see clause 8.11.1.4), the transmitter shall transmit the PROBE frame using parameters previously selected by the receiver, that is the parameters selected in the latest PROBE frame request (CM_CE_ProbeRequest.ind) or channel estimation initiation (CM_CE_Initiation.req). The transmitter shall use the default parameters if the receiver has not previously indicated these parameters.

8.11.1.6 Channel estimation completion

At any time after channel estimation initiation is confirmed, the receiver may send the transmitter the outcome of channel estimation using the CE_ParamUpdate.ind message.

Upon reception of CE_ParamUpdate.ind message, the transmitter shall incorporate the new channel estimation parameters (new BAT, etc.) as soon as possible.

If the transmitter does not receive any frame or message related to channel estimation in the duration of 200 ms after channel estimation confirmation, it may send the receiver a CE_ParamUpdateRequest.ind message to request to resend the result of channel estimation.

8.11.1.7 Channel estimation cancellation

At any time after channel estimation initiation is confirmed, the receiver may cancel the channel estimation process using CM_CE_Cancellation.ind message. This message shall be carried using a CTMG frame.

Alternatively, the receiver may use the ACK_CE_CTRL_TYPE field in an ACK frame to indicate cancellation. In either 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.ind, the transmitter may request the domain master to release bandwidth previously assigned for PROBE frame transmission by sending CE_ProbeSlotRelease.ind message.

The transmitter shall provide the domain master the channel estimation identifier (i.e., CE_BAT_ID) and channel estimation window (CE_STIME and CE_ETIME) associated with the channel estimation process.

The domain master shall only release bandwidth additionally assigned to the transmitter for PROBE frame transmission over a given channel estimation window.

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 CM_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 can send an MSG frame with the existing settings (e.g., old 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 CM_CE_Request.ind message. This message shall be carried using a CTMG frame.

The transmitter may specify the channel estimation window (CE_STIME & CE_ETIME). In this case the receiver shall use the same channel estimation window as the transmitter requested. Otherwise, the receiver can determine the channel estimation window at its own discretion.

8.11.3 Unsolicited CE_ParamUpdate.ind

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.ind message, provided that the BAT_ID is invalid at the time of sending the new BAT and the number of valid BAT_IDs after adding this one is less than or equal to the MAX_NUM_BAT_ID value sent by the transmitter. The receiver may use MSG frames or PROBE frames transmitted to other nodes to estimate the channel.

8.11.4 Channel adaptation via 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 the BAT currently 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 8.11.4.1.1).
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. After receiving the first MSG frame using the N_BAT_ID, the receiver shall consider O_BAT_ID is invalid (see clause 8.11.5).
3) After transmitting the PBU request, the receiver shall infer loss of the PBU request if either of the following conditions is satisfied:
   a) Five MSG frames using the same O_BAT_ID are received from the transmitter;
   b) No MSG frame with N_BAT_ID is received from the transmitter in 100 ms and one MSG frame is received after this time with O_BAT_ID.
4) 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.

8.11.4.1 PBU request

8.11.4.1.1 PBU request via a management message

The receiver may send the PBU request using the management message CE_PartialBatUpdate.ind, 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 retransmission is disabled in this example.

Upon reception of CE_PartialBatUpdate.ind message, the transmitter should incorporate new channel estimation parameters as soon as possible and then send the CE_PartialBatUpdate.cnf message to confirm the received CE_PartialBatUpdate.req message.

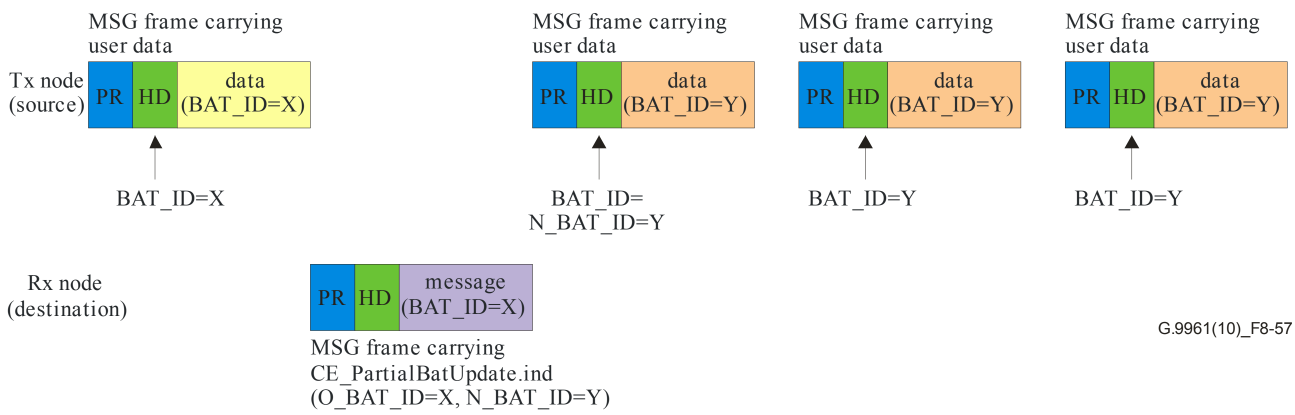


Figure 8-57 – Partial BAT update using management message

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 CM_BatIdMaintain.ind message carried using CTMG frame. The transmitter shall stop using BAT_IDs that are marked as invalid by the receiver as soon as possible. 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.ind), the transmitter shall send a CE_ParamUpdateRequest.ind message requesting the transmission of the BAT. The receiver can also invalidate a BAT_ID via the ACK_CE_CTRL field in the ACK frame (see clause 7.1.2.3.2.3.8 of [ITU-T G.9960]).

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) 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 MSG frames sent to the receiver. 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_ProbeSlotRequest.ind

The format of the MMPL of the CE_ProbeSlotRequest.ind message shall be as shown in Table 8-91.

Table 8-91 – Format of the MMPL of the CE_ProbeSlotRequest.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 updated by channel estimation. It shall be formatted as shown in Table 7-55 of [ITU-T G.9960].
Reserved		[7:5]	Reserved by ITU-T (Note).
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.

Table 8-91 – Format of the MMPL of the CE_ProbeSlotRequest.ind message

Field	Octet	Bits	Description
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_PARM	3 to 6	[23:0]	This field specifies a set of parameters for PROBE frame. It shall be coded as shown in Table 8-102.
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.ind

The format of the MMPL of the CE_ProbeSlotRelease.ind message shall be as shown in Table 8-92.

Table 8-92 – Format of the MMPL of the CE_ProbeSlotRelease.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 updated by channel estimation. It shall be formatted as shown in Table 7-55 of [ITU-T G.9960].
Reserved		[7:5]	Reserved by ITU-T (Note).
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.
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.ind

The format of the MMPL of the CE_ParamUpdate.ind message shall be as shown in Table 8-93.

Table 8-93 – Format of the MMPL of the CE_ParamUpdate.ind 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 7-55 of [ITU-T G.9960].
Bandplan ID		[7:5]	This field indicates the type of bandplan based on which the subsequent BAT entry is defined. It shall be formatted as shown in Table 7-10 of [ITU-T G.9960].
Minimum group ID	1	[2:0]	This field indicates the minimum GRP_ID (CE_GRP_MIN) associated with the new BAT (*G*), and determined during channel estimation confirmation. It shall be formatted as shown in Table 7-13 of [ITU-T G.9960].
Reserved		[7:3]	Reserved by ITU-T (Note 1).

Table 8-93 – Format of the MMPL of the CE_ParamUpdate.ind message

Field	Octet	Bits	Description
New block size	2	[1:0]	This field indicates the proposed BLKSZ associated with the new BAT. It shall be formatted as shown in Table 7-7 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 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 of [ITU-T G.9960] (Note 4).
New PSD ceiling	3	[4:0]	This field is the value of APSDC-M 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 of [ITU-T G.9960].
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) (Note 5).
CE_STIME_1	4	[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	5	[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+2	[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_ETIMEv	2V+3	[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.
$TIDX_{MIN}$	(2V+4) to (2V+6)	[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).
$TIDX_{MAX}$		[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). 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.
B_1	2V+7	[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).
		[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).
…	…	…	…
B_Z	2V+6 + Z	[3:0]	4-bit unsigned integer indicating the number of bits assigned to sub-carrier indices $TIDX_{MAX} – G$ to $TIDX_{MAX} – 1$ (Notes 6, 7).

Table 8-93 – Format of the MMPL of the CE_ParamUpdate.ind message

Field	Octet	Bits	Description
		[7:4]	4-bit unsigned integer indicating the number of bits assigned to sub-carrier indices $TIDX_{MAX}$ to $TIDX_{MAX} + G - 1$ (Notes 6, 9).
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 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. NOTE 6 – Sub-carrier index represents the physical index (clause 7.1.4.1 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.			

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 BAT retransmission. It shall be formatted as shown in Table 7-55 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.ind

The format of the MMPL of the CE_PartialBatUpdate.ind message shall be as shown in Table 8-95.

Table 8-95 – Format of the MMPL of the CE_PartialBatUpdate.ind 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 7-55 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 7-55 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.ind message

Field	Octet	Bits	Description
NUM_BAT_ENT	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).
GROUP_ID		[12:10]	This field indicates the minimum GRP_ID associated with the BAT corresponding to O_BAT_ID and N_BAT_ID (*G*). Partial BAT update shall not change the minimum GRP_ID. It shall be formatted as shown in Table 7-13 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 $T_V+G–1$.
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 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 the MSG frame. It shall be formatted as shown in Table 7-16 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.			

8.11.8 Control message formats for channel estimation

8.11.8.1 Format of CM_CE_Request.ind

The format of the CMPL of the CM_CE_Request.ind message shall be as shown in Table 8-97.

Table 8-97 – Format of the CMPL of the CM_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 ($b_7b_6b_5b_4b_3b_2b_1b_0$)	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 ($b_7b_6b_5b_4b_3b_2b_1b_0$)	Interpretation
00000000	End of MAC cycle.
00000001	T_0 + 1/256 of MAC cycle duration.
…	…
11111111	T_0 + 255/256 of MAC cycle duration.

8.11.8.2 Format of CM_CE_Initiation.req

The format of the CMPL of the CM_CE_Initiation.req message shall be as shown in Table 8-100.

Table 8-100 – Format of the CMPL of the CM_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 7-55 of [ITU-T G.9960].
CE_GRP_MIN		[7:5]	This field indicates the minimum value of sub-carrier grouping. It shall be formatted as shown in Table 7-13 of [ITU-T G.9960].

Table 8-100 – Format of the CMPL of the CM_CE_Initiation.req message

Field	Octet	Bits	Description
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 with channel estimation 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. 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.			

8.11.8.3 Format of CM_CE_Initiation.cnf

The format of the CMPL of the CM_CE_Initiation.cnf message shall be as shown in Table 8-101.

Table 8-101 – Format of the CMPL of the CM_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 7-55 of [ITU-T G.9960].
CE_GRP_MIN		[7:5]	This field indicates the minimum value of sub-carrier grouping. It shall be formatted as shown in Table 7-13 of [ITU-T G.9960].
CE_CNF_TYPE	1	[0]	This field indicates the type of channel estimation confirmation. It shall be set to one if channel estimation initiation is granted or set to zero otherwise.
CE_CNF_CODE		[3:1]	This field indicates the reason for channel estimation rejection. 001_2: CE_BAT_ID is invalid. 010_2: Bandwidth for PROBE frame transmission is not available. 000_2, 011_2 to 111_2: Reserved by ITU-T. If CE_CNF_TYPE is set to one, this field shall be set to 000_2.
Reserved		[7:4]	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.			

8.11.8.4 Format of CM_CE_ProbeRequest.ind

The format of the CMPL of the CM_CE_ProbeRequest.ind message shall be as shown in Table 8-102.

Table 8-102 – Format of the CMPL of the CM_CE_ProbeRequest.ind message

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 7-39 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 PROBE request. 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 7-40 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 of [ITU-T G.9960].
Reserved		[7]	Reserved by ITU-T (Note).
CE_PR_APSDC	2	[4:0]	This field indicates the APSDC-P requested by the receiver. It shall be formatted as described in clause 7.1.2.3.2.7.4 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.			

Table 8-103 – CE_PR_PRBFN field values

Value ($b_3b_2b_1b_0$)	**Interpretation**
0000	One PROBE frame
0001	Two PROBE frames
…	…
1111	Sixteen PROBE frames

8.11.8.5 Format of CM_CE_Cancellation.ind

The format of the CMPL of the CM_CE_Cancellation.ind message shall be as shown in Table 8-104.

Table 8-104 – Format of the CMPL of the CM_CE_Cancellation.ind 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 7-55 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 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 of [ITU-T G.9960]. It shall be set to 00_2 otherwise.

Table 8-104 – Format of the CMPL of the CM_CE_Cancellation.ind message

Field	Octet	Bits	Description
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 of [ITU-T G.9960]. It shall be set to 00_{02} otherwise.
Bandplan ID		[7:5]	When USE_RCM is set to one this field indicates the BNDPL based on which the RCM parameters are proposed. It shall be formatted as shown in Table 7-10 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 of [ITU-T G.9960]. It shall be set to 000_2 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.			

8.11.8.6 Format of CM_BatIdMaintain.ind

The format of the CMPL of the CM_BatIdMaintain.ind message shall be as shown in Table 8-105.

Table 8-105 – Format of the CMPL of the CM_BatIdMaintain.ind message

Field	Octet	Bits	Description
VALID_BAT_ID	0 and 1	[15:0]	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 16 is valid. The MSB of the VALID_BAT_ID shall be set if runtime BAT 31 is valid.
MAX_NUM_BAT_ID	2	[4:0]	This field contains the maximum number of valid BAT_ID that this node (SID) can support when transmitting to the destination node (DID). Valid values are from 0 to 16.
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.			

8.12 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. 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 247 data connections associated to service flows.

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.

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 FEC block size that the transmitter has selected for a connection shall be indicated in the PHY-frame header when the connection is established.

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, 7.1.2.3.2.3.2 and 7.1.2.3.2.3.3 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 FLCTRL field.

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.

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=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 minimum of the value indicated in the FLCTRL field and its own available window size.

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 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. For the connections that require reset of their transmission and reception windows (see clause 8.9.5.3.1), the nodes shall follow the procedure described in clause 8.12.1.1 with CNN_MNGMT set to 0011_2 for a management connection, and the procedure described in clause 8.12.1.2 with CNN_MNGMT set to 0111_2 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 and management broadcast connections.

8.12.8.1 Management broadcast connection

To establish a management broadcast 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 management broadcast connection. It is implementation dependent how this is achieved. To release a management broadcast 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 management broadcast 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 Data broadcast connection

To establish a data broadcast 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 data broadcast 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 data broadcast 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.

8.13 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

A domain master should be capable of detecting the presence of other domains operating in the same medium (either directly or via information sent by other devices in its own domain), and coordinating with them while guaranteeing that QoS requirements for existing service flows are met.

The protocol used for coordination of multiple domains is left for further study.

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 Multicast binding protocol

The multicast binding protocol shall be used to create multicast groups and manage multicast transmissions identified by (SID, MULTICAST_ID) tuples (see clause 8.7.1.2) among nodes communicating directly (i.e., not via a relay node). The protocol to support multicast relayed transmissions is for further study.

8.16.1 Initialization of a multicast group for a new multicast stream

A transmitting node of a multicast stream may initiate the multicast binding protocol upon detecting the presence of a multicast source (e.g., when IGMP query or multicast traffic is transmitted by the multicast source) when there are nodes that requested to receive the multicast stream (e.g., via an IGMP join message).

If the transmitter initiates the multicast binding protocol, it shall compute the BATs to be used for the multicast stream based on the BATs (see clause 8.11) reported by the receiver nodes that requested to receive this multicast stream. The transmitter shall then determine the number of multicast groups and the assignment of receivers to each multicast group.

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 multicast group, the transmitter assigns receivers to the Mc-ACK/NACK slots. The acknowledgement protocol state machine for multicast transmission shall be initialized as specified in clause 8.9.5.4.

NOTE 1 – The actual method for deciding on the number of 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 a MC_GrpInfoUpdate.ind message including information for each created multicast group. Upon reception of a MC_GrpInfoUpdate.ind message, each receiver that appears as a receiver of a multicast group shall confirm the message by sending a MC_GrpInfoUpdate.cnf to the transmitter.

In case MC_GrpInfoUpdate.cnf 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 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 multicast group.

When flow-control is not used on a 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 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 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 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 multicast group, the transmitter may decide to reassign multicast group members to different groups.

When flow-control is not used, the value of FLCTRL specified in the ACK, BACK and BMSG 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 receiver.

Before the multicast binding is completed for a new multicast stream, 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 multicast group or when a change in the membership of nodes of an existing 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.

8.16.2 Maintenance of multicast binding information

The transmitter shall send a MC_GrpInfoUpdate.ind message as specified in this clause to update receivers of a multicast group when change in BATs, or in the membership of receiver nodes, or in Mc-ACK slot assignment occurs.

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 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. If the receiver is no longer interested in that multicast stream while it has an assigned Mc-ACK slot, it shall set the FACK field to 111 (see clause 7.1.2.3.2.3.9.1.5 of [ITU-T G.9960]) and ACKI field to all ones (see clause 7.1.2.3.2.3.9.1.7 of [ITU-T G.9960]) to indicate to the transmitter that it is no longer interested in receiving the multicast stream, and its ACKI field shall be ignored by the transmitter.

The transmitter may split an existing multicast stream into several multicast groups when new receivers join. The transmitter shall assign a new multicast DID to each of the newly created multicast groups and shall send MC_GrpInfoUpdate.ind, which includes the information describing the new multicast groups, to all nodes associated with that multicast stream, using either separate unicast DIDs, broadcast DID or other multicast group DIDs.

Splitting of an existing multicast group or moving of receivers from one multicast group to another is for further study.

The transmitter shall follow the actions described in clause 8.16.1 each time it sends MC_GrpInfoUpdate.ind for informing on new multicast groups or for updating existing multicast group information.

The MC_GrpInfoUpdate.ind message sent by the transmitter shall include the list of all BAT_IDs that are to be active in the multicast group or groups (inside the McstGroupInfo field, see Table 8-107). New BAT_IDs in this list are accompanied with BATInfo fields (see Table 8-109). The transmitter shall not start using the new BATs until all the multicast group receivers have confirmed the change. Once a new BAT is used for transmission by the transmitter, the receivers of the multicast group shall invalidate the old BATs assigned to that multicast group.

In case the transmitter detects a change in the multicast binding information while awaiting confirmation from the receivers, it shall restart the procedure generating full binding information and retransmitting MC_GrpInfoUpdate.ind with a higher sequence number.

8.16.3 Termination of a multicast group

When none of the receivers that are part of a multicast group are interested in the multicast stream (e.g., no IGMP membership report messages are received from any of the receivers of the multicast group) the transmitter shall send MC_GrpInfoUpdate.ind to those receivers to release this multicast DID.

8.16.4 Multicast binding protocol flow

8.16.4.1 Message sequence – Initialization of a multicast group for a new multicast stream

Figure 8-58 shows an example of initialization of a multicast group when the multicast stream is not active. The multicast binding protocol messages are marked in dashed-red arrows.

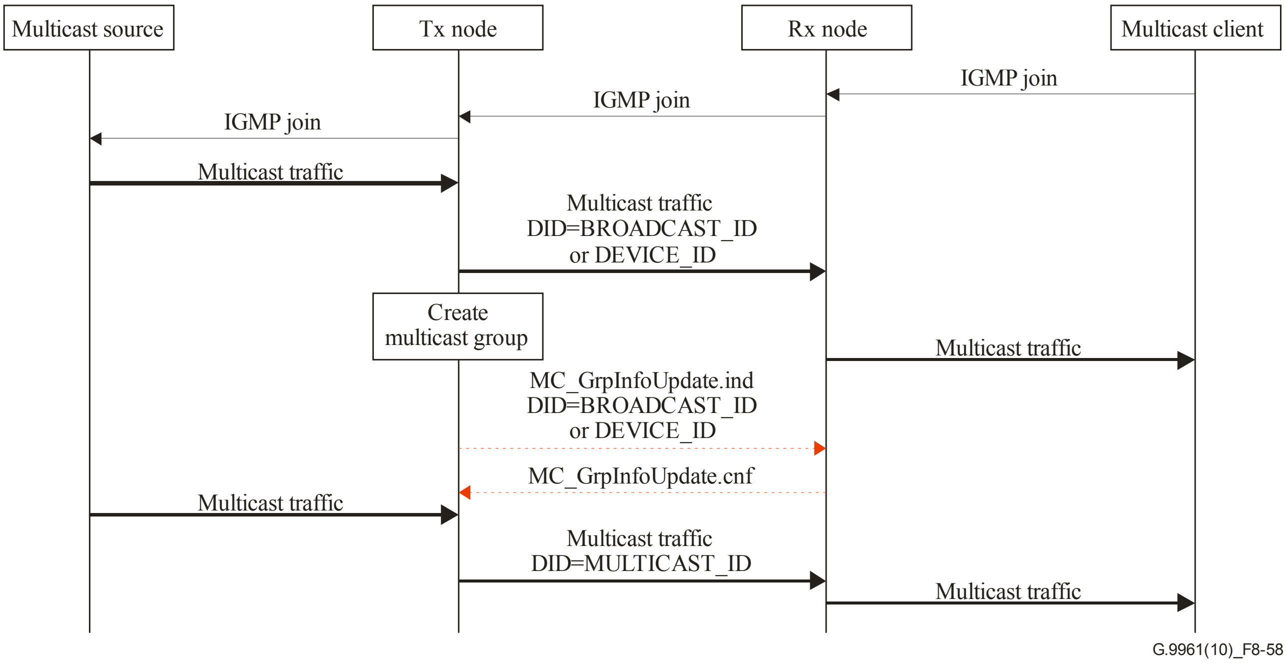


Figure 8-58 – Example of initialization of a multicast group

In this example, the MC_GrpInfoUpdate.ind is sent only when the multicast traffic starts until the point that the transmitter is not aware that the multicast source is above its A-interface. In case the transmitter is aware that the source of the multicast stream is above its A-interface, the transmitter could have started the multicast binding protocol without waiting for the actual multicast traffic.

NOTE – The transmitter may choose not to send the multicast traffic until the multicast binding protocol is complete or to send it via broadcast or unicast as shown in the example.

8.16.4.2 Message sequence – Split of a multicast stream into several groups

Figure 8-59 shows an example of an existing multicast stream that is transmitted using a single multicast group to two receivers. When node number three requests to join the multicast stream, the transmitter decides that it is better to allocate a new multicast group for this node. Hence, it has created another multicast DID and has informed node number three on the new group via the MC_GrpInfoUpdate.ind message. When the message was confirmed by node number three, the transmitter can start using the new multicast group in addition to the existing multicast group.

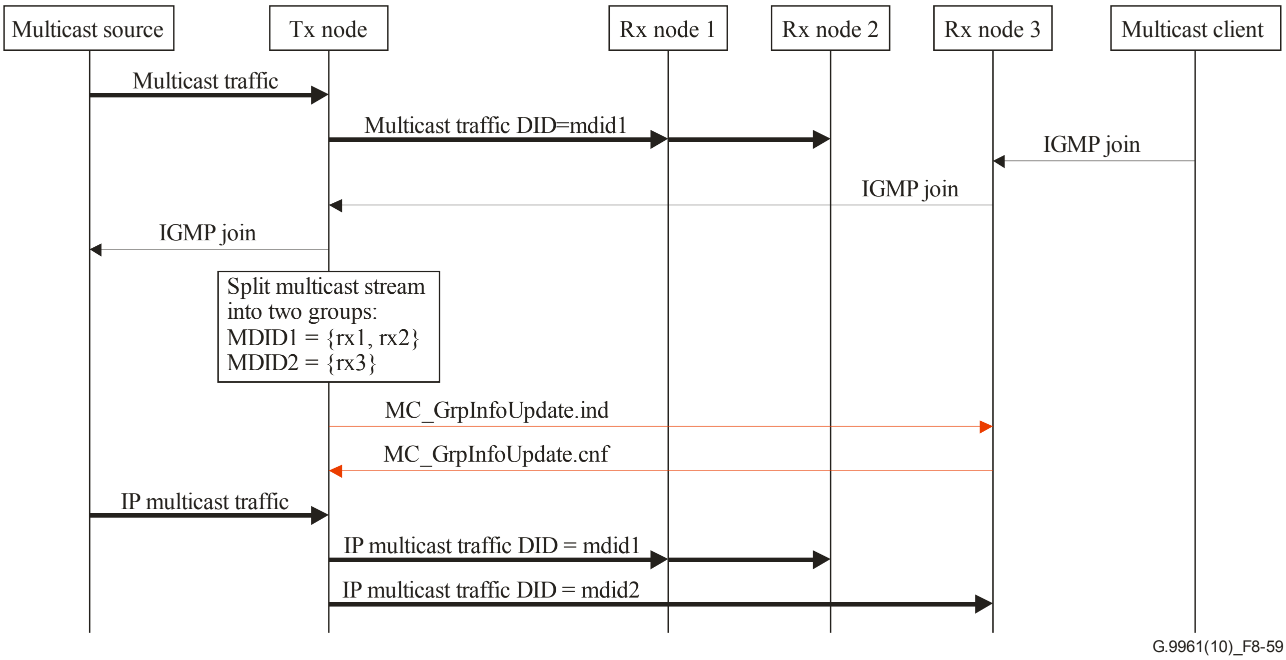


Figure 8-59 – Example of split of a multicast group

8.16.4.3 Message sequence – Establish a multicast group with flow-control disabled

Figure 8-60 shows an example of the establishment of a multicast group with flow-control disabled. In this example, the transmitter initiates the sequence by sending a MC_GroupInfoUpdate.ind message to the three multicast group members specifying the recommended minimum receive buffer size. The 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.

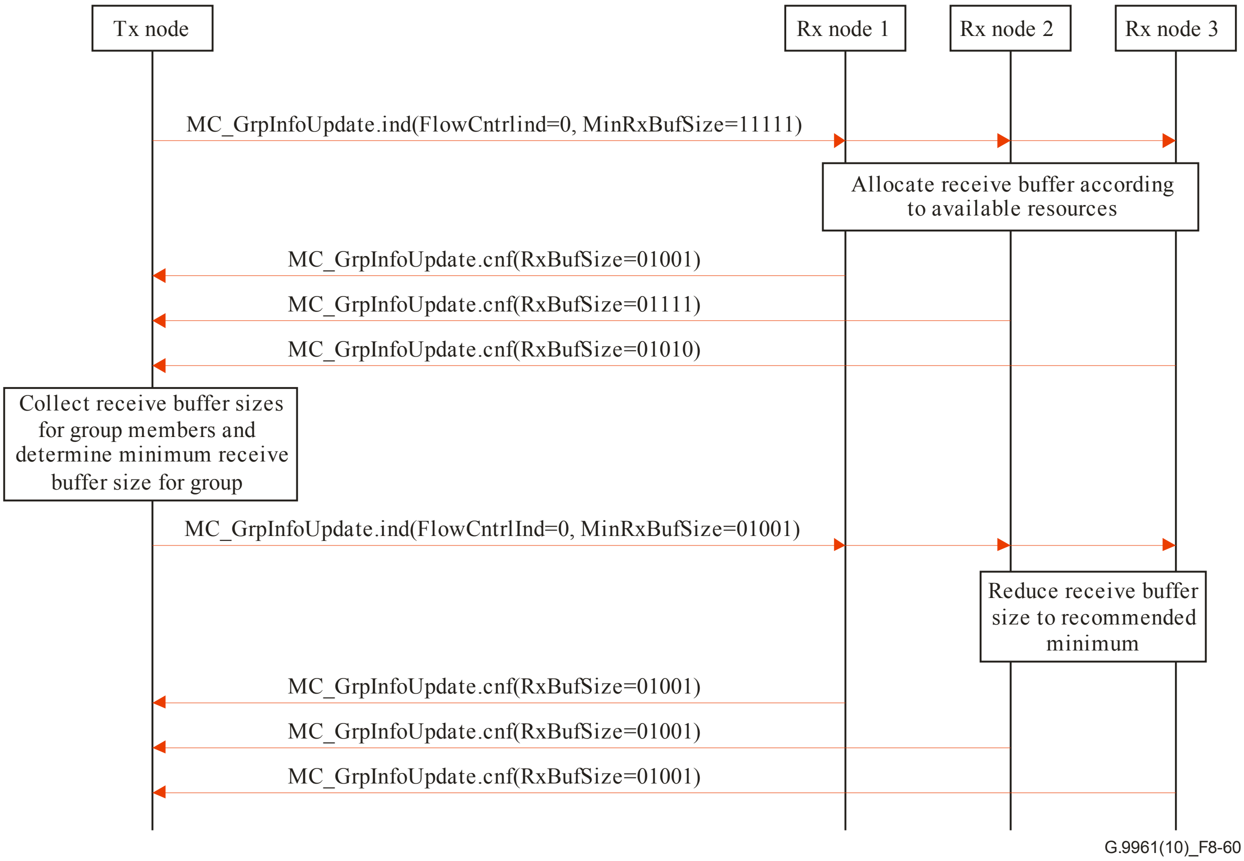


Figure 8-60 – Example of establishment of a multicast group with flow-control disabled

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 multicast group to its four receivers. When receiver number three reports on change in its recommended BAT, the transmitter has decided to update the BAT of the multicast group. Hence, it has informed the receivers on the change via the MC_GrpInfoUpdate.ind message. When the change was confirmed by all the receivers the transmitter can start using the new BAT.

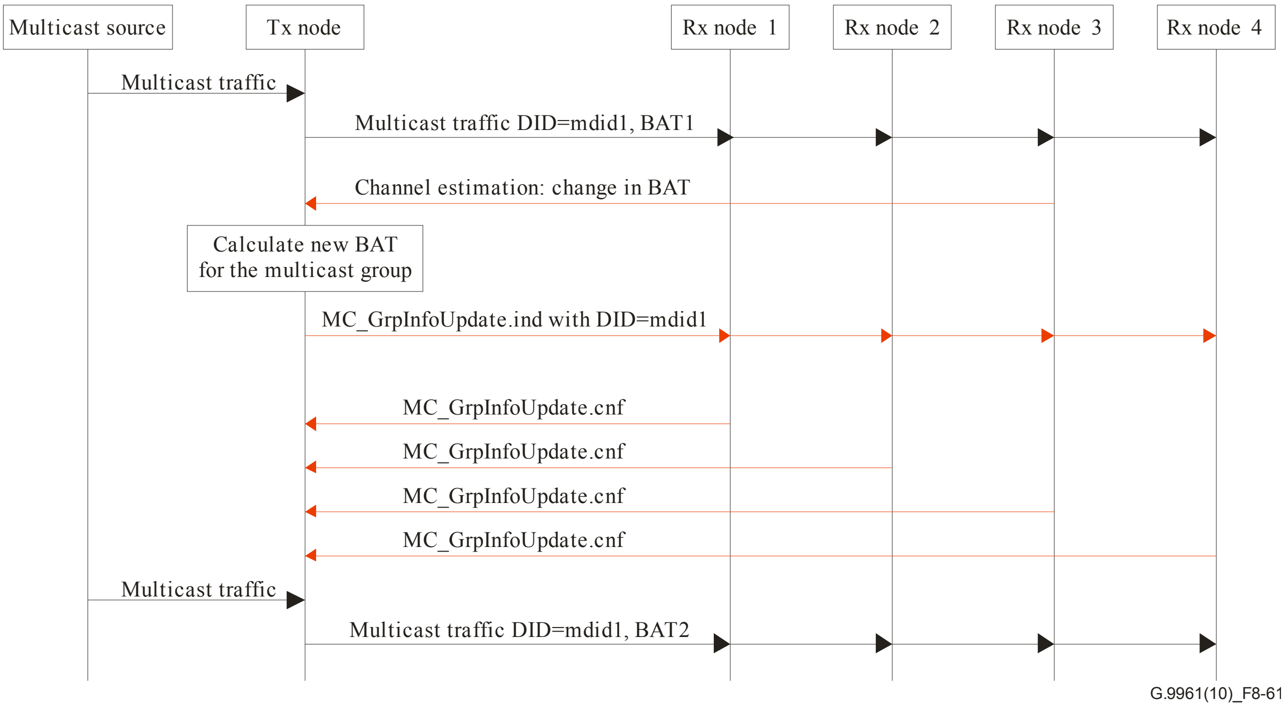


Figure 8-61 – Example of maintenance of multicast binding information

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 Multicast binding protocol messages

8.16.5.1 Multicast group information update indication

MC_GrpInfoUpdate.ind is a management message sent by the transmitter node to update all the receivers of a multicast stream about any change in the multicast binding information (e.g., creation of new groups, update of existing group parameters, etc.). Each McstGroupInfo within the message identifies a multicast group uniquely by its multicast DID and the source ID of the transmitter.

Each 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 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.
NumMcstGroups	1	[7:0]	Number of multicast groups described in this message.
McstGroupInfo[0]	2	Variable	Refer to Table 8-107.
…	…	…	…
McstGroupInfo[N]	...	Variable	
NumBATs	Variable	[7:0]	Number of BATs described. Zero indicates no BAT is described.
BATInfo[0]	…	Variable	Refer to Table 8-109.
…	…		
BATInfo[N]	…	Variable	

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_MCACK_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 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).
MinRxBufSize	2	[4:0]	Recommended minimum receiver buffer size expressed in LPDUs to be buffered by receivers in the multicast group (Note 2). The values of this field shall be the same as FLCTRL field for status report (Table 7-21 of [ITU-T G.9960]).
MinRxBufSize_BLKSZ		[5]	LPDU_size units for the MinRxBufSize field: 0 – 120 bytes. 1 – 540 bytes.
Reserved		[7:6]	Reserved by ITU-T (Note 1).
NumBatIds	3	[7:0]	Number of BAT_IDs minus one used for this 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 multicast group.
BAT_ID	n+3	[7:0]	The last of n BAT IDs used for this multicast group.
NumRxNode	n+4	[7:0]	Number of receive nodes m that are members of the multicast group. Zero indicates that this multicast DID is released.

Table 8-107 – Format of McstGroupInfo field

Field	Octet	Bits	Description
RxNodeInfo	n+5 and n+6	[15:0]	Info for the first of m receive nodes of the multicast group.
⋮	⋮	⋮	⋮
RxNodeInfo	n+m+4 and n+m+5	[15:0]	Info for the last of m receive nodes of the 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 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
RxDID	0	[7:0]	Device ID of a receive node of the 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 multicast group. 1-7 – Mc-ACK Slot ID if Mc-ACK is enabled according to the RPRQ of the 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 of [ITU-T G.9960].
Bandplan ID		[7:5]	This field indicates the type of bandplan used by the transmitter based on which the subsequent BAT entry is defined. It shall be formatted as shown in Table 7-10 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 of [ITU-T G.9960].
Reserved		[7:3]	Reserved by ITU-T (Note 1).

Table 8-109 – Format of BATInfo field

Field	Octet	Bits	Description
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.
NumBATEntries	2n+3 to 2n+4	[15:0]	Number of sub-carrier entries minus one contained in this message. Valid values are $0 \le m \le 4095$.
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: • 00_{16} = Success (indicating MC_GrpInfoUpdate.ind has been accepted). • 01_{16} = Failure – lack of resources. • 02_{16} – FF_{16} = Reserved by ITU-T.
RXBufSize	3	[4:0]	Available receiver buffer size (ACK_RX_CONF_WINDOW_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 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 multicast group shall report the value for the FLCTRL field (Table 7-21 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.			

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 periodical 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* 16-byte blocks: 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, …, 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 MIC = T. 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 MIC. Agreement therefore constitutes assurance of message authenticity (integrity).

5) Generate the counter blocks Ctr_0, Ctr_1,…, Ctr_m, where $m = ceiling\ (Plen/128)$.

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 $S = S_1 \| S_2 \| \ldots \| S_m$: this defines the string of encrypted counter blocks. Note that S_0 is skipped.

8) Compute $C = (P \oplus \text{MSB}_{Plen}(S)) \| (T \oplus \text{MSB}_{Tlen}(S_0))$: 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) and blocks B_0, 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 compliments the last payload block to 128 bits.

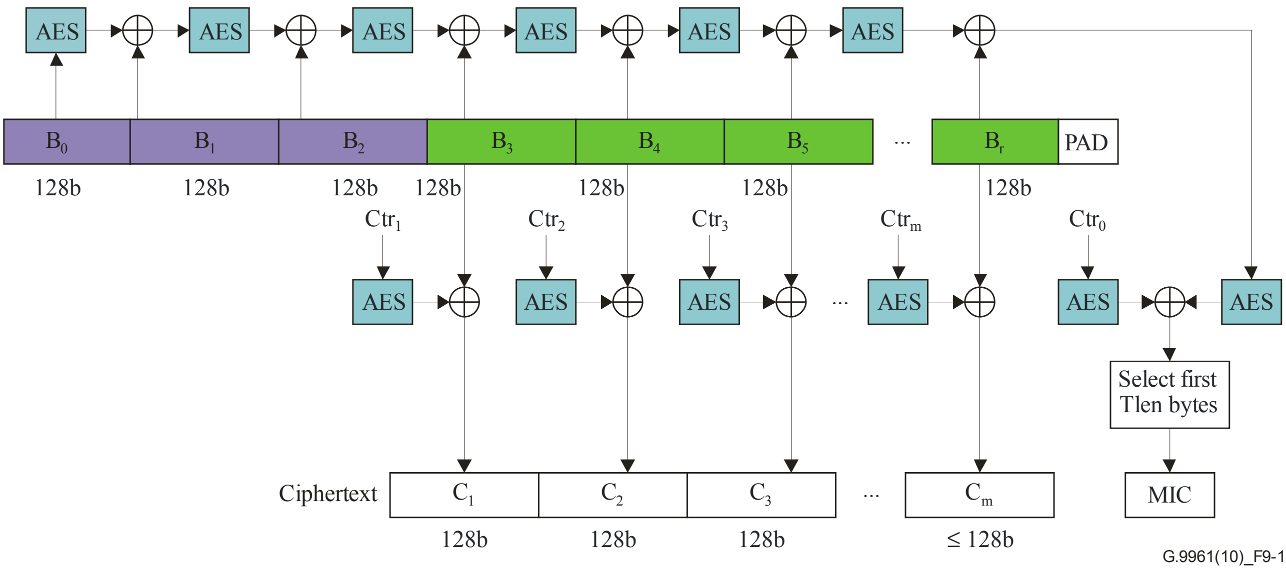


Figure 9-1 – Functional diagram of CCM encryption and message-authentication

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 *(Tlen)*, bytes	4, 8, 16
Payload size *(Plen)*, bytes	$\leq (2^{14} - 1)$
Associated data size (*Alen*), blocks	3

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 (Ctr_n);

– associated data blocks (B_0, B_1, and B_2);

– payload blocks (B_3 to B_r);

– encryption key.

The 16-byte counter blocks Ctr_0, Ctr_1…, 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 00_{16}. 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 – 12
Contents	Flags	Source MAC address	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 associated data blocks B_0-B_2 shall each be 16-bytes long. 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			

Blocks B_1, 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, expressed as an unsigned integer.
	2 and 3	Reserved by ITU-T (Note 2).
	4 to 9	Destination MAC address.
	10 to 15	Source MAC address.
B_2	0 to 3	Portion of LFH excluding bytes containing TTL and TSMP fields (Note 3).
	4 to (3 + V)	Additional unencrypted field. APDU (EAPC): *TG* bytes of EtherType/TAGs ($V = TG$, See Figure A.1). LCDU: 2 bytes of EtherType ($V = 2$, or equivalent).
	(4 + V) to 15	Remainder of the associated data (Reserved by ITU-T, Note 2).
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 – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver. NOTE 3 – Byte 0 to byte 3 of B_2 shall correspond to byte 0 to byte 3 of LFH, respectively (Table 8-1).		

All bytes of 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 17-th 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 key ID, 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.

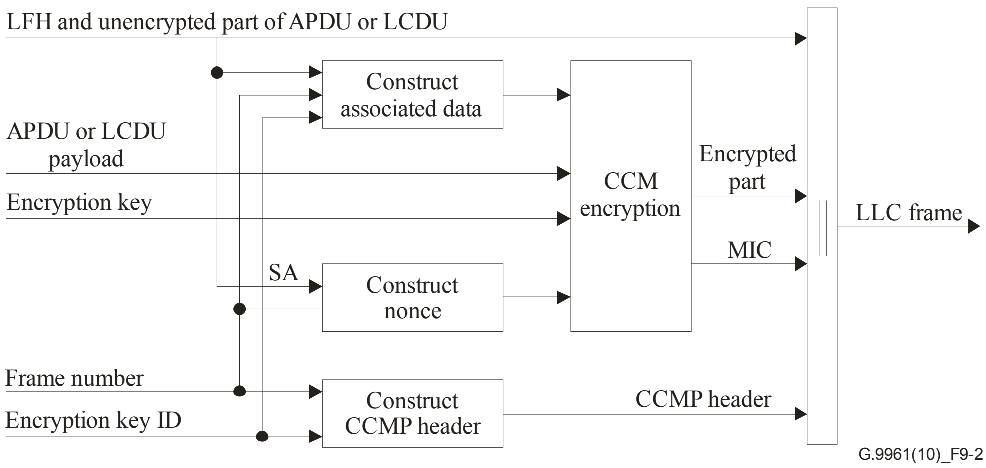


Figure 9-2 – Functional diagram of CCMP encryption

9.1.2.2 CCMP encryption format

The format of the encrypted LLC frame is presented in Figure 9-3 (see also Figure 8-4). The encrypted APDU (or LCDU) consists of four parts: CCMP header, unencrypted part, encrypted part (cipher text), and MIC.

LFH	CCMP header	Unencrypted part	Encrypted part	MIC
Clause 8.1.3.1.1	Clause 9.1.2.3		Clause 9.1.1	Clause 9.1.1

G.9961(10)_F9-3

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 (key ID), the length of the MIC, and the security frame number (FN). These three 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.
		[5:3]	Reserved by ITU-T (Note).
		[7:6]	Encryption key ID, formatted as an unsigned binary integer.
	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 key ID 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 range of the key ID is from 0 to 3. The format of the key ID is a 2-bit unsigned binary integer.

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.

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 security. AKM includes the following main procedures:

– authentication to the domain in secure mode;

– establishing point-to-point encryption keys for communication;

– periodic re-authentication and updating point-to-point 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. 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 can further apply for authentication to operate in secure mode. Authentication shall be performed as described in clause 9.2.2.

An authenticated node can establish encryption keys for secure unicast, multicast, and broadcast communications inside the domain and communications to nodes of other secure domains. 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 periodical re-authentication and updates of encryption keys.

A flowchart of AKM procedures is presented in Figure 9-4.

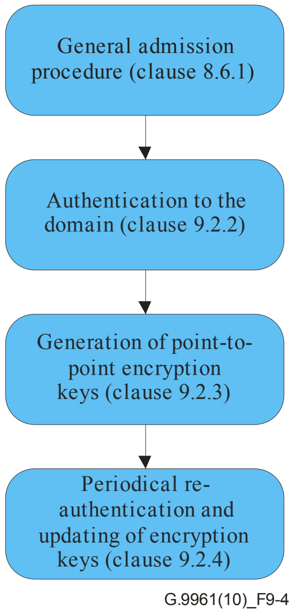


Figure 9-4 – Flowchart of AKM

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.

The SC can be configured to use a single encryption key per domain/network, the network membership key (NMK). 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 (both transmit and receive) with any other node operating in secure mode. After authentication, the node can be a part of the domain operating in secure mode.

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.

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 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, *K*, which shall be used for encryption of secure communications between the node and the SC.

NOTE 1 – The PAK protocol, with very high probability, returns a new encryption key after each run.

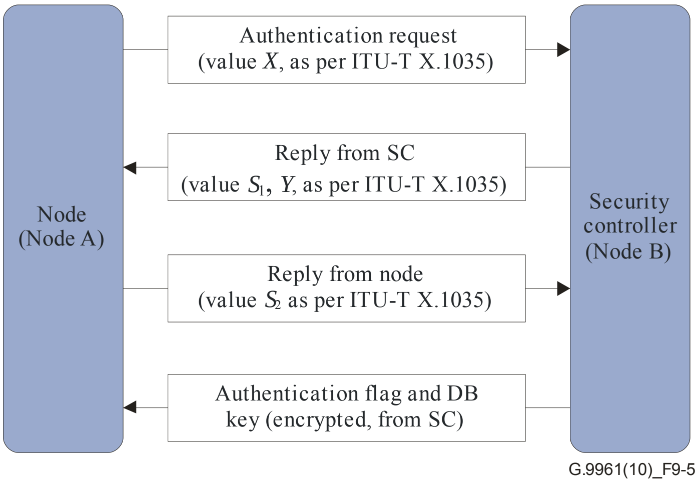


Figure 9-5 – PAK handshake procedure

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 initiates the authentication procedure with the SC by sending to the SC an authentication request (AUT_NodeRequest.req) message which includes the name of the node, the name of the SC, 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_NodeRequest.req message shall be sent unencrypted.

2) The SC verifies the received value of *X* and replies to the supplicant with an authentication prompt (AUT_Promp.ind) message, including values S_1 and *Y*, as per [ITU-T X.1035]. The message shall be sent unencrypted.

   NOTE 2 – The SC identifies the node providing the value of *X* using the node MAC address (SA of the LCDU carrying the AUT_NodeRequest.req message).

3) The node verifies the prompt, computes the value S_2, as per [ITU-T X.1035], and sends it to the SC in the authentication prompt verification (AUT_Verification.res) message. The message is sent unencrypted.

4) Using the exchanged variables S_1 and S_2 both nodes compute independently the 128-bit NSC encryption key (value of *K* as per [ITU-T X.1035]).

5) The SC sends to the supplicant the authentication confirmation (AUT_Confirmation.cnf) message, which includes the confirmation flag and the in-domain broadcast (DB) encryption key. The AUT_Confirmation.cnf message shall be sent encrypted by NSC.

If any one of the steps fails, SC shall send AUT_Confirmation.cnf with a confirmation flag set off. In case the confirmation flag is off, or the node cannot decrypt the AUT_Confirmation.cnf, or it did not receive AUT_Confirmation.cnf in the 200 ms after it sent the value S_2, the node shall consider authentication failed. The supplicant may start re-authentication in a time period greater than one second, and shall not transmit any data from the time it received AUT_Confirmation.cnf 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 node whose authentication was confirmed is allowed to broadcast and receive broadcast messages from other secure nodes of the domain using DB encryption key and 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 SC is configured to operate with NMK, it shall send the NMK to the authenticated node in AUT_Confirmation.cnf message (see clause 9.2.5.1.4).

9.2.2.1.1 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 may be used as the first relay between the SC and the node.

The node starts authentication by sending AUT_NodeRequest.req message encapsulated into an LCDU where the destination address is the MAC address of the SC, and uses the DID in the PHY frame carrying AUT_NodeRequest.req equal to the DEVICE_ID of the first relay (which is its registration proxy).

After the node has transmitted AUT_NodeRequest.req, it shall wait for an AUT_Promp.ind message (expected to come from the node that was used as the first relay). After AUT_Promp.ind is received, the node shall reply by AUT_Verification.res using again the same considerations as when it sends AUT_NodeRequest.req, and wait for AUT_Confirmation.cnf to complete the authentication process.

Messages AUT_Promp.ind and AUT_Confirmation.cnf are sent by the SC encapsulated in LCDUs with the destination address equal to the MAC address of the supplicant..

NOTE – To avoid multiple authentication attempts, a node may delay the start of authentication procedure after it gets registered to accommodate topology update initiated by the registration proxy (see clause 8.6.1.2).

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. The detailed requirements for selection of the values for PW are for further study.

Table 9-7 – ITU-T X.1035 – PAK parameters

ITU-T X.1035 parameter	Description	Length (bits)	Notes
Node name (A, B)	Value of the unicast DEVICE_ID granted to the node	8	
Password (PW)	User password of the authenticated node	96 bits (12 ASCII characters)	
p	Diffie-Hellman constant (prime value)	1024	Recommended by [ITU-T X.1035]
g	Diffie-Hellman generator	160	Recommended by [ITU-T X.1035] (value from TIA-683-D)
R_A, R_B	Secret exponents	384	Recommended by [ITU-T X.1035]
H_1	Hash functions of SHA-256 type	1152	Recommended by [ITU-T X.1035]
H_2		1152	Recommended by [ITU-T X.1035]
H_3, H_4, H_5		128	Recommended by [ITU-T X.1035]
K	NSC key	128	

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 sends a communication request (AKM_KeyRequest.req) message to the SC which includes the DEVICE_ID (or MULTICAST_ID in case of multicast) of the addressee node(s) it intends to communicate with. The message shall be encrypted with NSC of the supplicant.

2) The SC accepts the request and generates 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.

3) The SC sends the generated pair of NN keys to each of the authenticated addressees using the AKM_NewKey.req message; no key shall be generated for 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_KeyAck.cnf message to the SC. In case no AKM_KeyAck.cnf is received from a particular addressee during the time period of 200 ms, the SC may retransmit the message up to four times, and shall remove the addressee from the list if no AKM_KeyAck.cnf arrives after the last attempt or AKM_KeyAck.cnf brings a rejection code (NACK).

4) The SC replies to the supplicant with the confirmation (AKM_KeyConfirmation.req) message, which includes the generated pair of NN keys and DEVICE_ID(s) of the addressee(s) that acknowledged reception of the AKM_NewKey.req message. The AKM_KeyConfirmation.req message shall be encrypted using the NSC key of the supplicant.

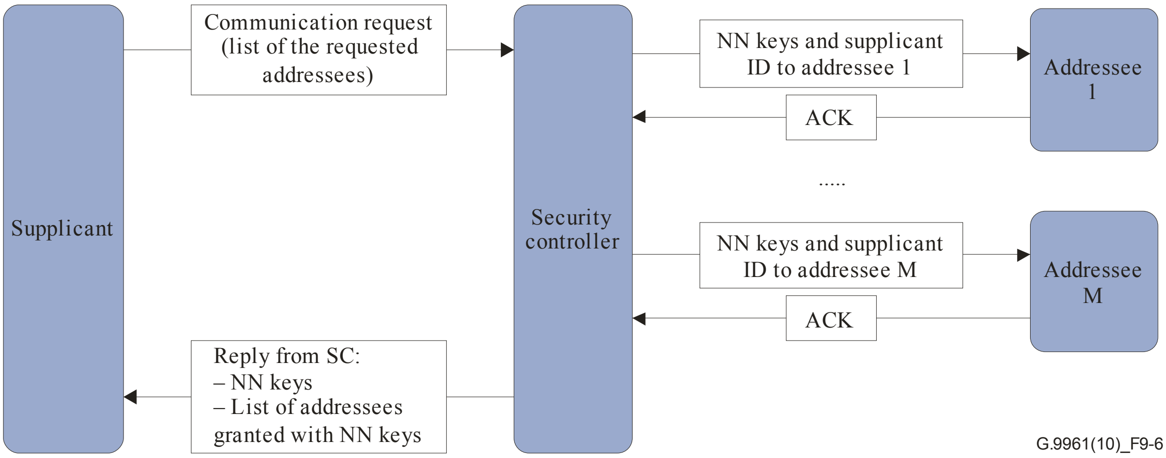


Figure 9-6 – Procedure for NN key generation for unicast (M=1) and multicast (M > 1)

If the supplicant does not receive the reply from SC (AKM_KeyConfirmation.req message) during 1 second, 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 sends to the SC a request (AKM_KeyAddRequest.req) message that includes the NN keys 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) The SC accepts the request, checks whether the addressee is authenticated, and sends the NN keys 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 AKM_NewKey.req message by sending an AKM_KeyAck.cnf message to the SC. In case no AKM_KeyAck.cnf is received from the addressee during the time period of 200 ms, the SC may retransmit the message up to four times, and shall remove the addressee from the list if no AKM_KeyAck.cnf arrives after the last attempt or if AKM_KeyAck.cnf brings a rejection code (NACK).

3) The SC replies to the supplicant with the confirmation (AKM_KeyConfirmation.req) message that includes the pair of NN keys and DEVICE_ID of the addressee, if it acknowledged reception of the AKM_NewKey.req message (if no addressee name is communicated in AKM_KeyConfirmation.req message, the addressee is not joined to the group). The AKM_KeyConfirmation.req message shall be encrypted using the NSC key of the supplicant.

9.2.4 Updating and termination of encryption keys

From time to time the SC may initiate a routine update of encryption keys. The frequency of routine updates is vendor discretionary, although the period of updates shall be much longer than the duration of the procedure to establish the corresponding key, but shall not exceed 24 hours. In addition, the key shall be updated to prevent repetition of FN for the same key (see clause 9.1.2.3).

9.2.4.1 Updating of NSC and NN keys

The key updating procedure shall be initiated by the SC. To initiate the procedure, the SC shall send a key update request AKM_KeyUpdate.req message to the node that initiated generation of the key(s) to be updated. The node receiving the AKM_KeyUpdate.req message shall reply to the SC by:

- initiating an authentication procedure with the SC, as described in clause 9.2.2, if AKM_KeyUpdate.req message indicates NSC key update;
- initiating a point-to-point key generation procedure with the relevant addressees, as described in clause 9.2.3, if AKM_KeyUpdate.req message indicates the NN keys update.

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 Termination of NSC and NN keys

The SC shall terminate all NSC keys associated with a node upon node resignation from the domain. The node shall terminate NN keys if the node-supplicant 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 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.

9.2.5 Messages supporting AKM procedures

9.2.5.1 Authentication messages

9.2.5.1.1 Authentication request message (AUT_NodeRequest.req)

The AUT_NodeRequest.req message is a unicast management message intended to be used for authentication request only. The format of the MMPL of the AUT_NodeRequest.req message shall be as shown in Table 9-8.

Table 9-8 – Format of the MMPL of the AUT_NodeRequest.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.
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.			

9.2.5.1.2 Authentication prompt message (AUT_Promp.ind)

The AUT_Promp.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_Promp.ind message shall be as shown in Table 9-9.

Table 9-9 – Format of the MMPL of the AUT_Promp.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 159	[1151:0]	Value of Y as per [ITU-T X.1035].
Status	160	[0]	Shall be set to zero if the X-value was accepted and to 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 Authentication prompt verification message (AUT_Verification.res)

The AUT_Verification.res 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.res message shall be as shown in Table 9-10.

Table 9-10 – Format of the MMPL of the AUT_Verification.res message

Field	Octet	Bits	Description
Value S_2	0 to 15	[127:0]	Value of S_2 as per [ITU-T X.1035].
Status	16	[0]	Shall be set to zero if both the S_1-value and Y-value were accepted and to 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 Authentication confirmation message (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. 01_2 – Single key per domain (NMK). 10_2, 11_2 – Reserved by ITU-T.
Confirmation flag		[3:2]	Shall be set to 11_2 if authenticated and any other value for "authentication fails": 00_2 – Reason undefined. 01_2 – Reason (reserved). 10_2 – Reason (reserved).
Reserved		[7:4]	Reserved by ITU-T (Note).

Table 9-11 – Format of the MMPL of the AUT_Confirmation.cnf message

Field	Octet	Bits	Description
DB key	1 to 16	[127:0]	Encryption key for broadcast communications.
NMK	17 to 32	[127:0]	NMK, if security mode is 01_2. This field shall not be present if security mode is 00_2.
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 Pair-wise authentication messages

9.2.5.2.1 Communication request message (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 (1 for unicast transmission and up to 248 for multicast transmission).
Addressee name	1	[7:0]	First addressee unicast DEVICE_ID.
Addressee name	2	[7:0]	Second addressee unicast DEVICE_ID.
…	…	…	…
Addressee name	N	[7:0]	N-th addressee unicast DEVICE_ID.
Attempt number	N+1	[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 keys with ID = 0, 1 and set to one to request keys with ID = 2, 3.
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.			

9.2.5.2.1.1 Add-a-node Request message (AKM_KeyAddRequest.req)

The AKM_KeyAddRequest.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_KeyAddRequest.req message shall be as presented in Table 9-13.

Table 9-13 – AKM_KeyAddRequest.req message format

Field	Octet	Bits	Description
Addressee name	0	[7:0]	The addressee unicast DEVICE_ID.
NN_{SA} key 0/2	1 to 16	[127:0]	Encryption key for supplicant-to-addressee direction with ID=0 if KeyID=0 and with ID=2 if KeyID=1
NN_{AS} key 0/2	17 to 32	[127:0]	Encryption key for addressee-to-supplicant direction with ID=0 if KeyID=0 and with ID=2 if KeyID=1
NN_{SA} key 1/3	33 to 48	[127:0]	Encryption key for supplicant-to-addressee direction with ID=1 if KeyID=0 and with ID=3 if KeyID=1

Table 9-13 – AKM_KeyAddRequest.req message format

Field	Octet	Bits	Description
NN_{AS} key 1/3	49 to 64	[127:0]	Encryption key for addressee-to-supplicant direction with ID=1 if KeyID=0 and with ID=3 if KeyID=1
Attempt number	65	[1:0]	Shall be set to 00_2 for the initial request and incremented for every next attempt.
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.			

9.2.5.2.2 Key communication message (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.
Number of keys	1	[1:0]	Number of keys provided by the SC represented as an unsigned integer minus 1.
Reserved		[7:2]	Reserved by ITU-T (Note).
NN_{SA} key 0/2	2 to 17	[127:0]	Encryption key for supplicant-to-addressee direction with ID=0 if KeyID=0 and with ID=2 if KeyID=1
NN_{AS} key 0/2	18 to 33	[127:0]	Encryption key for addressee-to-supplicant direction with ID=0 if KeyID=0 and with ID=2 if KeyID=1
NN_{SA} key 1/3	34 to 49	[127:0]	Encryption key for supplicant-to-addressee direction with ID=1 if KeyID=0 and with ID=3 if KeyID=1
NN_{AS} key 1/3	50 to 65	[127:0]	Encryption key for addressee-to-supplicant direction with ID=1 if KeyID=0 and with ID=3 if KeyID=1
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 Communication acknowledgement message (AKM_KeyAck.cnf)

The COM _ACK message is a unicast management message intended to be used to confirm delivery of the new encryption key to SC or to reject the communication request. The format of the MMPL of the AKM_KeyAck.cnf message shall be as shown in Table 9-15.

Table 9-15 – Format of the MMPL of the AKM_KeyAck.cnf message

Field	Octet	Bits	Description
ACK	0	[1:0]	00 – If the addressee successfully decoded the new encryption key. 01 – If the addressee is incapable of decoding the new encryption key. 10 – If the addressee decoded successfully the new encryption key, but denies communication with supplicant. 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 Confirmation message (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 shall be as 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, 11 – Reserved by ITU-T.
Reserved		[7:2]	Reserved by ITU-T (Note 1).
NMK 0/2	1 to 16	[127:0]	NMK with ID=0 if KeyID=0 and with ID=2 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 0/2	17 to 32	[127:0]	Encryption key for supplicant-to-addressee direction with ID=0 if KeyID=0 and with ID=2 if KeyID=1
NN_{AS} key 0/2	33 to 48	[127:0]	Encryption key for addressee-to-supplicant direction with ID=0 if KeyID=0 and with ID=2 if KeyID=1
NN_{SA} key 1/3	49 to 64	[127:0]	Encryption key for supplicant-to-addressee direction with ID=1 if KeyID=0 and with ID=3 if KeyID=1
NN_{AS} key 1/3	65 to 80	[127:0]	Encryption key for addressee-to-supplicant direction with ID=1 if KeyID=0 and with ID=3 if KeyID=1
Number of addressees	81	[7:0]	Number of addressees N (1 for unicast transmission and up to 248 for multi-cast transmission) (Note 2).
Addressee name	82	[7:0]	First addressee unicast DEVICE_ID.
Addressee name	83	[7:0]	Second addressee unicast DEVICE_ID.
…	…	…	…
Addressee name	81 + 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 – In case no addressee is authenticated, the list is empty and the field shall be set to zero.			

9.2.5.2.5 Resignation request message (SC_DMRes.req)

The SC_DMRes.req message is a unicast management message sent by the SC to the domain master and 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 SC and 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 – SC_DMRes.req message format

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 Confirmation of resignation message (SC_DMRes.cnf)

The SC_DMRes.cnf message is a unicast management message sent by the domain master to the SC to confirm resignation of the nodes requested by the SC. The MMPL of the SC_DMRes.cnf message shall be as presented in Table 9-18.

Table 9-18 – SC_DMRes.cnf message format

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 requested to be expelled from the domain.
	2	[7:0]	Status code (node resigned, not resigned).
…	…	…	…
Entry n	(2×n)–1	[7:0]	DEVICE_ID of the last node requested to be expelled from the domain.
	(2×n)	[7:0]	Status code (node resigned, not resigned).

9.2.5.3 Key updating messages

9.2.5.3.1 Re-authentication and key-update request (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 and NN keys or NMK only. 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
Type of the key	0	[1:0]	00 for NSC, 01 for NN or for NMK. 10, 11 are reserved by ITU-T.
Reserved		[7:2]	Reserved by ITU-T (Note).
Authenticator	1	[7:0]	Shall be set to FF_{16} if NSC is to be updated and set to DEVICE_ID of one of the addressees if NN is to be updated (both for unicast and multicast).
Attempt number	2 and 3	[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.			

Annex A

Application protocol convergence sub-layer

(This annex forms an integral part of this Recommendation.)

Application protocol convergence (APC) specific sub-layer 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.

LSB	MSB
6 octets	Destination address
6 octets	Source address
0-TG octets	EtherType/TAGs
2 octets	MAC client length/type
Application dependent	Service data unit (APDU payload)
4 octets	Frame check sequence (FCS)

G.9961(10)_FI-1-Ann-A

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 TAG 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. If AE provides neither frame_type, nor access_priority or user_priority primitives, the EtherType/TAGs field of the APDU shall be zero octets long.

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/type" field of the APDU. The length of the unencrypted part of the MPDU depends on the length *TG* of the EtherType/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 4 – 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 LSB 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.

In-band management data units generated by the DLL management entity shall follow the LCDU 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].
- 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.

Bibliography

[b-IETF RFC 791] IETF RFC 791 (1981), *Internet Protocol, DARPA Internet Program, Protocol Specification*.

[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*.

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
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	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	Terminals and subjective and objective assessment methods
Series Q	Switching and signalling
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 and next-generation networks
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2011