International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

G.9961

Corrigendum 1

(12/2011)

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

Corrigendum 1

Recommendation ITU-T G.9961 (2010) – Corrigendum 1

ITU-T

International
Telecommunication
Union

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

Corrigendum 1

Summary

Corrigendum 1 to Recommendation ITU-T G.9961 (2010) addresses several protocol issues. These include issues related to both the PHY layer and DLL multicast, channel estimation, tunnel termination, routing and topology management, management message OPCODEs and security.

History

Edition	Recommendation	Approval	Study Group
1.0	ITU-T G.9961	2010-06-11	15
1.1	ITU-T G.9961 (2010) Cor. 1	2011-12-16	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

Page

1) Clause 2 1

2) Clause 3 1

3) Clause 4 1

4) Clause 8.1.1 1

5) Clause 8.1.2 2

6) Clause 8.1.3.1.1 3

7) Clause 8.1.3.2 4

8) Clause 8.3 5

9) Clause 8.4 7

10) Clause 8.5 11

10.1) Clauses 8.5.3 and 8.5.4 11

10.2) Clauses 8.5.6.1 and 8.5.6.2 18

10.3) Clause 8.5.7 19

11) Clause 8.6 25

11.1) Clauses 8.6.1 to 8.6.2.2.2 25

11.2) Clause 8.6.2.2.5 35

11.3) Clause 8.6.2.3 46

11.4) Clauses 8.6.4 to 8.6.6 60

12) Clause 8.8 73

13) Clause 8.9 82

14) Clause 8.10 83

15) Clause 8.11 90

16) Clause 8.12 109

17) Clause 8.16 112

18) Clause 8.17 121

19) Clause 9 129

20) Annex A 142

Recommendation ITU-T G.9961

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

Corrigendum 1

1) Clause 2

Add the following reference to clause 2:

[ITU-T G.9963] Recommendation ITU-T G.9963 (2011), *Unified high-speed wireline-based home networking transceivers – Multiple input/multiple output specification.*

2) Clause 3

Revise the definition of connection in clause 3 as follows:

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

- Source ~~DEVICE_~~ID (SID),
- Destination ~~DEVICE_~~ID (DID),
- Multicast indication~~or~~ (MI)~~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 identifier (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.~~,

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

- Management queue flag (MQF).

3) Clause 4

Add abbreviations to clause 4 as follows:

…	
DA	Destination Address
…	
LAAT	Local Address Association Table
…	
MSID	Multicast Stream Identifier
…	
RAAT	Remote Address Association Table
…	
RMAP-A	Relayed Medium Access Plan – Active
RMAP-D	Relayed Medium Access Plan – Default

4) Clause 8.1.1

Revise the text in clause 8.1.1 as follows:

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 – 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 clause 8.1.1, renumber NOTE 2 as NOTE.

5) Clause 8.1.2

Revise the text in clause 8.1.2 as follows:

8.1.2 Application protocol convergence (APC)

…

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

…

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

The local ~~A~~address ~~A~~association ~~T~~table (LAAT) contains in its first entry the MAC address of the node itself (i.e., REGID) and, in the rest of the table, the~~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 ~~LAAT data is passed to the DLL management entity for network management purposes (see clause 8.5.3).

The remote address association table~~In its "remote" part,~~ (RAAT) stores MAC addresses of ~~the~~ other nodes in the domain and their associated clients~~ associated with other nodes~~ that were advertised on the network.

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

6) Clause 8.1.3.1.1

Revise the text in clause 8.1.3.1.1 and its sub-clauses as follows:

8.1.3.1.1 LLC frame header fields

The LLC frame header (LFH) is composed of the fields described in Table 8-1. The LFH is 6~~5~~ octets long if no time stamp is present and 10~~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
MCSTI		[14]	Multicast stream indicator	Clause 8.1.3.1.1.11
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
DestinationNode	4	[7:0]	Destination identifier that indicates the node(s) to which the LLC frame is finally destined.	Clause 8.1.3.1.1.10
BRCTI	5~~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~~6 to 9~~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.				

…

8.1.3.1.1.5 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 ~~in~~excludes the CCMP header and MIC. It is formatted as a 14-bit unsigned integer.

…

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's best estimate of the domain master's transmit ~~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.

8.1.3.1.1.10 DestinationNode

The DestinationNode field indicates one or more nodes to which the LLC frame is finally destined. In the case of unicast (BRCTI=0, MCSTI=0), it shall be set to the DEVICE_ID of the final destination node. In the case of multicast (BRCTI=0, MCSTI=1), it shall be set to the multicast stream identifier (MSID) assigned by the multicast source. In the case of broadcast (BRCTI=1, MCSTI=0), it shall be set to BROADCAST_ID by the originating node.

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

8.1.3.1.1.11 MCSTI

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

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

7) Clause 8.1.3.2

Revise the text in clause 8.1.3.2 and its sub-clauses as follows:

8.1.3.2 Generation of LPDUs

…

Multiple LLC frames carrying APDUs (data LLC frames) associated with the same data connection can be concatenated to form a data LLC frame block. LLC frames containing LCDUs (management LLC frames) that belong to the same management connection can be concatenated to form a management LLC frame block. The LLC frame block may also include LLC frames intended to be relayed that are associated with the same connection. The number of concatenated LLC frames for an LLC frame block is determined by DLL management entity and is vendor discretionary. The order of LLC frames of the same user priority in the LLC frame block (see Figure 8-5) containing APDUs shall be the same as the order of arrival of the APDUs sourcing these frames. The order of LLC frames in the LLC frame block containing LCDUs shall be the same as the order that these LCDUs are generated by the DLL management entity. The order of bytes in the LLC frame payload shall be the same as in sourcing APDUs or LCDUs, and in the same relative order, bytes shall be passed to the MAC as LPDUs that the MAC maps into MPDU. Mixing data LLC frames and management LLC frames into the same LLC frame block (mixed LLC frame block) is allowed only if the lowest user priority associated with the corresponding prioritized data connection is equal to or greater than six and the highest user priority associated with the corresponding prioritized data connection is equal to seven (i.e., in case three or more priority queues are supported as described in Table III.1 of [ITU-T G.9960]). In this case LCDUs shall be mapped to the same prioritized data connection where APDUs with user priority 7 for the same destination are mapped. Mixed LLC frame blocks are not allowed for data connections associated with service flows. A ~~M~~mixed LLC frame block ~~is~~ shall be 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.

…

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 s~~S~~equence n~~N~~umber (SSN) is a 16-bit field indicating the order of segments that are associated with a connection. The SSN shall be initialized to START_SSN (see clause 7.1.2.3.2.2.20 of [ITU-T G.9960]) ~~zero~~ for the first valid segment that belongs to a new connection and shall be incremented by 1 for each new valid segment that is associated with this connection that follows the current segment. 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) Clause 8.3

Revise the text in clause 8.3 and its sub-clauses as follows:

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

…

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

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

…

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 g~~G~~roup i~~I~~nformation extension (see clause 8.8.4.1.3). The maximum number of groups within a STXOP is ~~255~~ 127 groups due to the limitation on the number of TXOP descriptors in the MAP, describing TSs, within a single STXOP~~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.

…

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.

A node that intends to contend for transmission in the CBTS shall listen and signal in both PRS regardless of whether it has an MPDU ready for transmission prior to PRS0 or not. If no MPDU is ready for transmission prior to PRS0, this process shall be executed as if it has an MPDU with priority MA0 (i.e., no PR signal transmission in PRS0). If the node receives an MPDU of higher MA priority than the priority signalled in PRS0, it may participate in the priority resolution during PRS1 as if the MPDU was ready for transmission prior to PRS0.

~~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.~~

…

8.3.3.6.2 TS grid synchronization recovery

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

- if a sequential TS assignment rule is used, the node shall resynchronize with the TS grid identity using the CURRTS field of the received PHY-frame header.

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

…

8.3.5 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-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-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]).

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

…

8.3.7 Bidirectional transmissions

…

If using acknowledged bidirectional transmission, the BMSG PHY frames shall use the format described in Tables 7-43 and 7-51~~44~~ of [ITU-T G.9960], and the BACK PHY frames shall use the format described in Tables 7-45 and 7-52~~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).

…

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-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]).

…

9) Clause 8.4

Revise the text in clause 8.4 and its sub-clauses as follows:

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~~ec~~	5 ms~~ec~~	5 ms~~ec~~
CYCLE_MAX	Maximum duration of MAC cycle	2 AC cycles (Note 1)	100 ms~~ec~~	100 ms~~ec~~	100 ms~~ec~~
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
MIN_SYM_VAR_AIFG	The minimum number of ~~payload~~ symbols following the header required in a frame to use receiver specific T_{AIFG}, instead of $T_{AIFG\text{-}D}$, as the AIFG gap, between the frame and the following immediate acknowledgment.	2	5	5	2
T_{RCIFG}	Inter frame gap between RTS and CTS	110 µs~~ec~~	29 µs~~ec~~	29 µs~~ec~~	74 µs~~ec~~
T_{CCIFG}	Inter frame gap between CTS and MSG frame	110 µs~~ec~~	29 µs~~ec~~	29 µs~~ec~~	74 µs~~ec~~
$T_{BM2BAIFG}$	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

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

Parameter	Description	Medium			
		Power-line baseband (Note 2)	Coax BB	Coax RF	Phoneline
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~~ec~~	10 ns~~ec~~	10 ns~~ec~~	10 ns~~ec~~
MAP_TX_ SETUP_TIME	The minimal time between the MAP and the MAC cycle it describes	2 ms~~ec~~	2 ms~~ec~~	2 ms~~ec~~	2 ms~~ec~~
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~~ec~~	2.5 ms~~ec~~	2.5 ms~~ec~~	2.5 ms~~ec~~
DEFAULT_TBFCS_TIMEOUT	Default timeout used for closing the CBTS in "Timeout-based from CBTS start" mode	3.5 ms~~ec~~	2.5 ms~~ec~~	2.5 ms~~ec~~	2.5 ms~~ec~~
DEFAULT_ERR_CWOI_ TIMEOUT	Default timeout for error conditions in "CBTS without INUSE" (see clause 8.3.3.4.5.3)	4.9 ms~~ec~~	3.61 ms~~ec~~	3.61 ms~~ec~~	4.04 ms~~ec~~
REG_RESP_TIME~~OUT~~	The maximum time ~~out for~~within which the domain master ~~to~~ shall respond to registration request (see clause 8.6.1.1.1)	200 ms~~ec~~	200 ms~~ec~~	200 ms~~ec~~	200 ms~~ec~~
REG_RETRY_TIMEOUT	Timeout for node to retry registration (see clause 8.6.1.1.1)~~)~~	1 s~~ec~~	1 s~~ec~~	1 s~~ec~~	1 s~~ec~~
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~~ec~~	200 ms~~ec~~	200 ms~~ec~~	200 ms~~ec~~
MAX_RES_ATTEMPTS	Max number of resignation attempts	4	4	4	4
CNM_TIMEOUT	Timeout associated with release of connections	200 ms~~ec~~	200 ms~~ec~~	200 ms~~ec~~	200 ms~~ec~~
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~~ec~~	100 ms~~ec~~	100 ms~~ec~~	100 ms~~ec~~
N_{MCST}	Maximal number of retransmissions of the MC_GrpInfoUpdate.ind message	2	2	2	2
$T_{\mathrm{DM\_UPDATE}}$	The domain master broadcasts the updated topology information, within this time duration, after receiving topology updates.	~~4~~100 ms~~ec~~	~~4~~100 ms~~ec~~	~~4~~100 ms~~ec~~	~~4~~100 ms~~ec~~

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

Parameter	Description	Medium			
		Power-line baseband (Note 2)	Coax BB	Coax RF	Phoneline
T_{N_RSP}	A node replies to the request for topology information from the domain master within this time duration, after receiving the request.	~~4~~100 ms~~ec~~	~~4~~100 ms~~ec~~	~~4~~100 ms~~ec~~	~~4~~100 ms~~ec~~
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.	~~4~~100 ms~~ec~~	~~4~~100 ms~~ec~~	~~4~~100 ms~~ec~~	~~4~~100 ms~~ec~~
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~~ec~~	1 ms~~ec~~	1 ms~~ec~~	1 ms~~ec~~
JOIN_INTERVAL_T_0	The time interval after a node's initialization, during which the node refrains from transmitting and tries to detect MAP frames or RMAP frames associated with the domain that the node intends to join.	10 s~~ec~~	~~2~~10 s~~ec~~	~~2~~10 s~~ec~~	~~2~~10 s~~ec~~
SYM_BOOST_TYPE	Valid types of symbol boost	00_2, 01_2	N/A	N/A	00_2, 01_2
SYM_BOOST_AMOUNT	Valid amounts of symbol boost	0.0 dB, 0.8 dB	N/A	N/A	0.0 dB, 0.8 dB
MAX_RESP_TIME	The maximum time within which a node shall respond to the received management message.	100 ms~~ec~~	100 ms~~ec~~	100 ms~~ec~~	100 ms~~ec~~
MAX_WAIT_TIME	The time that a node shall wait for an expected response after transmitting a management message before inferring the loss of the transmitted message or the response from the responding node or both.	200 ms~~ec~~	200 ms~~ec~~	200 ms~~ec~~	200 ms~~ec~~
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 at the time of registration (see clause 8.6.1.1.4.1).					

8.4.1 General parameters for management message timeout

There are two general timeout-related parameters, MAX_RESP_TIME and MAX_WAIT_TIME, associated with the exchange of management messages. A node is referred to as the initiating node when it transmits a management message such as ".req" message. A node is referred to as the responding node when it is expected to respond back to the initiating node with a management message such as ".cnf" message. The parameter MAX_RESP_TIME specifies the maximum time available to the responding node until it responds to the initiating node. The parameter MAX_WAIT_TIME specifies the time that the initiating node waits after transmitting a management message, before inferring the loss of either the transmitted message or the response from the responding node or both. The relationship between management messages exchanged in a protocol with these parameters is illustrated in Figure 8-24.1. The values for these parameters are defined in Table 8-14, and shall apply to all management protocols defined in this Recommendation except the management protocols that specify values different from these parameters (e.g., registration protocol in clause 8.6.1).

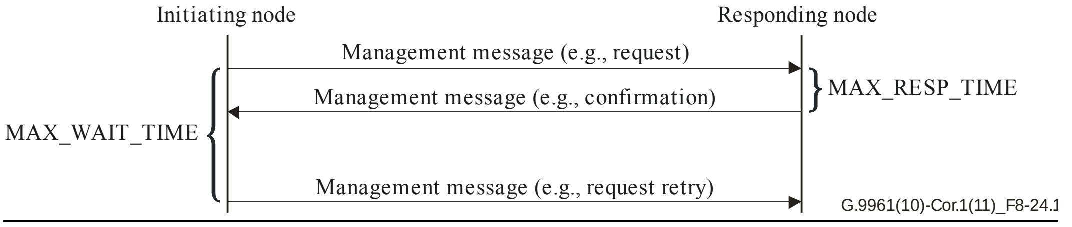

Initiating node
Responding node
Management message (e.g., request)
Management message (e.g., confirmation)
MAX_RESP_TIME
MAX_WAIT_TIME
Management message (e.g., request retry)
G.9961(10)-Cor.1(11)_F8-24.1

Figure 8-24.1 – General parameters for management message timeout

10) Clause 8.5

10.1) Clauses 8.5.3 and 8.5.4

Replace clauses 8.5.3 and 8.5.4 with the following:

8.5.3 Routing of ADPs

Each node shall inform the domain master 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 MAC addresses associated with applications above its A-interface as well as its own MAC address. This list is referred to as a local address association table (LAAT). Each node shall also maintain the list of MAC addresses associated with the AEs of other nodes in the domain and the MAC addresses of those nodes. This list is referred to as a remote address association table (RAAT). 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.

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

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 (case B of Table 8-14.1), 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 (case D of Table 8-14.1), the node shall associate this ADP with a destination MSID and it shall send the APDU using DLL multicast transmission. The node may send the APDU to the appropriate nodes using unicast transmissions until the DLL multicast paths toward the appropriate nodes are established. The node may send the APDU using a

combination of DLL multicast and DLL unicast transmissions until the relevant DLL multicast path is established.

NOTE 1 – The association between the group of MAC addresses and addressed nodes 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.

- If the destination address of the ADP is a standard broadcast address (FFFFFFFFFFFF$_{16}$) or a multicast address intended to reach all the nodes of the domain (case E of Table 8-14.1) or is not found in the LAAT or RAAT of the node (case G of Table 8-14.1), then the BRCTI bit in the LFH of the LLC frame carrying the corresponding APDU shall be set to one, so that the 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$_{16}$, the corresponding APDU shall also be forwarded to the local DLL management entity.

NOTE 2 – For ADP with EtherType different from 22E3$_{16}$ and the standard broadcast address as the DA of that ADP, sending the corresponding APDU to the local DLL management entity is vendor discretionary.

- If the destination address of a received ADP is found in the local AAT and it is not the MAC address of the node (case A of Table 8-14.1), the ADP shall be dropped without notification.
- If the destination address of a received ADP is the MAC address of the node (case C of Table 8-14.1), the node shall pass the corresponding APDU to its DLL management entity.
- If the destination address of a received ADP is the reserved MAC address 01-19-A7-52-76-96 (case F of Table 8-14.1), the node shall pass the corresponding APDU to its DLL management entity

Table 8-14.1 – Routing of ADPs

Case	**Ethernet frame type**	**ADP Destination address**	**Routing**	**Example**
A	Unicast frame	In LAAT, except node's MAC address	Drop the message	Any kind of traffic
B	Unicast frame	In RAAT	Look for the DestinationNode defined for this DA	Normal routing of frames coming through the A interface (can be normal Ethernet or remote in-band messages)
C	Unicast frame	Node's MAC address	Send to management	Local in-band message
D	Multicast frame	Multicast address	The node has the choice to treat this multicast transmission as several DLL unicast transmissions or using a DLL multicast stream	IGMP/MLD Ethernet frames
E	Broadcast frame	Broadcast address or multicast address for all nodes in domain	If EtherType = 22E3$_{16}$ send to management treat this broadcast transmission using BRT (BRCTI=1; DestinationNode = BROADCAST_ID) and route following the BRT rules	Normal broadcast

Table 8-14.1 – Routing of ADPs

Case	Ethernet frame type	ADP Destination address	Routing	Example
F	Unicast frame	Reserved address	Send to management	
G	Unicast Frame	Unknown	Send using BRT (like case E)	Any kind of traffic

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 a broadcasted APDU or LCDU that was received from the medium from a specified root nodes. 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 PHY 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})).

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, and set the DestinationNode of the LFH field to FF_{16}. The DA of the broadcasted frame may be any address, including the standard broadcast address ($FFFFFFFFFFFF_{16}$).

A node that receives a broadcast LLC frame (APDU or LCDU, BRCTI = 1) from the medium, shall first perform the filtering procedure according to the BRT as 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 (as indicated in some of the cases specified in Tables 8-14.2 and 8-14.3) 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 received from the medium a broadcast LLC frame that contains an LCDU with DestinationNode different from BROADCAST_ID or its own DEVICE_ID, it shall relay the LLC frame as indicated by the BRT (cases 1 and 2 of Table 8-14.2).

If a node received from the medium a broadcast LLC frame that contains an LCDU with DestinationNode equal to the node's DEVICE_ID (case 3 of Table 8-14.2), it shall recover this LCDU and treat it as an unicast frame for relaying purposes (see clause 8.5.7). The node shall not relay the broadcast LLC frame.

If a node received from the medium a broadcast LLC frame that contains an LCDU with DestinationNode equal to BROADCAST_ID:

- If the node is a leaf node:
  - ○ If the DA of that LCDU is the MAC address of the node, the standard broadcast address or the reserved MAC address 01-19-A7-52-76-96 (cases 6, 7 and 8 of Table 8-14.2), the node shall recover this LCDU and pass it to the DLL management. In addition, the node shall stop the broadcast of the LLC frame.
  - ○ In all other cases (cases 4, 5 and 9 of Table 8-14.2), the LLC frame shall be dropped and not relayed.

- If the node is a non-leaf node:
  - ○ If the DA of that LCDU is the standard broadcast address or the reserved MAC address 01-19-A7-52-76-96, the node shall recover this LCDU and pass it to the DLL management (cases 13 and 14 of Table 8-14.2). In addition, the node shall relay that LLC frame as indicated by the BRT.
  - ○ If the DA of that LCDU is the MAC address of the node (case 12 of Table 8-14.2), the node shall recover this LCDU and pass it to the DLL management. In addition, the node may relay the LLC frame as indicated by the BRT.
  - ○ In all other cases (cases 10, 11 and 15 of Table 8-14.2), the LLC frame shall be relayed as indicated by the BRT.

If a node received from the medium a broadcast LLC frame that contains an APDU with DestinationNode different from BROADCAST_ID or its own DEVICE_ID, it shall relay the LLC frame as indicated by the BRT (cases 16 and 17 of Table 8-14.3).

If a node received from the medium a broadcast LLC frame that contains an APDU with DestinationNode equal to the nodes DEVICE_ID (cases 18 and 19 of Table 8-14.3), it shall recover this APDU and treat it as an unicast frame for relaying purposes (see clause 8.5.7). The node shall not relay the broadcast frame.

If a node received from the medium a broadcast LLC frame that contains an APDU with DestinationNode equal to BROADCAST_ID, it shall:

- If the node is a leaf node:
  - ○ If the DA of that APDU is the address of the DLL management or the reserved MAC address 01-19-A7-52-76-96 (cases 22 and 24 of Table 8-14.3), the node shall recover this APDU and shall pass it to the DLL management. In addition, the node shall stop the broadcast of the LLC frame.
  - ○ If the DA of that APDU is the standard broadcast address (case 23 of Table 8-14.3), the node shall recover this APDU and shall pass it to the DLL management and to the A-interface. In addition, the node shall stop the broadcast of the LLC frame.
  - ○ If the DA of that APDU is in the LAAT (case 20 of Table 8-14.3), the node shall recover this APDU and shall pass it to the A-interface. In addition, the node shall stop the broadcast of the LLC frame.
  - ○ If the DA of that APDU is in the RAAT (case 21 of Table 8-14.3), the node shall recover this APDU and may pass it to the A-interface. In addition, the node shall stop the broadcast of the LLC frame.
  - ○ In all other cases (i.e., the DA of that APDU is unknown, case 25 of Table 8-14.3), the LLC frame shall be passed to the A-interface and not relayed.
- If the node is a non-leaf node
  - ○ If the DA of that APDU is the address of the DLL management (case 28 of Table 8-14.3), the node shall recover this APDU and shall pass it to the DLL management. In addition, the node may stop relaying the LLC frame.
  - ○ If the DA of that APDU is the address of the reserved MAC address 01-19-A7-52-76-96 (case 30 of Table 8-14.3), the node shall recover this APDU and shall pass it to the DLL management. In addition, the node shall relay the LLC frame as indicated in the BRT.
  - ○ If the DA of that APDU is the standard broadcast address (case 29 of Table 8-14.3), the node shall recover this APDU and shall pass it to the DLL management and to the A-interface. In addition, the node shall relay the LLC frame.

○ If the DA of that APDU is in the LAAT (case 26 of Table 8-14.3), the node shall recover this APDU and shall pass it to the A-interface. In addition, the node may relay the LLC frame as indicated in the BRT.

○ If the DA of that APDU is in the RAAT (case 27 of Table 8-14.3), the node may recover this APDU and may pass it to the A-interface. In addition, the node may relay the LLC frame as indicated in the BRT.

○ In all other cases (i.e., the DA of that APDU is unknown, case 31 of Table 8-14.3), the LLC frame shall be passed to the A interface and relay the LLC frame following the BRT.

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

Case	**Type of broadcast**	**Leaf/ Non-leaf**	**LCDU DA**	**Broadcasting actions**	**Example**
1	Broadcast frame intended for another node in the network (BRCTI = 1; MCSTI = 0; DestinationNode = $\text{DeviceID}_{\text{OtherNode}}$)	Leaf	–	Drop the frame (Note 2)	Unicast frame not found by a previous relay node in its internal unicast routing tables and relayed in broadcast
2		Non-Leaf	–	Follow BRT rules	Unicast frame not found by a previous relay node in its internal unicast routing tables and relayed in broadcast
3	Broadcast frame intended for this node (BRCTI = 1; MCSTI = 0; DestinationNode = $\text{DeviceID}_{\text{Node}}$)	Leaf/ Non leaf	–	Consider frame as non-broadcast (unicast) and follow the corresponding rules (cases 1-6 of Table 8-14.4). Stop the broadcast	End point of a unicast frame not found by a previous relay node in its internal unicast routing tables and relayed in broadcast
4	Broadcast frame intended for all the nodes (BRCTI = 1; MCSTI = 0; DestinationNode = BroadcastID)	Leaf	In LAAT except node's MAC address	Drop the frame	Not applicable
5			In RAAT	Drop the frame	
6			Node's MAC address	Pass the frame to DLL management. Stop the broadcast through BRT	
7			Broadcast address	Pass the frame to DLL management. Stop the broadcast through BRT	
8			Reserved address	Pass the frame to DLL management Stop the broadcast through BRT	Management message intended to all nodes
9			Unknown	Drop the frame	

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

Case	Type of broadcast	Leaf/ Non-leaf	LCDU DA	Broadcasting actions	Example
10		Non-Leaf	In LAAT except node's MAC address	Forward through BRT (Note 1)	Not applicable
11			In RAAT	Forward through BRT (NOTE – RAAT is not consulted)	
12			node's MAC address	Pass the frame to DLL management; Optional: forward through BRT	
13			Broadcast address	Pass the frame to DLL management and forward through BRT	
14			Reserved address	Pass the frame to DLL management and forward through BRT	Management message intended to all nodes
15			Unknown	Forward through BRT	
NOTE 1 – LAAT is not consulted. NOTE 2 – Following BRT rules leads to a drop.					

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

Case	Type of broadcast	Leaf/ Non-leaf	APDU DA	Broadcasting actions	Example
16	Broadcast frame intended for another node in the network (BRCTI = 1; MCSTI = 0; DestinationNode = DeviceID$_{OtherNode}$)	Leaf	–	Drop the frame	Unicast frame not found by a previous relay node in its internal unicast routing tables and relayed in broadcast
17		Non-Leaf	–	Follow BRT rules. Apply filtering	Unicast frame not found by a previous relay node in its internal unicast routing tables and relayed in broadcast
18	Broadcast frame intended for this node (BRCTI = 1; MCSTI = 0; DestinationNode = DeviceID$_{Node}$)	Leaf	–	Consider the frame as non-broadcast (unicast) and follow the corresponding rules (cases 1-6 of Table 8-14.5). Stop the broadcasting	End point of an unicast frame not found by a previous relay node in its internal unicast routing tables and relayed in broadcast

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

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

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

Case	Type of broadcast	Leaf/ Non-leaf	APDU DA	Broadcasting actions	Example
31			Unknown	Pass to A interface. Relay through BRT	Unknown Destination frames broadcast
NOTE 1 – LAAT is not consulted. NOTE 2 – RAAT is not consulted.					

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).

10.2) Clauses 8.5.6.1 and 8.5.6.2

Revise clauses 8.5.6.1 and 8.5.6.2 as follows:

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

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

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

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

10.3) Clause 8.5.7

Replace clause 8.5.7 as follows. Note that clauses 8.5.7.1 and 8.5.7.2 are new additions.

8.5.7 Relaying messages

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

8.5.7.1 Relaying of LCDU

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

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

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

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

b) In all other cases (cases 7, 8 and 12 of Table 8-14.4), the node shall drop the LLC frame.

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

If the frame has been received with BRCTI = 0, MCSTI = 0 and the DestinationNode is not present in the unicast routing tables (cases 14 and 15 of Table 8-14.4), the node shall set the BRCTI to 1 and broadcast the received LLC frame to the nodes that are specified in the branch list in the BRT the DestinationNode.

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

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

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

Case	Type of relaying	Leaf/ Non-leaf	LCDU DA	Relaying actions	Example
1	Unicast frame intended to the node (BRCTI = 0; MCSTI = 0; Destination Node = $\text{DeviceID}_{\text{Node}}$)	Leaf/ Non leaf	In LAAT except node's MAC address	Send to DLL management	
2			In RAAT	Send to DLL management	Management message with proxy (ADM_NodeRegisterRequest.req)
3			Node's MAC address	Send to management	Management message (or remote in-band message)
4			Broadcast address	Send to management	
5			Reserved address	Send to management	
6			Unknown	Send to DLL management	
7	Unicast frame with Destination Node = 0 (BRCTI = 0; MCSTI = 0; Destination Node = 0)	Leaf/Non leaf	In LAAT except node's MAC address	Drop the frame	Not pplicable
8			In RAAT	Drop the frame	Not applicable
9			node's MAC address	Send to management	ADM_DMRegistrResponse.cnf (with/without proxy)
10			Broadcast address	Send to management	Not applicable
11			Reserved address	Send to management	Not applicable
12			Unknown	Drop	
13	Unicast frame not intended to the node but with known Destination Node (BRCTI = 0; MCSTI = 0; Destination Node = $\text{DeviceID}_{\text{OtherNode}}$)	–	–	Use unicast routing tables	Normal "relaying"

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

Case	Type of relaying	Leaf/ Non-leaf	LCDU DA	Relaying actions	Example
14	Unicast frame not intended to the node but with an unknown Destination Node (BRCTI = 0; MCSTI = 0; Destination Node = Unknown)	Leaf	–	Drop the frame (Note 1)	During transient periods
15		Non-Leaf	–	Broadcast the unicast frame (Note 2)	During transient periods
16	Multicast frame where the relay node does not belong to the group (BRCTI = 0; MCSTI = 1; Destination Node = MulticastID) $\text{DeviceID}_{\text{Node}} \notin$ MulticastID	–	–	Follow multicast rules to forward the frame through the DLL multicast tree	
17	Multicast frame where the relay node does not know the MSID (BRCTI = 0; MCSTI = 1 Destination Node = MulticastID) Unknown MulticastID	–	–	Drop the frame	

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

Case	Type of relaying	Leaf/ Non-leaf	LCDU DA	Relaying actions	Example
18	Multicast frame where the relay node belongs to the group (BRCTI = 0; MCSTI = 1 Destination Node = MulticastID) $\text{DeviceID}_{\text{Node}} \in$ MulticastID	–	–	Apply same rules than unicast frame intended to the node (cases 1-6 of Table 8-14.4) and follow the multicast rules to forward the frame through the multicast tree	
NOTE 1 – Follow the BRT rules, leading to a drop. NOTE 2 – Keep DestinationNode and OriginatingNode; set BRCTI = 1 and route using the BRT.					

8.5.7.2 Relaying of APDU

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

If the frame has been received with BRCTI = 0, MCSTI = 0 and the DestinationNode is the DEVICE_ID of the receiving node, the node shall extract the APDU and:

a) If the DA is the same as the MAC address of the node or the standard broadcast address or the reserved MAC address 01-19-A7-52-76-96 (cases 3, 4 and 5 of Table 8-14.5), the node shall pass it to the DLL management.

b) If the DA is found in the LAAT of the node (cases 1 of Table 8-14.5), the node shall pass it to the A-interface

c) If the DA is found in the RAAT of the node (cases 2 of Table 8-14.5), the node may pass it to the A-interface

d) If the DA is unknown (case 6 of Table 8-14.5), the node shall send it through the A-interface.

If the frame has been received with BRCTI = 0, MCSTI = 0 and the DestinationNode is not the DEVICE_ID of the receiving node and it is found in the unicast routing tables (case 7 of Table 8-14.5), the node shall relay it to the appropriate node or nodes as indicated in the routing table.

If the frame has been received with BRCTI = 0, MCSTI = 0 and the DestinationNode is not the DEVICE_ID of the receiving node and it is not found in the unicast routing tables (cases 8, 9, 10 and 11 of Table 8-14.5) the node shall broadcast the received LLC frame by setting the BRCTI to 1 and maintaining the DestinationNode.

If the frame has been received with BRCTI = 0, MCSTI = 1 and with a known MSID (cases 12 and 14 of Table 8-14.5), the frame shall be relayed as specified in clause 8.17. In addition, if the node is a member of the DLL multicast stream (case 14 of Table 8-14.5), the node shall extract the APDU and follow the same rules as the case where the frame has been received with BRCTI = 0, MCSTI = 0 and the DestinationNode is the DEVICE_ID of the receiving node. If the frame has been received with an unknown MSID (case 13 of Table 8-14.5), the frame shall be dropped and inform the transmitter as specified in clause 8.17.

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

Table 8-14.5 – Relaying of LLC frames (APDU case)

Case	**Type of relaying**	**Leaf/ Non-leaf**	**APDU DA**	**Relaying action**	**Example**
1	Unicast frame intended to the node (BRCTI = 0; MCSTI = 0; Destination Node = $\text{DeviceID}_{\text{Node}}$)	Leaf/ non leaf	In LAAT except node's MAC address	Send through A interface	Normal data frame
2			In RAAT	Optional to send frame through A interface	Handover of equipment~~s~~ between different ITU-T G.9960 nodes
3			Node's MAC address	Send to management	firmware upgrade, ping, etc.
4			Broadcast address	Send to management	
5			Reserved address	Send to management	
6			Unknown	Send through A interface	Can happen in a corner case (e.g., ageing)
7	Unicast frame not intended to the node but with known Destination Node (BRCTI = 0; MCSTI = 0; Destination Node = $\text{DeviceID}_{\text{OtherNode}}$)	–	–	Use unicast routing tables	Normal "relaying"
8	Unicast frame with Destination Node = 0 (BRCTI = 0; MCSTI = 0; Destination Node = 0)	Leaf	–	Drop the frame	During transient periods
9		Non-Leaf	–	Broadcast the unicast frame (Note)	During transient periods

Table 8-14.5 – Relaying of LLC frames (APDU case)

Case	Type of relaying	Leaf/ Non-leaf	APDU DA	Relaying action	Example
10	Unicast frame not intended to the node but with unknown Destination Node (BRCTI = 0; MCSTI = 0; Destination Node = Unknown)	Leaf	–	Drop the frame	During transient periods
11		Non-Leaf	–	Broadcast the unicast frame (Note)	During transient periods
12	Multicast frame where the relay node does not belong to the group (BRCTI = 0; MCSTI = 1; Destination Node = MulticastID) $\text{DeviceID}_{\text{Node}} \notin$ MulticastID	–	–	Follow multicast rules to forward the frame through the multicast tree	
13	Multicast frame where the relay node does not know the group (BRCTI = 0; MCSTI = 1; Destination Node = MulticastID) Unknown MulticastID	–	–	Drop the frame	

Table 8-14.5 – Relaying of LLC frames (APDU case)

Case	Type of relaying	Leaf/ Non-leaf	APDU DA	Relaying action	Example
14	Multicast frame where the relay node belongs to the group (BRCTI = 0; MCSTI = 1; Destination Node = MulticastID) DeviceID$_{Node}$ ∈ MulticastID	–	–	Send to A interface and if it a relay node for this multicastID then forward it to the proper nodes	
NOTE – Keep DestinationNode and OriginatingNode; BRCSTI = 1					

11) Clause 8.6

11.1) Clauses 8.6.1 to 8.6.2.2.2

Revise the text as follows for clauses 8.6.1 to 8.6.2.2.2:

…

8.6.1 Network admission

Revise the last paragraph as follows:

…

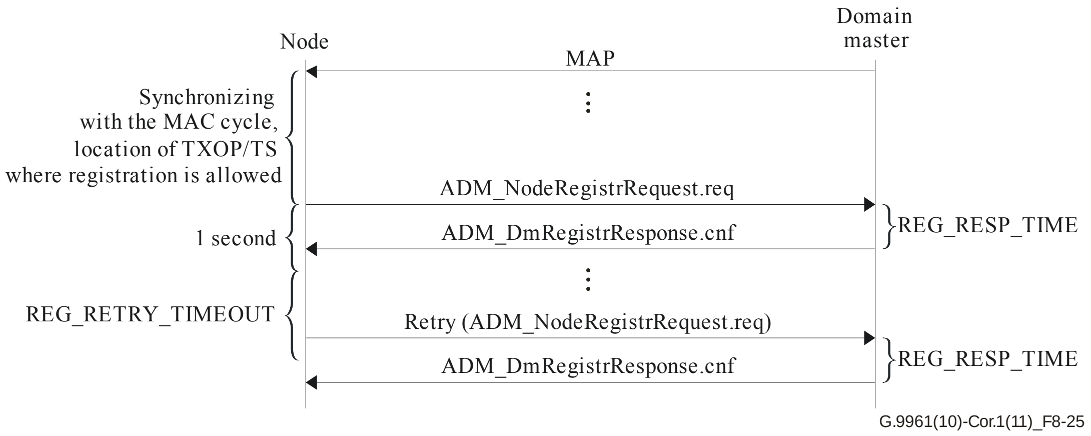
The DEVICE_ID (and OriginatingNode) of the registering node shall be set to zero. After registration (for an unsecure domain) or authentication (for a secure domain) is complete, the DEVICE_ID shall be set to the value assigned by the domain master, as described in clause 8.7.1.1. A node shall not establish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

8.6.1.1 Network admission protocol

8.6.1.1.1 Registration into the domain

Revise Figure 8-25 and paragraph 3 after the figure as follows:

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


Node
Domain master
MAP
Synchronizing with the MAC cycle, location of TXOP/TS where registration is allowed
ADM_NodeRegistrRequest.req
1 second
ADM_DmRegistrResponse.cnf
REG_RESP_TIME
REG_RETRY_TIMEOUT
Retry (ADM_NodeRegistrRequest.req)
ADM_DmRegistrResponse.cnf
REG_RESP_TIME
G.9961(10)-Cor.1(11)_F8-25

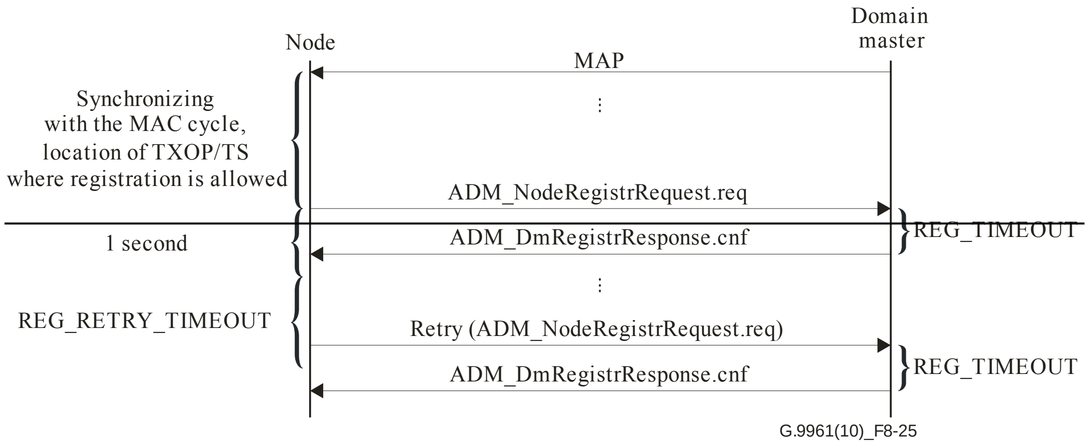

Node
Domain master
MAP
Synchronizing with the MAC cycle, location of TXOP/TS where registration is allowed
ADM_NodeRegistrRequest.req
1 second
ADM_DmRegistrResponse.cnf
REG_TIMEOUT
REG_RETRY_TIMEOUT
Retry (ADM_NodeRegistrRequest.req)
ADM_DmRegistrResponse.cnf
REG_TIMEOUT
G.9961(10)_F8-25

Figure 8-25 – Protocol diagram describing node registration

…

The domain master shall process the registration request and shall reply within REG_RESP_TIME~~OUT~~ 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. The DestinationNode of the LLC frame containing the ADM_DmRegistrResponse.cnf message shall be set to zero.

…

8.6.1.1.2 Periodic re-registrations

Revise paragraph 2 as follows:

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.

…

8.6.1.1.3.2 Forced resignation

Revise paragraph 2 as follows:

If the domain master does not receive the reply within 200 ms, it shall repeat the request and wait for the ADM_NodeResignRequest.req message again. If the reply is again not received within 200 ms, the domain master shall~~ broadcast the ADM_DmResign.cnf~~ transmit the updated TM_DomainRoutingChange.ind message with an update that reflects that the node has been forced to resign and cut off all available bandwidth assignment for the node~~ forced to resign~~. The nodes in the domain shall delete the entries associated with the resigned node from their lists (RAAT etc.).

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 (directly or via a proxy), 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.

…

8.6.1.1.4.2 Registration response message (ADM_DmRegistrResponse.cnf)

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

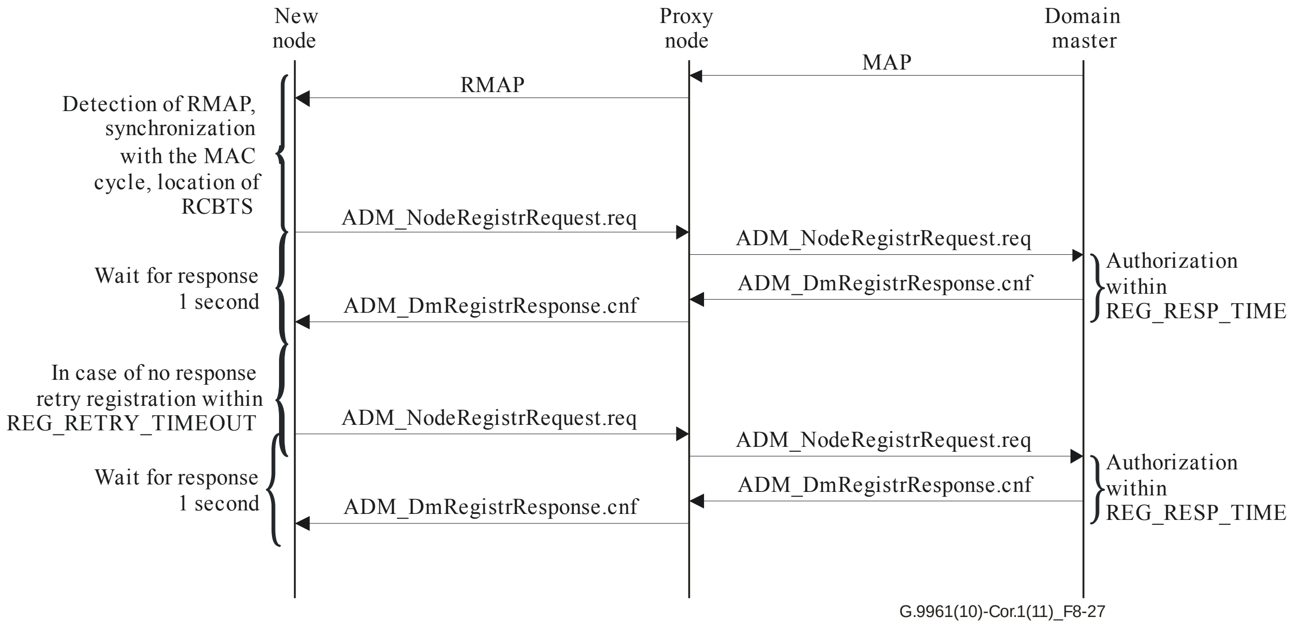
…

8.6.1.2 Admission via Proxy

Replace Figure 8-27 and edit the text from paragraph 2 after the figure as follows:

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.


New node
Proxy node
Domain master
MAP
RMAP
Detection of RMAP, synchronization with the MAC cycle, location of RCBTS
ADM_NodeRegistrRequest.req
ADM_NodeRegistrRequest.req
Authorization within REG_RESP_TIME
Wait for response 1 second
ADM_DmRegistrResponse.cnf
ADM_DmRegistrResponse.cnf
In case of no response retry registration within REG_RETRY_TIMEOUT
ADM_NodeRegistrRequest.req
ADM_NodeRegistrRequest.req
Authorization within REG_RESP_TIME
Wait for response 1 second
ADM_DmRegistrResponse.cnf
ADM_DmRegistrResponse.cnf
G.9961(10)-Cor.1(11)_F8-27

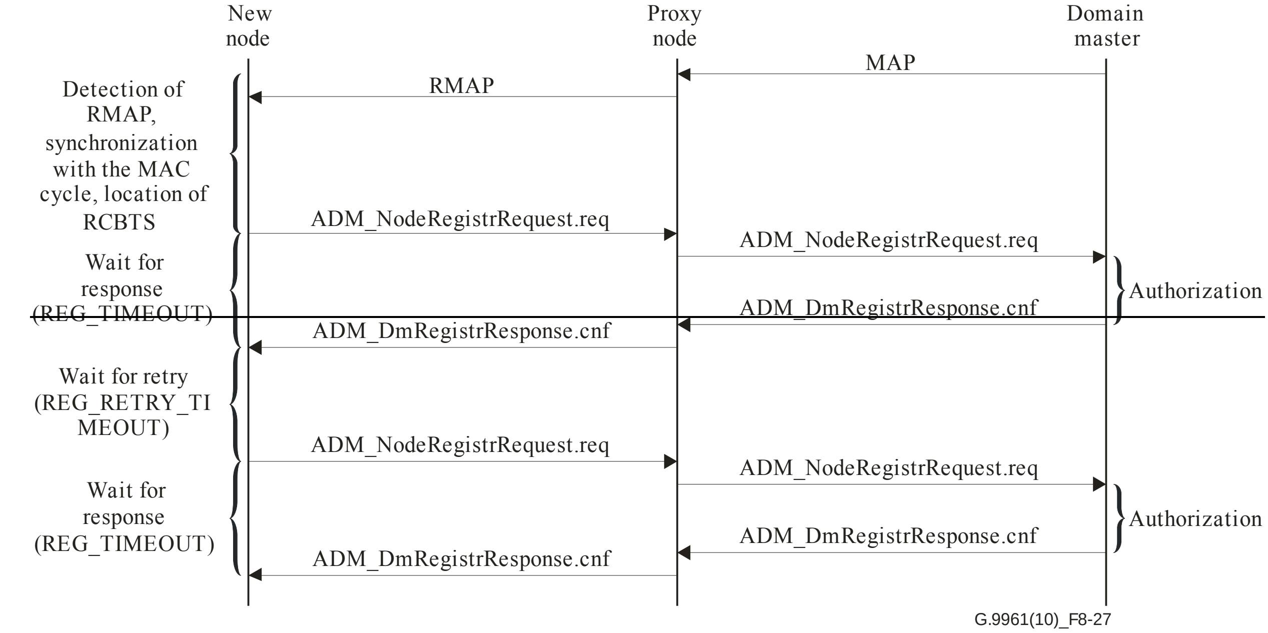

New node
Proxy node
Domain master
MAP
RMAP
Detection of RMAP, synchronization with the MAC cycle, location of RCBTS
ADM_NodeRegistrRequest.req
ADM_NodeRegistrRequest.req
Wait for response (REG_TIMEOUT)
Authorization
ADM_DmRegistrResponse.cnf
ADM_DmRegistrResponse.cnf
Wait for retry (REG_RETRY_TIMEOUT)
ADM_NodeRegistrRequest.req
ADM_NodeRegistrRequest.req
Wait for response (REG_TIMEOUT)
Authorization
ADM_DmRegistrResponse.cnf
ADM_DmRegistrResponse.cnf
G.9961(10)_F8-27

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

…

After receiving the ADM_NodeRegistrRequest.req from the node, the proxy shall relay it towards the domain master. To facilitate relaying of ADM_NodeRegistrRequest.req, the registering node shall use the addressing scheme shown in Table 8-20.1~~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~~.

Table 8-20.1 – Addressing fields of the ADM_NodeRegistrRequest.req from registering node to proxy node

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

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

The proxy shall relay the received ADM_NodeRegistrRequest.req message by recreating a new ADM_NodeRegistrRequest.req with the same contents as the received one and shall send it to the DM using the addressing scheme shown in the Table 8-20.2.

Table 8-20.2 – Addressing fields of the ADM_NodeRegistrRequest.req from proxy to the DM

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

The registering hidden node shall send ADM_NodeRegistrRequest.req only during the CBTS for which registration is allowed, using medium access rules for registration described in clause 8.3.3.4.8. The domain master shall process the ADM_NodeRegistrRequest.req message and reply to the registration proxy with a registration response message (ADM_DmRegistrResponse.cnf, see clause 8.6.1.1.4.2) ~~. The destination address of the LCDU carrying ADM_DmRegistrResponse.cnf shall be the REGID of the proxy~~using the addressing scheme shown in Table 8-20.3.

Table 8-20.3 – Addressing fields of the ADM_NodeRegistrRequest.cnf from DM to the proxy node

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

The registration proxy node shall then unicast the received ADM_DmRegistrResponse.cnf message to the new node using the addressing scheme shown in Table 8-20.4.

Table 8-20.4 – Addressing fields of the ADM_NodeRegistrRequest.cnf from the proxy node to the registering node

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

~~The 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.

…

8.6.2.2.1 Flow establishment

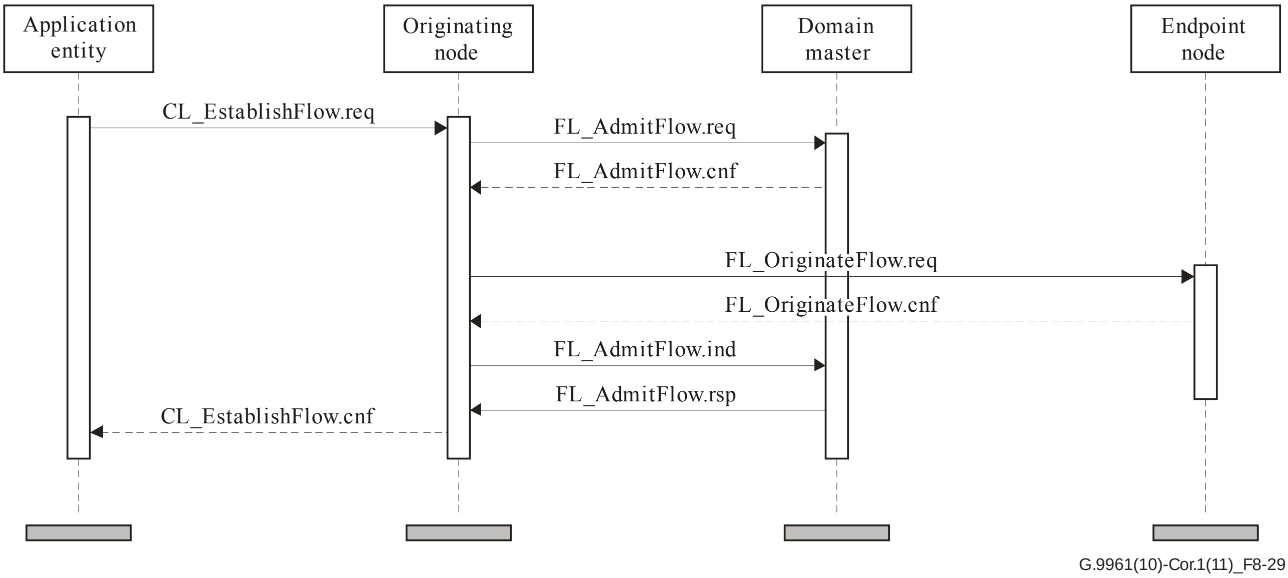
Replace Figures 8-29 and 8-31 and revise the text as follows:

…

- If the endpoint node is unable to support the new flow, it shall notify the originating node by sending the FL_OriginateFlow.cnf message with a failure code. The originating node shall notify the domain master about the flow establishment failure by sending the FL_AdmitFlow.ind message to it and release the allocated FLOW_ID. The domain master shall send the FL_AdmitFlow.rsp message to the originating node and then release the reserved bandwidth for the allocated FLOW_ID.
  - ○ Otherwise, if the endpoint node is able to support the new flow, it shall notify the originating node by sending the FL_OriginateFlow.cnf message with the success code. In case of bidirectional flow, the FL_AdmitFlow.ind message shall contain the FLOW_ID for reverse flow. The originating node shall notify the domain master that

the flow was established successfully by sending the FL_AdmitFlow.ind message with a success code. In case of bidirectional flow the FL_AdmitFlow.ind message shall contain the FLOW_ID for reverse flow in addition to the FLOW_ID for the forward flow. The Domain Master shall acknowledge the FL_AdmitFlow.ind by sending the FL_AdmitFlow.rsp message to the originating node.

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



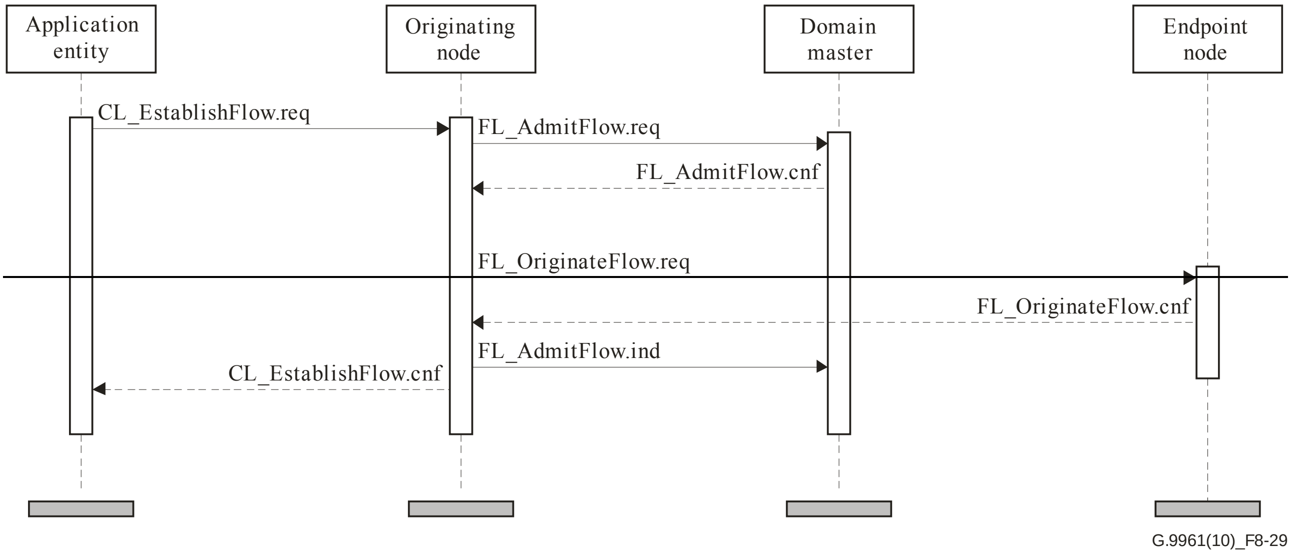


Figure 8-29 – Successful flow establishment

…

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

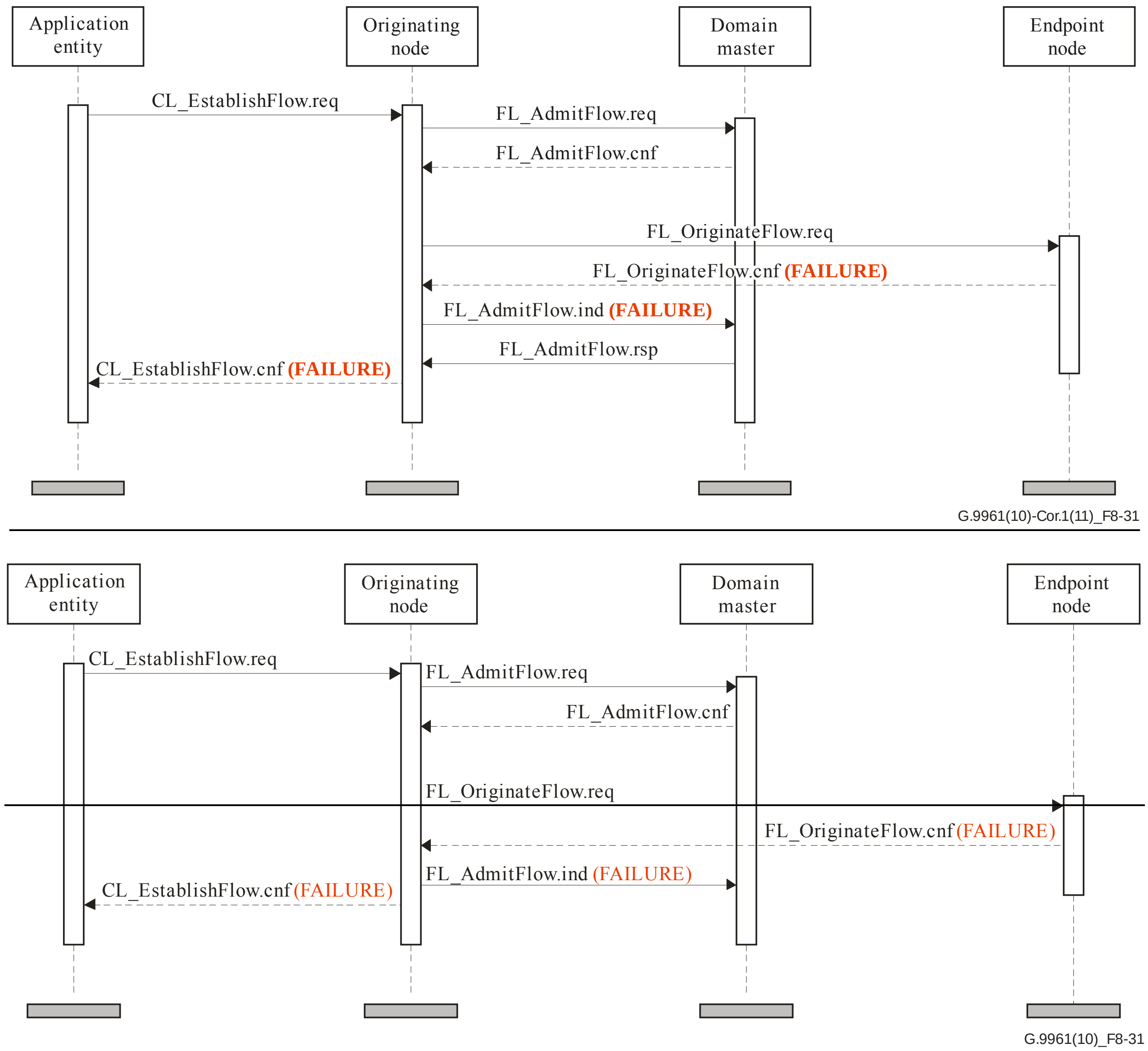


Figure 8-31 – Failure in flow establishment due to rejection by endpoint node

…

8.6.2.2.2 Flow establishment via relay nodes

Replace Figures 8-33, 8-34 and 8-36 and revise the text as follows:

…

i) When the originating node receives the FL_OriginateFlow.cnf message with the success code it shall send the FL_AdmitFlow.ind message to the domain master to notify that the tunnel establishment has completed successfully. The FL_AdmitFlow.ind shall include all the established flows that compose the tunnel. If the request to set up the flow was explicitly sent by the application entity, ~~T~~the originating node shall send to the ~~higher~~ application entity the CL_EstablishFlow.cnf message with success code to the application entity.

j) After the domain master receives the FL_AdmitFlow.ind it shall acknowledge it by sending the FL_AdmitFlow.rsp message and allocate the actual bandwidth resources required to serve the flows that compose the tunnel.

…

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.

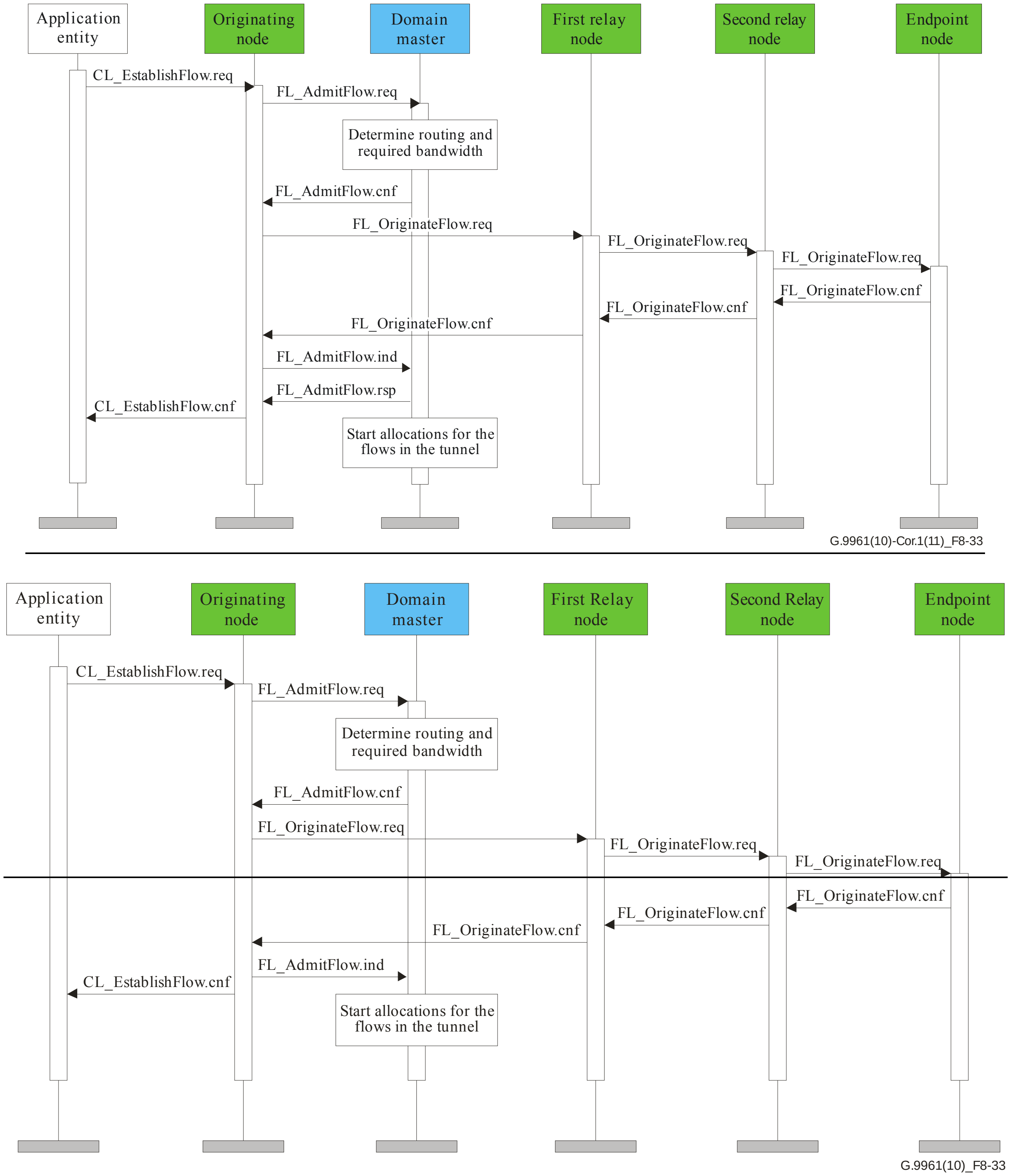
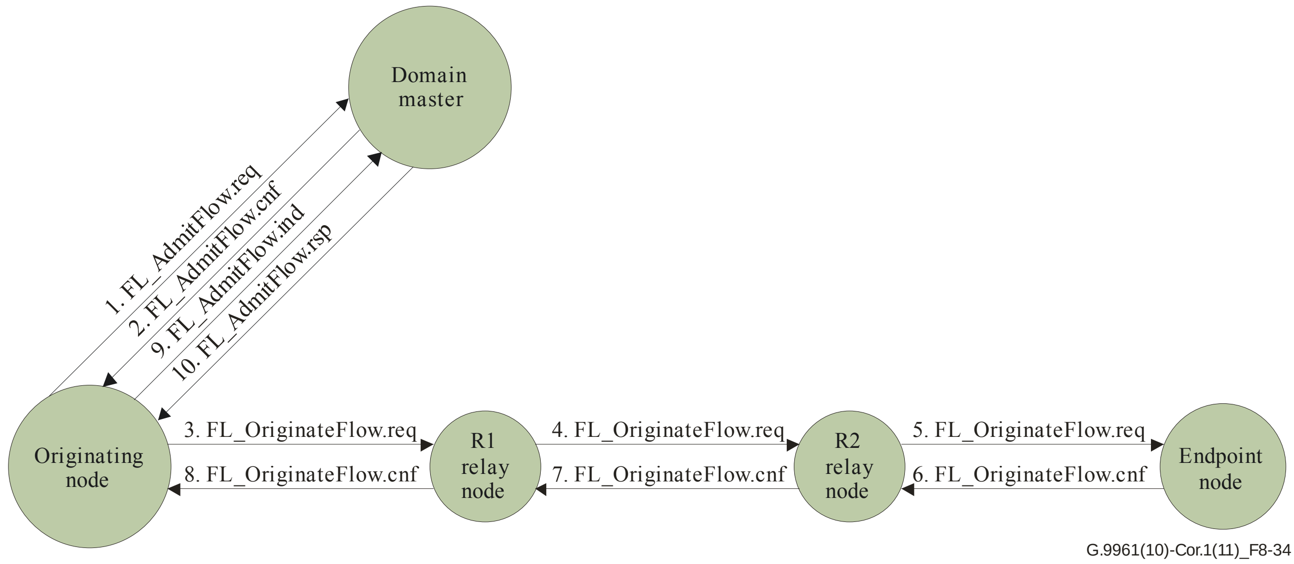


Figure 8-33 – Example of a successful tunnel establishment with two relay nodes

…

Figure 8-34 illustrates a successful tunnel establishment.



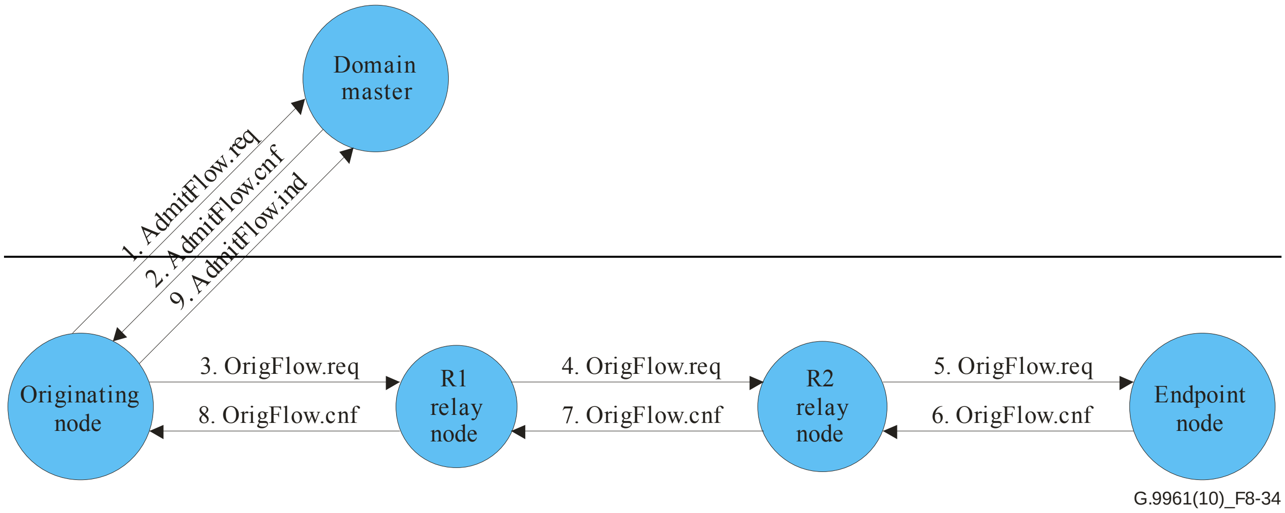
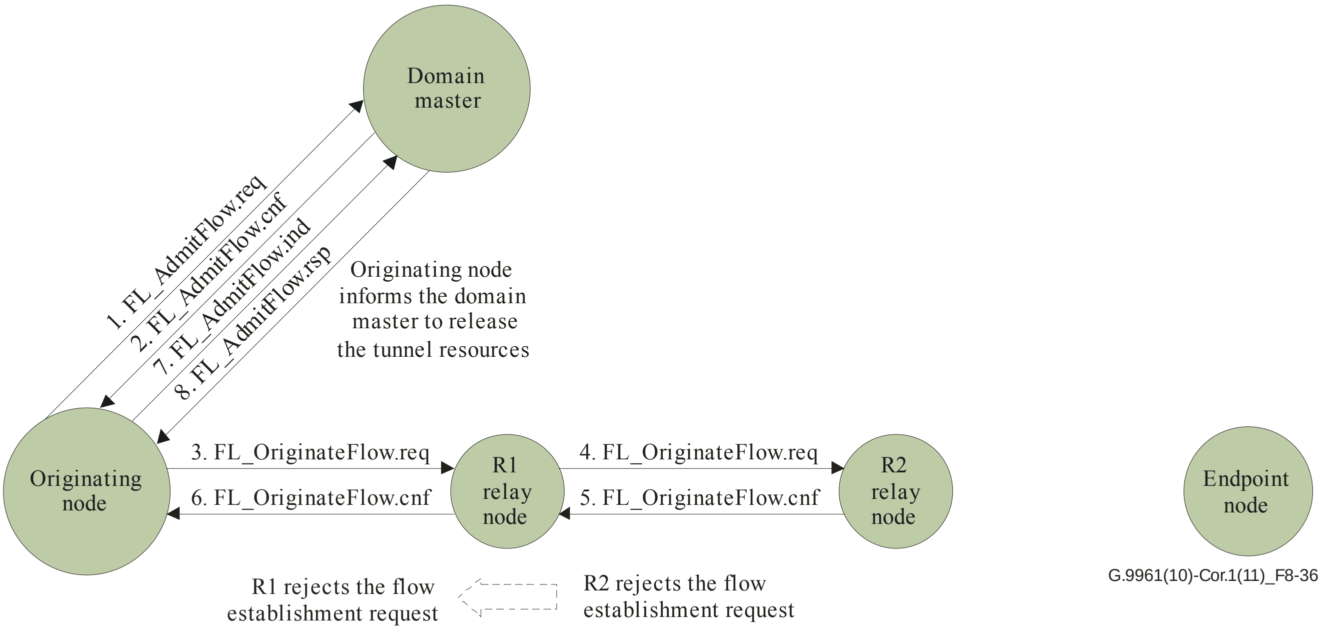


Figure 8-34 – Example of a successful tunnel establishment via two relay nodes

…

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



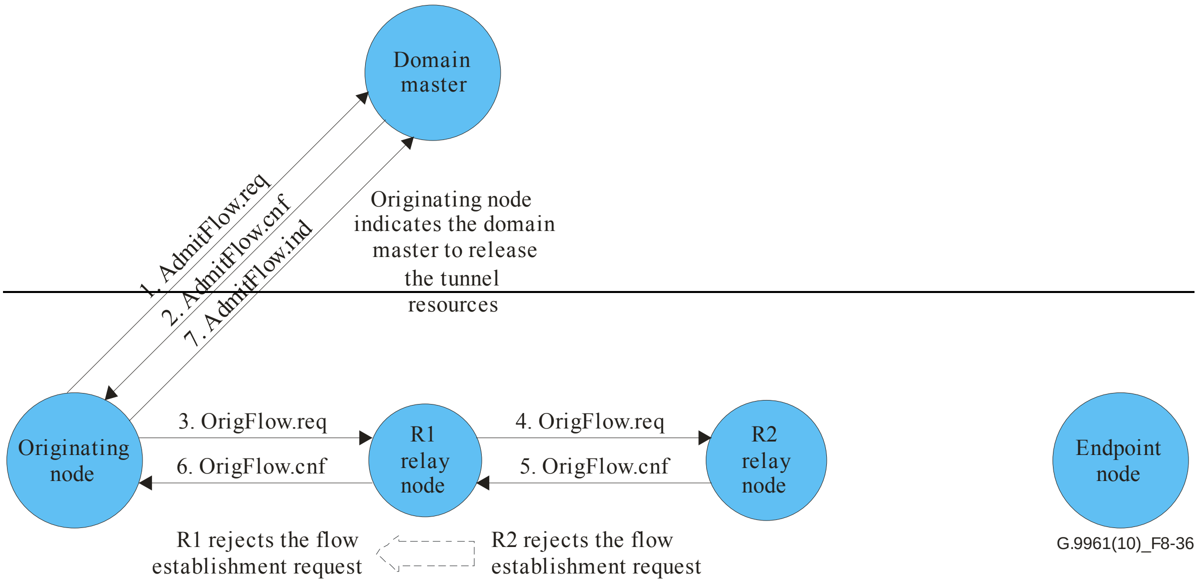


Figure 8-36 – Example of a tunnel establishment failure in the case where the second relay node, R2, rejects the flow establishment request

…

11.2) Clause 8.6.2.2.5

Add clause 8.6.2.2.5 as follows:

8.6.2.2.5 Tunnel reconstruction or termination

The following situations require a tunnel to be terminated:

- After the application entity on the originating node no longer has data to send using a particular flow, it may signal to the originating node that the flow has ended or the originating node may infer that the application entity has finished sending data.

- When a tunnel has been established following automatic traffic classification and the originating node determines that the application entity has finished sending data associated with the tunnel.
- When the domain master determines that there is insufficient bandwidth to support that tunnel.
- When the domain master determines that the originating node or the endpoint node has left the domain.
- Any node in the tunnel that infers a broken link and informs the domain master about the broken link, the domain master may request the originating node to restore the tunnel via another alternative route or request the originating node to terminate the tunnel.

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

8.6.2.2.5.1 Normal tunnel termination

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

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

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

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

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

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

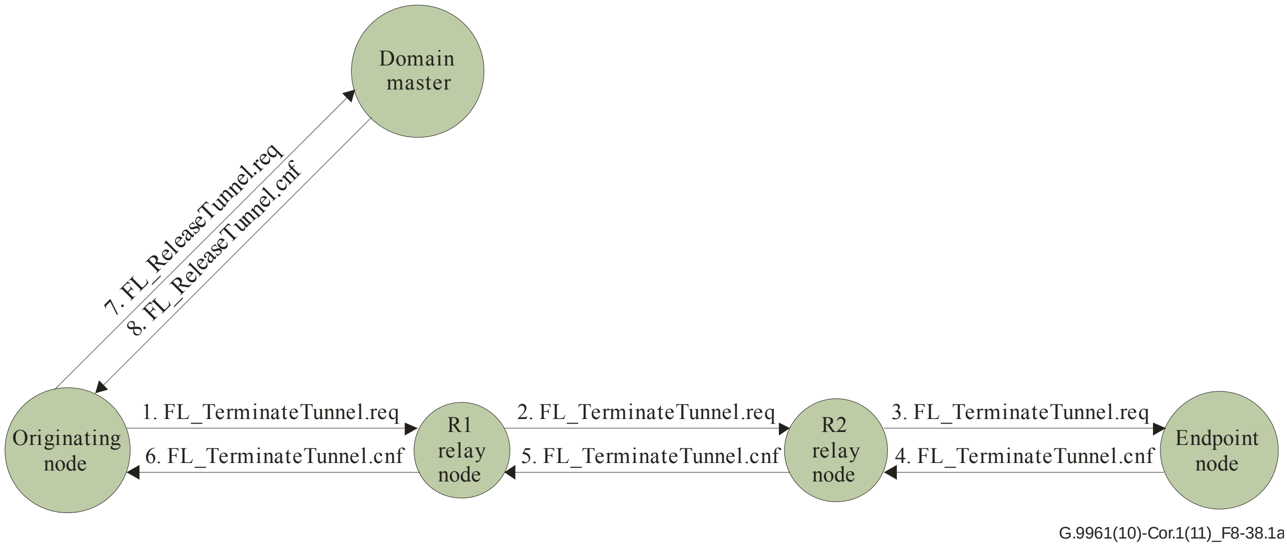

Domain master
7. FL_ReleaseTunnel.req
8. FL_ReleaseTunnel.cnf
Originating node
1. FL_TerminateTunnel.req
6. FL_TerminateTunnel.cnf
R1 relay node
2. FL_TerminateTunnel.req
5. FL_TerminateTunnel.cnf
R2 relay node
3. FL_TerminateTunnel.req
4. FL_TerminateTunnel.cnf
Endpoint node
G.9961(10)-Cor.1(11)_F8-38.1a

Figure 8-38.1a – Normal tunnel termination

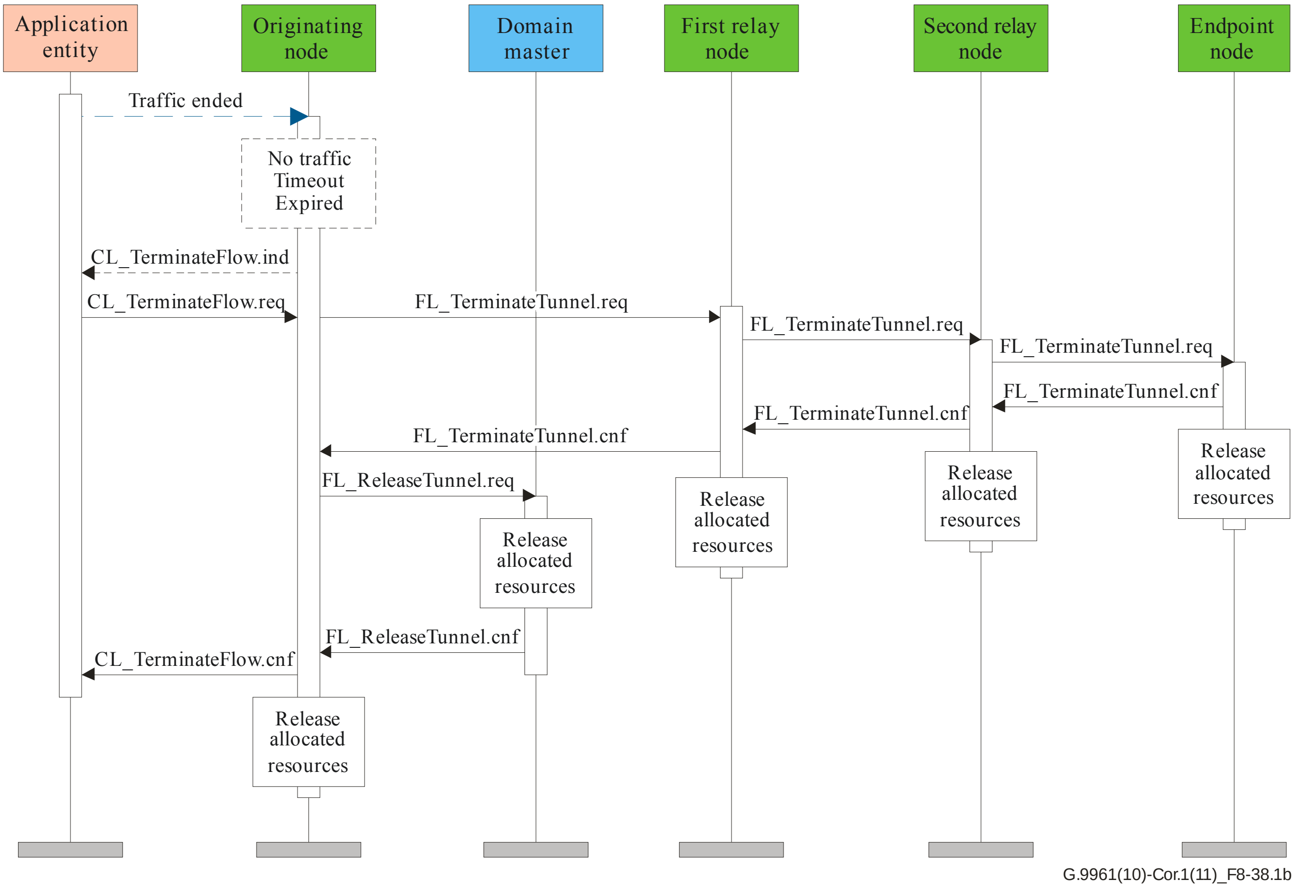


Figure 8-38.1b – MSC of normal termination of a tunnel

8.6.2.2.5.2 Tunnel reconstruction or termination due to a broken link

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

a) When a node that is participating in a tunnel determines that a link to the next node in a tunnel is broken, it shall send an FL_BrokenTunnel.ind message to the domain master to inform about the broken link. The DM shall reply with an FL_BrokenTunnel.rsp message to confirm receiving the FL_BrokenTunnel.ind message.

b) The domain master shall calculate an alternative new path from the originating node towards the endpoint node.

- ○ If the domain master succeeds in calculating an alternative path, it shall reserve the needed resources to support the new alternative path for the tunnel and shall build an FL_DM_RenewTunnel.req message that contains the new alternative path for the tunnel and send the FL_DM_RenewTunnel.req message to the originating node.

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

- ○ Otherwise, if the domain master did not succeed in calculating or allocating an alternative path to reconstruct the broken tunnel, it shall send an FL_TerminateFlow.req message to the originating node, to terminate the tunnel as specified in clause 8.6.2.2.5.3.

c) Upon receiving the FL_ DM_RenewTunnel.req message, the originating node shall build an FL_RenewTunnel.req message based on the received FL_DM_RenewTunnel.req message and send it to the next relay node according to the path as specified in the received FL_DM_TerminateFlow.req message. The FLOW_ID allocated by the originating node to the renewed tunnel shall be the same as the FLOW_ID allocated to the original tunnel with a broken link.

d) If the broken link is between the originating node and the first relay node, as shown in Figure 8-38.2, then the originating node shall replace the old broken flow ('a') with the new established flow 'd' with the new alternative relay node (AR1). The originating node shall establish the flow with AR1 by sending the FL_RenewTunnel.req message to a new alternative relay node (AR1). The originating node shall delete the flow ('b') between the relay node (R1) and the next relay node (R2) by indirectly sending via relay nodes (because the link to R1 is broken) an FL_DeleteFlow.req message to the old relay node (R1).

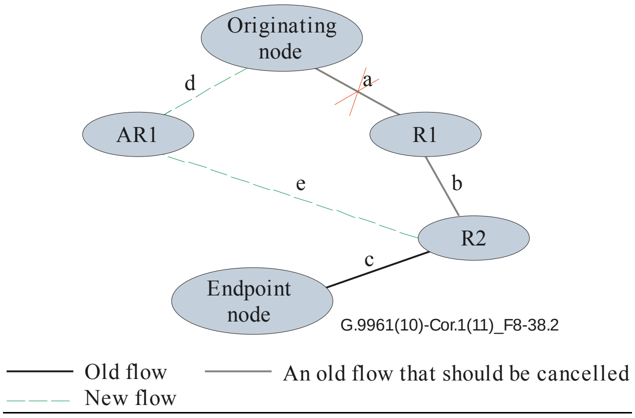

Originating node
d
a
AR1
R1
e
b
R2
c
Endpoint node
G.9961(10)-Cor.1(11)_F8-38.2
Old flow
New flow
An old flow that should be cancelled

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

e) A relay node participating in this tunnel that receives the FL_RenewTunnel.req message and has a valid flow for this tunnel towards the next node in the path as specified in the received FL_RenewTunnel.req message (as relay node R1 in Figure 8-38.3), shall update the FL_RenewTunnel.req message with the actual RelFLOW_ID ('b') that it has with the next node (R2), and then it shall relay the FL_RenewTunnel.req message to the next node (R2).

f) A relay node (as for example relay node R2 in Figure 8-38.3) participating in a tunnel, that receives an FL_RenewTunnel.req message that specifies that it has to reconstruct the tunnel by establishing a flow to a new relay node (AR3) shall do the following actions:

- It shall allocate a FLOW_ID for the new established flow, update the received FL_RenewTunnel.req message with the allocated FLOW_ID (RelFLOW_ID) and send the updated FL_RenewTunnel.req message to the next new relay node (AR3) to establish the flow.
- It shall bind the actual flow ('b' in Figure 8-38.3) that it has with the previous node in the tunnel with the new established flow ('f' in Figure 8-38.3).

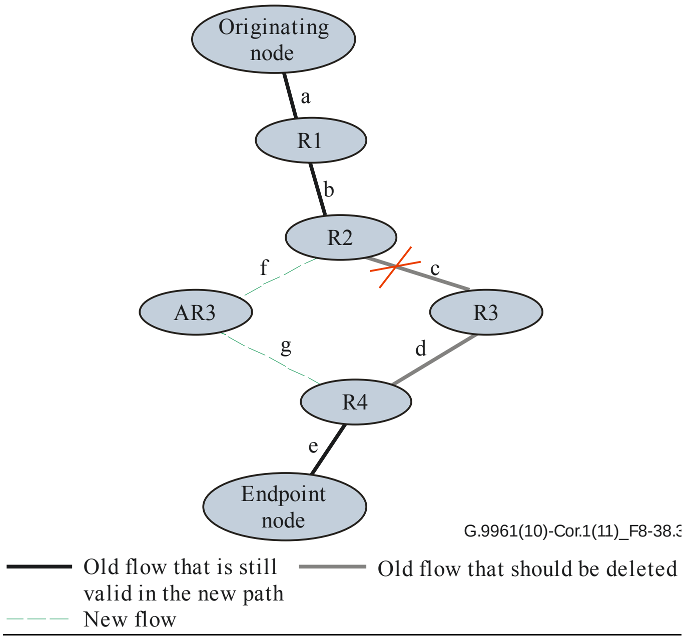

Originating node
a
R1
b
R2
f
c
AR3
R3
g
d
R4
e
Endpoint node
G.9961(10)-Cor.1(11)_F8-38.3
Old flow that is still valid in the new path
Old flow that should be deleted
New flow

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

g) A node (for example AR3 in Figure 8-38.3) that receives an FL_RenewTunnel.req message that requests that the node has to reconstruct a tunnel shall do the following:

   i) It shall save the FLOW_ID (flow f in Figure 8-38.3) that is specified in the FL_RenewTunnel.req message that was established by the node that sent it the FL_RenewTunnel.req message (R2 Figure 8-38.3).

   ii) It shall allocate a FLOW_ID (flow g in Figure 8-38.3) for the flow that it has to establish with the next relay node (R4 in Figure 8-38.3).

   iii) It shall update the received FL_RenewTunnel.req message with the allocated FLOW_ID (RelFLOW_ID) and send the updated FL_RenewTunnel.req message to the next new relay node (R4) to establish the flow.

   iv) It shall bind the flow ('f' in Figure 8-38.3) that it has with the previous node (R2) with the flow ('g' in Figure 8-38.3) that it establishes with the next node (R4).

h) If a relay node that receives an FL_RenewTunnel.req message which specifies that it has to reconstruct the tunnel by establishing a flow to a new relay node cannot establish the new requested flow (lack of internal resources), it shall reply to the node that sent it the FL_RenewTunnel.req message with an FL_RenewTunnel.cnf message with a failure code and abort the tunnel reconstruction procedure. The FL_RenewTunnel.cnf with the failure code shall be relayed by the relay nodes in the tunnel toward the originating node.

i) when a relay node (for example R4 Figure 8-38.3) participating in a tunnel receives an FL_RenewTunnel.req message from a new relay node (AR3 Figure 8-38.3) in the tunnel and it has a valid flow towards the next node in the path specified in FL_RenewTunnel.req message, it shall release the old flow ('d') that it has with the old relay node and bind the new flow specified in the FL_RenewTunnel.req message ('g') with the next flow ('e') that it has with the next node. It shall update the field RelFLOW_ID in the FL_RenewTunnel.req message with the actual FLOW_ID ('e') that it has with the next node (endpoint in Figure 8-38.3) and relay the updated FL_RenewTunnel.req message to the next node (endpoint in Figure 8-38.3).

j) Each relay node along the tunnel path that receives an FL_RenewTunnel.req message shall execute one of the previous steps (d, e, f, g and i) according to its topology state and conditions. It means that the originating node shall do the operations defined in step d and a relay node participating in this tunnel that receives the FL_RenewTunnel.req message and has a valid flow for this tunnel towards the next node in the path as specified in the received FL_RenewTunnel.req message shall execute step e and so on.

k) When the endpoint node receives the FL_RenewTunnel.req message, it shall reply to the relay node that sent it the FL_RenewTunnel.req message with an FL_RenewTunnel.cnf message. If it has an old flow with another relay node for that tunnel, it shall replace the old flow with the new flow (RelFLOW_ID) that is specified in the received FL_RenewTunnel.req message. If the flow is a bidirectional flow, the endpoint shall update the FL_RenewTunnel.cnf message with the BFLOW_ID of the flow in the reverse direction.

l) Each relay node in the tunnel that receives an FL_RenewTunnel.cnf with a success code shall update the FL_RenewTunnel.cnf message by adding to the flows list that compose the tunnel its own established RelFLOW_ID that it has established. If the flow is a bidirectional flow, the relay node shall update the FL_RenewTunnel.cnf message with the assigned BFLOW_ID of the flow in the reverse direction. Then the relay node shall send the FL_RenewTunnel.cnf message as a reply to the node that has sent it the FL_RenewTunnel.req message. This step shall be executed by all intermediate nodes until the FL_RenewTunnel.cnf is sent to the originating node.

m) When the originating node receives the FL_ RenewTunnel.cnf message with a success code it shall build and send an FL_ DM_RenewTunnel.cnf message to the domain master to notify that the tunnel reconstruction has been completed successfully. The FL_DM_RenewTunnel.cnf message shall include all the flows that implement the reconstructed tunnel.

n) The originating node shall send an FL_DeleteFlow.req message to each one of the nodes that were part of the original tunnel and not part of the restored tunnel.

o) After the domain master receives the FL_DM_RenewTunnel.cnf message it shall allocate the bandwidth resources required to serve the flows in the reconstructed tunnel and release allocated resources of the deleted flows.

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

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

8.6.2.2.5.2.1 MSC of tunnel reconstruction for case 1

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

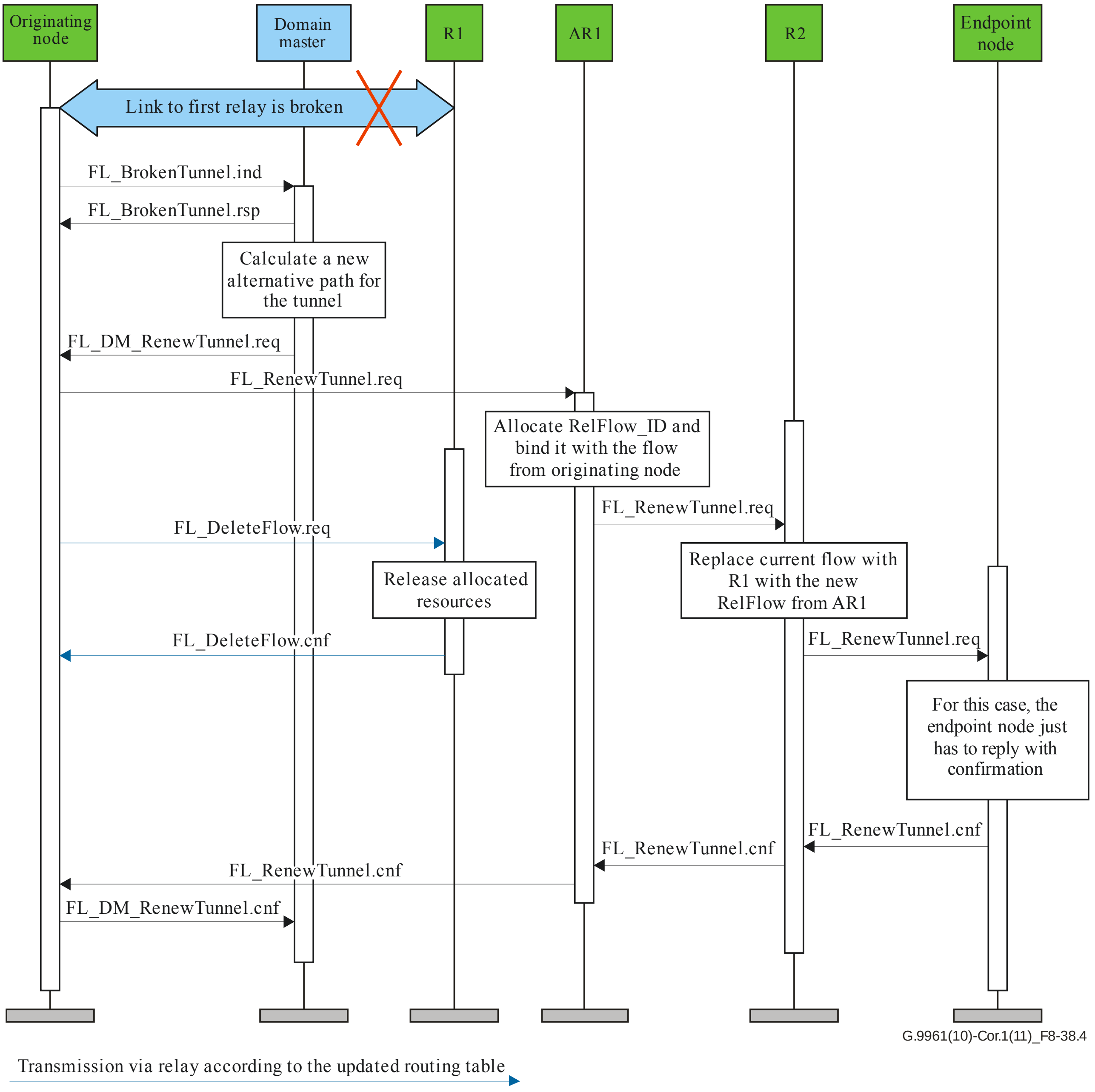
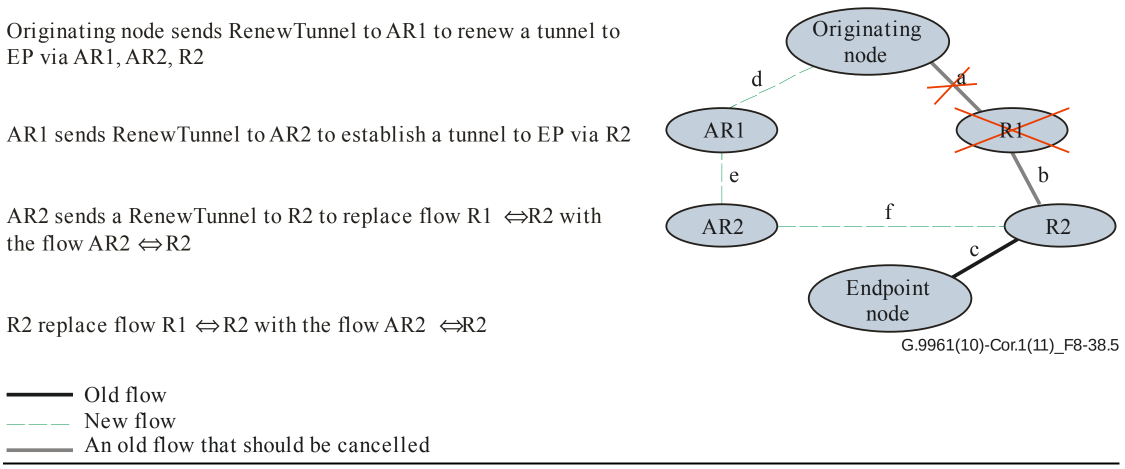

Originating node
Domain master
R1
AR1
R2
Endpoint node
Link to first relay is broken
FL_BrokenTunnel.ind
FL_BrokenTunnel.rsp
Calculate a new alternative path for the tunnel
FL_DM_RenewTunnel.req
FL_RenewTunnel.req
Allocate RelFlow_ID and bind it with the flow from originating node
FL_RenewTunnel.req
FL_DeleteFlow.req
Replace current flow with R1 with the new RelFlow from AR1
Release allocated resources
FL_DeleteFlow.cnf
FL_RenewTunnel.req
For this case, the endpoint node just has to reply with confirmation
FL_RenewTunnel.cnf
FL_RenewTunnel.cnf
FL_RenewTunnel.cnf
FL_DM_RenewTunnel.cnf
G.9961(10)-Cor.1(11)_F8-38.4
Transmission via relay according to the updated routing table

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

8.6.2.2.5.2.2 Tunnel reconstruction for case 2

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



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

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

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

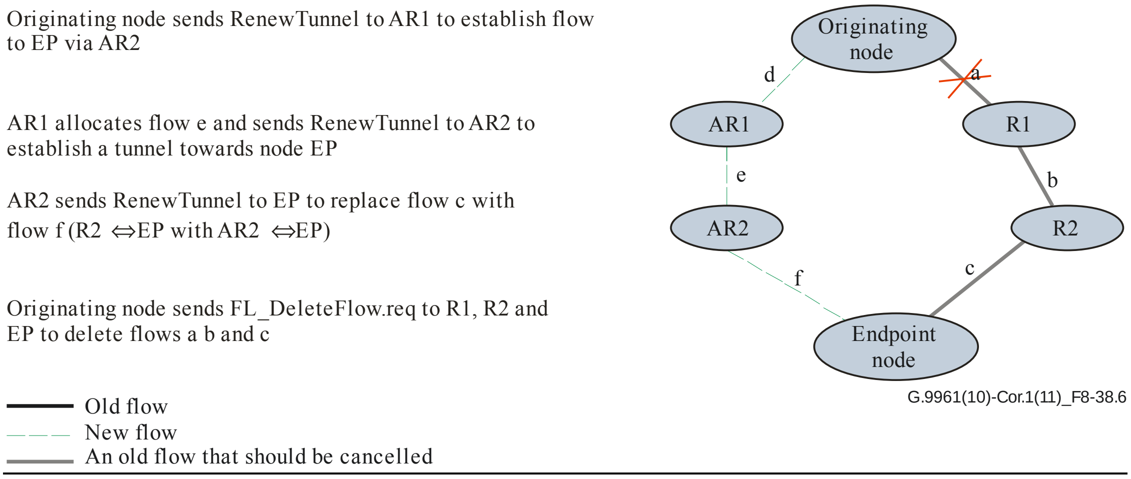
R2 replace flow R1 ⇔R2 with the flow AR2 ⇔R2

Originating node
d
a
AR1
R1
e
b
AR2
f
R2
c
Endpoint node
G.9961(10)-Cor.1(11)_F8-38.5
Old flow
New flow
An old flow that should be cancelled

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

8.6.2.2.5.2.3 Tunnel reconstruction for case 3

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



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

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

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

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

Originating node
d
a
AR1
R1
e
b
AR2
R2
f
c
Endpoint node
G.9961(10)-Cor.1(11)_F8-38.6
Old flow
New flow
An old flow that should be cancelled

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

8.6.2.2.5.2.4 Tunnel reconstruction for case 4

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

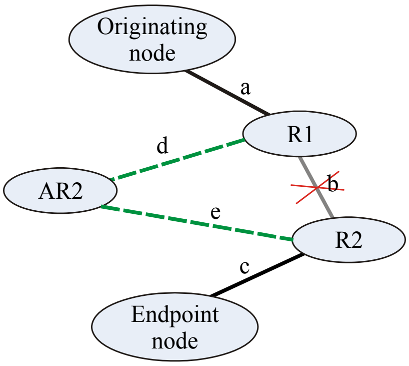
Link b is broken

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

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

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

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

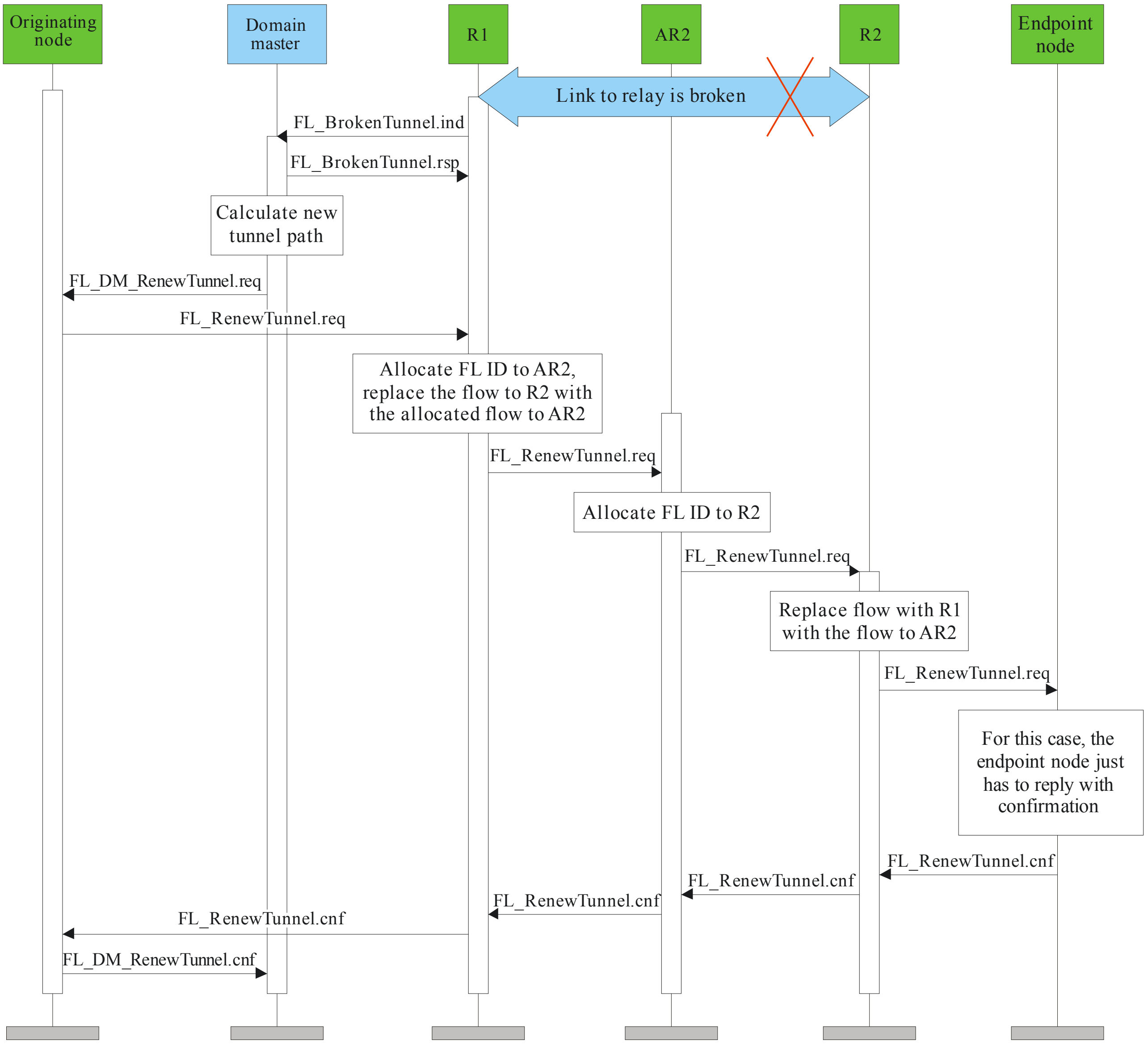


G.9961(10)-Cor.1(11)_F8-38.7

━━ Old flow
╍╍ New flow
━━ An old flow that should be cancelled

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

The following MSC describes the protocol behaviour for case 4



G.9961(10)-Cor.1(11)_F8-38.8

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

8.6.2.2.5.3 Domain master triggered tunnel termination

If the domain master decides to terminate a tunnel due to lack of resources to support the tunnel, it shall send an FL_TerminateFlow.req to the originating node, with the ReasonCode set to 01_{16} (termination by domain master due to lack of resources). The originating node upon receiving the FL_TerminateFlow.req message shall start the tunnel termination protocol as specified in clause 8.6.2.2.5.1 but with one difference: the originating node shall send the FL_TerminateFlow.cnf message to the domain master instead of sending the FL_ReleaseFlow.req message.

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

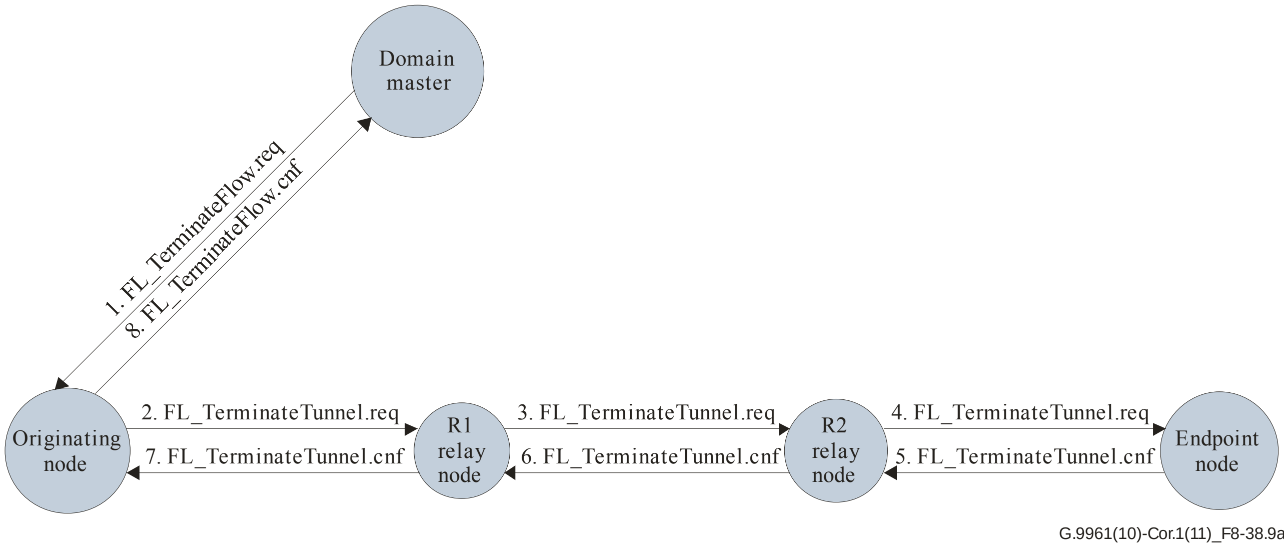

Domain master
1. FL_TerminateFlow.req
8. FL_TerminateFlow.cnf
Originating node
2. FL_TerminateTunnel.req
7. FL_TerminateTunnel.cnf
R1 relay node
3. FL_TerminateTunnel.req
6. FL_TerminateTunnel.cnf
R2 relay node
4. FL_TerminateTunnel.req
5. FL_TerminateTunnel.cnf
Endpoint node
G.9961(10)-Cor.1(11)_F8-38.9a

Figure 8-38.9a – Domain master triggers tunnel termination

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

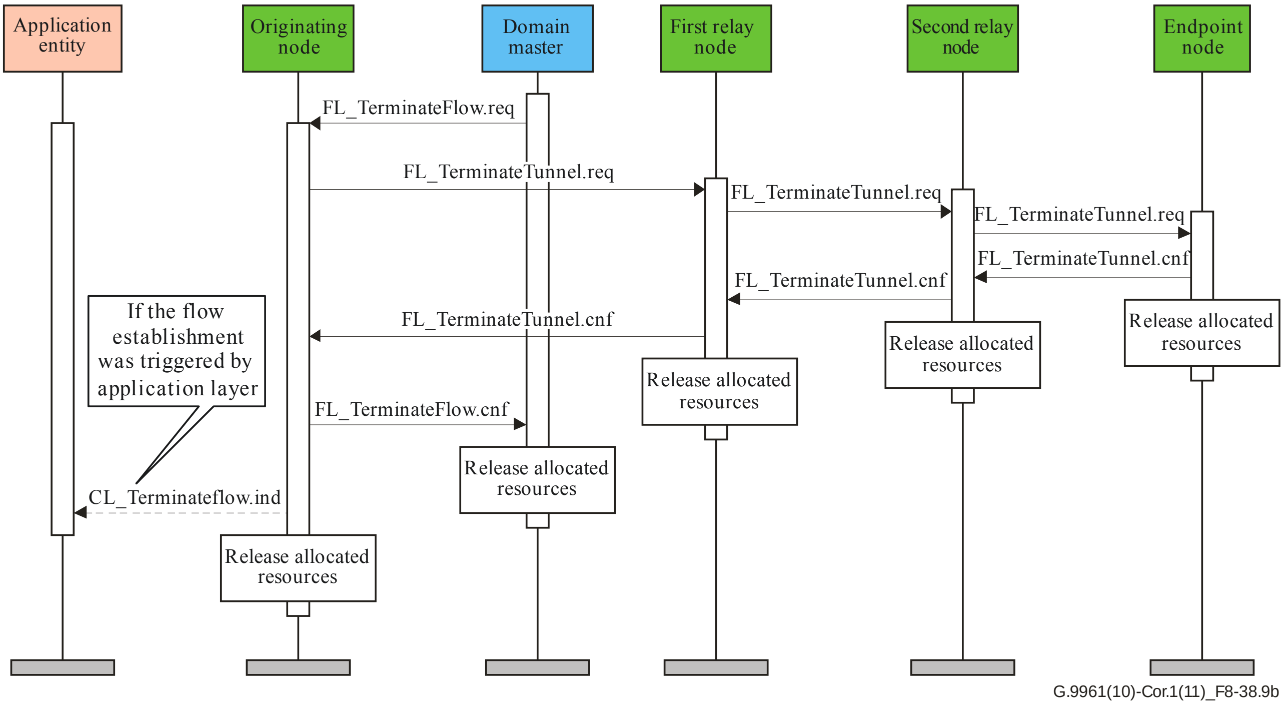


Figure 8-38.9b – MSC for tunnel termination triggered by the domain master

8.6.2.2.5.4 Tunnel termination due to disconnection of the originating node

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

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

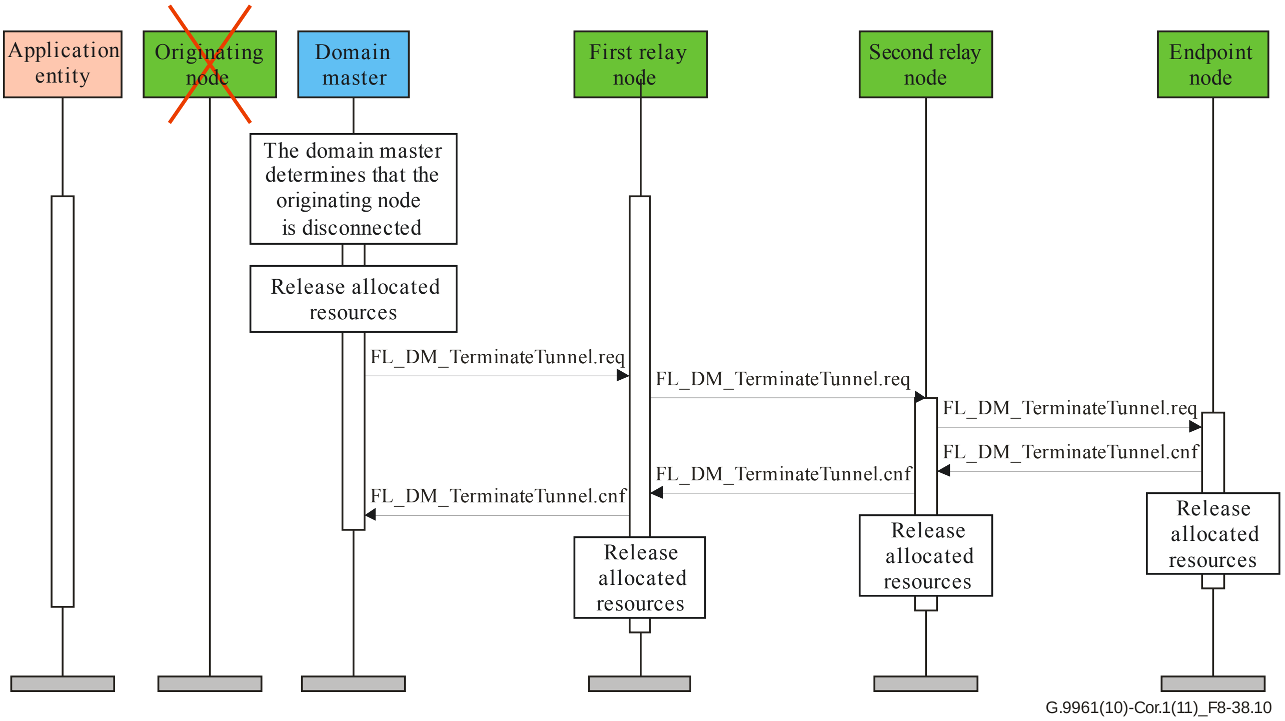

Application entity
Originating node
Domain master
First relay node
Second relay node
Endpoint node
The domain master determines that the originating node is disconnected
Release allocated resources
FL_DM_TerminateTunnel.req
FL_DM_TerminateTunnel.req
FL_DM_TerminateTunnel.req
FL_DM_TerminateTunnel.cnf
FL_DM_TerminateTunnel.cnf
FL_DM_TerminateTunnel.cnf
Release allocated resources
Release allocated resources
Release allocated resources
G.9961(10)-Cor.1(11)_F8-38.10

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

…

11.3) Clause 8.6.2.3

Revise clause 8.6.2.3 as follows and add clauses 8.6.2.3.18 to 8.6.2.3.28:

8.6.2.3.1 Format of CL_EstablishFlow.req

…

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

Field	Octet	Bits	Description
DA_F	0 to 5	[47:0]	Flow Destination MAC address. APDUs whose destination MAC address is specified in this field should be transmitted via this flow.
Classifiers	6 to (7+j)	See Table 8-22	This field shall contain traffic classifiers. APDUs whose destination MAC address is the specified MAC address and header conforms to the specified classifiers should be transmitted via this flow.
TSpec	Variable	See Table 8-24	Traffic specification for this flow may include the following fields: Traffic priority, maximum information rate, maximum traffic burst, committed information rate, tolerated jitter, maximum latency, unsolicited grant interval, unsolicited polling interval and APDU size. N – The length of this field is variable according to the actual number of included traffic specification fields. The TSpec format is as specified in Table 8-24.

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

Field	Octet	Bits	Description
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 8-24 – Format for the TSpec field

Field	Octet	Bits	Description
Length	0	[7:0]	The length of the TSpec sub-fields following this field expressed in number of octets in the range between 2 and 255.
TSpecBitMask	1 and 2	[15:0]	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: **Bit** **TSpec attribute** 0 Traffic priority 1 Maximum information rate 2 Maximum traffic burst 3 Committed information rate 4 Tolerated jitter 5 Maximum latency 6 Grant interval 7 Polling interval 8 APDU size 9 to 15 Reserved by ITU-T If bit 3 is set (CIR field is present), then bit 0 shall also be set (TrafficPriority is present).

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 be present if the CIR field is present. This field shall only be present if TSpecBitMask bit 0 is set to one.
MIR	Variable	[31:0]	Specifies the maximum information rate in bit/s, represented as a 32-bit unsigned integer. This field shall only be present if TSpecBitMask bit 1 is set to one.
MaxTBurst	Variable	[15:0]	Specifies the maximum traffic burst (see clause 8.6.2.1) in kbytes, represented as a 16-bit unsigned integer. This field shall only be present if TSpecBitMask bit 2 is set to one.
CIR	Variable	[31:0]	Specifies the committed information rate (see clause 8.6.2.1) in bit/s, represented as a 32-bit unsigned integer. This field shall only be present if TSpecBitMask bit 3 is set to one.
ToleratedJitter	Variable	[7:0]	Specifies the tolerated jitter in ms, represented as an 8-bit unsigned integer. This field shall only be present if TSpecBitMask bit 4 is set to one.
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

…

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} = Failure – classifier rule passed in CL_EstablishFlow.req is not supported. • 08~~7~~$_{16}$ – 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).
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.			

…

8.6.2.3.5 Format of CL_FlowTerminated.ind

…

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 (o~~Originating node has inferred that the traffic flow has ended~~)~~. • 01_{16} = Terminated at request of the domain master. • 02_{16}-FF_{16} = Reserved.

…

8.6.2.3.8 Format of FL_AdmitFlow.req

…

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 ~~r~~Rate	Variable	See Table 8-33	The actual PHY data rate used by the transmitter, specified in bits per ~~symbol~~ second 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~~ TX ~~r~~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 ~~r~~Rate_B	Variable	See Table 8-33	TX~~ r~~ Rate for the reverse direction (Note).
Tunnel	Variable	[7:0]	00_{16} – direct flow admission is requested 01_{16} – tunnel flow admission is requested
EndPoint	Variable	[7:0]	DEVICE_ID of the endpoint node.
NOTE – These fields appear only if Bidirectional field is set to 01_{16}.			

Table 8-33 – Format of the TX rate field

Field	Octet	Bits	Description
NumCEWindows	0	[4:0]	Number of items in the following list. Each item contains information for one channel estimation window. Each item includes three fields: CE_STime, CE_ETime and BitsPerSecond~~ymbol~~. The list shall not exceed n=32 items.
EstimOverhead		[7:5]	Estimated DLL overhead in percentage represented as an unsigned integer minus 1 (Note 1). A value of zero represents 1% overhead. A value of 7 represents ≥ 8 % overhead.
CE_STime	1	[7:0]	Start time as specified in Table 8-98 for first channel estimation window.
CE_ETime	2	[7:0]	End time as specified in Table 8-99 for first channel estimation window.
BitsPerSecond~~ymbol~~	3 and 4	[15:0]	~~Number of~~The PHY data rate in bits per ~~symbol~~second for the first channel estimation window in ~~the~~ steps of 32 kbit/s~~ps~~ (Note 2).
⋮	⋮	⋮	⋮
CE_STime	4n–3	[7:0]	Start time as specified in Table 8-98 for last channel estimation window.
CE_ETime	4n–2	[7:0]	End time as specified in Table 8-99 for last channel estimation window.
BitsPerSecond~~ymbol~~	4n–1 to 4n	[15:0]	~~Number of~~The PHY data rate in bits per ~~symbol~~second for the last channel estimation window in ~~the~~ steps of 32 kbit/s~~ps~~ (Note 2).
NOTE 1 – Defined as (Number of bytes crossing the PMI – number of bytes crossing the A-interface)/Number of bytes crossing the A-interface * 100% associated with a flow, including retransmission. The estimation of this parameter shall be vendor discretionary. NOTE 2 – BitsPerSecond = ($k_P \cdot R \cdot F_{SC}$) / ($1 + N_{GI}/N$) where k_P is the number of loaded bits (see clause 7.1.3.3.1 of [ITU-T G.9960]), R is the code rate (see clause 7.1.3.2 of [ITU-T G.9960]), F_{SC} is the sub-carrier spacing, N_{GI} is the guard interval, and N is the number of sub-carriers (see clause 7.1.4.6 of [ITU-T G.9960]) for a payload OFDM symbol transmitted over a specified channel estimation window.			

8.6.2.3.9 Format of FL_AdmitFlow.cnf

…

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.

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

Field	Octet	Bits	Description
			• 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$_{16}$ = Failure – the specified Endpoint is not accessible or not registered. • 08$_{16}$ = Rejected – the domain master is now in the middle of a handover process. 09~~7~~$_{16}$ – 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

…

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.

Table 8-35 – Format of the MMPL of the FL_AdmitFlow.ind message

Field	Octet	Bits	Description
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 endpoint node or by~~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 – This~~ese~~ field~~s~~ appears only if the 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.

…

8.6.2.3.13 Format of FL_TerminateFlow.req

The message is sent to request a flow termination.~~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
TerminatorID	0	[7:0]	DEVICE_ID of the originating node or of the domain master that triggers the flow termination.
DeviceID	1~~0~~	[7:0]	DEVICE_ID of the originating node~~ or of the endpoint node in case of bidirectional flow~~.
FlowID	2~~1~~	[7:0]	FLOW_ID of the flow to be terminated~~assigned by the originating node or of the endpoint node in case of bidirectional flow~~.
ReasonCode	3~~2~~	[7:0]	Reason why the flow is being terminated: • 00$_{16}$ = Normal termination initiated by the originating node~~ or by endpoint node in case of bidirectional flow~~. • 01$_{16}$ = Termination by domain master due to ~~over-subscription of the channel~~lack of resources. • 02$_{16}$ = Termination by domain master ~~due to~~because the tunnel is broken and reconstruction is not possible. • 03$_{16}$ = Termination by the domain master because the endpoint is disconnected. • 04~~3~~$_{16}$ – FF$_{16}$ = Reserved.

8.6.2.3.14 Format of FL_TerminateFlow.cnf

This message shall be sent as a reply to~~is sent by either the Domain Master or the endpoint node in response to~~ a received FL_TerminateFlow.req message. 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 node originating ~~node or of the endpoint node in case of bidirectional flow~~the flow.
FlowID	1	[7:0]	FLOW_ID of the terminated flow~~assigned by the originating node or of the endpoint node in case of bidirectional flow~~.
Status	2	[7:0]	Status of termination. • 00$_{16}$ = Termination ~~is~~ completed successfully. • 01$_{16}$ = Flow termination failed due to a broken link with the endpoint. • 02$_{16}$ = Flow unknown. 03~~5~~$_{16}$ – FF$_{16}$ = Reserved.

…

8.6.2.3.18 Format of FL_AdmitFlow.rsp

This message is sent by the the domain master to the originating node to acknowledge the reception of FL_AdmitFlow.ind. The message shall contain the established flow from the originating node towards the endpoint node (or towards the first relay in case of a tunnel), and the reverse flow from the endpoint node to the originating node (or towards the first relay in the reverse path) in case of bidirectional flow.

The format of the MMPL of the FL_AdmitFlow.rsp message shall be as shown in Table 8-44.1.

Table 8-44.1 – Format of the MMPL of the FL_AdmitFlow.rsp 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.
Bidirectional	2	[7:0]	Set to 1 in case the admitted flow is a bidirectional flow.
FlowID_B	3	[7:0]	FLOW_ID assigned by the endpoint node for the reverse flow (Note).
NOTE – These fields appear only if the bidirectional field is set to one.			

8.6.2.3.19 Format of FL_BrokenTunnel.ind

A node that is in the tunnel path, that determines that the link to the next node in the tunnel is broken shall send this message to the domain master.

The format of the MMPL of the FL_BrokenTunnel.ind message shall be as shown in Table 8-44.2.

Table 8-44.2 – Format of the MMPL of the FL_BrokenTunnel.ind message

Field	Octet	Bits	Description
ReportID	0	[7:0]	DEVICE_ID of the reporting node that indicates the broken link in the tunnel.
OriginateID	1	[7:0]	DEVICE_ID of the originating node.
FlowID	2	[7:0]	FLOW_ID that identifies the tunnel.
BrokenLink	3	[7:0]	DEVICE_ID of the disconnected node.
BrokenFlow	4	[7:0]	FLOW_ID of the flow of the broken link.

8.6.2.3.20 Format of FL_BrokenTunnel.rsp

This response message is sent by the domain master to confirm receiving the FL_BrokenTunnel.ind message. The format of the MMPL of the FL_BrokenTunnel.rsp message shall be as shown in Table 8-44.3.

Table 8-44.3 – Format of the MMPL of the FL_BrokenTunnel.rsp message

Field	Octet	Bits	Description
ReportID	0	[7:0]	DEVICE_ID of the reporting node that indicates the broken link in the tunnel.
OriginateID	1	[7:0]	DEVICE_ID of the originating node.
FlowID	2	[7:0]	FLOW_ID that identifies the tunnel.
BrokenLink	3	[7:0]	DEVICE_ID of the node that the link to it is broken.
BrokenFlow	4	[7:0]	FLOW_ID of the flow of the broken link.

8.6.2.3.21 Format of FL_ReleaseTunnel.req

This message is sent by an originating node of a tunnel to the domain master to request the domain master to release the tunnel resources after the tunnel termination is completed.

The format of the MMPL of the FL_ReleaseTunnel.req message shall be as shown in Table 8-44.4.

Table 8-44.4 – Format of the MMPL of the FL_ReleaseTunnel.req message

Field	Octet	Bits	Description
OrigDeviceID	0	[7:0]	DEVICE_ID of the originating node.
FLOW_ID	1	[7:0]	FLOW_ID that identifies the tunnel.

8.6.2.3.22 Format of FL_ReleaseTunnel.cnf

The domain master shall send this message to the originating node, after it receives the FL_ReleaseTunnel.req message from the originating node.

The format of the MMPL of the FL_ReleaseTunnel.cnf message shall be as shown in Table 8-44.5.

Table 8-44.5 – Format of the MMPL of the FL_ReleaseTunnel.cnf message

Field	Octet	Bits	Description
OrigDeviceID	0	[7:0]	DEVICE_ID of the originating node.
FLOW_ID	1	[7:0]	FLOW_ID that identifies the tunnel.
ReleaseCode	2	[7:0]	Condition after releasing of tunnel resources: • 00_{16} – Resources are successfully released. • 01_{16} – Tunnel is unknown. • 02_{16} – FF_{16} Reserved.

8.6.2.3.23 Format of FL_DM_RenewTunnel.req

This message is sent by the domain master to the originating node to reconstruct a broken tunnel. This message contains a list of relay nodes from the originating node until the endpoint node that comprise the updated tunnel path.

The format of the MMPL of the FL_DM_RenewTunnel message shall be as shown in Table 8-44.6.

Table 8-44.6 – Format of the MMPL of the FL_DM_RenewTunnel.req message

Field	Octet	Bits	Description
DeviceID	0	[7:0]	DEVICE_ID of the originating node.
FlowID	1	[7:0]	FLOW_ID assigned by the originating node that identifies the tunnel.
EndPoint	2	[7:0]	DEVICE_ID of the endpoint node.
NumRelays	3	[7:0]	Number of relay nodes (n) in the routeList.
RelayNode[0]	4	[7:0]	DEVICE_ID of the first relay node in the list.
….	…	…	…
RelayNode[n-1]	3+n	[7:0]	DEVICE_ID of the last relay node in the list.

8.6.2.3.24 Format of FL_DM_RenewTunnel.cnf

This message shall be sent by the originating node to the domain master after the originating node receives the FL_RenewTunnel.cnf message from the first relay node in the reconstructed tunnel.

The format of the MMPL of the FL_ DM_RenewFlow.cnf message shall be as shown in Table 8-44.7.

Table 8-44.7 – Format of the MMPL of the FL_DM_RenewTunnel.cnf message

Field	Octet	Bits	Description
DeviceID	0	[7:0]	DEVICE_ID of the originating node.
FlowID	1	[7:0]	FLOW_ID ID assigned by the originating node that identifies the tunnel.
StatusCode	2	[7:0]	Status of the reconstruction procedure: • 00_{16} – Success. • 01_{16} – Failure due to node capabilities limitation. • 02_{16} – Failure due to node resources limitation. • 03_{16} – Failure due to broken link. • 04_{16}–ff_{16} – Reserved.
EndPoint	3	[7:0]	DEVICE_ID of the endpoint node.
Bidirectional	5	[7:0]	Specifies the flow type: • 00_{16} – Unidirectional flow. • 01_{16} – Bidirectional flow. • 02_{16}–ff_{16} – Reserved.
NumRelays	6	[7:0]	Number of relays in the list (n) in case of success. if StatusCode is not success this field shall contain zero.
DeviceID[0]	7	[7:0]	DeviceID of the last relay node in the tunnel (Note 1).
FlowID[0]	8	[7:0]	FLOW_ID assigned by the last relay node toward the endpoint (Note 1).
BFlowID[0]	9	[7:0]	FLOW_ID of the inverse bidirectional flow assigned by the endpoint toward the last relay node when the flow is a bidirectional flow (Notes 1 and 2).
….			
DeviceID[n–1]	7+ 3×(n–1)	[7:0]	DeviceID of the first relay in the tunnel (Note 1).

Table 8-44.7 – Format of the MMPL of the FL_DM_RenewTunnel.cnf message

Field	Octet	Bits	Description
FlowID[n–1]	8+ 3×(n–1)	[7:0]	FLOW_ID of the assigned by first relay node in the tunnel (Note 1).
BFlowID[n–1]	9+ 3×(n–1)	[7:0]	FLOW_ID of the inverse bidirectional flow assigned by the endpoint toward the last relay node when the flow is a bidirectional flow (Notes 1 and 2).
DeviceID	7	[7:0]	DeviceID of the node that could not establish a flow (Note 3).
NOTE 1 – This field should exist in the message only if StatusCode is success. NOTE 2 – This field should exist in the message only if the Bidirectional field contains $01_{16.}$ NOTE 3 – This field shall exist in the message only if StatusCode is not success. The failure status corresponding to this node is communicated in the field StatusCode.			

8.6.2.3.25 Format of the FL_RenewTunnel.req message

This message is sent by the originating node or by relay nodes in a reconstructed tunnel to reconstruct a broken tunnel. The format of the MMPL of the FL_RenewTunnel.req message shall be as shown in Table 8-44.8.

Table 8-44.8 – Format of the MMPL of the FL_RenewTunnel.req message

Field	Octet	Bits	Description
DeviceID	0	[7:0]	DEVICE_ID of the originating node.
FlowID	1	[7:0]	FLOW_ID assigned by the originating node that identifies the tunnel.
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.
EndPoint	Variable	[7:0]	DEVICE_ID of the endpoint hidden node.
RouteList	Variable	See Table 8-30	Routing list toward the destination endpoint.
NumRelays	Variable	[7:0]	Number of relays in the reconstructed tunnel (n).
RelDeviceID[0]	Variable	[7:0]	DeviceID of the first relay node.
RelFlowID[0]	Variable	[7:0]	FLOW_ID assigned by the first relay node.
….	Variable		
RelDeviceID[n-1]	Variable	[7:0]	DeviceID of the nth relay node in the reconstructed tunnel.
RelFlowID[n-1]	Variable	[7:0]	FLOW_ID assigned by nth relay node in the reconstructed tunnel.

8.6.2.3.26 Format of the FL_RenewTunnel.cnf message

This message confirms the tunnel reconstruction request. The format of the MMPL of the FL_RenewTunnel.cnf message shall be as shown in Table 8-44.9.

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

Field	Octet	Bits	Description
DeviceID	0	[7:0]	DEVICE_ID of the originating node.
FlowID	1	[7:0]	FLOW_ID assigned by the originating node that identifies the reconstructed tunnel.
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.
NumRelays	3	[7:0]	Number of relays in the reconstructed tunnel (n). If StatusCode is not success this field shall contain zero.
RelDeviceID[0]	4	[7:0]	DeviceID of the first relay node (Note 1).
RelFlowID[0]	5	[7:0]	FLOW_ID assigned by the first relay node (Note 1).
….			
RelDeviceID[n–1]	4+ 2×(n–1)	[7:0]	DeviceID of the nth relay node in the reconstructed tunnel (Note 1).
RelFlowID[n–1]	5+ 2×(n–1)	[7:0]	FLOW_ID assigned by nth relay node in the reconstructed tunnel (Note 1).
Bidirectional	6+ 2×(n–1)	[7:0]	When set to 01_{16} it indicates that the flow is bidirectional flow. When set to 00_{16} it indicates that the flow is unidirectional flow.
RelBFlowID[0]	1+ 6+ 2×(n–1)	[7:0]	FLOW_ID of the inverse bidirectional flow assigned by the endpoint node (Notes 1 and 2).
RelBFlowID[1]	2+ 6+ 2×(n–1)	[7:0]	FLOW_ID of the inverse bidirectional flow assigned by the last relay node (Notes 1 and 2).
...			
RelBFlowID[n]	n+ 6+ 2×(n–1)	[7:0]	FLOW_ID of the inverse bidirectional flow assigned by the first relay node when the flow is a bidirectional flow (Notes 1 and 2).

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

Field	Octet	Bits	Description
DeviceID	7	[7:0]	DeviceID of the node that could not establish a flow (Note 3).
NOTE 1 – This field should exist in the message only if StatusCode is success. NOTE 2 – This field should exist in the message only if the Bidirectional field contains $01_{16.}$ NOTE 3 – This field shall exist in the message only if StatusCode is not success. The failure status corresponding to this node is communicated in the field StatusCode.			

8.6.2.3.27 Format of the FL_DeleteFlow.req message

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

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

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

8.6.2.3.28 Format of the FL_DeleteFlow.cnf message

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

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

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

…

11.4) Clauses 8.6.4 to 8.6.6

Revise the text as follows for clauses 8.6.4 to 8.6.6.5.3:

8.6.4 Routing and topology management

…

Nodes shall transmit topology messages in accordance with the topology update interval (see ~~Table 8-82~~clause 8.6.4.2) and upon particular events and management procedures in the domain that change or may potentially change the topology.

…

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

8.6.4.1 Domain master operation for routing and topology management

8.6.4.1.1 Domain master operation in CRTM mode

…

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

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

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

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

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

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

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

The domain master may receive TM_NodeTopologyChange.ind messages via the mechanism described in clause 8.6.4.2.1.1. After the domain master receives the topology change messages and updates its routing tables, it shall generate and distribute a new TM_DomainRoutingChange.ind message. This message shall include the updated routing table, and shall also indicate the nodes that generated the TM_NodeTopologyChange.ind messages that requested an acknowledgement (i.e., with AckType set to 00_2 or 10_2) via the TM_DomainRoutingChange.ind (see Table 8-46), 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. The message sequence number of the

TM_ReturnDomainRouting.cnf message header and the message sequence number specified in the last transmitted TM_DomainRoutingChange.ind message header are unrelated.

…

8.6.4.1.1.2 Distribution of routing tables

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

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

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

…

8.6.4.2 Endpoint node topology maintenance

Each node in the domain shall participate in the topology maintenance by sending a message TM_NodeTopologyChange.ind with the AckType field set to 00_2 (see Table 8-46) 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 contents of the LAAT~~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.

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

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

8.6.4.2.1 Endpoint node topology maintenance in CRTM mode

A node shall reply with 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 within T_{N_RSP} ms (see clause 8.4) after reception of the message TM_NodeTopologyChange.req from the domain master.

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

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

~~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 Table 8-82 and Table 8-83).~~

At its own discretion, a node may request an acknowledgement via TM_DomainRoutingChange.ind, that the domain master has received the periodically sent TM_NodeTopologyChange.ind, by setting the AckType field to 10_2 (see Table 8-46). If a node requests an acknowledgement, then the domain master shall include the DEVICE_ID of this node and the sequence number of the topology change message (see Table 8-50) in the next TM_DomainRoutingChange.ind message.

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

…

8.6.4.2.1.1 Communication of endpoint topology change in CRTM mode

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

…

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

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

…

8.6.4.3.1 Format of TM_NodeTopologyChange.ind

…

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~~3:0]	Shall be set to: – ~~0~~0_{16} if report includes all the topology parameters available to the node (full report) – ~~0~~1_{16} if the report includes any fraction of topology parameters available to the node – ~~0~~2_{16} if the report includes only parameters that changed relatively to the last report Other values are for further study.
AckType		[5:4]	This field shall be set to: • 00_2: Event driven update, ACK requested • 01_2: Periodic update, no ACK requested • 10_2: Periodic update, ACK is requested • 11_2: Reserved by ITU-T ACK in this case means acknowledgement via TM_DomainRoutingChange.ind is requested (see Table 8-50).
Reserved		[7:6]	Reserved by ITU-T (Note 1)
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	[39~~1~~:0]	List of M fields, 54 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 8-48 – 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~~BitsPerSecond	1 to 4~~3~~	[~~2~~31:0]	Bits [15~~1~~:0] indicate the PHY data rate in bits per second from the reporting node to the detected node; Bits [~~2~~31:16~~2~~] 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 – Both data rates shall follow the same formula as Note 2 in Table 8-33. If the data rate with the particular detected node is not available, the value of this parameter shall be set to FFFF$_{16}$~~; The value shall be set to zero if the detected data rate is less than 0.5 Mbit/s~~.			

…

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~~ a ~~message~~ TM_NodeTopologyChange.req message from the domain master or backup domain master.~~,~~

If the requested topology information has not changed since sending the last TM_NodeTopologyChange.ind message, ~~T~~the MMPL of the TM_NodeTopologyChange.cnf message shall contain the sequence number of the last transmitted TM_NodeTopologyChange.ind message and the rest of the MMPL shall contain the requested components from ~~be~~ the ~~same as~~ MMPL of the last transmitted TM_NodeTopologyChange.ind message (Table 8-46).

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

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

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

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

8.6.4.3.4 Format of TM_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 changes in a node's topology information. The MMPL of the TM_DMBackup.ind message shall contain the sequence number of the received TM_NodeTopologyChange.ind message whose MMPL information is contained in the Info field. The format of the TM_DMBackup.ind message shall be as shown in Table 8-49.2~~be the same as MMPL of the TM_NodeTopologyChange.ind message (Table 8-46)~~.

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

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

8.6.4.3.5 Format of TM_DomainRoutingChange.ind

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

…

~~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~~ domain master after the previous transmission of domain routing change message that requested an acknowledgement via this message (see Table 8-46).
DEVICE_ID[0]	1	[7:0]	DEVICE_ID of the first node that requested an acknowledgement via this~~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 of the first node that requested an acknowledgement via this~~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 that requested an acknowledgement via this~~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 of the m-th node that requested an acknowledgement via this message~~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.

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

Field	Octet	Bits	Description
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.
BRT_Size	Variable	[15:0]	Length in bytes of all the BRT entries of the node plus one. This length includes the NumBRTEntries and the BRTEntry[i] fields.
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.

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

Field	Octet	Bits	Description
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.
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.			

…

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 8-53.1~~0~~, 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.

Table 8-53.1 – Format of the MMPL of the TM_ ReturnDomainRouting.cnf message

Field	Octet	Bits	Description
ReqRep	0	[7:0]	This field contains the original received ReqRep field as specified in the TM_ReturnDomainRouting.req message.
OrigSequence	1 and 2	[15:0]	Sequence number of the last transmitted TM_DomainRoutingChange.ind message.
Info	Variable	Variable	This field contains the TM_DomainRoutingChange.ind message components as specified in Table 8-50 according to the specified ReqRep field.

…

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 (i.e., JOIN_INTERVAL_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)

…

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)~~Variable	~~[(8*N)-1:0]~~Variable	Data related to domain management provided by the acting domain master as ~~presented~~ shown 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.~~
Number of Records	2	[7:0]	Number of data records (R). Each data record is of variable length. The length of each record is specified in the first field of the record.
Backup Data Record[0]	Variable	Table 8-60.1	First record in the list. The record structure format is defined in Table 8-60.1.
…			…
Backup Data Record[R-1]	Variable	Table 8-60.1	Last record in the list. The record structure format is defined in Table 8-60.1.

Table 8-60.1 – Format of the Backup Data Record Structure

Field	Octet	Bits	Description
Record Length	0	[7:0]	Record length in bytes.
Record type	1	[7:0]	This field contains the type of record. The valid values are: 0: Service flow. 1-255: reserved.
Record data	Variable	Table 8-60.2	Contains the relevant information according to Record type. For Record type of service flow the format is defined in Table 8-60.2

Table 8-60.2 – Format of the Service flow record

Field	Octet	Bits	Description
Service flow data	Variable	Table 8-32	This field contains the service flow information as defined in Table 8-32

…

12) Clause 8.8

Revise the text in clause 8.8 and its sub-clauses as follows:

8.8 Medium access plan (MAP) frame

…

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 01-19-A7-52-76-96$_{16}$, the DestinationNode of the LLC frame shall be set to zero.

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-A frame each MAC cycle and may transmit additional MAP frames (MAP-A or MAP-D) 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~~ shall distribute the MAP for all nodes registered to a domain by transmitting ~~default MAP frames (MAP-D) or both default and active~~ MAP-A frames ~~(MAP-A)~~. The domain master shall distribute the MAP and other information necessary for registration by transmitting default MAP-D frames. 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 frame in addition to~~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~~ shall include only the relevant information needed by the registering nodes to synchronize with the MAC cycle, to learn the regional transmission parameters, and to learn the header segmentation (see clauses 8.6.7 and 8.8.4.1.1) of MAP-A and RMAP-A. The content of the MAP-D shall include the followings:

- TXOP descriptor(s) and TXOP attributes extensions describing all MAP-A, RMAP-A and MAP-D transmissions,
- TXOP descriptor(s) and TXOP attributes extensions describing transmit opportunity (e.g., RCBTS) for registering node, and
- Auxiliary information necessary for registration – Domain name, PSD-related domain info, Registration code, and Timer-related domain info (see Table 8-73).

~~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~~. The RMAP-D frame shall be constructed following the same requirements as MAP-D.

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-A frame 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-A frame. 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 1 – 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 the RMAP-D transmissions so that during each (JOIN_INTERVAL_T_0)/2 interval all nodes that are MAP relay capable transmit RMAP-D~~,~~ at least once 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~~.

NOTE 2 – This ensures that a joining node that can detect RMAP-D from only one node in a domain that it intends to join, can still detect, at least two consecutive RMAP-D transmissions within the JOIN_INTERVAL_T_0, which is sufficient to synchronize its transmit clock with the node transmitting the RMAP-D and decoding the MAP.

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. During this time, the domain master shall avoid scheduling RMAP-D transmission since it affects the content of the MAP-D (e.g., dynamic SM changes).

…

8.8.4 TXOP descriptor

…

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		[5:4]	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.
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.			

…

8.8.4.1.3 Group information extension data

…

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. The valid range is 1 to 127. A value of zero indicates default sequential behaviour, described in clause 8.3.3.2.2.
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. The valid range is 1 to 127. 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.			

…

8.8.5 Auxiliary information field

…

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.

NOTE – Some changes in auxiliary information (e.g., SM change) may lead to changes in transmission parameters (e.g., BAT). In this case, a node should adjust its transmission parameters as soon as possible using existing protocols (e.g., CE_PartialBatUpdate.req) to minimize the impact.

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.

…

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~~04$_{16}$
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 descriptor sub-field is not ~~used~~ present. 1, when PSD shaping descriptor sub-field is ~~used~~ present.
Regional SM		[1]	0, when ~~sub-carrier masking~~SM descriptor sub-field is not ~~used~~ present. 1, when ~~sub-carrier masking~~SM descriptor sub-field is ~~used~~ present.
Regional TX power limit		[2]	0, when ~~standard transmit~~TX power limit sub-field is ~~used~~not present (see clause 7.2.6). 1, when TX power limit sub-field is ~~used~~ present.
Regional Amateur radio bands		[3]	0, when ~~all international~~ Amateur radio band~~s are masked~~ descriptor sub-field is not present. 1, when ~~one or more bands are not masked~~ Amateur radio band descriptor sub-field is present.
Symbol boost indicator		[4]	0, when Symbol boost parameters sub-field is not present. 1, when Symbol boost parameters sub-field is present.
Reserved		[7:54]	Reserved by ITU-T (Note 2).
Amateur radio band descriptor	~~4 and 5~~Variable	[9:0]	~~Zero octets~~This field shall not be present if the regional Amateur radio bands field~~bit 3 of octet 3~~ is set to zero, otherwise it 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).

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

Field	Octet	Bits	Description
TX power limit	~~6~~Variable	[7:0]	~~Zero octets~~This field shall not be present if the regional TX power limit field~~bit 2 of octet 3~~ is set to zero, otherwise it represents the value of maximum transmit power in dBm, represented as 0.1dBm per unit
PSD shaping descriptor	~~7 to (6+L)~~Variable	[(8*L) – 1:0]	~~Zero octets~~This field shall not be present if the regional PSD shaping mask field~~bit 0 of octet 3~~ is set to zero, otherwise see Table 8-78 (Note 3).
SM descriptor	~~(7+L) to (6+L+M)~~ Variable	[(8*M) – 1:0]	~~Zero octets~~This field shall not be present if the regional SM field~~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).
Symbol boost parameters	Variable	[7:0]	This field shall not be present if the symbol boost indicator field is set to zero, otherwise see Table 8-79.1.

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).

Table 8-78 – PSD shaping descriptor

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 $400_{16} = 1024$.

NOTE 5 – In order to remove the regional PSD shaping mask, octets 0 to 3 shall be set to 0.

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 $400_{16} = 1024$.

NOTE 3 – In order to remove the regional SM, octets 0 to 3 shall be set to 0.

Table 8-79.1 – Symbol boost parameters

Octet	Bits	Description
0	[1:0]	SYM_BOOST_TYPE (Note 1) 00_2 – disabled 01_2 – preamble and first OFDM symbol of the PFH 10_2 – preamble only 11_2 – Reserved by ITU-T (Note 3)
	[5:2]	SYM_BOOST_AMOUNT (Note 1) When SYM_BOOST_TYPE is set to 00_2, 0 to 15 – Reserved by ITU-T When SYM_BOOST_TYPE is set to 01_2, 0 to 9 – 0 to 2.0 dB in 0.2 dB steps (Note 2) 10 to 15 – Reserved by ITU-T When SYM_BOOST_TYPE is set to 10_2, this field represents the symbol boost amount with the following possible values: 0 to 15 – 0 to 3.0 dB in 0.2 dB step (Note 2) When SYM_BOOST_TYPE is set to 11_2, 0 to 15 – Reserved by ITU-T
	[7:6]	Reserved by ITU-T (Note 3)
NOTE 1 – Additional constraints on the values of SYM_BOOST_TYPE and SYM_BOOST_AMOUNT are specified in clause 8.4. NOTE 2 – The maximum symbol boost amount shall be limited to comply with regional regulations. NOTE 3 – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver. NOTE 4 – In order to disable symbol boost, octet 0 shall be set to 0.		

…

8.8.5.7 Backup domain master ID sub-field

The format of the backup 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.

…

8.8.5.8.1 Topology update interval sub-field

This sub-field ~~indicates~~ contains the topology update report interval and requested report scope 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~~ periodic interval	0	[7:0]	Specifies the periodic interval the nodes shall send the TM_NodeTopologyChange.ind message. The field is represented as an unsigned integer in units of 0.1 s. If this field is set to zero, the node shall not send periodically the topology report.~~Minimum interval, $T_{MIN\_DISCOVER}$, in s represented as an unsigned integer~~

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

Field	Octet	Bits	Description
RequestReport	1	[7:0]	Bit 0 – if it is set to one the nodes shall include the visibility information in the TM_NodeTopologyChange.ind message that shall be periodically transmitted. Bit 1 – if it is set to one the complete AAT information shall be included in the TM_NodeTopologyChange.ind -message that shall periodically be transmitted. Other bits are reserved by ITU-T and shall be set to zero.~~Maximum interval, $T_{MAX\_DISCOVER}$, in s represented as an unsigned integer~~

…

13) Clause 8.9

Revise the text in clause 8.9 and its sub-clauses as follows:

…

8.9.1.1 Immediate acknowledgment

…

A gap of T_{AIFG-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.

…

8.9.1.2 Delayed acknowledgment

If delayed-ACK is required, the receiver may transmit~~transmission of~~ the acknowledgement ~~shall be deferred to~~in 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, the corresponding ACK PHY frame shall be considered as having an MPDU priority equal to 7~~:~~.

- ~~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 either:

- one or more errors were detected in those LPDUs of the received frame that were not received correctly in the previous transmissions, or~~;~~
- the SSN could not be determined by the node for at least one of the LPDUs with errors in the received frame~~if errors are detected only in the LPDUs that were already received correctly in previous transmissions,~~.

Otherwise, the frame ~~should~~ shall 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.

…

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~~ 1024 for data connections~~ with LPDUs of size 540 bytes and 564 for data connections with LPDUs of size 120 bytes (see Table 7-21)~~, ~~16 for management connections with LPDUs of size 540 bytes,~~ and 32 for management connections~~ with LPDUs or size 120 bytes~~.

…

14) Clause 8.10

Revise the text in clause 8.10 and its sub-clauses as follows:

8.10.1 Management message format

…

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. An LCDU that contains a MAP message shall be carried only in the MAP or RMAP frame.~~ 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).
	STD Version~~Reserved~~	3	[7:0]	The format of this message is according to the specified ITU-T G.9961 version.~~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 (Note 2).
	Segment number		[7:4]	Segment number, represented as an unsigned integer between 0_{16} and F_{16}; set to 0_{16} for the first segment and if message is not segmented (Note 2).
	Sequence number	5 and 6	[15:0]	Sequence number of the segmented message in a format of 16-bit unsigned integer (Notes 2 and 3).
	Repetition number	7	[3:0]	Repetition number of the segmented message formatted as a 4-bit unsigned integer whose initial value is 0. Each time a segment is retransmitted by the originating node this field shall be incremented.
	FSB		[4]	Force Sequence Bit. See clause 8.10.1.2
	Reserved		[7:5~~0~~]	Reserved by ITU-T (Note 4~~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 – This field is not applicable for a MAP message, and shall be set to zero.

NOTE 3 – The meaning of the sequence number depends on the OPCODE. See clause 8.10.1.2.

NOTE 4~~2~~ – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.

8.10.1.1 Management message OPCODEs

Management message OPCODEs are formatted as 12-bit unsigned integers. Valid values of OPCODEs are presented in Table 8-88. OPCODEs are categorized (typically by their associated protocol or procedure) according to the value of their eight MSBs.

Table 8-88 – OPCODEs of management messages

Category	Message name	OPCODE (hex)	Description	MMPL Reference
Admission (01X)	ADM_NodeRegistrRequest.req	010	Registration request	Clause 8.6.1.1.4.1
	ADM_DmRegistrResponse.cnf	011	Registration response	Clause 8.6.1.1.4.2
	ADM_NodeResignRequest.req	012	Resignation request	Clause 8.6.1.1.4.3
	ADM_DmResign.cnf	013	Registration announcement	Clause 8.6.1.1.4.4
	ADM_DmForcedResign.req	014	Forced resignation request	Clause 8.6.1.1.4.5
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
	AKM_DomainKeyUpdate.ind	02C	Indication to update the domain-wide encryption keys	Clause 9.2.5.3.2
	AKM_NewKey.ind	02D	Indication that the new encryption key is available for use	Clause 9.2.5.2.7

Table 8-88 – OPCODEs of management messages

Category	Message name	OPCODE (hex)	Description	MMPL Reference
Topology maintenance (03X)	TM_NodeTopologyChange.ind	030	Topology report from a node	Clause 8.6.4.2.1
	TM_NodeTopologyChange.req	031	Request sent by the domain master to a particular node requesting its topology report	Clause 8.6.4.3.2
	TM_NodeTopologyChange.cnf	032	Topology report from a node in response to the message TM_NodeTopologyChange.req	Clause 8.6.4.3.3
	TM_DomainRoutingChange.ind	033~~1~~	Optimal routing update from the domain master	Clause 8.6.4.3.5
	TM_ReturnDomainRouting.req	034~~2~~	Request for routing update from the node to the domain master	Clause 8.6.4.3.6
	TM_ReturnDomainRouting.cnf	035~~3~~	Reply on routing request by the Domain master	Clause 8.6.4.3.7
	TM_DMBackup.ind	036~~4~~	Topology report from a node sent by backup domain master to a node	Clause 8.6.4.3.4
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
	MC_GrpRemove.req	052	Multicast leave request from the transmitter	Clause 8.16.5.3
	MC_GrpRemove.cnf	053	Multcast leave confirmation from the receiver	Clause 8.16.5.4
	DMC_Path.req	054	DLL multicast path establishment request	Clause 8.17.6.1
	DMC_Path.cnf	055	DLL multicast path establishment confirmation	Clause 8.17.6.2
	DMC_PathReject.cnf	056	DLL multicast path establishment rejection	Clause 8.17.6.3
	DMC_EnforcePath.req	057	DLL multicast enforced path establishment request	Clause 8.17.6.4
	DMC_ReleasePath.req	058	A request to release a DLL multicast client node from its MSID	Clause 8.17.6.5

Table 8-88 – OPCODEs of management messages

Category	Message name	OPCODE (hex)	Description	MMPL Reference
	DMC_ReleasePath.cnf	059	Confirmation of the release of a DLL multicast client node from its MSID	Clause 8.17.6.6
	DMC_PathAlive.ind	05A	DLL multicast path alive indication	Clause 8.17.6.7
	DMC_BrokenLink.ind	05B	DLL multicast broken link indication	Clause 8.17.6.8
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_ProbeSlot~~Request.ind~~Assign.req	070	Channel estimation bandwidth assignment request	Clause 8.11.7.1
	CE_ProbeSlotRelease.~~ind~~req	071	Channel estimation bandwidth release request	Clause 8.11.7.2
	CE_ParamUpdate.~~ind~~req	072	Channel estimation parameters update request	Clause 8.11.7.3
	CE_ParamUpdateRequest.ind	073	Request for c~~C~~hannel estimation parameter ~~request~~update	Clause 8.11.7.4
	CE_PartialBatUpdate.~~ind~~req	074	Partial BAT update ~~indication~~request	Clause 8.11.7.5
	CE_ACESymbols.ind	075	Request for an ACE symbol attachment	Clause 8.11.7.6
	CE_ProbeSlotAssign.cnf	076	Channel estimation bandwidth assignment confirmation	Clause 8.11.7.7
	CE_ProbeSlotRelease.cnf	077	Channel estimation bandwidth release confirmation	Clause 8.11.7.8
	CE_ParamUpdate.cnf	078	Channel estimation parameters update confirmation	Clause 8.11.7.9
	CE_PartialBatUpdate.cnf	079	Partial BAT update confirmation	Clause 8.11.7.10

Table 8-88 – OPCODEs of management messages

Category	Message name	OPCODE (hex)	Description	MMPL Reference
Neighbouring networks coordination (08X)	For further study	For further study	For further study	For further study
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~~Reserved	0A0	~~Flow establishment request~~Reserved by ITU-T	~~Clause 8.6.2.3.1~~
	~~CL_EstablishFlow.cnf~~Reserved	0A1	~~Flow establishment confirmation~~Reserved by ITU-T	~~Clause 8.6.2.3.2~~
	FL_AdmitFlow.req	0A2	Flow admission request	Clause 8.6.2.3.8
	FL_AdmitFlow.cnf	0A3	Flow admission confirmation	Clause 8.6.2.3.9
	FL_AdmitFlow.ind	0A4	Flow admission indication	Clause 8.6.2.3.10
	FL_AdmitFlow.rsp	0A5	Flow admission acknowledgement	Clause 8.6.2.3.18
	FL_OriginateFlow.req	0A6~~4~~	Flow origination request	Clause 8.6.2.3.6
	FL_OriginateFlow.cnf	0A7~~5~~	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~~Reserved	0C0	~~Flow termination request and confirmation~~Reserved by ITU-T	~~Clause 8.6.2.3.3~~
	~~CL_TerminateFlow.cnf~~Reserved	0C1	Reserved by ITU-T	~~Clause 8.6.2.3.4~~
	~~CL_FlowTerminated.ind~~Reserved	0C2	Reserved by ITU-T	~~Clause 8.6.2.3.5~~
	FL_TerminateFlow.req	0C3	Request flow termination	Clause 8.6.2.3.13
	FL_TerminateFlow.cnf	0C4	Confirm flow termination	Clause 8.6.2.3.14
	FL_ BrokenTunnel .ind	0C5	Indicate broken tunnel	Clause 8.6.2.3.19
	FL_ BrokenTunnel.rsp	0C6	Response to indication	Clause 8.6.2.3.20
	FL_ReleaseTunnel.req	0C7	Request Release Tunnel	Clause 8.6.2.3.21
	FL_ReleaseTunnel.cnf	0C8	Confirm Release Tunnel	Clause 8.6.2.3.22
	FL_DM_RenewTunnel.req	0C9	DM renew tunnel request	Clause 8.6.2.3.23

Table 8-88 – OPCODEs of management messages

Category	Message name	OPCODE (hex)	Description	MMPL Reference
	FL_DM_RenewTunnel.cnf	0CA	Confirm DM renew tunnel	Clause 8.6.2.3.24
	FL_RenewTunnel.req	0CB	Renew tunnel request	Clause 8.6.2.3.25
	FL_RenewTunnel.cnf	0CC	Confirm Renew tunnel	Clause 8.6.2.3.26
	FL_DeleteFlow.req	0CD	Delete Flow request	Clause 8.6.2.3.27
	FL_DeleteFlow.cnf	0CE	Confirm Delete Flow	Clause 8.6.2.3.28
Media Access Plan (0DX)	MAP	0D0	MAP message	Clause 8.8
Channel Estimation 2 (0EX)	CE_Request.ind	0E0	Channel estimation trigger	Clause 8.11.7.11
	CE_Initiation.req	0E1	Channel estimation initiation request	Clause 8.11.7.12
	CE_Initiation.cnf	0E2	Channel estimation initiation confirmation	Clause 8.11.7.13
	CE_ProbeRequest.ind	0E3	Request for PROBE frame transmission	Clause 8.11.7.14
	CE_Cancellation.req	0E4	Channel estimation cancellation request	Clause 8.11.7.15
	CE_BatIdMaintain.ind	0E5	BAT ID maintenance	Clause 8.11.7.16
	CE_Cancellation.cnf	0E6	Channel estimation cancellation confirmation	Clause 8.11.7.17
	Reserved	0E7 – 0EF	Reserved by ITU-T	
Reserved	Reserved	0F0~~A0-F~~7FF	Reserved by ITU-T	
MIMO (8XX – 9XX)	Reserved for use by ITU-T G.9963	800 – 9FF		
Reserved	Reserved	A00 – 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.

NOTE – The sequence number space used by an originating node for a given OPCODE is the same regardless of the destination (e.g., single counter per OPCODE). However, any message that is retransmitted has the same sequence number as that of the original transmission (see second bullet below).

The management of the sequence numbers by an originating node for a given OPCODE shall follow these rules:~~depends on the protocol associated with such OPCODE.~~

- When the MMPL of a message is modified, the message sequence number shall be incremented by 1.
- When the MMPL of a message is not modified, meaning that the same message content is retransmitted, the message sequence number shall be the same as the original transmitted message and the repetition number shall be incremented by 1.
- When a message is relayed, the sequence number and the repetition number fields shall not be modified.

When the field Force Sequence Bit (FSB) of the MMH is set to one, it indicates that the receiver shall process this message without performing any sequence filtering. The receiver shall also consider the sequence number of this message as the latest valid sequence number associated with the transmitter's DeviceID and OPCODE of the message.

NOTE – The increment in the value of the message sequence number is independent of the value of the FSB field.

~~However, t~~The following segmentation rules apply to any segmented LCDU:

- ~~t~~The segmentation shall be done in the ascending order of octets.
- ~~a~~All the segments shall have the same sequence number.
- ~~t~~The segmentation shall not be changed if the LCDU is retransmitted, unless a new sequence number is generated.
- The segmentation shall not be changed if the LCDU is relayed (the sequence number shall remain the same).

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. If the FSB field of the MMH is set to one, the received LCDU shall be considered as the newest. If the FSB field of the MMH is set to zero, ~~S~~sequence numbers of LCDUs with the same OPCODE shall be compared according to the following rules:

- The first LCDU received from a node shall be considered as a new message containing new information. The node shall perform the operations required by the protocol that defines that OPCODE.
- If the sequence number of the new received LCDU is the same as the sequence number of the LCDU already kept by the node, the new received LCDU shall be considered to be equal to the LCDU kept by the node.
- If the sequence number of the new received LCDU is higher than the sequence number of the LCDU already kept by the node and the difference between the numbers is, in absolute value, less than half of SequenceModulus, the new received LCDU shall be considered to be newer. Otherwise it shall be considered to be older.
- If the sequence number of the new received LCDU is lower than the sequence number of the LCDU already kept by the node and the difference between the numbers is, in absolute value, lower than half of SequenceModulus, the new received LCDU shall be considered to be older. Otherwise it shall be considered to be newer.

In any of the above cases, the actions to perform by the node that receives the LCDU depend on the protocol that defines that OPCODE.

NOTE – A transmitter may use the FSB bit to force synchronization with the receiver. Once the transmitter gets confirmation that the receiver is synchronized, it should set FSB to zero.

…

8.10.2.1 Control message OPCODEs

Control message OPCODEs are formatted as 10-bit unsigned integers. Valid values of OPCODEs are for further study.~~presented in Table 8-90. OPCODEs are categorized (typically by their associated protocol or procedure) according to the value of their 6 MSBs.~~

~~Table 8-90/G.9961 – 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_CE_BatIdMaintain.ind~~	~~015~~	~~BAT ID maintenance~~	~~Clause 8.11.8.6~~

Table 8-90 – Placeholder table

(This table has been intentionally left blank

15) Clause 8.11

Revise the text in clause 8.11 and its sub-clauses as follows:

8.11 Channel estimation protocol

The channel estimation protocol describes the procedure of measuring the characteristics of the channel between the transmitter (source) and the receiver (destination) nodes. The procedure involves initiation of channel estimation, transmissions of PROBE frames, and selection of parameters.

Channel estimation can be done in two phases:

- Channel discovery – Initial channel estimation.
- Channel adaptation – Subsequent channel estimation to adapt changing channel.

The protocols used for channel discovery and channel adaptation can be started either by the transmitter or by the receiver. The core part of the channel estimation protocol is identical in these two cases, and is always initiated by the receiver (receiver-initiated channel estimation). The transmitter can request the receiver to initiate channel estimation (transmitter-requested channel estimation).

During the initiation process, the transmitter and receiver jointly determine input parameters for channel estimation such as channel estimation window (a fraction of a MAC cycle over which channel estimation should be executed), the minimum value of G ($G_{\min}$, see clause 7.1.4.2.4 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 for PROBE frame transmission request via ACK_CE_CTRL as described in clause 8.11.1.4), the transmitter transmits the PROBE frame using parameters previously selected by the receiver. The receiver is not required to request PROBE frames if it chooses other means such as MSG frames or PROBE frames transmitted to other nodes to estimate the channel. ~~The 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;
- ~~g~~Guard interval for payload;
- PSD ceiling.

The receiver may cancel the channel estimation process without generating new channel estimation parameters.

The protocol provides several options to expedite the channel estimation process for faster channel adaptation. For example, the channel estimation initiation process (clause 8.11.1.1) can be omitted in case of channel adaptation where no new input parameter negotiation is necessary. The receiver can create a new BAT by sending an unsolicited CE_ParamUpdate.req (clause 8.11.3.1) or update the existing BAT by sending a CE_PartialBatUpdate.req (clause 8.11.3.2). The receiver can request PROBE frame transmission without going through the channel estimation initiation process (clause 8.11.4).

8.11.1 Receiver-initiated channel estimation

The following procedure describes the receiver-initiated channel estimation process:

1) The receiver initiates the channel estimation process by sending the transmitter a ~~CM_~~CE_Initiation.req message. The receiver may request a PROBE frame transmission in this message (channel estimation initiation, see clause 8.11.1.1).

2) Upon reception of the channel estimation initiation request, if the transmitter does not have transmit opportunities for a given channel estimation window, it shall request the domain master to allocate bandwidth for PROBE frame transmission by sending a CE_ProbeSlot~~Request.ind~~Assign.req message. The domain master shall confirm that it received the bandwidth request by replying with the CE_ProbeSlotAssign.cnf message (bandwidth request, see clause 8.11.1.2).

3) Depending on the availability of the bandwidth, the transmitter may grant or reject the channel estimation initiation request by sending the receiver a ~~CM_~~CE_Initiation.cnf message (channel estimation initiation confirmation, see clause 8.11.1.3).

4) Upon reception of the CE_Initiation.cnf message indicating channel estimation initiation confirmation, the receiver may request the transmitter to send additional PROBE frames by sending a ~~CM_~~CE_ProbeRequest.ind message or through the ACK_CE_CTRL field in the PFH of an ACK frame (request for PROBE frame ~~request~~transmission, see clause 8.11.1.4).

5) Upon reception of the request for PROBE frame ~~request~~transmission, the transmitter shall transmit the PROBE frame as the receiver requested (PROBE frame transmission, see clause 8.11.1.5).

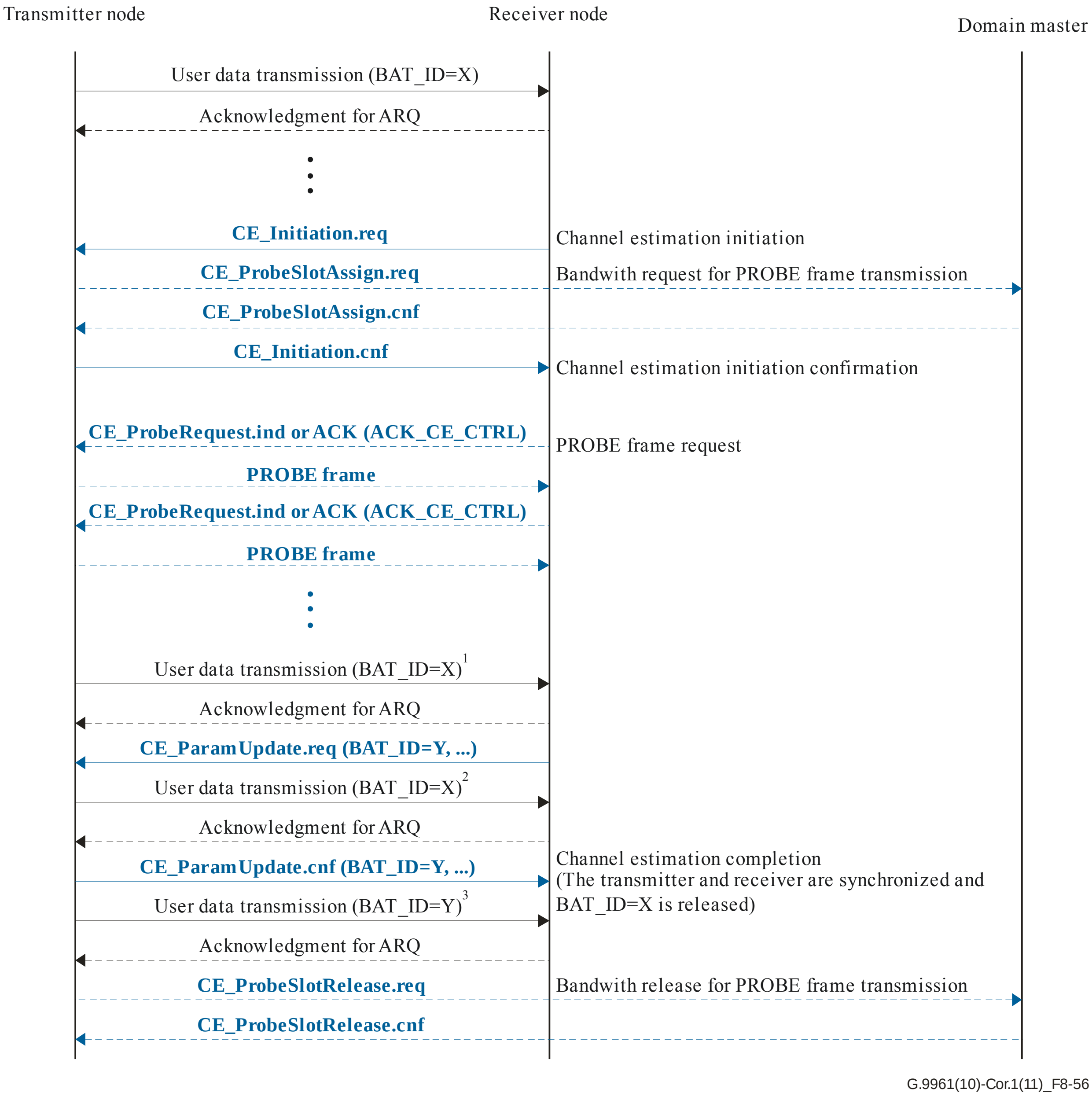
6) Steps 4 and 5 can repeat until the receiver sends the transmitter the final outcome of channel estimation using the CE_ParamUpdate.req~~ind~~ message. The transmitter shall confirm reception of the new parameters by sending a CE_ParamUpdate.cnf message (channel estimation completion, see clause 8.11.1.6). Steps 4 and 5 may be skipped altogether if the receiver does not need additional PROBE frames.

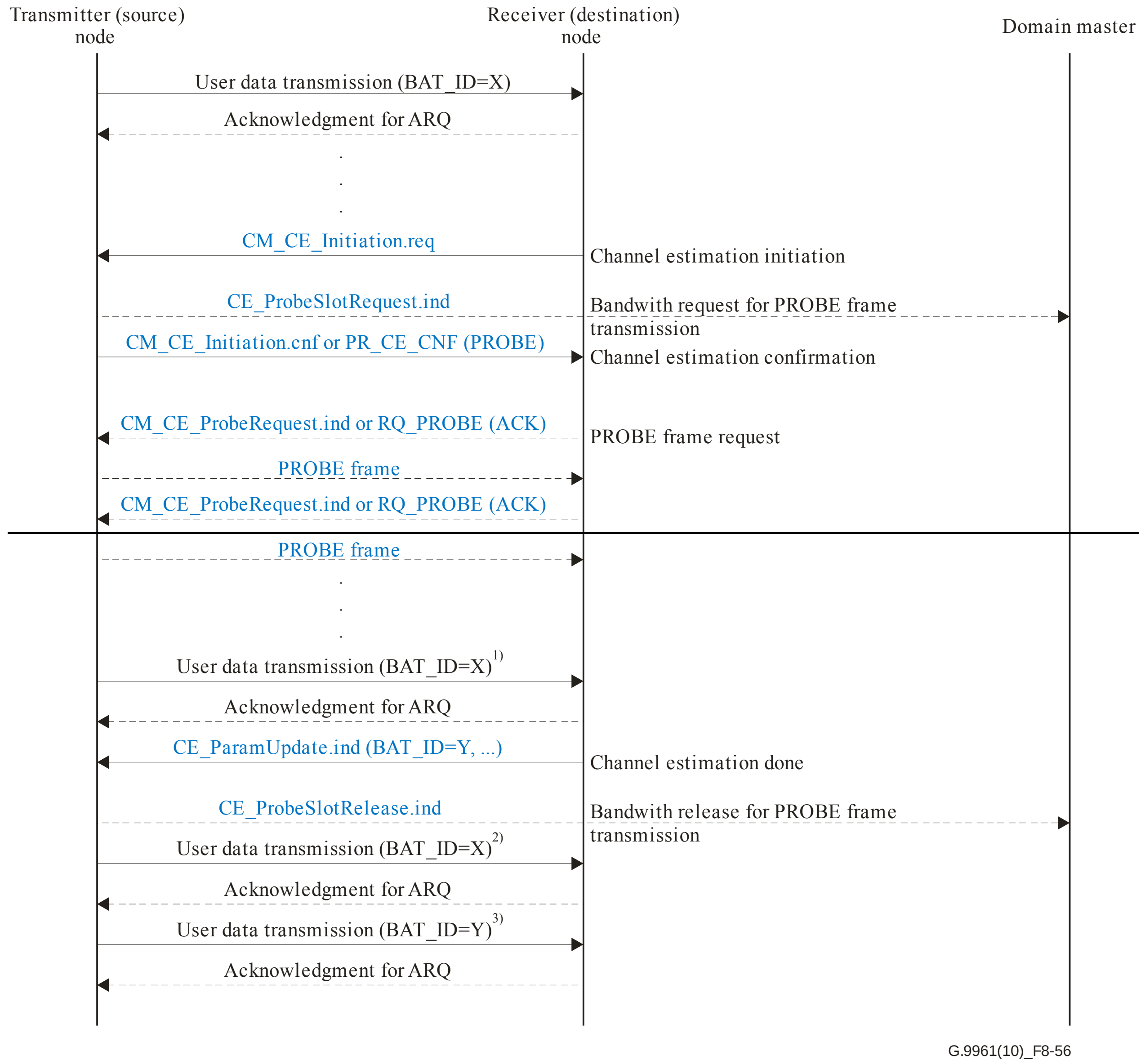
7) The receiver may cancel the channel estimation process anytime after it receives the channel estimation initiation confirmation by sending a ~~CM~~ CE_Cancellation.~~ind~~req message~~ or by setting a flag in an ACK frame~~ (channel estimation cancellation, see clause 8.11.1.7).

8) Upon reception of a CE_ParamUpdate.req~~ind~~ message, ~~the transmitter~~ if the transmitter has been allocated extra bandwidth for the PROBE frame transmission, it shall ~~may~~ send a CE_ProbeSlotRelease.req~~ind~~ message to the domain master to release the bandwidth used for PROBE frame transmission. The domain master shall confirm the bandwidth release request by replying with a CE_ProbeSlotRelease.cnf message (bandwidth release, see clause 8.11.1.8).

The transmitter ~~can~~may send ~~an MSG~~ frames carrying payload with the existing settings (e.g., ~~old~~any valid runtime BAT or predefined BAT) at any time during this process.

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


Transmitter node
Receiver node
Domain master
User data transmission (BAT_ID=X)
Acknowledgment for ARQ
CE_Initiation.req
Channel estimation initiation
CE_ProbeSlotAssign.req
Bandwith request for PROBE frame transmission
CE_ProbeSlotAssign.cnf
CE_Initiation.cnf
Channel estimation initiation confirmation
CE_ProbeRequest.ind or ACK (ACK_CE_CTRL)
PROBE frame request
PROBE frame
CE_ProbeRequest.ind or ACK (ACK_CE_CTRL)
PROBE frame
User data transmission (BAT_ID=X)1
Acknowledgment for ARQ
CE_ParamUpdate.req (BAT_ID=Y, ...)
User data transmission (BAT_ID=X)2
Acknowledgment for ARQ
CE_ParamUpdate.cnf (BAT_ID=Y, ...)
Channel estimation completion
(The transmitter and receiver are synchronized and
BAT_ID=X is released)
User data transmission (BAT_ID=Y)3
Acknowledgment for ARQ
CE_ProbeSlotRelease.req
Bandwith release for PROBE frame transmission
CE_ProbeSlotRelease.cnf
G.9961(10)-Cor.1(11)_F8-56


Transmitter (source) node
Receiver (destination) node
Domain master
User data transmission (BAT_ID=X)
Acknowledgment for ARQ
CM_CE_Initiation.req
Channel estimation initiation
CE_ProbeSlotRequest.ind
Bandwith request for PROBE frame transmission
CM_CE_Initiation.cnf or PR_CE_CNF (PROBE)
Channel estimation confirmation
CM_CE_ProbeRequest.ind or RQ_PROBE (ACK)
PROBE frame request
PROBE frame
CM_CE_ProbeRequest.ind or RQ_PROBE (ACK)
PROBE frame
User data transmission (BAT_ID=X)1)
Acknowledgment for ARQ
CE_ParamUpdate.ind (BAT_ID=Y, ...)
Channel estimation done
CE_ProbeSlotRelease.ind
Bandwith release for PROBE frame transmission
User data transmission (BAT_ID=X)2)
Acknowledgment for ARQ
User data transmission (BAT_ID=Y)3)
Acknowledgment for ARQ
G.9961(10)_F8-56

[1] The transmitter can transmit data using the existing BAT anytime during channel estimation process.

[2] 1st user data transmission after CE_ParamUpdate.ind may still use old channel estimation parameters.

[3] The transmitter decides when to apply updated channel estimation parameters within a given constraint.

NOTE – Dotted-lines indicate optional communications.

Figure 8-56 – Receiver-initiated channel estimation

8.11.1.1 Channel estimation initiation

The receiver initiates the channel estimation process by sending the transmitter a ~~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 default PROBE frame. If the CE_PRB_RQST field is not set to one, parameters for the default PROBE frame shall be as follows: CE_PR_PRBTYPE = 0001_2;

CE_PR_PRBFN = 0000$_2$; CE_PR_PRBSYM = 0011$_2$; CE_PR_PRBGI = 111$_2$ and CE_PR_APSDC = 31.

The receiver may resend the ~~CM_~~CE_Initiation.req message, if it does not receive the ~~a~~ ~~CM_~~CE_Initiation.cnf message within 200 ms~~ec~~.

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_ProbeSlot~~Request.ind~~Assign.req message.

The transmitter shall provide the domain master the channel estimation identifier (i.e., CE_BAT_ID, Transmitter_ID, and Receiver_ID), channel estimation window (CE_STIME and CE_ETIME), and PROBE frame parameters (CE_PRB_PARM) as provided by ~~CM_~~CE_Initiation.req message.

The transmitter shall provide the priority of the bandwidth request in the CE_ProbeSlotAssign.req message by setting the CE_PRIORITY field to the highest priority of the user data traffic that the transmitter has to send to the specified receiver.

The domain master shall confirm the bandwidth request by replying to the transmitter with a CE_ProbeSlotAssign.cnf indicating whether or not the request is granted within 100 ms after it receives the CE_ProbeSlotAssign.req message.

The domain master should allocate bandwidth so that at least one PROBE frame with requested parameters can be transmitted during the channel estimation window. The additional TSs or TXOPs shall only be used for PROBE frame transmissions (see clause 8.8.4.1.1). If the domain master has granted extra bandwidth for PROBE frame transmission, it should keep this bandwidth until it receives the bandwidth release request from the transmitter (see clause 8.11.1.8).

If the transmitter does not receive the CE_ProbeSlotAssign.cnf message, it may resend the CE_ProbeSlotAssign.req message multiple times before it transmits the channel estimation initiation confirmation.

8.11.1.3 Channel estimation initiation confirmation

The transmitter shall confirm~~s~~ the channel estimation initiation request ~~process~~ by sending the receiver a ~~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~~in the CE_Initiation.req message. 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 $i = G \times j, G \times j+1, \ldots, G \times j+G–1$, 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, t~~The transmitter shall send a ~~CM_CE_Initiation.cnf message using a CTMG frame. The transmitter shall send CM_~~CE_Initiation.cnf message within 100 ms~~ec~~ after it receives a ~~CM_~~CE_Initiation.req message. If the transmitter needs to request the bandwidth for PROBE frame transmission, the transmitter shall send a CE_Initiation.cnf message within 200 ms~~ec~~.

8.11.1.4 Request for PROBE frame ~~request~~ transmission

Once a channel estimation initiation request has been confirmed, the receiver may request the transmitter to send additional PROBE frames by sending a ~~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~~the CE_PRB_PARM field~~s~~ of the ~~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]~~8.11.4).

The receiver may not request PROBE frames at all if it uses ~~MSG~~ other frames carrying payload (e.g., MSG, BMSB, BACK) to estimate the channel.

8.11.1.5 PROBE frame transmission

Upon reception of ~~the~~ a request for PROBE frame ~~request~~transmission, the transmitter shall transmit PROBE frames ~~as the receiver requested~~ as soon as possible as described in clause 8.11.4.

~~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 request ~~is~~ has been confirmed, the receiver may send the transmitter the outcome of channel estimation using the CE_ParamUpdate.~~ind~~req message. The transmitter shall confirm reception of the new parameters by replying with the CE_ParamUpdate.cnf message within 100 ms.

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

If the transmitter does not receive a~~ny frame or~~ message that is related to channel estimation (i.e., CE_ProbeRequest.ind or CE_ParamUpdate.req), or does not receive a request for PROBE frame transmission via an ACK frame, within ~~the duration of~~ 200 ms after the channel estimation initiation request has been confirmed~~confirmation~~, it may send the receiver a CE_ParamUpdateRequest.ind message to request the receiver to resend the result of the specified channel estimation.

If the transmitter does not receive either CE_ParamUpdate.req or CE_Cancellation.req within 400 ms after the channel estimation initiation request has been confirmed, it shall abort the channel estimation process.

8.11.1.7 Channel estimation cancellation

At any time after channel estimation initiation ~~is~~ request has been confirmed, the receiver may cancel the channel estimation process using ~~CM~~ CE_Cancellation.req~~ind~~ message. The transmitter shall confirm receiving the cancellation request within 100 ms by replying with the CE_Cancellation.cnf message. If the receiver does not receive the CE_Cancellation.cnf message within 200 ms, it may resend the CE_Cancellation.req message.~~This message shall be carried using a CTMG frame.~~

If the receiver does not receive either CE_ParamUpdate.cnf or CE_Cancellation.cnf within 400 ms after the channel estimation initiation request has been confirmed, it shall abort the channel estimation process and consider the CE_BAT ID as invalid (see clause 8.11.5).

~~Alternatively, the receiver may use the ACK_CE_CTRL_TYPE field in an ACK frame to indicate cancellation.~~ In ~~either~~ this case, the channel estimation is finished without generating a new BAT.

8.11.1.8 Channel estimation bandwidth release

Upon reception of the CE_ParamUpdate.~~ind~~req or CE_Cancellation.req message, the transmitter ~~may~~ shall request the domain master to release any bandwidth previously assigned for PROBE frame transmission by sending CE_ProbeSlotRelease.~~ind~~req message.

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

The domain master shall confirm receiving the CE_ProbeSlotRelease.req message within 100 ms by replying with the CE_ProbeSlotRelease.cnf message. If the domain master does not receive a CE_ProbeSlotRelease.req message from the transmitter within 800 ms after the bandwidth was assigned, it shall release the bandwidth allocated to the transmitter for PROBE frames. The domain master shall only release bandwidth additionally assigned to the transmitter for PROBE frame transmission ~~over a given~~for the associated channel estimation identifier~~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~~may send ~~an MSG~~ frames carrying payload with the existing settings (e.g., ~~old~~any valid runtime BAT or pre-defined BAT) any time during this process.

8.11.2.1 Channel estimation request

The transmitter triggers the channel estimation process by sending the receiver ~~CM~~ CE_Request.ind message. ~~This message shall be carried using a CTMG frame.~~

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

If the transmitter does not receive CE_Initiation.req within 200 ms~~ec~~ after CE_Request.ind is sent, it may resend the channel estimation request message.

8.11.3 Shortened channel estimation processes~~Unsolicited CE_ParamUpdate.ind~~

8.11.3.1 Unsolicited CE_ParamUpdate.req

It is not required to exchange PROBE frames between transmitter and receiver in order to exchange a new BAT between them. The receiver may send a new BAT at any time to the transmitter by sending a CE_ParamUpdate.req~~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 or other frames carrying payload (e.g., MSG, BMSG, BACK) transmitted to other nodes to estimate the channel.

Upon receiving the CE_ParamUpdate.req message, the transmitter shall reply by sending the CE_ParamUpdate.cnf message within 100 ms indicating whether the transmitter adopts the new BAT or rejects the new BAT due to lack of resources.

If the receiver does not receive CE_ParamUpdate.cnf within 200 ms, it may retry with the same or different CE_ParamUpdate.req message.

8.11.3.2~~4~~ ~~Channel adaptation via p~~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~~any valid 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 and reply with the PBU confirmation. After receiving the first ~~MSG~~ frame carrying payload using the N_BAT_ID, the receiver shall consider O_BAT_ID is invalid (see clause 8.11.5).

~~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,~~

~~No MSG frame with N_BAT_ID is received from the transmitter in 100 msec and one MSG frame is received after this time with O_BAT_ID.~~

3~~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.

8.11.3.2.1~~4.1~~ PBU request

~~8.11.4.1.1 PBU request via management message~~

The receiver may send the PBU request using the~~ management message,~~ CE_PartialBatUpdate.~~ind~~req message in which the receiver can request bit allocation changes for up to 1024 sub-carriers. Figure 8-57 illustrates an example of partial BAT update using this approach. Note that ~~retransmission~~ acknowledgement is disabled in this example.

If the receiver does not receive CE_PartialBatUpdate.cnf within 200 ms, it may retry with the same or different CE_PartialBatUpdate.req message.

8.11.3.2.2 PBU confirmation

Upon reception of CE_PartialBatUpdate.~~ind~~req message, the transmitter should incorporate the new channel estimation parameters as soon as possible and then send the CE_PartialBatUpdate.cnf message to confirm the received CE_PartialBatUpdate.req message within 100 ms. The receiver shall infer loss of the PBU request if the PBU confirmation is not received within 200 ms after it transmits CE_PartialBatUpdate.req. The transmitter may switch to N_BAT_ID before sending the CE_PartialBatUpdate.cnf message.

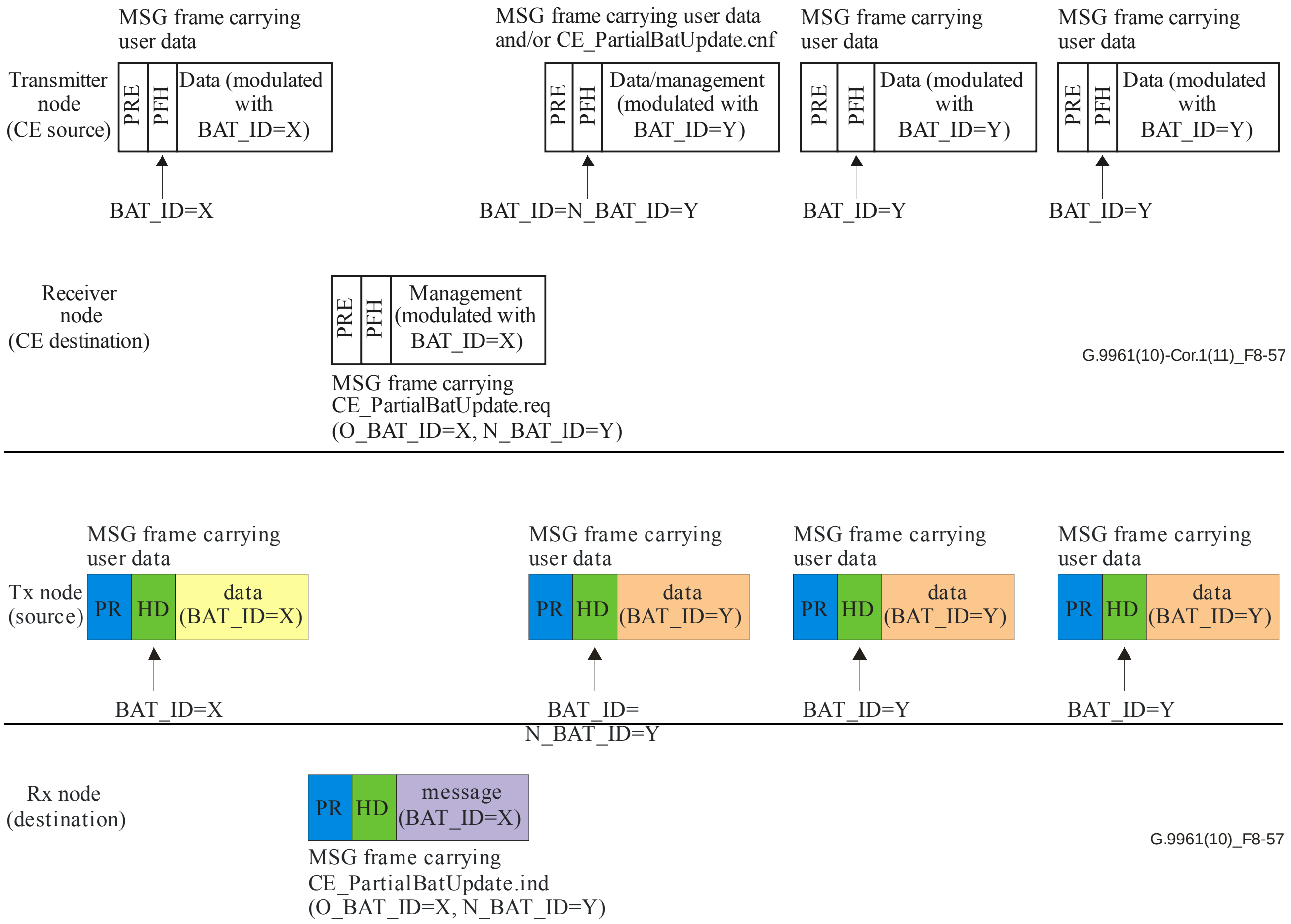

Transmitter node (CE source)
MSG frame carrying user data
PRE
PFH
Data (modulated with BAT_ID=X)
BAT_ID=X
MSG frame carrying user data and/or CE_PartialBatUpdate.cnf
Data/management (modulated with BAT_ID=Y)
BAT_ID=N_BAT_ID=Y
MSG frame carrying user data
Data (modulated with BAT_ID=Y)
BAT_ID=Y
MSG frame carrying user data
Data (modulated with BAT_ID=Y)
BAT_ID=Y
Receiver node (CE destination)
Management (modulated with BAT_ID=X)
MSG frame carrying CE_PartialBatUpdate.req (O_BAT_ID=X, N_BAT_ID=Y)
G.9961(10)-Cor.1(11)_F8-57
Tx node (source)
MSG frame carrying user data
PR
HD
data (BAT_ID=X)
BAT_ID=X
MSG frame carrying user data
data (BAT_ID=Y)
BAT_ID= N_BAT_ID=Y
MSG frame carrying user data
data (BAT_ID=Y)
BAT_ID=Y
MSG frame carrying user data
data (BAT_ID=Y)
BAT_ID=Y
Rx node (destination)
message (BAT_ID=X)
MSG frame carrying CE_PartialBatUpdate.ind (O_BAT_ID=X, N_BAT_ID=Y)
G.9961(10)_F8-57

Figure 8-57 – Example of ~~P~~partial BAT update using management message

8.11.4 Channel estimation using PROBE frames

The receiver can request the transmitter for PROBE frame transmission at any time after registration without going through the channel estimation initiation process.

To request PROBE frames, the receiver may use CE_ProbeRequest.ind messages or the ACK_CE_CTRL field in the PFH of an ACK frame (see clause 7.1.2.3.2.3.8 of [ITU-T G.9960]). Upon reception of a request for PROBE frame transmission, the transmitter should transmit PROBE frames as soon as possible.

If the receiver requests a PROBE frame through a specific management message, the transmitter shall transmit the PROBE frame using parameters selected by the receiver, that is, the parameters selected in the latest request for PROBE frame transmission (CE_ProbeRequest.ind) or channel estimation initiation (CE_Initiation.req).

If the receiver requests a PROBE frame through an ACK frame, the transmitter shall use the default PROBE frame. The transmitter shall use the default PROBE frame for all ACK frame-based requests for PROBE frame transmission by the receiver. In this case, the transmitter may use an entire MAC cycle to transmit PROBE frames, regardless of a particular channel estimation window associated with the BAT_ID under channel estimation.

The parameters for the default PROBE frame are determined by the receiver through the CE_Initiation.req message as described in clause 8.11.1.1. Alternatively, they can be updated by setting a bit in the CE_ProbeRequest.ind message as described in Table 8-102.

When a transmitter receives a request for PROBE frame transmission from a receiver while handling previous requests for PROBE frame transmission from the same receiver, it should ignore the new request if the requested parameters are the same as the old ones, regardless of the value of the BAT_ID under estimation.

NOTE – The transmitter should try to cover as much of the channel estimation window as possible when generating PROBE frames.

When the receiver requests a PROBE frame via ACK frames, it may request multiple times by sending multiple ACK frames by setting ACK_CE_CTRL until it receives the PROBE frame. The transmitter should ignore new requests for PROBE frame transmission coming from the receiver in order to avoid unnecessary PROBE transmissions.

After PROBE transmissions, the receiver may send the outcome of channel estimation to the transmitter in case it is needed, using an unsolicited CE_ParamUpdate.req (clause 8.11.3.1) or a partial BAT update (clause 8.11.3.2).

8.11.5 BAT_ID maintenance

The receiver is responsible for tracking the list of valid and invalid BAT_IDs. The receiver informs the transmitter of the valid BAT_IDs in the VALID_BAT_ID field by sending a CE~~M~~_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 all the BAT_IDs are marked as invalid, the transmitter may use RCM mode. In this case, the transmitter should use the parameters indicated in the CE_BatIdMaintain.ind message.

If a BAT_ID is marked as valid by the receiver but the transmitter does not have a BAT associated with it (e.g., the transmitter fails to receive CE_ParamUpdate.~~ind~~req), the transmitter shall send a CE_ParamUpdateRequest.ind message requesting the transmission of the BAT.

The receiver ~~can ~~may instruct the transmitter to stop using~~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]). The transmitter shall then consider the BAT_ID as invalid.

The receiver may invalidate a BAT_ID as part of the channel estimation cancellation process (see clause 8.11.1.7).

8.11.6 ACE symbol insertion

The receiver may request the transmitter to attach up to seven ACE symbols (see clause 7.1.2.1) 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 that are allowed to carry ACE symbols. The receiver may use the same procedure to remove or change the number of ACE symbols.

8.11.7 Management message formats for channel estimation

8.11.7.1 Format of CE_ProbeSlot~~Request.ind~~Assign.req

The format of the MMPL of the CE_ProbeSlot~~Request.ind~~Assign.req message shall be as shown in Table 8-91.

Table 8-91 – Format of the MMPL of the CE_ProbeSlot~~Request.ind~~Assign.req

Field	Octet	Bits	Description
Transmitter ID	0	[7:0]	The DEVICE_ID of the node requesting the bandwidth allocation for probe transmissions.
Receiver_ID	1	[7:0]	The DEVICE_ID of the receiver node in the channel estimation procedure.

Table 8-91 – Format of the MMPL of the CE_ProbeSlot~~Request.ind~~Assign.req

Field	Octet	Bits	Description
CE_BAT_ID	2~~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	3~~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	4~~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~~5 to 7~~6~~	[23:0]	This field specifies a set of parameters for PROBE frame. It shall be coded as shown in Table 8-102.
CE_PRIORITY	8	[2:0]	This field specifies the highest user priority of the traffic the transmitter has to transmit to the specified receiver.
Reserved		[7:3]	Reserved by ITU-T (Note).
NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

8.11.7.2 Format of CE_ProbeSlotRelease.req~~ind~~

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

Table 8-92 – Format of the MMPL of the CE_ProbeSlotRelease.req~~ind~~ message

Field	Octet	Bits	Description
Transmitter ID	0	[7:0]	The DEVICE_ID of the node requesting the bandwidth allocation for probe transmissions.
Receiver ID	1	[7:0]	The DEVICE_ID of the receiver node in the channel estimation procedure.
CE_BAT_ID	2~~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	3~~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	4~~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.req~~ind~~

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

Table 8-93 – Format of the MMPL of the CE_ParamUpdate.req~~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~~ at the channel estimation initiation 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).
VALID_BAT_ID	2 to 4	[23:0]	This field contains a bitmap indicating which runtime BATs are valid (including the New BAT ID) for this node (SID) when receiving from the destination node (DID). Each bit is associated with one runtime BAT. The LSB of the VALID_BAT_ID shall be set to one if runtime BAT 8 is valid. The MSB of the VALID_BAT_ID shall be set to one if runtime BAT 31 is valid.
NUM_TX_ AVAIL_BATS	5	[4:0]	This field contains the number of runtime BATs, assuming *G* =1, that this node (SID) can support when transmitting to the destination node (DID). Valid values are from 0 to 24.
Reserved		[7:5]	Reserved by ITU-T (Note 1).
New block size	6~~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	7~~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$_1$	8~~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$_1$	9~~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.
…	…	…	…

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

Field	Octet	Bits	Description
CE_STIME$_v$	2V+6~~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+7~~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+84) to (2V+10~~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+11 ~~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$_{MIN}$ + 2*G* – 1 (Notes 6, 7, 8).
…	…	…	…
B$_Z$	2V+10 ~~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).
		[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 of the channel estimation result. 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.req~~ind~~

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

Table 8-95 – Format of the MMPL of the CE_PartialBatUpdate.req~~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).
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~~current GRP_ID associated with the BAT corresponding to O_BAT_ID and N_BAT_ID (G). Partial BAT update shall not change the ~~minimum~~current 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	(2V+2) to (2V+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 all frames sent to the receiver that are allowed to carry ACE symbols~~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.			

8.11.7.7 Format of CE_ProbeSlotAssign.cnf

The format of the MMPL of the CE_ProbeSlotAssign.cnf message shall be as shown in Table 8-96.1.

Table 8-96.1 – Format of the MMPL of the CE_ProbeSlotAssign.cnf message

Field	Octet	Bits	Description
Transmitter ID	0	[7:0]	The DEVICE_ID of the node requesting the bandwidth allocation for probe transmissions.
Receiver ID	1	[7:0]	The DEVICE_ID of the receiver node in the channel estimation procedure.
CE_BAT_ID	2	[4:0]	This field indicates the BAT_ID associated with the runtime BAT to which bandwidth was required for probing.
Request Status		[7:5]	0 – Bandwidth request is confirmed (Note). 1 – Request is rejected. 2 to 7 – Reserved by ITU-T.
NOTE – Bandwidth allocation will be identified in the MAP using the Transmitter_ID (SID), Receiver_ID (DID) and channel estimation only indication set in the TXOP attributes extension (see clause 8.8.4.1.1).			

8.11.7.8 Format of CE_ProbeSlotRelease.cnf

The format of the MMPL of the CE_ProbeSlotRelease.cnf message shall be as shown in Table 8-96.2.

Table 8-96.2 – Format of the MMPL of the CE_ProbeSlotRelease.cnf message

Field	Octet	Bits	Description
Transmitter ID	0	[7:0]	The DEVICE_ID of the node requesting the bandwidth allocation for probe transmissions.
Receiver ID	1	[7:0]	The DEVICE_ID of the receiver node in the channel estimation procedure.
CE_BAT_ID	2	[4:0]	This field indicates the BAT_ID associated with the runtime BAT for which the bandwidth has to be released.
Request Status		[7:5]	0 – Request is confirmed. 1 – Request is rejected (unknown BAT identity) (Note). 2 to 7 – Reserved by ITU-T.
NOTE – There is no bandwidth allocated for the identified channel estimation procedure. The identification is defined by the Transmitter_ID, Receiver_ID, and CE_BAT_ID.			

8.11.7.9 Format of CE_ParamUpdate.cnf

The format of the MMPL of the CE_ParamUpdate.cnf message shall be as shown in Table 8-96.3.

Table 8-96.3 – Format of the MMPL of the CE_ParamUpdate.cnf message

Field	Octet	Bits	Description
New BAT ID	0	[4:0]	This field indicates the BAT_ID specified in the received CE_ParamUpdate.req message.
Reserved		[7:5]	Reserved by ITU-T (Note).
NUM_AVAIL_BATS	1	[4:0]	This field contains the number of available runtime BATs, assuming *G* = 1, that this node (SID) can support when transmitting to the destination node (DID). It excludes the BAT associated with the New BAT ID. Valid values are from 0 to 23.
Request Status		[7:5]	0 – BAT successfully updated. 1 – Update is rejected (no more resources). 2 – Update is rejected (New BAT ID already exists). 3 to 7 – Reserved by ITU-T.
NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

8.11.7.10 Format of CE_PartialBatUpdate.cnf

The format of the MMPL of the CE_PartialBatUpdate.cnf message shall be as shown in Table 8-96.4.

Table 8-96.4 – Format of the MMPL of the CE_PartialBatUpdate.cnf message

Field	Octet	Bits	Description
CE_BAT_ID	0	[4:0]	This field indicates the CE_BAT_ID specified in the CE_PartialBatUpdate.req message
Reserved		[7:5]	Reserved by ITU-T (Note)
NUM_AVAIL_BATS	1	[4:0]	This field contains the number of available runtime BATs, assuming *G* = 1, that this node (SID) can support when transmitting to the destination node (DID). It excludes the BAT associated with the CE_BAT_ID. Valid values are from 0 to 23.
Request Status		[7:5]	0 – BAT successfully updated 1 – Request rejected (no more resources) 2 – Request rejected (O_BAT_ID does not exist) 3 – Request rejected (N_BAT ID already exist) 4 to 7 – Reserved by ITU-T
NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

~~8.11.8 Control message formats for channel estimation~~

8.11.7.11~~8.1~~ Format of ~~CM_~~CE_Request.ind

The format of the ~~C~~MMPL of the ~~CM_~~CE_Request.ind message shall be as shown in Table 8-97.

Table 8-97 – Format of the ~~C~~MMPL 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.			

…

8.11.7.12~~8.2~~ Format of ~~CM_~~CE_Initiation.req

The format of the ~~C~~MMPL of the ~~CM_~~CE_Initiation.req message shall be as shown in Table 8-100.

Table 8-100 – Format of the ~~C~~MMPL 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].
CE_STIME	1	[7:0]	This field indicates the time at which the transmitter can start PROBE frame transmissions, and it shall be coded as shown in Table 8-98.
CE_ETIME	2	[7:0]	This field indicates the time at which the transmitter shall end PROBE frame transmissions, and it shall be coded as shown in Table 8-99.
CE_PRB_RQST	3	[0]	This field shall be set to one if the receiver wants PROBE frames along with channel estimation initiation confirmation. It shall be set to zero otherwise.
Reserved		[7:1]	Reserved by ITU-T (Note).
CE_PRB_PARM	4 to 6	[23:0]	This field specifies a set of parameters for PROBE frame. It shall be coded as shown in Table 8-102. This field shall be set to 000000_{16} if CE_PRB_RQST is set to zero.
NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

8.11.7.13~~8.3~~ Format of ~~CM_~~CE_Initiation.cnf

The format of the ~~C~~MMPL of the ~~CM_~~CE_Initiation.cnf message shall be as shown in Table 8-101.

Table 8-101 – Format of the ~~C~~MMPL 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).~~
NUM_AVAIL_BATS	1	[4:0]	This field contains the number of available runtime BATs, assuming *G* = 1, that this node (SID) can support when transmitting to the destination node (DID). It excludes the BAT associated with the CE_BAT_ID. Valid values are from 0 to 23.
Request Status		[7:5]	0 – Channel estimation initiation is confirmed. 1 – Rejected (CE_BAT_ID is valid and currently in use). 2 – Rejected (Bandwidth for PROBE frame transmission is not available). 3 – Rejected (Bandwidth request for probe frame transmission is pending). 4 – Rejected (Channel estimation window is currently not available). 45 to 7 – Reserved by ITU-T (Note).
NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

8.11.7.14~~8.4~~ Format of CE_ProbeRequest.ind

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

Table 8-102 – Format of the MMPL of the CE_ProbeRequest.ind message

Field	Octet	Bits	Description
CE_BAT_ID	0	[4:0]	This field indicates the BAT_ID associated with the runtime BAT to be created by channel estimation. It shall be formatted as shown in Table 7-55 of [ITU-T G.9960].
CE_PRB_DEFAULT_IND		[5]	When this field is set to one, the parameters provided in this message (CE_PRB_PARM) replace the existing parameters for the default PROBE frame for this node (SID) when receiving from the destination node (DID).
Reserved		[7:6]	Reserved by ITU-T (Note).
CE_PRB_PARM	1 to 3	[23:0]	See Table 8-102.1.
NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

Table 8-102.1 – Format of the ~~C~~MMPL 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 request for PROBE ~~request~~frame transmission. The field shall be coded as shown in Table 8-103. The transmitter may send multiple PROBE frames within a single channel estimation window.
CE_PR_PRBSYM	1	[3:0]	This field indicates the PRBSYM requested by the receiver. It shall be formatted as shown in Table 7-40 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 7.1.2.3.2.7.1.4 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.			

…

8.11.7.15~~8.5~~ Format of ~~CM_~~CE_Cancellation.req~~ind~~

The format of the ~~C~~MMPL of the ~~CM_~~CE_Cancellation.req~~ind~~ message shall be as shown in Table 8-104.

Table 8-104 – Format of the ~~C~~MMPL of the ~~CM_~~CE_Cancellation.req~~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.
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.
RCM_BAT_ID		[3]	When USE_RCM is set to one, this field indicates the pre-defined BAT associated with RCM. It shall be set to the following value: zero, when pre-defined BAT Type 1 is used. one, when pre-defined BAT Type 2 is used.
Reserved		[7:4~~3~~]	Reserved by ITU-T (Note).
NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

8.11.7.16~~8.6~~ Format of C~~M_~~E_BatIdMaintain.ind

The format of the ~~C~~MMPL of the C~~M~~E_BatIdMaintain.ind message shall be as shown in Table 8-105.

Table 8-105 – Format of the ~~C~~MMPL of the C~~M_~~E_BatIdMaintain.ind message

Field	Octet	Bits	Description
VALID_BAT_ID	0 to 2~~1~~	[~~15~~23:0]	This field contains a bitmap indicating which runtime BATs are valid for this node (SID) when receiving from the destination node (DID). Each bit is associated with one runtime BAT. The LSB of the VALID_BAT_ID shall be set if runtime BAT 8~~16~~ is valid. The MSB of the VALID_BAT_ID shall be set if runtime BAT 31 is valid.

Table 8-105 – Format of the ~~C~~MMPL of the C~~M~~_E_BatIdMaintain.ind message

Field	Octet	Bits	Description
~~MAX_~~NUM_TX_AVAIL_BATS~~_ID~~	3~~2~~	[4:0]	This field contains the ~~maximum~~ number of ~~valid~~ available runtime BATs~~_ID~~, assuming *G*=1, that this node (SID) can support when transmitting to the destination node (DID). Valid values are from 0 to 24~~16~~.
Reserved		[7:5]	Reserved by ITU-T (Note 1)
New block size	4	[1:0]	This field indicates the proposed BLKSZ associated with RCM, if there is no available runtime BAT (Note 2). It shall be formatted as shown in Table 7-7 of [ITU-T G.9960]. It shall be set to 0 otherwise.
New FEC rate		[4:2]	This field indicates the proposed FEC_RATE associated with RCM, if there is no available runtime BAT (Note 2). It shall be formatted as shown in Table 7-12 of [ITU-T G.9960]. It shall be set to 0 otherwise.
Bandplan ID		[7:5]	This field indicates the BNDPL based on which the RCM parameters are proposed, if there is no available runtime BAT (Note 2). It shall be formatted as shown in Table 7-10 of [ITU-T G.9960]. It shall be set to 0 otherwise.
Repetitions	5	[2:0]	This field indicates the proposed number of repetitions associated with RCM, if there is no available runtime BAT (Note 2). It shall be formatted as shown in Table 7-8 of [ITU-T G.9960]. It shall be set to 0 otherwise
RCM_BAT_ID		[3]	This field indicates the pre-defined BAT associated with RCM, if there is no available runtime BAT (Note 2). It shall be set to the following value: zero, when pre-defined BAT Type 1 is used one, when pre-defined BAT Type 2 is used
Reserved		[7:4]	Reserved by ITU-T (Note 1)
NOTE 1 – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver. NOTE 2 – Runtime BATs might only be available for specified time periods (see Table 8-93).			

8.11.7.17 Format of CE_Cancellation.cnf

The format of the MMPL of the CE_Cancellation.cnf message shall be as shown in Table 8-105.1.

Table 8-105.1 – Format of the MMPL of the CE_Cancellation.cnf message

Field	Octet	Bits	Description
CE_BAT_ID	0	[4:0]	This field indicates the BAT_ID specified in the received CE_Cancellation.req message.
Request Status		[7:5]	0 – Channel estimation is successfully cancelled 1 – no ongoing channel estimation for this CE_BAT_ID. 2 to 7 – Reserved by ITU-T (Note)
NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

16) Clause 8.12

Revise the text in clause 8.12 and its sub-clauses as follows:

8.12 Connection management

Connection management is a mechanism used by the transmitter and the receiver to exchange information about the availability of resources to manage the communication. Connections may be established by the transmitter and may be released by the transmitter or the receiver.

Segments exchanged between devices shall be part of a connection except in the following cases:

- MAP or RMAP messages.
- APDUs and LCDUs conveyed in PHY-frames with the CNN_MNGMT field equal to 1111, for example, messages exchanged as part of the network admission protocol described in clause 8.6.1.1.1. In this case, the LLC frames contained in the MPDU shall be complete and the CONNECTION_ID shall be set to 255 (see clause 7.1.2.3.2.2.12 of [ITU-T G.9960]). Segments contained in this MPDU shall not use the acknowledgement protocol defined in clause 8.9.

NOTE – More cases may be added in future versions of this Recommendation.

A connection shall be established prior to exchange of any segment associated with that connection.

A data connection associated with a priority queue is uniquely identified by the tuple:

(SID > 0, DID > 0, PRI-Q, MQF = 0) and is known as a prioritized data connection, where PRI-Q is specified in Table III.1 of [ITU-T G.9960] as a function of the user priority mapped to the connection and the number of priority queues (traffic classes) supported from the source node to the destination node (i.e., for user priority 6 and 3 priority queues, PRI-Q is equal to 2).

A data connection associated with a service flow is uniquely identified by the tuple:

(SID > 0, FLOW_ID, MQF = 0).

A management connection is uniquely identified by the tuple (SID > 0, DID > 0, MQF = 1).

Each connection shall use an independent SSN sequence numbering. At any given time, there may be multiple open "connections" between a transmitter and a receiver in a network:

- zero or one management connection (for exchange of LCDUs);
- between zero and eight prioritized data connections (for exchange of APDUs that have not been mapped to flows. LCDUs may be mixed with APDUs in a prioritized data connection; see clause 8.1.3.2);
- between zero and 243~~7~~ unicast data connections, identified by CONNECTION_ID (see clauses 7.1.2.3.2.2.12 and 7.1.2.3.2.3.9.1.4 of [ITU-T G.9960]) value in the range 8 to 250, associated ~~to~~ with service flows;~~.~~
- between zero and 254 multicast connections, identified by MI=1, DID=MULTICAST_ID and CONNECTION_ID=252;~~.~~
- two broadcast connections one for data and one for management.

Table 8-105.2 – Values of connection identifiers for different types of connections

Connection identifier	Type of connection
0 to 7	Prioritized data connection.
8 to 250	Service flow.
251	Management connection (unicast or broadcast).
252	Multicast connection.
253, 254	Reserved by ITU-T.
255	Broadcast data connection (Note).
NOTE – In the case where the data does not belong to any connection (i.e. the CNN_MNGMT field is set to 1111), CONNECTION_ID shall be set to 255.	

Only one connection (either the management connection or a prioritized data connection) for delivering LCDUs may be established between a source node and destination node. A multicast connection shall not be used for delivering LCDUs.

Connections are unidirectional.

Connections may be established either with or without acknowledgements. A connection with acknowledgements is a connection that uses the acknowledgement protocol described in clause 8.9. Establishment of two connections identified by the same tuple, one with ACKs and the other without ACKs, is not allowed.

A given PHY frame may carry segments from the management connection and from not more than one data connection. Data and management segments can be differentiated by the MQF field in the LPDU header.

The CONNECTION_ID field in the PHY-frame header identifies the connection. The CONNECTION_ID field shall be set to the FLOW_ID for connections associated to service flows, or it shall be set to the value of PRI-Q for prioritized data connections. The valid values of CONNECTION_ID field for different types of connections are as shown in Table 8-105.2.

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

PHY frames carrying connection management information in which no payload is allowed (see Table 7-17 of [ITU-T G.9960]) shall have the MPDU priority equal to 7.

…

8.12.1.1 Establishment of the management connection

…

If the receiver has resources for the new connection, it shall respond with a PHY frame with FT=ACK, RXRST_MNGMT=1, FLCTRLT=<Status report>, FLCTRL_CONN=1 and FLCTRL equal to the number of LPDUs that the receiver can buffer for this connection. The transmitter shall set ACK_TX_CONF_WINDOW_SIZE (see clause 8.9.4.2) to the value received in the RX_CONN_WIN_SIZE ~~FLCTRL~~ field. The number of LPDUs that the receiver can buffer for this connection (indicated by the FLCTRL field during the lifetime of the connection) shall not exceed the maximum acknowledge window size that the receiver can support for the connection (indicated by RX_CONN_WIN_SIZE during connection setup).

8.12.1.2 Establishment of a data connection

…

If the receiver has resources for the new connection, it shall respond with a PHY frame with FT=ACK, RXRST_DATA=1, FLCTRLT=<Status report>, FLCTRL_CONN=0~~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 RX_CONN_WIN_SIZE ~~FLCTRL~~ field and its own available window size (see clause 7.1.2.3.2.3.8 of [ITU-T G.9960]). The number of LPDUs that the receiver can buffer for this connection, indicated by the FLCTRL field during the lifetime of the connection, shall not exceed the maximum acknowledge window size that the receiver can support for the connection indicated by RX_CONN_WIN_SIZE during connection setup.

…

8.12.8 Broadcast connections

…

The following clauses describe the establishment and release of both data ~~and~~ management and broadcast management connections.

8.12.8.1 Broadcast ~~M~~management ~~broadcast~~ connection

To establish a broadcast 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 broadcast management ~~broadcast~~ connection. It is implementation dependent how this is achieved. To release a broadcast 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 broadcast 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 b~~Broadcast data connection

To establish a broadcast 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 broadcast 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 broadcast 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.

17) Clause 8.16

Revise the text in clause 8.16 and its sub-clauses as follows:

8.16 PHY ~~M~~multicast binding protocol

The PHY multicast binding protocol ~~shall be used to~~enables a transmitter node to transmit the same PHY frames to several nodes that might share a common BAT. The multicast binding protocol enables the creation~~e~~ of a PHY multicast group~~s~~ and the management of the membership of nodes in the PHY multicast group. The PHY multicast transmissions are identified by (SID, MULTICAST_ID) tuples (see clause 8.7.1.2) which identify the transmitter node and the receiver nodes that receive directly the PHY frame from the transmitter node.~~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 PHY multicast group~~ for a new multicast stream~~

A transmitting node of a PHY multicast stream may initiate the PHY multicast binding protocol when it needs to transmit the same data to several nodes directly (in the same hop).~~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~~When a node initiates the multicast binding protocol, it shall compute the common BATs to be used for the PHY multicast ~~stream~~group based on the BATs (see clause 8.11) reported by the receiver nodes in the group.~~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 PHY multicast group, the transmitter shall assign~~s~~ receivers to the Mc-ACK/NACK slots. The acknowledgement protocol state machine for PHY multicast transmission shall be initialized as specified in clause 8.9.5.4.

NOTE 1 – The actual method for deciding on the number of PHY multicast groups and the BATs used for each group and the assignment of nodes to the Mc-ACK slots is beyond the scope of this Recommendation.

The transmitter shall then send MC_GrpInfoUpdate.ind message to all the nodes that will be part of the PHY multicast group including information about the BATs to be used within the PHY multicast group and the receiver nodes that are members of the group~~for each created multicast group~~. Upon reception of a MC_GrpInfoUpdate.ind message, each receiver that appears as a receiver of a PHY multicast group shall confirm the message by sending 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 PHY multicast group by setting the appropriate value of the FlowControlInd field in the MC_GrpInfoUpdate.ind message. The decision as to whether flow control should be enabled or not is beyond the scope of this Recommendation.

NOTE 2 – Flow-control may be disabled if Mc-ACK slots have not been allocated to all members of the PHY multicast group.

When flow-control is not used on a PHY multicast group a transmitter shall advertise the recommended receive buffer size in the MC_GrpInfoUpdate.ind message. The initial recommended receive buffer size (ACK_RX_CONF_WINDOW_SIZE) for a PHY multicast group shall be specified by the transmitter to have a maximum value (set in the MinRxBufSize field in

Table 8-107). Upon reception of the MC_GrpInfoUpdate.ind message receivers shall respond by specifying their available receive buffer sizes (ACK_RX_CONF_WINDOW_SIZE) in the MC_GrpInfoUpdate.cnf message. The transmitter shall collect all the receive buffer sizes advertised by all PHY multicast group members and shall adjust the recommended receive buffer size advertised in the MC_GrpInfoUpdate.ind message. The adjusted value of the MinRxBufSize field (see MinRxBufSize in Table 8-107) in the MC_GrpInfoUpdate.ind message shall be set to the minimum of the receive buffer size of all members of the PHY multicast group. Upon reception of the adjusted MC_GrpInfoUpdate.ind message receivers may reduce the size of their receive buffers to the specified value. The new receive buffer size used by the receiver shall be reported in the corresponding MC_GrpInfoUpdate.cnf message.

NOTE 3 – Based on the advertised receive buffer sizes of members of the PHY multicast group the transmitter may decide to reassign PHY multicast group members to different groups.

When flow-control is not used the value of FLCTRL specified in the ACK~~, 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 receiving nodes that are assigned a Mc-ACK slot. The initial recommended receive buffer size (ACK_RX_CONF_WINDOW_SIZE) for a PHY multicast group shall be specified by the transmitter to have a maximum value (set in the MinRxBufSize field in Table 8-107). The receiving nodes that are not assigned a Mc-ACK slot shall respond by specifying their available buffer sizes (ACK_RX_CONF_WINDOW_SIZE) in the MC_GrpInfoUpdate.cnf message. The transmitter shall collect the receive buffer sizes advertised by all receiving nodes that are not assigned a Mc-ACK slot and shall adjust the recommended receive buffer size advertised in the MC_GrpInfoUpdate.ind message. The adjusted value of the MinRxBufSize field (see MinRxBufSize in Table 8-107) in the MC_GrpInfoUpdate.ind message shall be set to the minimum of the receive buffer size of those receiving nodes of the PHY multicast group. Upon reception of the adjusted MC_GrpInfoUpdate.ind message these receivers may reduce the size of their receive buffers to the specified value. The new receive buffer size used by these receivers shall be reported in the corresponding MC_GrpInfoUpdate.cnf message. The transmitter shall limit the number of LPDUs transmitted within each PHY frame according to the transmit window corresponding to this group, to the minimum of the receive buffer size indicated in the MC_GrpInfoUpdate.cnf message by the receivers that are not assigned an Mc-ACK slot and the values indicated in the FLCTRL field by the receivers that are assigned Mc-ACK slots~~receiver~~.

Before the multicast binding is completed for a new PHY multicast ~~stream~~group the transmitter may send the multicast stream traffic using the BROADCAST_ID as DID, or by making unicast transmissions to the multicast receivers.

During initialization of a PHY multicast group or when a change in the membership of nodes of an existing PHY multicast group occurs the transmitter may use broadcast DID when sending the protocol messages. The reserved MAC address 01-19-A7-52-76-96 shall be used as the DA in the LCDU delivering the MC_GrpInfoUpdate.ind message. The DestinationNode of the LLC frame corresponding to the LCDU delivering the MC_GrpInfoUpdate.ind message shall be set to zero.

8.16.2 Maintenance of multicast binding information

The transmitter shall send MC_GrpInfoUpdate.ind message as specified in this clause to update receivers of a PHY multicast group when there is a change in BATs, or in the membership of receiver nodes, or in Mc-ACK slot assignment~~ 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 PHY multicast group associated with a multicast stream shall continue acknowledging in its assigned slot until its assignment for this Mc-ACK slot is terminated by an MC_GrpInfoUpdate.ind message. ~~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 of a PHY multicast group may remove any receiver that is a member of the PHY multicast group at any time by sending an MC_GrpRemove.req message to that receiver. The receiver shall send the MC_GrpRemove.cnf message to the transmitter, confirming that it is no longer a member of the multicast group. If the receiver is assigned an Mc-ACK slot of the PHY multicast group, it shall continue acknowledging in its assigned slot until its assignment for this Mc-ACK slot is terminated by an MC_GrpInfoUpdate.ind message. From the time of receiving MC_GrpRemove.req until the assignment of its Mc-ACK slot is terminated by an MC_GrpInfoUpdate.ind, the receiver shall set the FACK field to 111_2 (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]).

The transmitter may split an existing PHY multicast ~~stream~~ group into several PHY multicast groups, for example, when new receivers with very different BAT join. The transmitter shall assign a new multicast DID to each of the newly created PHY multicast groups and shall send MC_GrpInfoUpdate.ind, which includes the information describing the new PHY multicast groups, to all nodes associated with that PHY multicast ~~stream~~group, 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 PHY multicast groups or for updating existing PHY multicast group information.

The transmitter shall allocate a new BAT ID for a PHY multicast group when a change is required in any of the active BAT IDs of the PHY multicast group.

The MC_GrpInfoUpdate.ind message sent by the transmitter shall include the list of all BAT_IDs that are to be active in the PHY multicast group ~~or groups~~ (inside the McstGroupInfo field, see Table 8-107). This consists of those BAT_IDs that are to be retained and new BAT_IDs to be added. BAT_IDs to be removed shall be excluded from the list. New BAT_IDs in this list are accompanied ~~with~~ by BATInfo fields (see Table 8-109). The transmitter shall not start using the new BATs until all the PHY multicast group receivers have confirmed the change. Once ~~a new BAT is used for transmission by~~ the transmitter actually uses a new BAT in transmission, the receivers of the PHY multicast group shall invalidate the old BATs assigned to that PHY multicast group that were excluded from the list in the McstGroupInfo field of the last received MC_GrpInfoUpdate.ind message.

In case the transmitter detects a change in the PHY multicast ~~binding~~group 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

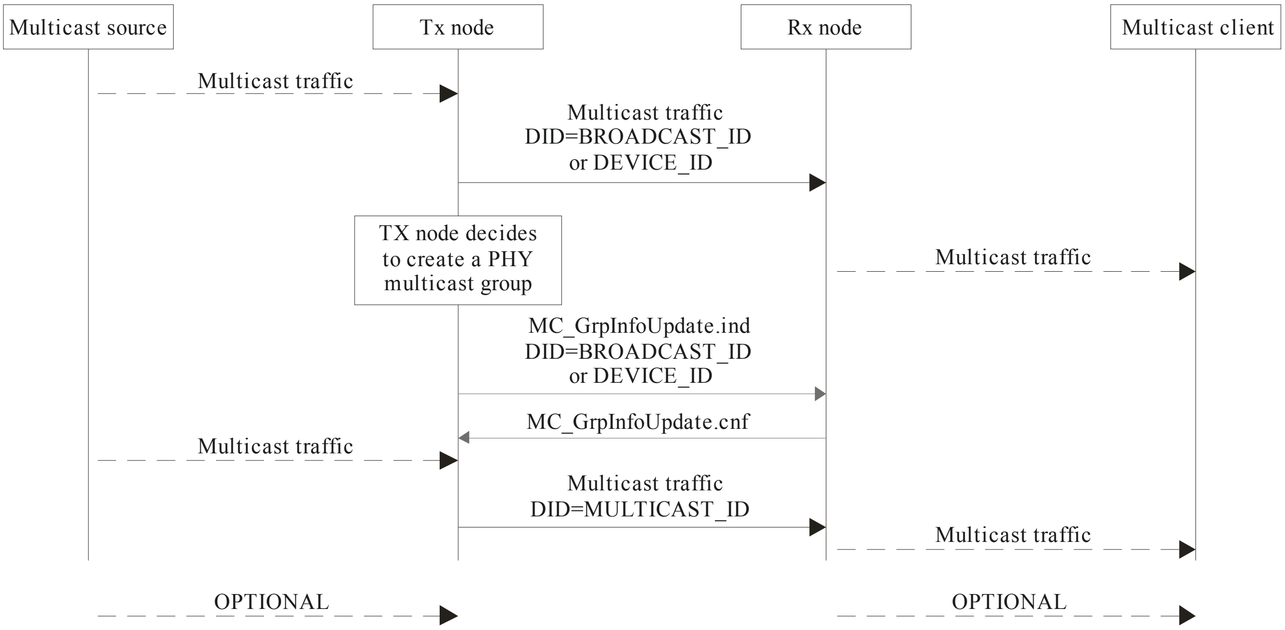
When a transmitter wishes to terminate a PHY multicast group it ~~none of the receivers that are part of a multicast group are interested in the multicast stream (e.g. no IGMP membership report messages are received from any of the receivers of the multicast group) the transmitter~~ shall send MC_GrpInfoUpdate.ind to the PHY multicast group members~~those receivers~~ to release ~~this~~the PHY multicast DID. The receivers shall respond with an MC_GrpInfoUpdate.cnf message. Upon receiving the MC_GrpInfoUpdate.cnf messages from all the PHY multicast group members the transmitter shall terminate the PHY multicast group and release the multicast DID.

When all members have left a PHY multicast group a transmitter shall terminate the PHY multicast group and release the multicast DID.

8.16.4 PHY ~~M~~multicast binding protocol flow

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

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



G.9961(10)-Cor.1(11)_F8-58

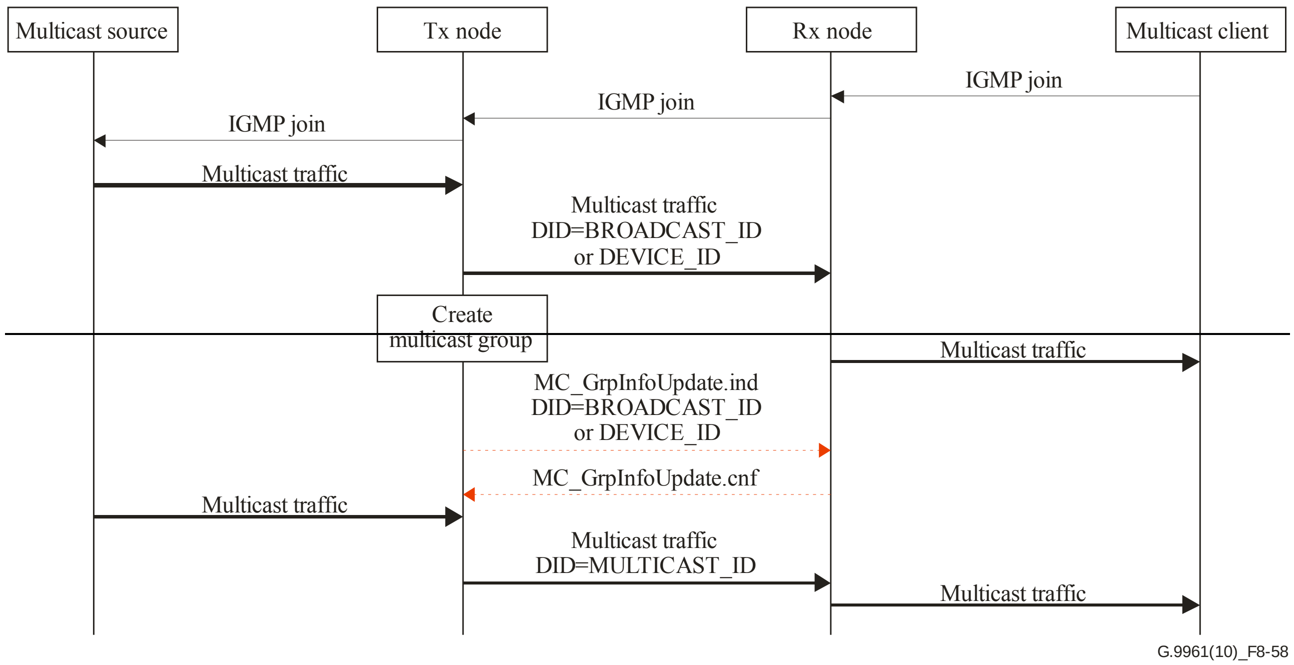


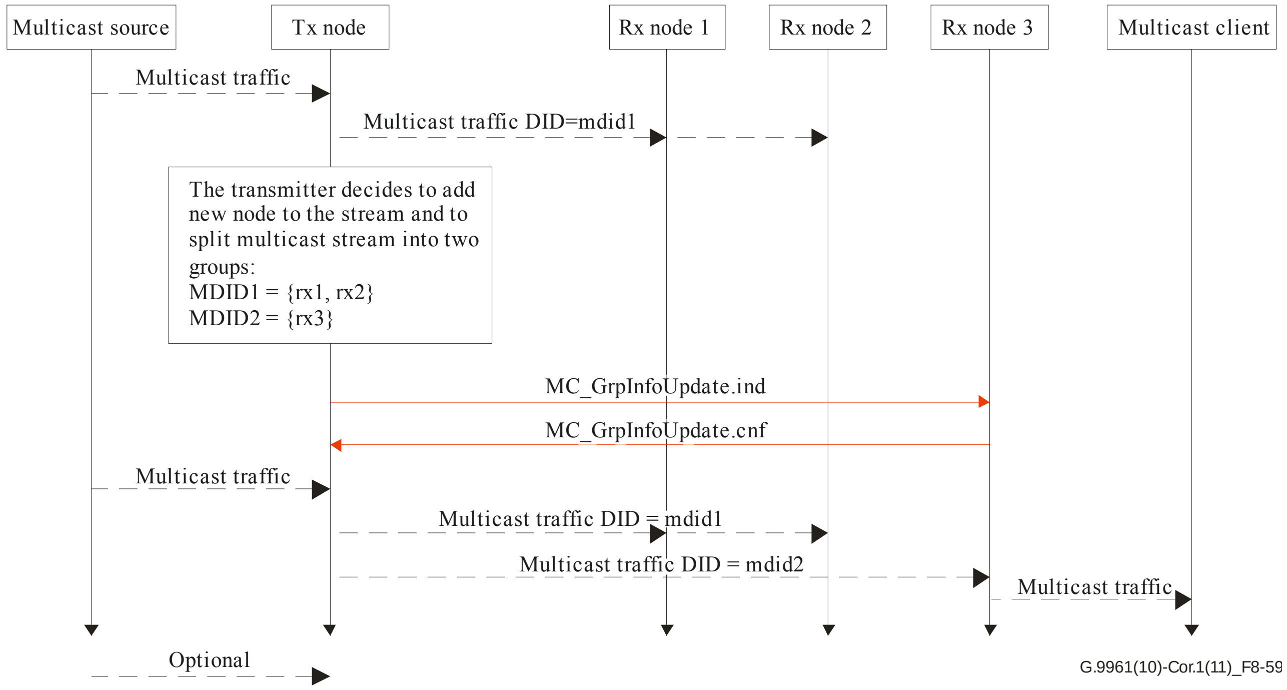
Figure 8-58 – Example of initializing a PHY multicast group

In this example the MC_GrpInfoUpdate.ind is sent after the transmitter decides to use the PHY multicast group mechanism to deliver a multicast stream to a set of receiver nodes.~~only when the multicast traffic starts since until that point 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 PHY multicast groups

Figure 8-59 shows an example of an existing multicast stream that is transmitted using a single PHY multicast group to two receivers. When the transmitter needs to add node number three ~~requests~~ to join the multicast stream, the transmitter decides that it is better to allocate a new PHY multicast group for this node. Hence, it ~~has~~ creates~~d~~ another PHY multicast DID and ~~has~~ informs~~ed~~ node number three on the new group via the MC_GrpInfoUpdate.ind message. When ~~the message was confirmed by~~ node number three confirms the received message, the transmitter ~~can~~ starts using the new PHY multicast group in addition to the existing multicast group.


Multicast source
Tx node
Rx node 1
Rx node 2
Rx node 3
Multicast client
Multicast traffic
Multicast traffic DID=mdid1
The transmitter decides to add new node to the stream and to split multicast stream into two groups:
MDID1 = {rx1, rx2}
MDID2 = {rx3}
MC_GrpInfoUpdate.ind
MC_GrpInfoUpdate.cnf
Multicast traffic
Multicast traffic DID = mdid1
Multicast traffic DID = mdid2
Multicast traffic
Optional
G.9961(10)-Cor.1(11)_F8-59

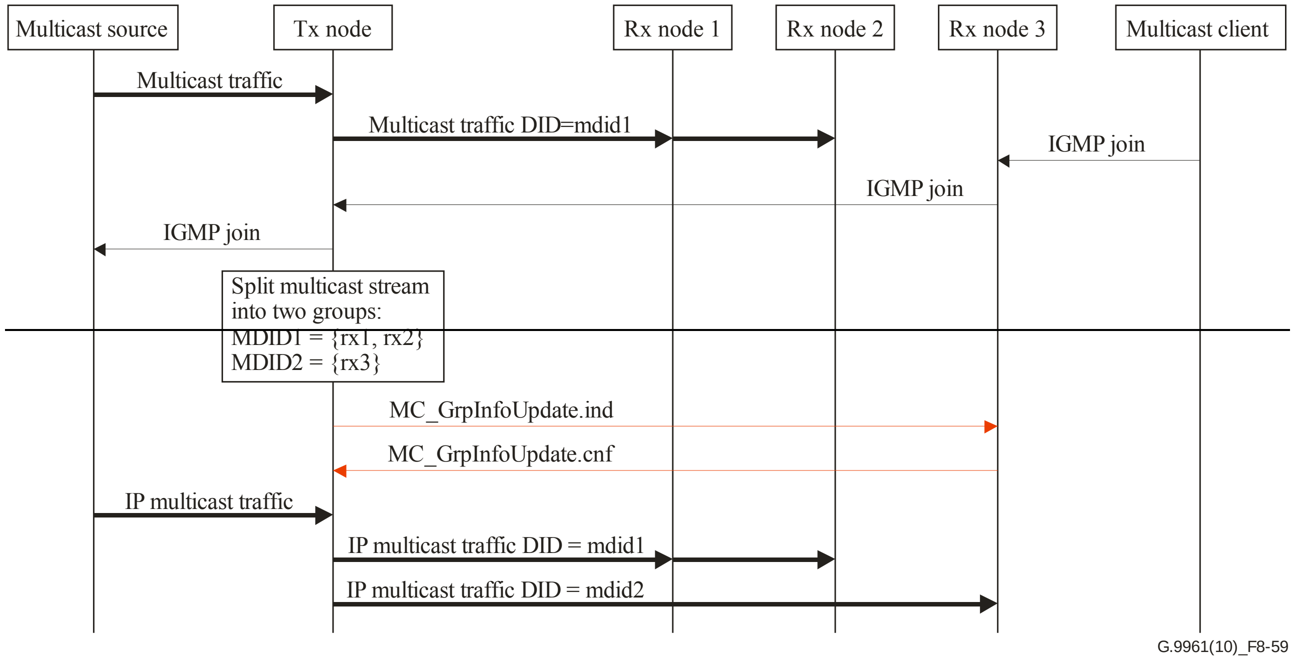


Figure 8-59 – Example of split of a PHY multicast group

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

Figure 8-60 shows an example of the establishment of a PHY multicast group with flow-control disabled. In this example the transmitter initiates the sequence by sending a MC_GroupInfoUpdate.ind message to the three PHY multicast group members specifying the recommended minimum receive buffer size. The PHY multicast group members allocate receive buffers for the multicast flow and respond with a MC_GroupInfoUpdate.cnf specifying the actual size of the allocated receive buffer. The transmitter collects the results from the MC_GroupInfoUpdate.cnf messages, calculates the new minimum receive buffer size and advertises it to the group using another MC_GroupInfoUpdate.ind message. Finally, the receivers adjust their receive buffer allocations according to the specified minimum receiver buffer size and reply to the transmitter using the updated MC_GroupInfoUpdate.cnf messages.

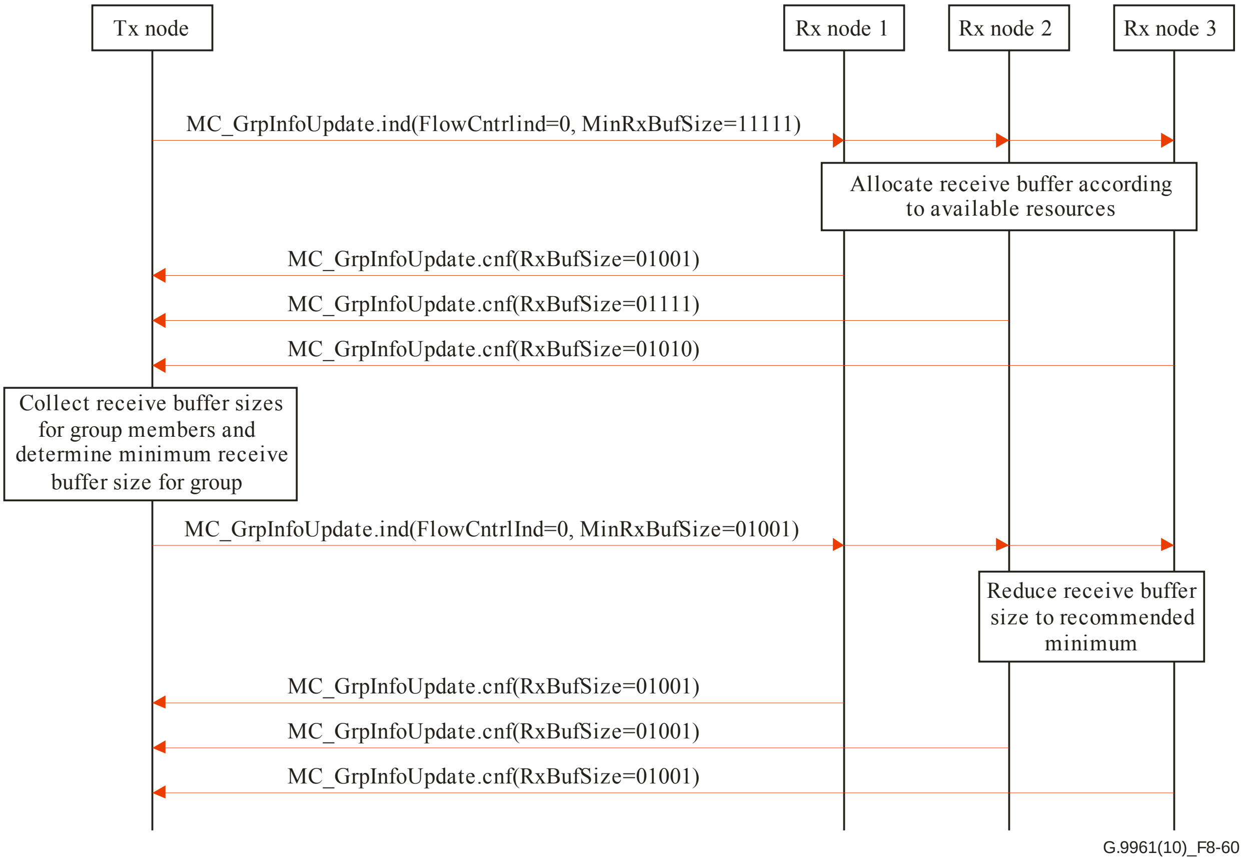
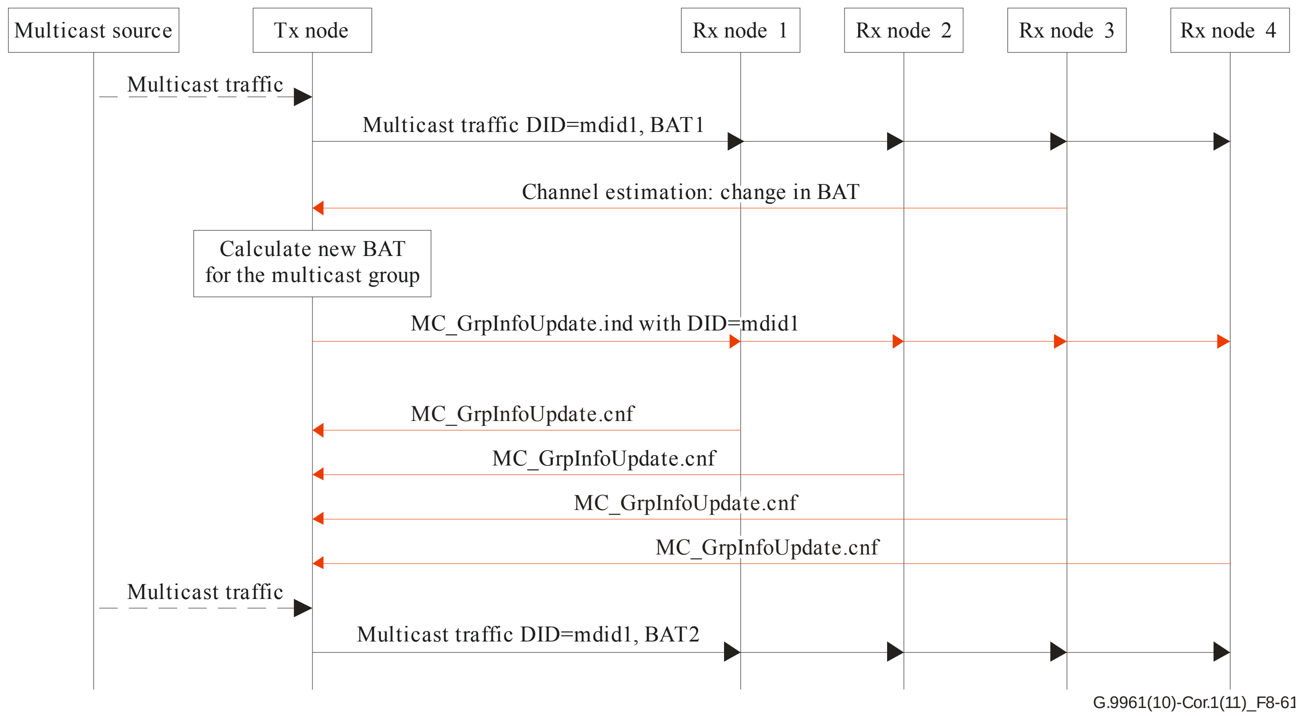

Tx node
Rx node 1
Rx node 2
Rx node 3
MC_GrpInfoUpdate.ind(FlowCntrlind=0, MinRxBufSize=11111)
Allocate receive buffer according to available resources
MC_GrpInfoUpdate.cnf(RxBufSize=01001)
MC_GrpInfoUpdate.cnf(RxBufSize=01111)
MC_GrpInfoUpdate.cnf(RxBufSize=01010)
Collect receive buffer sizes for group members and determine minimum receive buffer size for group
MC_GrpInfoUpdate.ind(FlowCntrlInd=0, MinRxBufSize=01001)
Reduce receive buffer size to recommended minimum
MC_GrpInfoUpdate.cnf(RxBufSize=01001)
MC_GrpInfoUpdate.cnf(RxBufSize=01001)
MC_GrpInfoUpdate.cnf(RxBufSize=01001)
G.9961(10)_F8-60

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

8.16.4.4 Message sequence – Maintenance of multicast binding information

Figure 8-61 shows an example of an existing multicast stream that is transmitted using a single PHY multicast group to its four receivers. When receiver number three reports on change in its recommended BAT, the transmitter ~~has~~ decides~~d~~ to update the BAT of the PHY multicast group. Hence, it ~~has~~ informs~~ed~~ the receivers on the change via the MC_GrpInfoUpdate.ind message. ~~When~~After the change ~~wa~~is confirmed by all the receivers the transmitter can start using the new BAT.



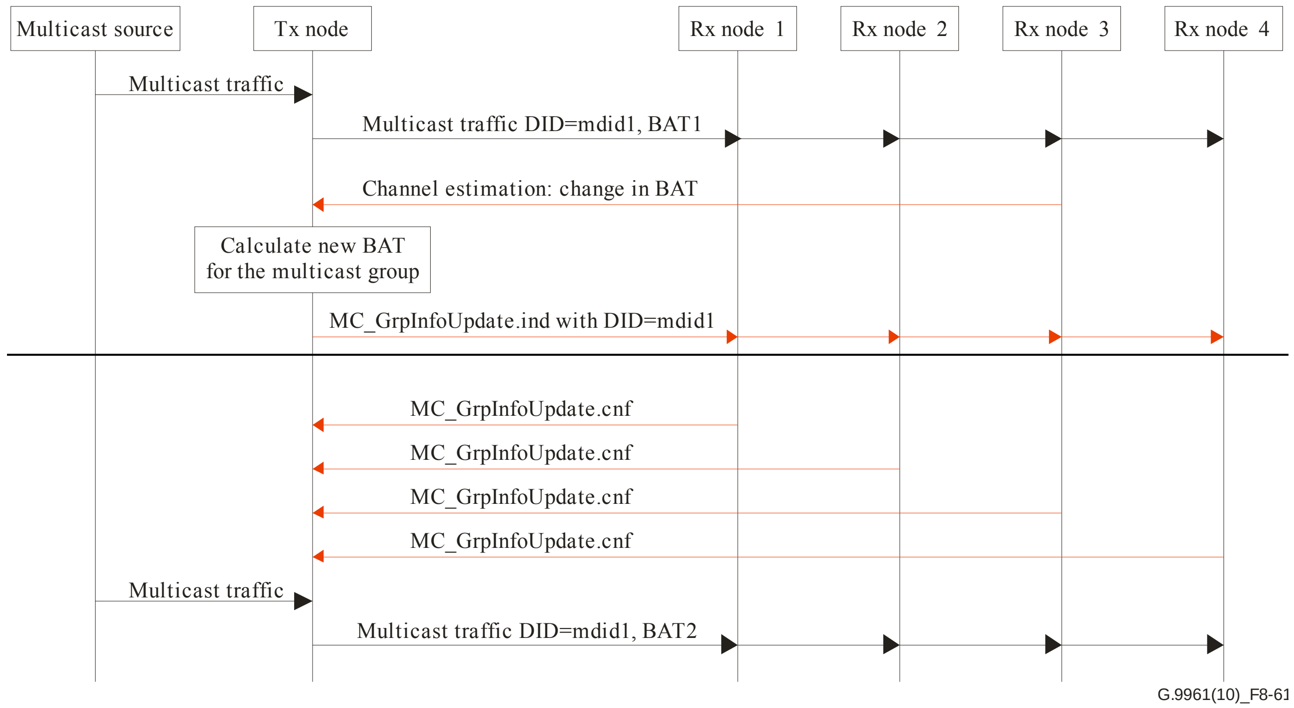


Figure 8-61 – Example of maintenance of PHY multicast binding information

NOTE – The transmitter sends MC_GrpInfoUpdate.ind to the multicast DID. Also, the transmitter may continue transmitting the multicast stream using the original BAT until the multicast binding is completed.

8.16.5 PHY ~~M~~multicast binding protocol messages

8.16.5.1 PHY ~~M~~multicast group information update indication

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

Each PHY multicast group contains the list of all receivers of the group each uniquely identified by its device ID. Each receiver information contains the McAckSlot assignment if Mc-ACK is enabled for the PHY multicast group.

The format of the MMPL of the MC_GrpInfoUpdate.ind message shall be as shown in Table 8-106.

Table 8-106 – Format of the MMPL of the MC_GrpInfoUpdate.ind message

Field	Octet	Bits	Description
Source ID	0	[7:0]	The Device ID of the transmitter.
~~NumMcstGroups~~	~~1~~	~~[7:0]~~	~~Number of multicast groups described in this message.~~
McstGroupInfo~~[0]~~	~~2~~1	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 PHY multicast group: 0 – The flow control mechanism shall not be used. 1 – The flow control mechanism shall be used
Reserved		[7:6]	Reserved by ITU-T (Note 1)
MinRxBufSize	2	[4:0]	Recommended minimum receiver buffer size expressed in LPDUs to be buffered by receivers in the PHY multicast group. (Note 2) The values of this field shall be the same as FLCTRL field for status report (Table 7-21 of [ITU-T G.9960])

Table 8-107 – Format of McstGroupInfo Field

Field	Octet	Bits	Description
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 PHY multicast group as allocated by the transmitter. The number of BAT_IDs shall not exceed n=32.
BAT_ID	4	[7:0]	The first of n BAT IDs used for this PHY multicast group.
⋮	⋮	⋮	⋮
BAT_ID	n+3	[7:0]	The last of n BAT IDs used for this PHY multicast group.
NumRxNode	n+4	[7:0]	Number of ~~R~~receive n~~N~~odes m that are members of the PHY multicast group. Zero indicates that this multicast DID is released.
RxNodeInfo	n+5 and n+6	[15:0]	Info for the first of m receive nodes of the PHY multicast group.
⋮	⋮	⋮	⋮
RxNodeInfo	n+m+4 and n+m+5	[15:0]	Info for the last of m receive nodes of the PHY multicast group.

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

NOTE 2 – The receiver buffer size (MinRxBufSize) specified by a transmitter for a PHY multicast group may be reduced over time by sending another MC_GrpInfoUpdate.ind message with a new MinRxBufSize value. When the flow control mechanism is not used, a receiver may reduce the size of the receiver buffer to the new value specified by the transmitter.

When the flow control mechanism is used, a receiver shall ignore the value of this field.

Table 8-108 – Format of RxNodeInfo field

Field	Octet	Bits	Description
RxDeviceID	0	[7:0]	Device ID of a receive node of the PHY multicast group
McAckSlot	1	[2:0]	Mc-Ack Slot assigned to this node 0 – Use NACK Slot if NACK is enabled according to RPRQ of the PHY multicast group 1-7 – Mc-ACK Slot ID if Mc-ACK is enabled according to the RPRQ of the PHY multicast group
Reserved		[7:3]	Reserved by ITU-T (Note)
NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

…

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 PHY 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.			

8.16.5.3 Format of MC_GrpRemove.req

Message MC_GrpRemove.req is a management message that shall be sent by the transmitter of the PHY multicast group to a receiver of the PHY multicast group to request the receiver to leave the PHY multicast group.

The format of the MMPL of the MC_GrpRemove.req message shall be as shown in Table 8-111.

Table 8-111 – Format of the MMPL of the MC_GrpRemove.req message

Field	Octet	Bits	Description
RxDeviceID	0	[7:0]	DEVICE_ID of the receiver node that is requested to leave the PHY multicast group.
TxDeviceID	1	[7:0]	DEVICE_ID of the transmitter of the PHY multicast group.
MulticastID	2	[7:0]	MULTICAST_ID of the group.

8.16.5.4 Format of MC_GrpRemove.cnf

Message MC_GrpRemove.cnf is a management message that shall be sent by the receiver of a PHY multicast group to the transmitter of the PHY multicast group to confirm the reception of the MC_GrpRemove.req message.

The format of the MMPL of the MC_GrpRemove.cnf message shall be as shown in Table 8-112.

Table 8-112 – Format of the MMPL of the MC_GrpRemove.cnf message

Field	Octet	Bits	Description
RxDeviceID	0	[7:0]	DEVICE_ID of the receiver node that confirmed the request to leave the PHY multicast group.
TxDeviceID	1	[7:0]	DEVICE_ID of the transmitter of the PHY multicast group.
MulticastID	2	[7:0]	MULTICAST_ID for the group.

18) Clause 8.17

Add the new clause 8.17 as follows:

8.17 DLL multicast stream

A source node that decides to establish a DLL multicast stream shall establish a multicast path towards each client of the DLL multicast stream. The paths towards the client nodes may include relay nodes that are bound to the path and the DLL multicast stream identification (MSID). The source node that establishes a DLL multicast group shall generate the DLL multicast stream identifier (MSID) that together with the DEVICE_ID of the source of the DLL multicast stream uniquely identifies the DLL multicast stream. The members of a DLL multicast group are identified by the source node of the DLL multicast stream. The source of a DLL multicast stream, shall transmit the traffic of the DLL multicast stream to the members of the DLL multicast group according to established paths as described in the following sections.

8.17.1 DLL multicast stream establishment

An ITU-T G.9960 node that determines that it has to transmit a multicast stream to client nodes in the domain, shall establish a path to each one of the client nodes. The source node that generates the DLL multicast stream shall first allocate an MSID that together with the DEVICE_ID of the source node, shall uniquely identify the DLL multicast stream. The source node shall also initialize the Transaction ID for that DLL multicast stream to zero. The source node shall increment the Transaction ID for each new DLL multicast path it establishes for that DLL multicast stream.

The source node shall establish the path towards a client node as follows:

If the source node has a direct link to the client node according to the current unicast routing table, it shall send a DMC_Path.req message to the client node to bind it with the specified MSID multicast stream and the multicast stream MAC Address. The client node shall reply with a DMC_Path.cnf message that contains the same Transaction_ID that was specified in the DMC_Path.req message and shall bind itself to the established path identified by the MSID, the DEVICE_ID of the source node and the multicast MAC address (DA). The source node upon receiving the DMC_Path.cnf message shall bind the path and complete the path establishment procedure. If the source node does not receive a DMC_Path.cnf message after a vendor discretionary period, which is larger than MAX_WAIT_TIME, it may repeat the request through a new DMC_Path.req with a different Transaction_ID.

NOTE – DMC_Path.req should be sent using connections with acknowledgements in order to avoid long setup times for DLL multicast trees because of lost messages.

If the source node does not have a direct link to the client node, it shall determine the first relay node towards the client node according to the current unicast routing table and send a DMC_Path.req message to that node.

The DMC_Path.req message shall contain the following fields: the DEVICE_ID of the source node of the DLL multicast, the allocated MSID, the DEVICE_ID of the client node, the MAC address of the multicast stream and the Transaction_ID. The source node of the multicast stream shall address the DMC_Path.req message to the first relay node by setting the DA to the MAC address of that node.

A relay node that receives a DMC_Path.req message and has a direct link with the client node shall bind the DEVICE_ID of the source of the DLL multicast stream, the MSID, and the sender node's DEVICE_ID with the DEVICE_ID of the client endpoint node, and shall replace the DA of the LCDU of the DMC_Path.req message with the client node's MAC address and transmit the DMC_Path.req message to the client node.

A relay node that receives a DMC_Path.req message and does not have a direct link to the client node shall bind the DEVICE_ID of the source of the DLL multicast stream, the MSID, and the sender node's DEVICE_ID with the DEVICE_ID of the next relay node towards the client node according to the unicast routing table. It shall then replace the DMC_Path.req LCDU's DA by the MAC address of the next relay node and send the updated DMC_Path.req message to that node.

Upon reception of the DMC_Path.req message, the client node shall reply to the node that sent this message with a DMC_Path.cnf message and shall bind itself to the specified DLL multicast stream identified by the DEVICE_ID of the source DLL multicast stream, the MSID, and the sender node's DEVICE_ID.

A relay node that receives the DMC_Path.cnf message shall mark the binding of the DLL multicast stream path identified by the DEVICE ID of the source DLL multicast stream node and the MSID as valid. The relay node shall then append its DEVICE_ID to the Path_List field in the MMPL of the received DMC_Path.cnf message. The relay node shall transmit the updated DMC_Path.cnf message to the node from which it has received the DMC_Path.req message, which can be either a relay node or the node originating the DLL multicast stream.

Once the originating node receives the DMC_Path.cnf message, it has the complete path of this bound client from the received DMC_Path.cnf message. This completes the path establishment procedure. The source node may then start sending the multicast stream packets towards the client node(s) either directly or via the first relay node according to the established path.

Each relay node shall identify LLC frames corresponding to a DLL multicast stream according to the OriginatingNode and the MSID specified in the LFH. The relay node shall then relay any received LLC frames of that DLL multicast stream to all the nodes it has bound to this DLL multicast stream according to the binding information that it has configured during the DLL multicast stream path establishment. The relay node shall only relay LLC frames corresponding to a DLL multicast stream path for which its binding is marked as valid.

When the multicast source node or any other relay node in the DLL multicast paths receives an updated routing table, it shall not update the current multicast paths. A relay node shall correct an established multicast path only by explicit order received from the multicast source node as defined in clause 8.17.3.

8.17.2 Preventing loops and packet duplications

The paths of a specific DLL multicast stream shall be established in a tree topology that ensures that a node shall not receive duplicate multicast packets from different paths and prevent the source node or any relay node from unnecessarily duplicating transmissions. The topology of the DLL multicast stream tree is built under a principal rule that each node shall receive packets of a specific (OriginatingNode, MSID) only from one node. The DLL multicast stream paths tree shall be built according to this rule by executing the following procedure in path establishment: When a source node binds a new client node to an existing DLL multicast stream, it shall send towards it the DMC_Path.req message as defined in the previous clause. Any relay node on the path towards the newly joined client node shall verify that it always receives the DMC_Path.req for this specific (OriginatingNode, MSID) from the same sender node. In case it receives a DMC_Path.req message from a node different from the sender node to which it is currently bound, it shall reply with the DMC_PathReject.cnf message towards the source node. The DMC_PathReject.cnf message shall contain the rejecting node's DEVICE_ID, the DEVICE_ID of the node that sent it DMC_Path.req message and the rejection reason (duplication source).

When the source multicast node receives the DMC_PathReject.cnf message, it may decide to release the entire tree or the branch and rebuild it again, or to enforce establishment of the path until the rejecting relay node is based on the existing path. If the source node decides to enforce the existing path, it shall send the DMC_EnforcePath.req towards the relay node that encountered the problem via the original path. The source node shall address the DMC_EnforcePath.req message to the first relay node in the path toward the rejecting node. The DMC_EnforcePath.req message shall contain the full path until the rejecting node and the client node. Each relay node that receives the DMC_EnforcePath.req message shall forward the message to the next relay node according to the specified path toward the rejecting node. When the rejecting node receives the DMC_EnforcePath.req message it shall create a DMC_Path.req message, filling it with the information received in the DMC_EnforcePath.req message, and forward the message to the next relay node according to the current routing table. From this phase, the path establishment procedure towards the client node shall continue as specified in the previous section. The client node shall reply with the DMC_Path.cnf message to the relay node that sent it the DMC_Path.req message. All the relay nodes on the path towards the source node upon receiving the DMC_Path.cnf message, shall execute the bind, update the Path_List field in the DMC_Path.cnf message with their own DEVICE_ID and forward the DMC_Path.cnf message towards the source node.

A specific relay that has to forward a specific MSID stream traffic to several nodes that are bound with this MSID may establish a PHY multicast group. In this case, the node may create or update PHY multicast groups when it receives a DMC_Path.req message to transmit the data to the next relay nodes or client nodes. In that case, the PHY multicast group shall only include bound client nodes and relay nodes that are in its bind list for this MSID in its current hop.

Node A decides to add node N to the '***MSID*** DLL multicast stream.
According to the current routing table, node A sends the DMC_Path.req message to node D.

According to the current routing table, node D sends the DMC_Path.req message to node R.

According to the current routing table Relay node R sends the DMC_Path.req message to node E.

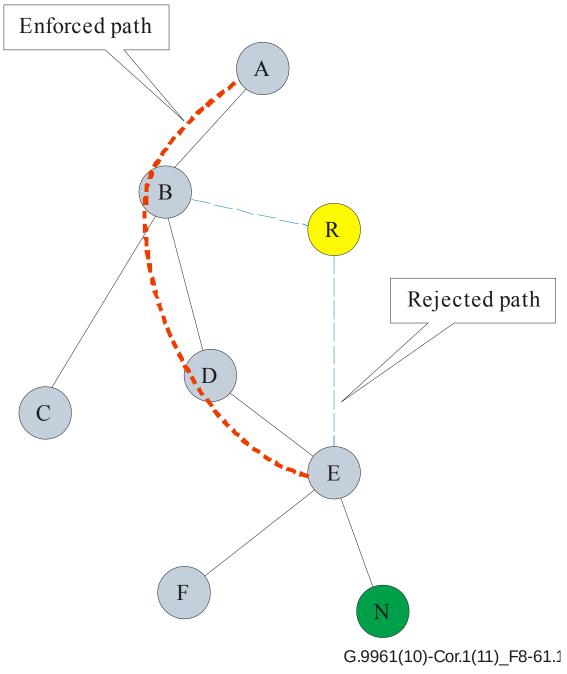
Node E knows that the legitimate source node for the specified '***MSID***' is node D, therefore it rejects the DMC_Path.req message by sending the DMC_PathReject.cnf message to node R.

Node R receives the DMC_PathReject.cnf message and sends a DMC_PathReject.cnf message node D.

Node D receives the DMC_PathReject.cnf message and sends a DMC_PathReject.cnf message node A.

Node A sends the DMC_EnforcePath.req message to node B to establish the path towards node N with a specified path until node E. Node B sends the DMC_EnforcePath.req message to node D (and not to node R) according to the specified path in the DMC_EnforcePath.req message.

Node D sends the DMC_EnforcePath.req message to node E and node E sends DMC_Path.req message to node N.



G.9961(10)-Cor.1(11)_F8-61.1

Figure 8-62 – Example of the mechanism for preventing loops in the DLL multicast tree

8.17.3 Releasing client node from MSID

When the source node of a DLL multicast stream decides to release a client node from its MSID, it shall increment the Transaction ID of that DLL multicast stream and shall send a DMC_ReleasePath.req message to the respective client node or to its first relay node in case the client node is accessed via relay node(s). Each node that receives the DMC_ReleasePath.req shall release the specified node from this bind list and forward the message towards the client node. Each node that received the message shall reply with the DMC_ReleasePath.cnf message to the node that sent it the DMC_ReleasePath.req message. Each relay node that does not have any nodes in its bind list shall release itself from the MSID multicast stream and indicate it in the replied DMC_ReleasePath.cnf message.

8.17.4 Recovery from a DLL multicast broken path

In the case where one of the relay nodes determines that the path of a specific MSID is broken it shall inform the domain master via a normal topology update message and the source node of the DLL multicast stream via a DMC_BrokenLink.ind message.

The multicast source node may correct the broken path according to a newly received updated routing table from the domain master. The source node may correct an existing path by sending DMC_ReleasePath.req to the relevant nodes and then it shall send new DMC_Path.req to the relevant nodes.

8.17.5 Aging DLL multicast path process

In order to prevent a situation where a multicast source node leaves the network and all the respective nodes in the multicast stream path are still holding MSID resources, an aging mechanism shall be used. The source node of each DLL multicast stream (MSID) shall periodically send a management message, DMC_PathAlive.ind, via the established MSID DLL multicast stream paths tree to the first node of each path (client node or relay node). Each node in the tree that receives this message shall reset its aging timer for that DLL multicast stream and shall transmit the DMC_PathAlive.ind message to each of the nodes that are bound to this DLL multicast stream, identified by (OriginatingNode, MSID), according to the binding information that it has configured during the DLL multicast stream paths establishment. Each node in the path that does not receive a DMC_PathAlive.ind message within a period of DMC_PATH_AGING_PERIOD (1 second) shall remove itself from this DLL multicast stream and release all of its MSID resources.

8.17.6 DLL Multicast protocol messages

8.17.6.1 DMC_Path.req message format

The format of the DMC_Path.req management message shall be as shown in Table 8-113.

Table 8-113 – Format of the MMPL of the DMC_Path.req message

Field	Octet	Bits	Description
Source ID	0	[7:0]	The DEVICE_ID of the source node of the DLL multicast stream.
MSID	1	[7:0]	The multicast identification allocated by the source of the DLL multicast stream.
ClientID	2	[7:0]	The DEVICE_ID of the client node of the DLL multicast stream source node.
MulticastAddress	3-8	[47:0]	MAC address of the multicast stream.
Transaction_ID	9	[7:0]	Identifies this path transaction.

8.17.6.2 DMC_Path.cnf message format

The format of the DMC_Path.cnf management message shall be as shown in Table 8-114.

Table 8-114 – Format of the MMPL of the DMC_Path.cnf message

Field	Octet	Bits	Description
Source ID	0	[7:0]	The DEVICE_ID of the source node of the DLL multicast stream.
MSID	1	[7:0]	The multicast identification allocated by the source of the DLL multicast stream.
ClientID	2	[7:0]	The DEVICE_ID of the client node of the DLL multicast stream source node.
Transaction_ID	3	[7:0]	Identifies this path establishment transaction. It shall contain the same value that was specified in the corresponding DMC_Path.req message.
NumOfNodes	4	[7:0]	Specifies the number of relay nodes (n) in the Path_List from the source node towards the client node.

Table 8-114 – Format of the MMPL of the DMC_Path.cnf message

Field	Octet	Bits	Description
Path_List[0]	5	[7:0]	This entry in the list contains the DEVICE_ID of the last relay node in the established path from the source node towards the client node.
Path_List[n-1]	4+n	[7:0]	This entry in the list contains the DEVICE_ID of the first -relay in the established path from the source node towards the client node.

8.17.6.3 DMC_PathReject.cnf message format

The format of the DMC_PathReject.cnf management message shall be as shown in Table 8-115.

Table 8-115 – Format of the MMPL of the DMC_PathReject.cnf message

Field	Octet	Bits	Description
Source ID	0	[7:0]	The DEVICE_ID of the source node of the DLL multicast stream.
MSID	1	[7:0]	The multicast identification allocated by the source of the DLL multicast stream.
ClientID	2	[7:0]	The DEVICE_ID of the client node of the DLL multicast stream source node
Transaction_ID	3	[7:0]	Identifies this path establishment transaction specified in the DMC_Path.req message.
RejectingNodeId	4	[7:0]	The DEVICE_ID of the relay node that rejects the DMC_Path.req message
Rejection_code	5	[7:0]	00_{16} – The request path is conflicted because there is already a path established for the specified multicast stream with a different source node. 01_{16} – The node is not able to support additional multicast streams. 02_{16} to FF_{16} – Reserved by ITU-T.

8.17.6.4 DMC_EnforcePath.req message format

The format of the DMC_EnforcePath.req management message shall be as shown in Table 8-116.

Table 8-116 – Format of the MMPL of the DMC_EnforcePath.req message

Field	Octet	Bits	Description
Source ID	0	[7:0]	The DEVICE_ID of the source node of the DLL multicast stream.
MSID	1	[7:0]	The multicast identification allocated by the source of the DLL multicast stream.
ClientID	2	[7:0]	The DEVICE_ID of the client node of the DLL multicast stream source node.
MulticastAddress	3-8	[47:0]	MAC address of the multicast stream

Table 8-116 – Format of the MMPL of the DMC_EnforcePath.req message

Field	Octet	Bits	Description
Transaction_ID	9	[7:0]	Identifies this path establishment transaction. It shall contain the same value as in the original DMC_Path.req for this path.
NumOfNodes	10	[7:0]	Specifies the number of relay nodes (n) in the Path_List from the source node towards the rejecting node.
Path_List[0]	11	[7:0]	This entry in the list contains the DEVICE_ID of the first relay node in the established path from the source node towards the rejecting node.
Path_List[n–1]	10+n	[7:0]	This entry in the list contains the DEVICE_ID of the last relay in the established path from the source node towards the rejecting node.

8.17.6.5 DMC_ReleasePath.req message format

The format of the DMC_ReleasePath.req management message shall be as shown in Table 8-117.

Table 8-117 – Format of the MMPL of the DMC_ReleasePath.req message

Field	Octet	Bits	Description
Source ID	0	[7:0]	The DEVICE_ID of the source node of the DLL multicast stream.
MSID	1	[7:0]	The multicast identification allocated by the source of the DLL multicast stream.
ClientID	2	[7:0]	The DEVICE_ID of the client node of the DLL multicast stream source node to be release from the path.
Transaction_ID	3	[7:0]	Identifies this path transaction.
NumOfNodes	4	[7:0]	Specifies the number of relay nodes (n) in the Path_List from the source node towards the client node.
Path_List[0]	5	[7:0]	This entry in the list contains the DEVICE_ID of the first relay node in the established path from the source node towards the client node.
Path_List[n–1]	4+n	[7:0]	This entry in the list contains the DEVICE_ID of the last relay node in the established path from the source node towards the client node.

8.17.6.6 DMC_ReleasePath.cnf message format

The format of the DMC_ReleasePath.cnf management message shall be as shown in Table 8-118.

Table 8-118 – Format of the MMPL of the DMC_ReleasePath.cnf message

Field	Octet	Bits	Description
Source ID	0	[7:0]	The DEVICE_ID of the source node of the DLL multicast stream.
MSID	1	[7:0]	The multicast identification allocated by the source of the DLL multicast stream.
ClientID	2	[7:0]	The DEVICE_ID of the client node of the DLL multicast stream source node to be release from the path.
Transaction_ID	3	[7:0]	Identifies this path establishment transaction. It shall contain the same value as in the corresponding DMC_ReleasePath.req for this path.
RelayNodeStatus	4	[7:0]	Specifies the status of the relay node that sends this message. 0: The relay node released itself from the specified DLL multicast stream. 1: The relay node still belongs to the specified DLL multicast stream. 2 to 255: Reserved by ITU-T.

8.17.6.7 DMC_PathAlive.ind message format

The format of the DMC_PathAlive.ind management message shall be as shown in Table 8-119.

Table 8-119 – Format of the MMPL of the DMC_PathAlive.ind message

Field	Octet	Bits	Description
Source ID	0	[7:0]	The DEVICE_ID of the source node of the DLL multicast stream.
MSID	1	[7:0]	The multicast identification allocated by the source of the DLL multicast stream.

8.17.6.8 DMC_BrokenLink.ind message format

This message is sent by a node that needs to report to the source of a DLL multicast stream that a link towards a multicast client node is broken.

The format of the DMC_BrokenLink.ind message shall be as shown in Table 8-120.

Table 8-120 – Format of the MMPL of the DMC_BrokenLink.ind message

Field	Octet	Bits	Description
Source ID	0	[7:0]	The DEVICE_ID of the source node of the DLL multicast stream.
Reporting_DeviceID	1	[7:0]	The DEVICE_ID of the node reporting the broken link.
Broken_DeviceID	2	[7:0]	DEVICE_ID of the node with which the link is broken.

Table 8-120 – Format of the MMPL of the DMC_BrokenLink.ind message

Field	Octet	Bits	Description
StatusCode	3	[7:0]	0: the reporting node experienced a broken link. 1: the reporting node has no bind information for this MSID. 2 to 255: Reserved by ITU-T.
NumberAffectedMSID	4	[7:0]	Number n of MSIDs affected by the broken link.
MSID0	5	[7:0]	The multicast identification of the first affected MSID.
…	…	…	…
MSIDn	variable	[7:0]	The multicast identification of the nth affected MSID.

19) Clause 9

Revise clause 9 and its sub-clauses as follows:

…

9.1.1.1 CCM encryption

…

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~~ block~~s: the unencrypted part of the data unit and additional data to be protected by the MIC.

…

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 $Y_0 = \mathrm{CIPH}_K(B_0)$: apply the block-cipher algorithm [NIST FIPS 197] with the key *K*.

3) For $i = 1$ to *r*, do $Y_i = \mathrm{CIPH}_K(B_i \oplus Y_{i-1})$: chaining the blocks.

4) Set $T = \mathrm{MSB}_{Tlen}(Y_r)$: the *Tlen* most significant bits of the final round of this computation.

NOTE 1 – These first four steps constitute the cipher-block chaining that calculates the value of *T* to generate MIC~~ = 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 received 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 $S_j = \mathrm{CIPH}_K(Ctr_j)$: apply the block-cipher algorithm with the key *K*.

7) Set $S = S_1 \parallel S_2 \parallel \ldots \parallel S_m$: this defines the string of encrypted counter blocks. Note that S_0 is skipped.

8) Compute $C = (P \oplus \mathrm{MSB}_{Plen}(S)) \parallel (T \oplus \mathrm{MSB}_{Tlen}(S_0))$: 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~~,~~ contains a nonce (*N*). B_1 and B_2 contain associated data bits (*A*). The AES-blocks stand for AES-128 functions. Those are fed by 128-bit counter blocks (Ctr_0-Ctr_m). The PAD compliments the last payload block to 128 bits.

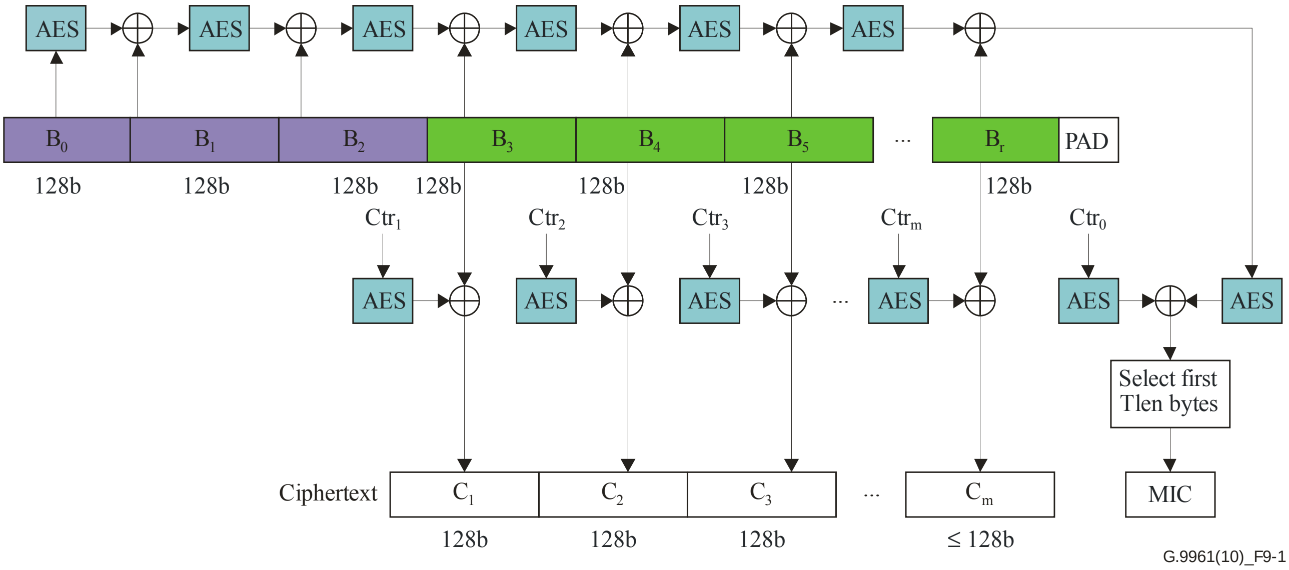


Figure 9-1 – Functional diagram of CCM encryption and message-authentication

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	$\le (2^{14} - 1)$
Associated data size (*Alen*), bytes~~blocks~~	See Table 9-5~~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);
- Nonce block (B_0)
- associated data blocks (~~B_0,~~ B_1~~,~~ and B_2);

– payload blocks (B_3 to B_r);

– encryption key.

…

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-byte~~s long.~~ nonce ~~B~~block B_0 shall have a format as presented in Table 9-4. The length of the encrypted payload in octets (*Plen*) shall be represented as a 16-bit unsigned integer with the LSB mapped to the LSB of byte 15 of B_0.

Table 9-4 – Format of block B_0

Byte number	0	1, 2…, 13	14, 15
Content	Flags (Note)	Nonce	Length of the payload (*Plen*)
NOTE – The content of the Flags byte is: Bit [7] – Reserved by ITU-T for NIST, shall be set to zero. Bit [6] – Shall be set to one. Bits [5:3] – Shall indicate the length of the MIC encoded as: 001_2 – 4-byte MIC. 011_2 – 8-byte MIC. 111_2 – 16-byte MIC. All other values are reserved by ITU-T. Bits [2:0] – Shall be set to 001_2			

The two 16-byte associated data ~~B~~blocks B_1~~,~~ and B_2 shall have a format as presented in Table 9-5. Byte 0 is the first byte and byte 15 is the last byte.

Table 9-5 – Format of blocks B_1, B_2

Block	Bytes	Contents (Note 1)
B_1	0 and 1	Length of associated data in bytes (*Alen*), expressed as an unsigned integer (Note 2).
	2 and 3	Reserved by ITU-T (Note 2).
	4 to 9	Destination MAC address.
	10 to 15	Source MAC address.

Table 9-5 – Format of blocks B_1, B_2

Block	Bytes	Contents (Note 1)
B_2	0 to 4~~3~~	Portion of LFH excluding bytes containing TTL and TSMP fields (Note 4~~3~~).
	~~4~~5 to (~~3~~4 + V)	Additional unencrypted field. APDU (EAPC): TG bytes of EtherType/TAGs plus 2 bytes of MAC client length/type (V = TG +2, See Figure A.1). LCDU: 2 bytes of EtherType (V = 2, or equivalent).
	(~~4~~5 + V) to 15	~~Remainder of the associated data (~~Reserved by ITU-T~~,~~ (Note 3~~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 – For APDU (EAPC), *Alen* shall include byte 2 of B_1 to byte 6+TG of B_2 (21+TG bytes). For LCDU, *Alen* shall include byte 2 of B_1 to byte 6 of B_2 (21 bytes). NOTE 3~~2~~ – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver. NOTE 4~~3~~ – Byte 0 to byte 4~~3~~ of B_2 shall correspond to byte 0 to byte 4~~3~~ of LFH, respectively (Table 8-1).		

All bytes of the nonce and the associated data blocks shall be formatted MSB first: the first bit of the byte 0 is MSB (bit 7) and the last bit of the byte 15 is LSB (bit 0).

Payload blocks (B_3 to B_r) are 16-byte long and shall contain bytes of the APDU or LCDU to be encrypted (see clause 9.1.2.2, encrypted part of APDU or LCDU). The APDU or LCDU bytes shall be mapped to payload blocks in sequential order, so that the first byte of the APDU or LCDU to be encrypted is mapped to byte 0 of B_3, the second byte of the payload is mapped to byte 1 of B_3, the 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 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.

…

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~~ five parts: LFH, CCMP header, unencrypted part, encrypted part (cipher text), and MIC as shown in Figure 9-3.

…

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 type of the encryption key, the length of the MIC, and the security frame number (FN). These ~~three~~ four parameters are necessary for decryption.

The length of the MIC shall be selected according to the procedure defined in clause 9.2.3.

Table 9-6 – CCMP header format

Field	Octet	Bits	Description
CCMP header	0	[2:0]	Length of the MIC encoded as: 001 – 4-byte MIC. 011 – 8-byte MIC. 111 – 16-byte MIC. All other values are reserved by ITU-T.
		[~~5:~~3]	Reserved by ITU-T (Note).
		[5:4]	The type of encryption key: 00 – NN key or NMK. 10 – DB key. 01 – NSC key. 11 – Reserved by ITU-T.
		[~~7:~~6]	Encryption key ID, formatted as an unsigned binary integer.
		[7]	Reserved by ITU-T (Note).
	1 to 5	[39:0]	40-bit FN, formatted as an unsigned binary integer.
NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

The 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 keyID shall be formatted as a 1-bit unsigned binary integer. ~~The range~~Valid values of the key-ID ~~is from~~are 0 ~~to 3~~and 1.~~ 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.

In order to allow some time for the FN update procedure before the FN repeats itself, whenever an FN covers 95% of its maximum value, the key update procedure for the corresponding key shall be started. This would apply to all the keys, namely NN, NMK, DB and multicast keys. In the case of NMK, DB and multicast keys it is possible that multiple nodes realize that the keys need to be updated at the same time, so a random delay in the range 0 to 5s is added between when a node realizes that it needs the key to be updated and when it communicates this to the SC. The node shall communicate to the SC that the key needs to be updated only if the key is not updated by the time its random delay timer expires.

For the DB, the NMK and multicast keys the FN is initialized to a value that is composed of the node's DEVICE_ID in the most significant byte of the FN field and zeros in all other bytes of the FN field. This will ensure that a node will repeat an FN value already used by another node for the same key with a very low probability.

NOTE – On the receive side, the FN may not appear to be sequential, if the order in which packets are encrypted and transmitted is different.

9.2 Authentication and key management procedures

9.2.1 Overview

…

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 9.2.2.2. 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.

…

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 only be used for encryption of secure communications between the node and the SC.

NOTE 1 – The NSC key is used only for communication with the SC function of the node. For secure communications with other clients associated with a node containing the SC function, either NMK or NN keys are used, as defined in clause 9.2.3.

NOTE 2~~1~~ – The PAK protocol, with very high probability, returns a new encryption key after each run.

…

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~~ one second 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 and DB encryption key to the authenticated node in AUT_Confirmation.cnf message (see clause 9.2.5.1.4). The authenticated node is allowed to broadcast and receive broadcast messages using the DB encryption key and may communicate with other nodes using the NMK encryption key in this mode.

…

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 MCSTI~~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. In the case of multicast keys, the supplicant shall provide an MSID. In the case of all other keys, the MSID shall be set to 0.

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. In the case of multicast keys the NN$_{SA}$ and NN$_{AS}$ keys shall be same.

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.

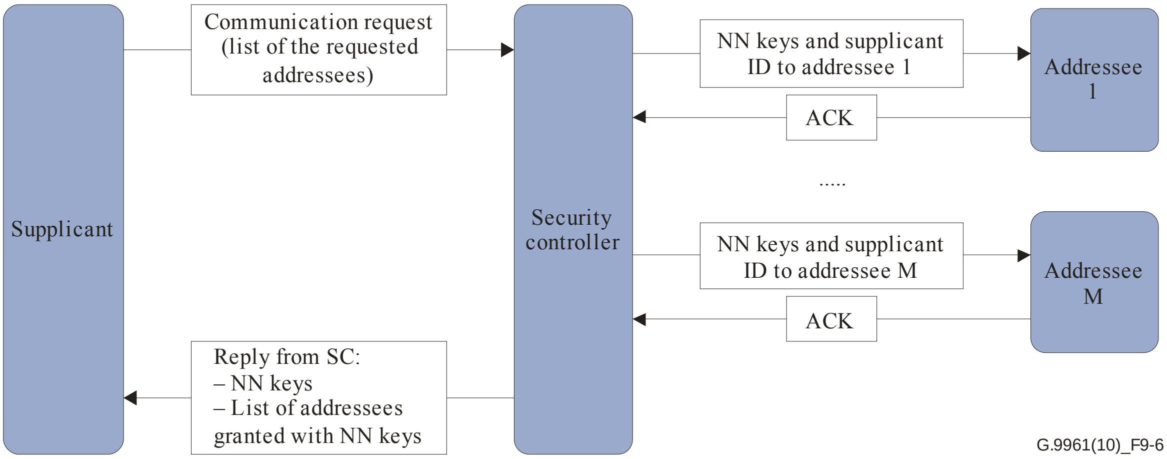

Supplicant
Communication request (list of the requested addressees)
Reply from SC:
– NN keys
– List of addressees granted with NN keys
Security controller
NN keys and supplicant ID to addressee 1
ACK
.....
NN keys and supplicant ID to addressee M
ACK
Addressee 1
Addressee M
G.9961(10)_F9-6

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 5~~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.

   An addressee shall not start using a new NN_{AS} key until it has received a transmission from the supplicant using the corresponding NN_{SA} key. If the supplicant has no frame to transmit in 100 ms after it receives the AKM_KeyConfirmation.req message, it has to transmit the AKM_NewKey.ind message to the addressee indicating that new NN keys are established. The AKM_NewKey.ind message shall be sent unencrypted.

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~~ but the ~~period~~ interval of the updates shall be ~~much~~ longer than ~~the duration of the procedure to establish the corresponding key~~an hour ~~but~~ and 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). In case the SC suspects a security breach, it may update the security keys immediately.

A transmitting node shall not use an old key to encrypt APDUs that arrived at the A-interface, or LCDUs that were generated after the key was updated.

…

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 or re-authentication of the node.

The domain master may resign any node from the domain based on security considerations using the forced resignation procedure described in clause 8.6.1.1.3.2. The SC shall use the SC_DMRes.req message to request resignation of the node from the domain.

9.2.4.3 Updating of the DB and NMK keys

The SC shall update the DB keys and NMK, and communicate the updated keys to all authenticated nodes in the domain, by unicasting the AKM_DomainKeyUpdate.ind message. This message shall always be sent encrypted with the NSC key of the corresponding destination node.

Any authenticated node can also request an updated set of DB and NMK by sending to the SC an AKM_KeyUpdate.req message. The SC shall reply by sending the AKM_KeyConfirmation.req message with security mode set to 01 (NMK and DB keys) or 10 (DB keys only).

…

9.2.5.1.2 Authentication prompt message (AUT_Promp.ind)

The AUT_Promp.ind~~_PRM~~ 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~~_PRM~~ message shall be as shown in Table 9-9.

…

9.2.5.1.4 Authentication confirmation message (AUT_Confirmation.cnf)

…

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).
DB Key ID		[4]	The current DB key ID to use for encryption.
NMK Key ID		[5]	The current NMK key ID to use for encryption.
Reserved		[7:64]	Reserved by ITU-T (Note).
DB0 key	1 to 16	[127:0]	Encryption key for broadcast communications with keyId 0.
DB1 key	17 to 32	[127:0]	Encryption key for broadcast communications with keyId 1.
NMK0	33 to 48	[127:0]	NMK with keyID 0, if security mode is set to 01_2. This field shall be skipped if security mode is set to 00_2.
NMK1	~~17~~49 to 64~~32~~	[127:0]	NMK with keyID 1, if security mode is set to 01_2. This field shall ~~not~~ be ~~present~~ skipped if security mode is set to 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).
Multicast stream identifier	1	[7:0]	Shall be set to the multicast stream identifier (MSID) for multicast keys. Otherwise it shall be set to 0.
Addressee name	2~~1~~	[7:0]	First addressee unicast DEVICE_ID.
Addressee name	3~~2~~	[7:0]	Second addressee unicast DEVICE_ID.
…	…	…	…
Addressee name	N+1	[7:0]	N-th addressee unicast DEVICE_ID.
Attempt number	N+2~~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 key~~s~~ with ID = 0~~, 1~~ and set to one to request key~~s~~ with ID = 1~~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.
Multicast stream identifier	1	[7:0]	Shall be set to the multicast stream identifier (MSID) for multicast keys. Otherwise it shall be set to 0
NN$_{SA}$ key 0/1~~2~~	~~1~~2 to 17~~6~~	[127:0]	Encryption key for Supplicant-to-Addressee direction with ID=0 if KeyID=0 and with ID=1~~2~~ if KeyID=1
NN$_{AS}$ key 0/1~~2~~	18~~7~~ to 33~~2~~	[127:0]	Encryption key for Addressee-to-Supplicant direction with ID=0 if KeyID=0 and with ID=1~~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~~
~~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~~34	[1:0]	Shall be set to 00_2 for the initial request and incremented for every next attempt.
KeyID		[2]	Set to zero for key with ID = 0 and set to one for key with ID = 1
Reserved		[7:3~~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).
Multicast stream identifier	2	[7:0]	Shall be set to the multicast stream identifier (MSID) for multicast keys. Otherwise it shall be set to 0
NN$_{SA}$ key 0/1~~2~~	~~2~~3 to 18~~7~~	[127:0]	Encryption key for supplicant-to-addressee direction with ID=0 if KeyID=0 and with ID=1~~2~~ if KeyID=1
NN$_{AS}$ key 0/1~~2~~	19~~8~~ to 34~~3~~	[127:0]	Encryption key for addressee-to-supplicant direction with ID=0 if KeyID=0 and with ID=1~~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~~
KeyID	35	[0]	Set to zero for key with ID = 0 and set to one for key with ID = 1
Reserved		[7:1]	Reserved by ITU-T (Note)
NOTE – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver.			

9.2.5.2.3 Communication acknowledgement message (AKM_KeyAck.cnf)

The AKM_KeyAck.cnf~~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 ~~to~~ of decoding~~ed~~ the new encryption key (NACK). 10 – If the addressee decoded successfully the new encryption key, but denies communication with supplicant (NACK). 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~~,~~ – Update of the DB keys. 11 – Reserved by ITU-T.
KeyID		[2]	Set to zero for key with ID = 0 and set to one for key with ID = 1
Reserved		[7:3~~2~~]	Reserved by ITU-T (Note 1).
DB 0/1	1 to 16	[127:0]	DB key with ID=0 if KeyID=0 and with ID=1 if KeyID=1.
NMK 0/1~~2~~	17 to 32~~16~~	[127:0]	NMK with ID=0 if KeyID=0 and with ID=1~~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 0/1~~2~~	~~17~~33 to 48~~32~~	[127:0]	Encryption key for supplicant-to-addressee direction with ID=0 if KeyID=0 and with ID=1~~2~~ if KeyID=1
NN$_{AS}$ key 0/1~~2~~	~~33~~49 to 64~~48~~	[127:0]	Encryption key for addressee-to-supplicant direction with ID=0 if KeyID=0 and with ID=1~~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	65~~81~~	[7:0]	Number of addressees N (1 for unicast transmission and up to 248 for multi-cast transmission) (Note 2).
Addressee name	66~~82~~	[7:0]	First addressee unicast DEVICE_ID.
Addressee name	67~~83~~	[7:0]	Second addressee unicast DEVICE_ID.
…	…	…	…
Addressee name	~~81~~65 + N	[7:0]	N-th addressee unicast DEVICE_ID.
NOTE 1 – Bits that are reserved by ITU-T shall be set to zero by the transmitter and ignored by the receiver. NOTE 2 – In case no addressee is authenticated, the list is empty and the field shall be set to zero.			

…

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). 00_{16}: node has successfully self resigned 01_{16}: node has been expelled by DM because self resignation failed 02_{16} to FF_{16}: reserved by ITU-T.
…	…	…	…
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). 00_{16}: node has successfully self resigned 01_{16}: node has been expelled by DM because self resignation failed 02_{16} to FF_{16}: reserved by ITU-T.

9.2.5.2.7 Confirmation of a new key message (AKM_NewKey.ind)

The AKM_NewKey.ind message is a unicast management message that shall only be used to inform the addressee that the supplicant received the NN key and communication using this new NN key is available. The MMPL of the AKM_NewKey.ind message shall be empty.

9.2.5.3 Key updating messages

9.2.5.3.1 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, or~~and~~
- NN keys or NMK, or
- DB key~~ 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 for DB, 11 ~~are~~ is reserved by ITU-T.
KeyID		[2]	Set to 0 to request keys with ID = 0 and set to 1 to request keys with ID = 1
Reserved		[7:3~~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).

Table 9-19 – Format of the MMPL of the AKM_KeyUpdate.req message

Field	Octet	Bits	Description
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.			

9.2.5.3.2 Domain wide key-update indication (AKM_DomainKeyUpdate.ind)

The AKM_ DomainKeyUpdate.ind message is a unicast management message intended to be used for indicating the update of the DB keys and NMK. The format of the MMPL of the AKM_DomainKeyUpdate.ind message shall be as shown in Table 9-11 (AUT_Confirmation.cnf) with the value of the confirmation flag field set to 11_2.

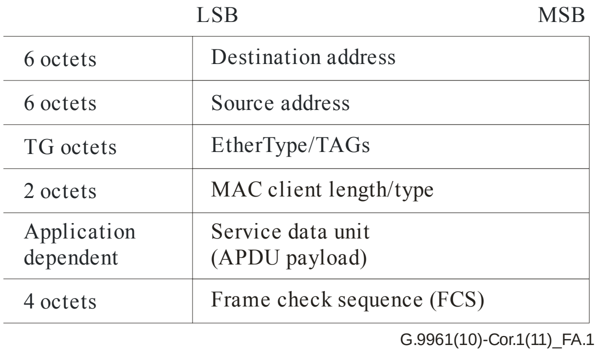
20) Annex A

Revise clause A.1.1 as follows:

A.1.1 Frame conversion

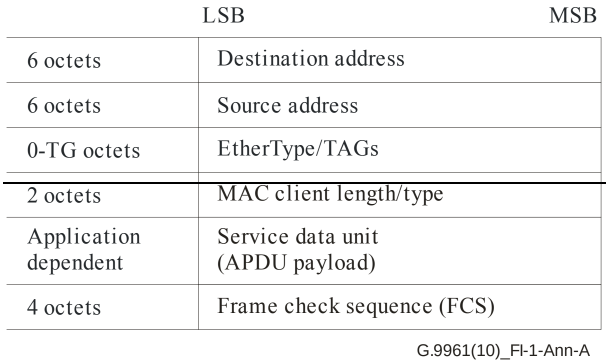
…

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
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)-Cor.1(11)_FA.1



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 ~~M~~APDU depends on the length *TG* of the EtherType/TAGs field of the APDU (see clause 9.1.2.2).

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, 2012