International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

G.8121.2/Y.1381.2

(11/2013)

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

Packet over Transport aspects – MPLS over Transport aspects

SERIES Y: GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS AND NEXT-GENERATION NETWORKS

Internet protocol aspects – Transport

Characteristics of MPLS-TP equipment functional blocks supporting ITU-T G.8113.2/Y.1372.2 OAM mechanisms

Recommendation ITU-T G.8121.2/Y.1381.2

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
Ethernet over Transport aspects	G.8000–G.8099
MPLS over Transport aspects	**G.8100–G.8199**
Quality and availability targets	G.8200–G.8299
Service Management	G.8600–G.8699
ACCESS NETWORKS	G.9000–G.9999

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

Recommendation ITU-T G.8121.2/Y.1381.2

Characteristics of MPLS-TP equipment functional blocks supporting ITU-T G.8113.2/Y.1372.2 OAM mechanisms

Summary

Recommendation ITU-T G.8121.2/Y.1381.2 specifies both the functional components and the methodology that should be used in order to specify the MPLS-TP layer network functionality of network elements based on the protocol-neutral constructs defined in Recommendation ITU-T G.8121/Y.1381 and on the tools defined in Recommendation ITU-T G.8113.2/Y.1372.2.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T G.8121.2/Y.1381.2	2013-11-06	15	11.1002/1000/12018-en

Keywords

Atomic functions, equipment functional blocks, MPLS-TP, MPLS-TP layer network.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

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 not 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

Recommendation ITU-T G.8121.2/Y.1381.2

Characteristics of MPLS-TP equipment functional blocks supporting ITU-T G.8113.2/Y.1372.2 OAM mechanisms

1 Scope

This Recommendation describes both the functional components and the methodology that should be used in order to describe MPLS-TP layer network functionality of network elements; it does not describe individual MPLS-TP network equipment.

This Recommendation provides protocol-specific extensions of the protocol-neutral constructs defined in [ITU-T G.8121] to support the OAM tools defined in [ITU-T G.8113.2].

This Recommendation provides a description of the MPLS-TP functional technology using the same methodologies that have been used for other transport technologies (e.g., SDH, OTN and Ethernet)[1].

This Recommendation, along with [ITU-T G.8121], specifies a library of basic building blocks and a set of rules by which they may be combined in order to describe digital transmission equipment. The library comprises the functional building blocks needed to specify completely the generic functional structure of the MPLS-TP layer network. In order to be compliant with this Recommendation, equipment needs to be describable as an interconnection of a subset of these functional blocks contained within this Recommendation. The interconnections of these blocks should obey the combination rules given.

Not every atomic function defined in this Recommendation is required for every application. Different subsets of atomic functions may be assembled in different ways according to the combination rules given in this Recommendation to provide a variety of different capabilities. Network operators and equipment suppliers may choose which functions must be implemented for each application.

2 References

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

[ITU-T G.805] Recommendation ITU-T G.805 (2000), *Generic functional architecture of transport networks*.

[ITU-T G.806] Recommendation ITU-T G.806 (2012), *Characteristics of transport equipment – Description methodology and generic functionality*.

[ITU-T G.8101] Recommendation ITU-T G.8101/Y.1355 (2013), *Terms and definitions for MPLS transport profile*.

[ITU-T G.8110.1] Recommendation ITU-T G.8110.1/Y.1370.1 (2011), *Architecture of the Multi-Protocol Label Switching transport profile layer network*.

[1] This ITU-T Recommendation is intended to be aligned with the IETF MPLS RFCs normatively referenced by this Recommendation.

[ITU-T G.8113.2] Recommendation ITU-T G.8113.2/Y.1372.2 (2012), *Operations, administration and maintenance mechanisms for MPLS-TP networks using the tools defined for MPLS.*

[ITU-T G.8121] Recommendation ITU-T G.8121/Y.1381 (2013), *Characteristics of MPLS-TP equipment functional blocks*.

[IETF RFC 4379] IETF RFC 4379 (2006), *Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures*.

[IETF RFC 5880] IETF RFC 5880 (2010), *Bidirectional Forwarding Detection (BFD).*

[IETF RFC 5884] IETF RFC 5884 (2010), *Bidirectional Forwarding Detection (BFD) for MPLS Label Switched Paths (LSPs).*

[IETF RFC 6435] IETF RFC 6435 (2011), *MPLS Transport Profile Lock Instruct and Loopback Functions*.

[IETF RFC 6374] IETF RFC 6374 (2011), *Packet Loss and Delay Measurement for MPLS Networks*.

[IETF RFC 6375] IETF RFC 6375 (2011), *A Packet Loss and Delay Measurement Profile for MPLS-Based Transport Networks*.

[IETF RFC 6426] IETF RFC 6426 (2011), *MPLS On-Demand Connectivity Verification and Route Tracing*.

[IETF RFC 6427] IETF RFC 6427 (2011), *MPLS Fault Management Operations, Administration, and Maintenance (OAM).*

[IETF RFC 6428] IETF RFC 6428 (2011), *Proactive Connectivity Verification, Continuity Check and Remote Defect Indication for the MPLS Transport Profile*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 access point: [ITU-T G.805]

3.1.2 adapted information: [ITU-T G.805]

3.1.3 associated channel header: [ITU-T G.8101]

3.1.4 bottom of stack: [ITU-T G.8101]

3.1.5 characteristic information: [ITU-T G.805]

3.1.6 client/server relationship: [ITU-T G.805]

3.1.7 connection: [ITU-T G.805]

3.1.8 connection point: [ITU-T G.805]

3.1.9 explicitly TC-encoded-PSC LSP: [ITU-T G.8101]

3.1.10 G-ACh label: [ITU-T G.8101]

3.1.11 generic associated channel: [ITU-T G.8101]

3.1.12 label: [ITU-T G.8101]

3.1.13 label inferred PHB scheduling class LSP: [ITU-T G.8101]

3.1.14 label stack: [ITU-T G.8101]

3.1.15 label switched path: [ITU-T G.8101]

3.1.16 label value: [ITU-T G.8101]

3.1.17 layer network: [ITU-T G.805]

3.1.18 matrix: [ITU-T G.805]

3.1.19 MPLS label stack: [ITU-T G.8101]

3.1.20 network: [ITU-T G.805]

3.1.21 network connection: [ITU-T G.805]

3.1.22 per-hop behaviour: [ITU-T G.8101]

3.1.23 reference point: [ITU-T G.805]

3.1.24 subnetwork: [ITU-T G.805]

3.1.25 subnetwork connection: [ITU-T G.805]

3.1.26 termination connection point: [ITU-T G.805]

3.1.27 time to live: [ITU-T G.8101]

3.1.28 traffic class: [ITU-T G.8101]

3.1.29 trail: [ITU-T G.805]

3.1.30 trail termination: [ITU-T G.805]

3.1.31 transport: [ITU-T G.805]

3.1.32 transport entity: [ITU-T G.805]

3.1.33 transport processing function: [ITU-T G.805]

3.1.34 unidirectional connection: [ITU-T G.805]

3.1.35 unidirectional trail: [ITU-T G.805]

3.2 Terms defined in this Recommendation

None.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

AIS Alarm Indication Signal

APS Automatic Protection Switching

BFD Bidirectional Forwarding Detection

CC Continuity Check

CCCV Continuity Check and Connectivity Verification

CoS Class of Service

CSF Client Signal Fail

CV Connectivity Verification

DLM Direct Loss Measurement

DM Delay Measurement

DSMap Downstream Mapping

EMF	Equipment Management Function
FEC	Forwarding Equivalence Class
G-ACh	Generic Associated Channel
GAL	G-ACh Label
GFP	Generic Framing Procedure
ILM	Inferred Loss Measurement
LCK	Locked
LI	Lock Instruct
LKI	Lock Instruct
LKR	Lock Report
LM	Loss Measurement
LStack	Label Stack
MCC	Maintenance Communication Channel
MEP	Maintenance entity group (MEG) End Point
MIP	Maintenance entity group (MEG) Intermediate Point
MPLS	Multi-Protocol Label Switching
MPLS-TP	Multi-Protocol Label Switching – Transport Profile
MT	Multi-Protocol Label Switching – Transport Profile
MTDe	MPLS-TP MEP Diagnostic function
MTU	Maximum Transmit Unit
OAM	Operation, Administration and Maintenance
ODCV	On-Demand Connectivity Verification
PDU	Protocol Data Unit
PHB	Per Hop Behaviour
PM	Performance Monitoring
QTF	Querier's Timestamp Format
RDI	Remote Detect Indication
Req	Request
Resp	Response
RPTF	Responder's Preferred Timestamp Format
RTF	Responder's Timestamp Format
SCC	Signalling Communication Channel
SQI	Session Query Interval
SSF	Server Signal Fail
TC	Traffic Class
TLV	Type Length Value
TS	Timestamp

TSFmt	Timestamp Format
TTL	Time-To-Live

5 Conventions

The diagrammatic convention for connection-oriented layer networks described in this Recommendation is that of [ITU T G.805].

6 Supervision

6.1 Defects

6.1.1 Summary of entry/exit conditions for defects

The defect entry and exit conditions are based on events. Occurrence or absence of specific events may raise or reset specific defects.

The events used by this Recommendation are defined in Table 6-1 of [ITU-T G.8121]. The events, unexpPeriod, CSF-LOS, CSF-FDI and CSF-RDI that are described in Table 6-1 of [ITU-T G.8121] are out of the scope of this Recommendation.

7 Information flow across reference points

Information flow for MPLS-TP functions is defined in clause 9. A generic description of information flow is defined in clause 7 of [ITU-T G.806].

8 MPLS-TP processes

8.1 G-ACh process

In the case where OAM packets are encapsulated using a generic associated channel (G-ACh), the G-Ach process is described in clause 8.1 of [ITU-T G.8121]. Encapsulation of OAM packets using IP/UDP or other mechanisms is for further study.

8.2 TC/Label processes

See clause 8.2 of [ITU-T G.8121].

8.3 Queueing process

See clause 8.3 of [ITU-T G.8121].

8.4 MPLS-TP-specific GFP-F processes

See clause 8.4 of [ITU-T G.8121].

8.5 Control word (CW) processes

See clause 8.5 of [ITU-T G.8121].

8.6 OAM-related processes used by server adaptation functions

8.6.1 Selector process

See clause 8.6.1 of [ITU-T G.8121].

8.6.2 AIS insert process

The AIS insert process generates MT_CI traffic units containing the AIS signal. MI_AIS_Period specifies the period between successive AIS messages, in seconds between 1 and 20. MI_AIS_CoS specifies the priority for AIS messages. MI_Local_Defect specifies whether an alternative path is available – that is, it is set to true when either the server layer does not provide any protection, or when both the working and protect paths have faults. The AIS insert process behaviour depends on the aAIS consequent action.

NOTE – It is expected that MI_Local_Defect can be set correctly by the EMF without explicit interaction by the end user. The value can be precomputed as described in [IETF RFC 6427].

The AIS insert process is described in clause 8.6.2 of [ITU-T G.8121], and is shown in Figure 8-1.

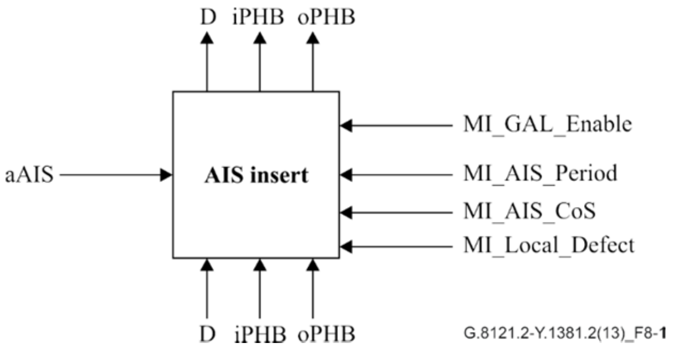

Figure 8-1 – AIS insert process

Figure 8-2 defines the behaviour of the AIS insert process:

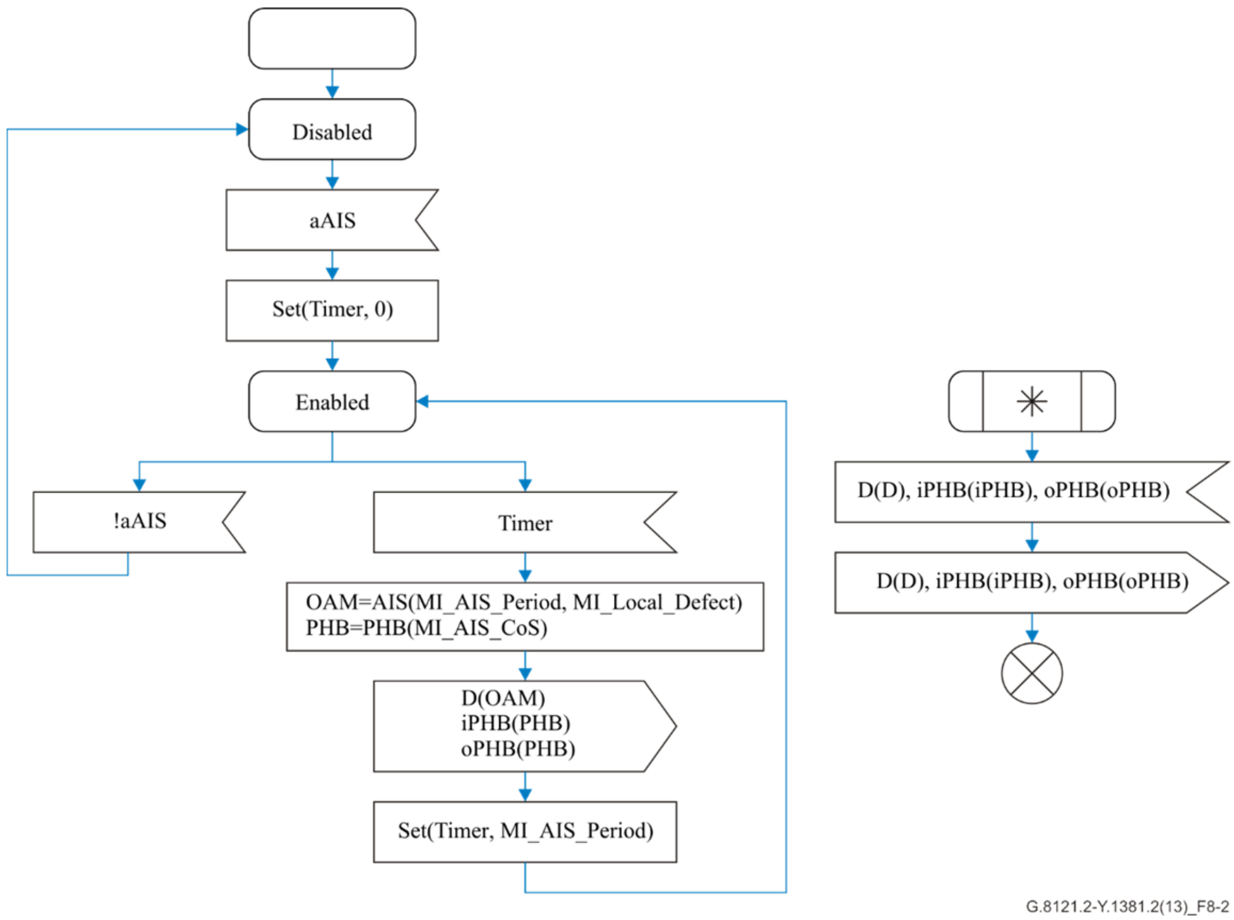

Figure 8-2 – AIS insert behaviour

The AIS function creates an AIS frame, by first creating an AIS PDU, and then encapsulating it in a G-ACh and depending on MI_GAL_Enable, a GAL, as described in clause 8.1 of [ITU-T G.8121]. It then inserts it into the data traffic stream. The PHB(CoS) function returns the PHB with the lowest drop precedence within the class of service defined by the CoS input parameter.

The AIS PDU is created according to the format described in [IETF RFC 6427]. The fields are filled in as follows:

- Vers: set to 1.
- Reserved: set to 0.
- Message Type: set to AIS.
- Flags: The L flag is set to 1 if MI_Local_Defect is true, and is otherwise set to 0. The remaining flags are set to 0.
- Refresh Timer: set to MI_AIS_Period.
- Total TLV Length: set to 0.

Inclusion of the IF_ID and Global_ID TLVs is for further study.

The PHB(CoS) function returns the PHB with the lowest drop precedence within the class of service defined by the CoS input parameter.

8.6.3 LCK generation process

The LCK generation process generates MT_CI traffic units containing the lock signal, i.e., containing LKR messages. MI_LCK_Period specifies the period between successive LKR messages in seconds between 1 and 20. MI_LCK_CoS specifies the priority for LKR messages.

NOTE – IETF uses "LKR" (Lock Report) equivalently to the ITU-T use of "LCK".

The LCK generation process is described in clause 8.6.3 of [ITU-T G.8121] and its behaviour is shown in Figure 8-3.

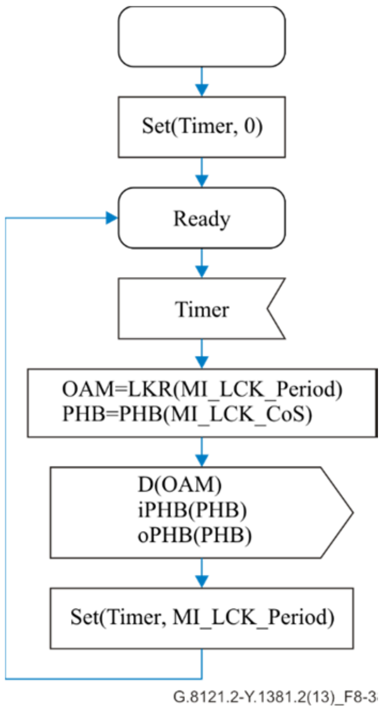

Figure 8-3 – LCK generation behaviour

The LKR function creates an LKR frame, by first creating an LKR PDU, and then encapsulating it in a G-ACh and depending on MI_GAL_Enable, a GAL, as described in clause 8.1 of [ITU-T G.8121].

The LKR PDU is created according to the format described in [IETF RFC 6427]. The fields are filled in as follows:

- Vers: set to 1.
- Reserved: set to 0.
- Message Type: set to LKR.
- Flags: set to 0.
- Refresh Timer: set to MI_LCK_Period.
- Total TLV Length: set to 0.

Inclusion of the IF_ID and Global_ID TLVs is for further study.

The PHB(CoS) function returns the PHB with the lowest drop precedence within the class of service defined by the CoS input parameter.

8.7 OAM-related processes used by adaptation functions

8.7.1 MCC/SCC mapping insert and de-mapping process

See clause 8.7.1 of [ITU-T G.8121].

8.7.2 APS insert and extract process

See clause 8.7.2 of [ITU-T G.8121].

8.7.3 CSF insert and extract process

See clause 8.7.3 of [ITU-T G.8121].

8.8 Proactive and on-demand OAM-related processes

As described in clause 8.8 of [ITU-T G.8121], there are 6 processes for proactive and on-demand OAM:

- Proactive OAM source control
- Proactive OAM sink control
- On-demand OAM source control
- On-demand OAM sink control
- OAM PDU generation
- OAM PDU reception

Each of these consists of a number of protocol-specific subprocesses, as described in [ITU-T G.8121]. Appendix I provides the table that indicates the relationship between processes and subprocesses and indicates where these (sub)processes are implemented in the termination functions (MT_TT, MTDe_TT, and MTDi_TT).

The OAM Mux subprocess is responsible for multiplexing together (PDU, TTL, PHB) signals from other subprocesses, and passing them to the G-ACh insertion process along with the appropriate channel type. Similarly, the OAM Demux subprocess receives (PDU, PHB, LStack, channel type) signals from the G-ACh extraction process, and passes on the (PDU, PHB, LStack) signals to the other subprocesses as appropriate, depending on the channel type.

The following subclauses describe the other subprocesses listed in Appendix I. They are organized by function (e.g., CCCV, on-demand CV, etc.), with all the subprocesses relevant to a particular function described together.

8.8.1 CC/CV processes

An overview of the CC/CV processes is shown in Figure 8-4.

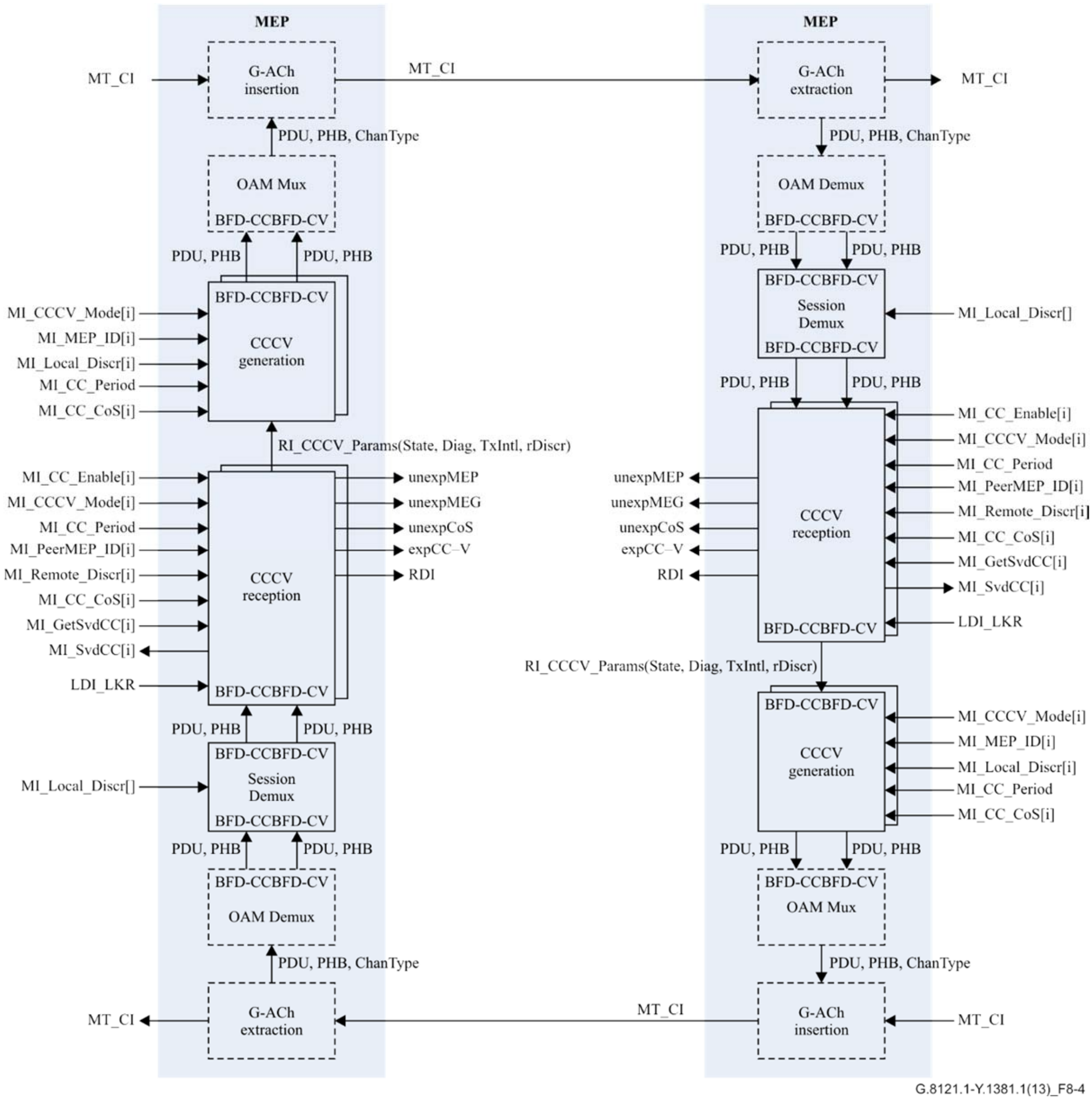

Figure 8-4 – Overview of CC/CV processes

The CCCV reception process controls the operation of the CCCV protocol. It operates when MI_CC_Enable is TRUE, according to the value of MI_CCCV_Mode. MI_CCCV_Mode takes one of the following values:

- COORD – Coordinated mode; operate a single co-ordinated BFD session
- SRC – Independent source; operate as the source MEP in an independent BFD session
- SINK – Independent sink; operate as the sink MEP in an independent BFD session

NOTE – [IETF RFC 6428] defines two modes for bidirectional LSPs operation, i.e., Coordinated mode and independent mode. In independent mode, separate sessions are used for each direction and a given MEP operates as the source for one session and the sink for the other session. Thus, there are three possible values for MI_CCCV_Mode as shown above.

Multiple instances of the CCCV reception process may be created for multiple BFD sessions; when operating in independent mode, it is expected that a pair of instances are created, one acting as the source and one as the sink.

MI_CC_Period specifies the desired period between successive BFD-CC messages, and MI_PeerMEP_ID specifies the MEP ID value to expect in received messages, in one of the formats described in [IETF RFC 6428].

The CCCV generation process sends periodic BFD-CC and BFD-CV messages, when MI_CC_Enable is TRUE. There is a separate instance of the process for each corresponding instance of the CCCV reception process. MI_MEP_ID and MI_Local_Discr specify the local MEP ID and session discriminator values to send in the packets.

The session Demux process demultiplexes received BFD-CC and BFD-CV messages to the correct instance of the CCCV reception process, based on the "Your discriminator" field in the received BFD-CC or BFD-CV packet. Demultiplexing of received packets where the "Your discriminator" field is 0 is for further study.

8.8.1.1 CCCV reception process

The CCCV reception process controls the operation of the BFD protocol, according to MI_CC_Enable and MI_CCCV_Mode. Multiple instances of the CCCV reception process can be instantiated. Each one has a corresponding instance of the CCCV generation process; the contents and period for sending CCCV packets are controlled via the RI_CCCV_Params(state, diag, TX-interval,your-discriminator) signal.

The CCCV reception process is described in Figures 8-5a, 8-5b and 8-5c. In Disabled state, all received BFD-CC and BFD-CV packets are discarded and no packets are sent. In Enabled state, received BFD-CC packets are processed, and received BFD-CV packets are processed when the BFD state machine is UP. BFD-CC and BFD-CV packets are sent, except if the process is operating in SINK mode. When MI_CC_Enabled is set to FALSE, the process moves to Disabling state so that the ADMIN_DOWN diagnostic code can be signalled to the peer MEP. The process stays in Disabling state for three times the transmit interval, before moving to Disabled state. In Disabling state, BFD-CC packets are sent, but received BFD-CC and BFD-CV packets are used only for updating the timer.

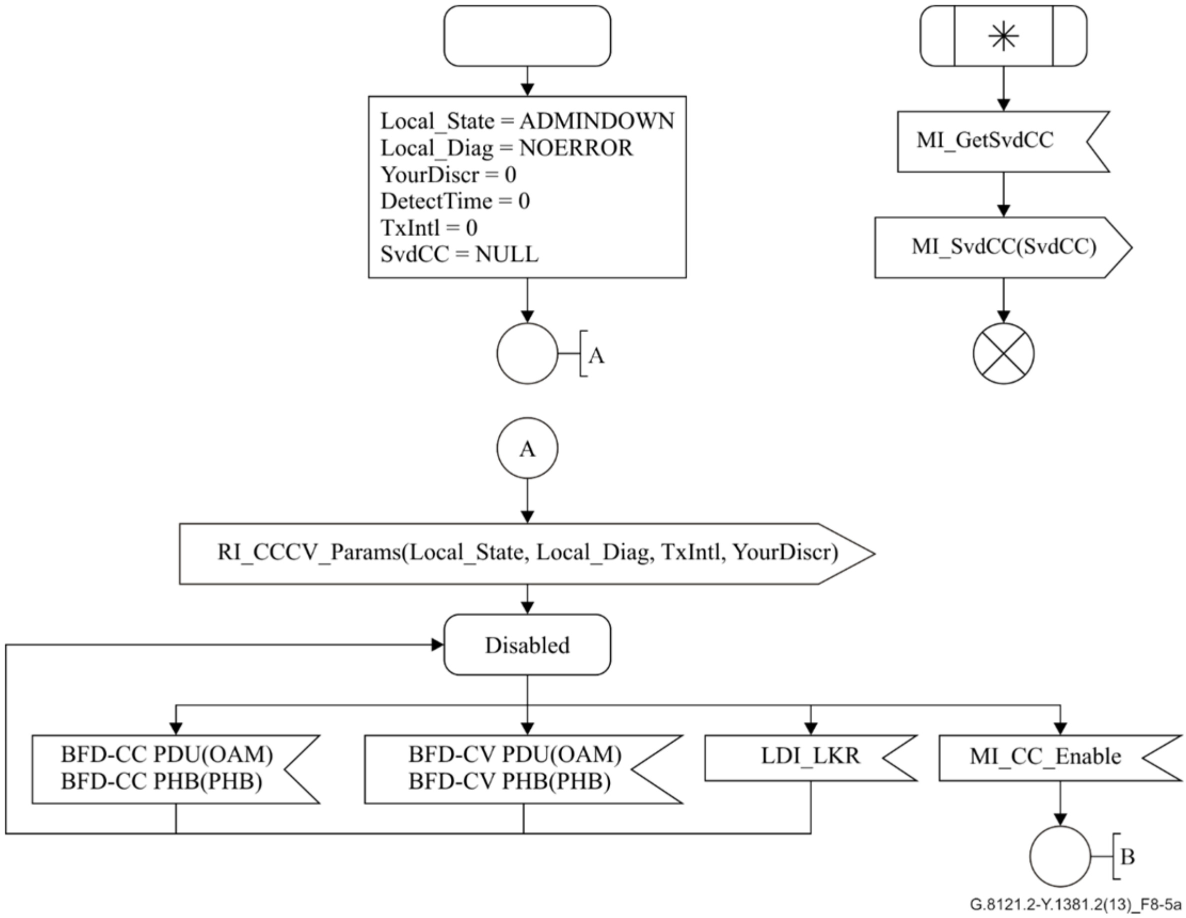

Figure 8-5a – CCCV reception process

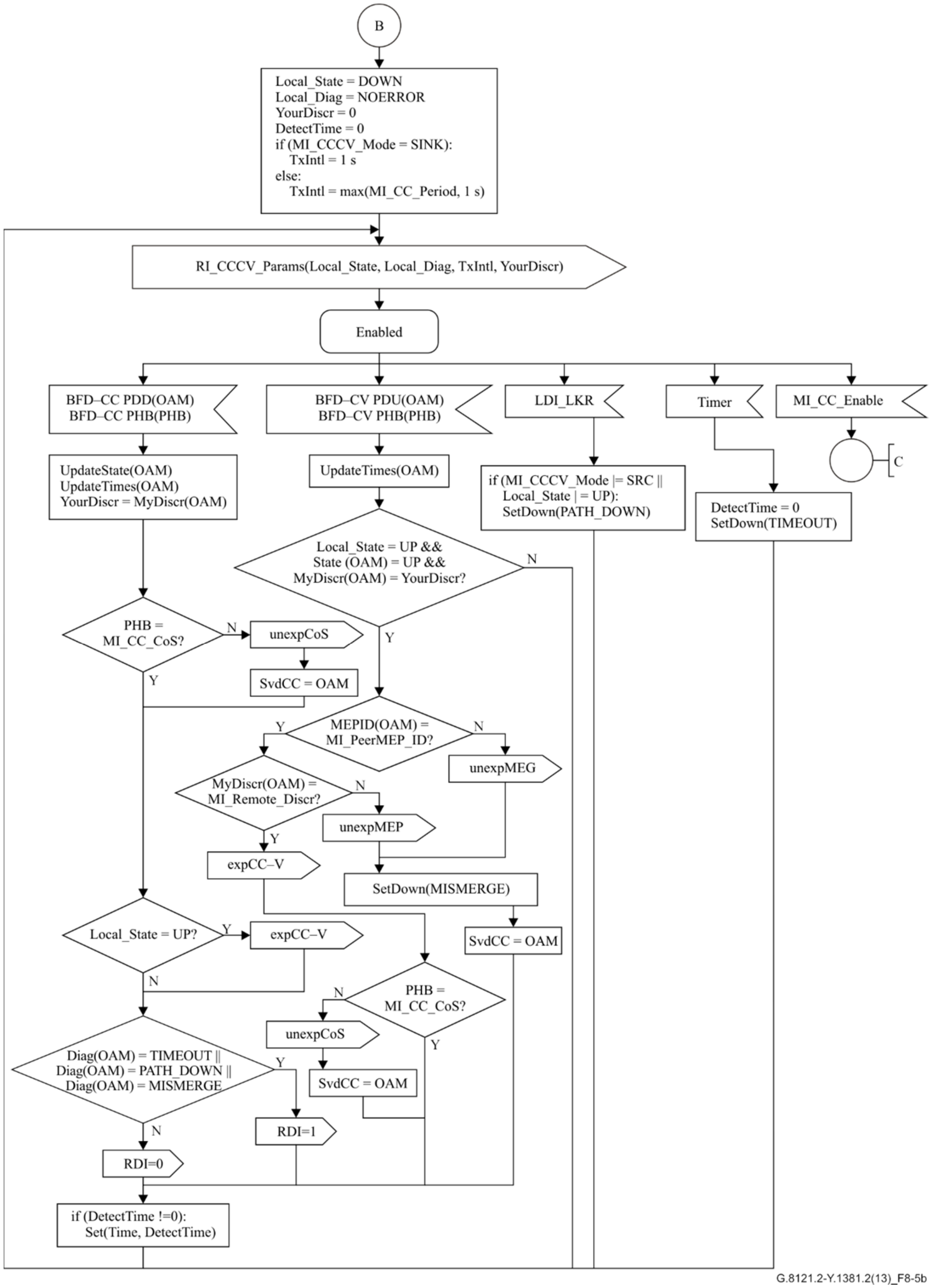

Figure 8-5b – CCCV reception process

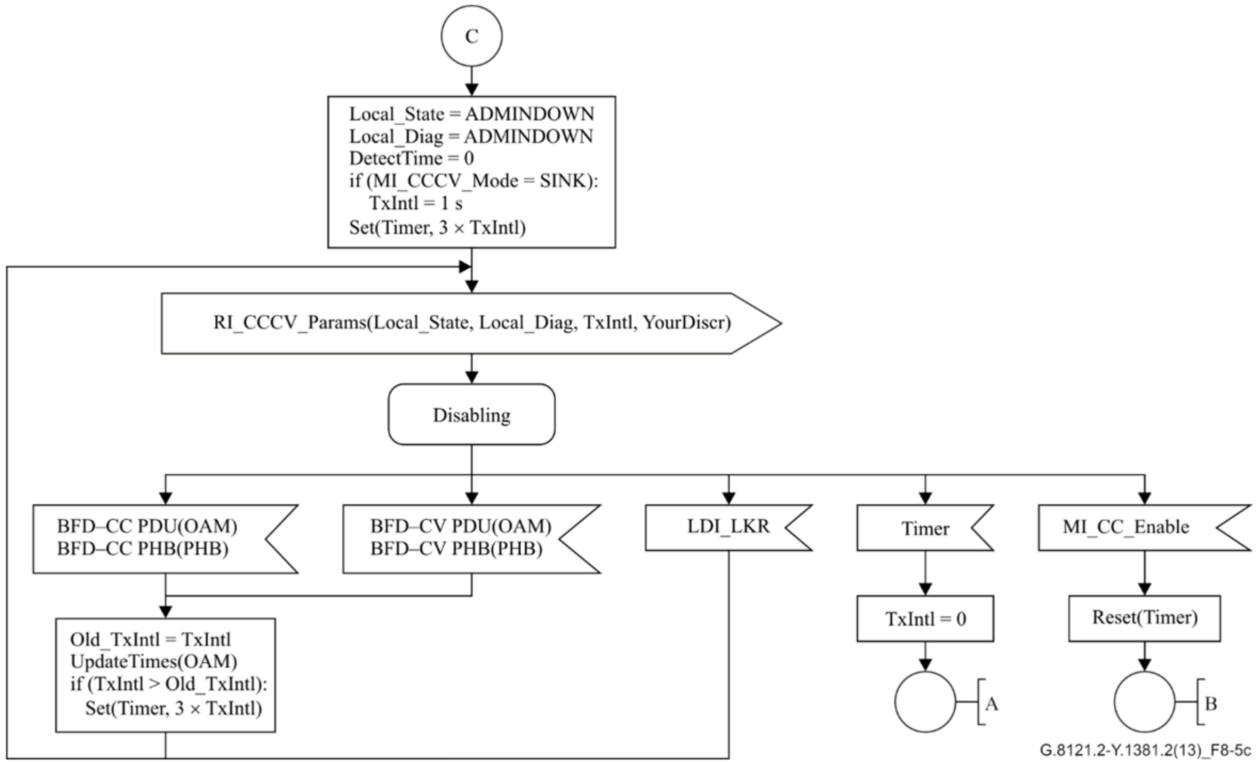

Figure 8-5c – CCCV reception process

The values of State and Diag correspond with those in [IETF RFC 5880] and [IETF RFC 6428].

The functions 'SetDown', 'UpdateState' and 'UpdateTimes' are described by the following pseudocode:

```
SetDown(new_diag) {
    if (Local_State != DOWN) {
        Local_State = DOWN
        if (Local_Diag != PATH_DOWN || new_diag != TIMEOUT) {
            Local_Diag = new_diag
        }
        if (MI_CCCV_Mode = SINK) {
            TxIntl = 1s
        }
    }
}

UpdateState(OAM) {
    if (State(OAM) = ADMINDOWN) {
        SetDown(NBR_DOWN)
    } else {
        if (Local_State = DOWN) {
            if (State(OAM) = DOWN) {
                Local_State = INIT
                Local_Diag = NOERROR
```

```
        } else if (State(OAM) = INIT ||
                   (MI_CCCV_Mode = SINK && State(OAM) = UP)) {
            Local_State = UP
            Local_Diag = NOERROR
        }
    } else if (Local_State = INIT) {
        if (State(OAM) = INIT || State(OAM) = UP) {
            Local_State = UP
            Local_Diag = NOERROR
        }
    } else {
        // Local_State must be UP
        if (state(OAM) = DOWN && MI_CCCV_Mode != SRC) {
            SetDown(NBR_DOWN)
        }
    }
  }
}

UpdateTimes(OAM) {
    if (MI_CCCV_Mode = SRC) {
        DetectTime = 0
    } else {
        DetectTime = 3 x max(MI_CC_Period, DesiredMinTxInterval(OAM))
    }
    if (MI_CCCV_Mode = SINK) {
        if (State(OAM) != LocalState) {
            TxIntl = 1s
        } else {
            TxIntl = 0
        }
    } else {
        TxIntl = max(MI_CC_Period, RequiredMinRxInterval(OAM))
    }
}
```

Use of authentication for CC/CV is for further study.

Use of the BFD Poll/Final mechanism for changing the value of TxIntl is for further study.

8.8.1.2 CCCV generation process

The CCCV generation process is responsible for generating BFD-CC and BFD-CV packets, according to the parameters set by the corresponding CCCV reception process in the RI_CCCV_Params(state, diag, TX-interval,your-discriminator) signal. When the TX-interval (TxIntl) is set to 0, no BFD-CC or BFD-CV packets are generated. Otherwise, BFD-CC packets are generated at the specified interval, and BFD-CV packets are generated if the state is up, at an interval of 1 s.

The CCCV generation process is described in Figure 8-6.

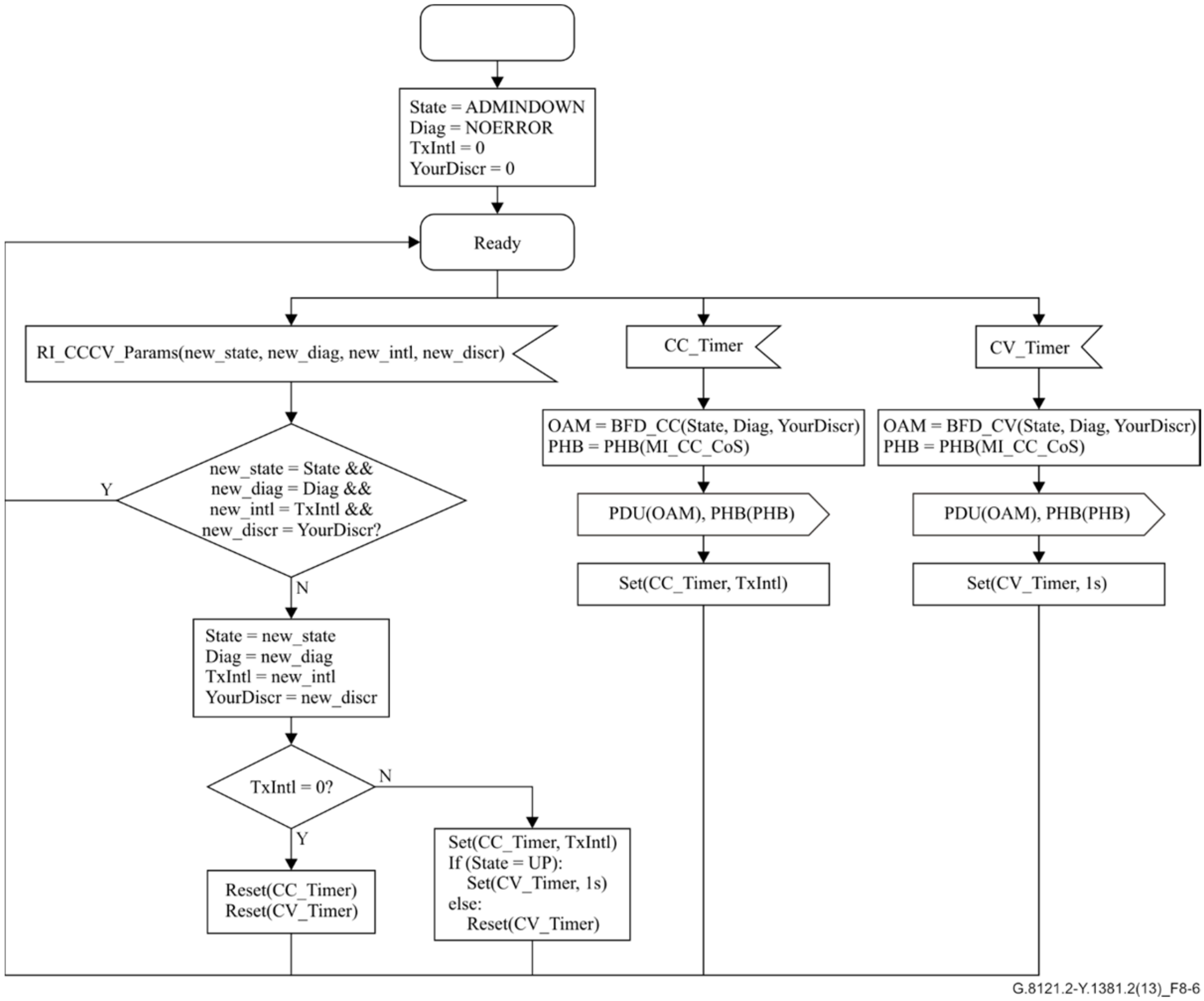

Figure 8-6 – CCCV generation process

The BFD_CC function creates a BFD control packet according to the format described in [IETF RFC 5880]. The fields are filled in as follows:

- Vers: set to 1.
- Diag: set to the value of Diag.
- Sta: set to the value of State.
- P, F, A, D, M flags: set to 0.
- C flag: set appropriately depending on the implementation.
- Detect Mult: set to 3.

- Length: set to 24.
- My Discriminator: set to MI_Local_Discr.
- Your Discriminator: set to YourDiscr.
- Desired Min Tx Interval: set to 0 if MI_CCCV_Mode is SINK, otherwise set to MI_CC_Period.
- Required Min Rx Interval: set to 0 if MI_CCCV_Mode is SRC, otherwise set to MI_CC_Period.
- Required Min Echo Rx Interval: set to 0.

No authentication section is added. Use of authentication is for further study.

The BFD_CV function creates a BFD control packet in the same way as the BFD_CC function, and then appends an MEP source ID TLV as described in [IETF RFC 6428], containing the value of MI_MEP_ID.

The PHB(CoS) function returns the PHB with the lowest drop precedence within the class of service defined by the CoS input parameter.

8.8.1.3 Session demux process

The session demux process receives BFD-CC and BFD-CV packets from the OAM demux process. It performs the following checks on the packet:

- If the version number is not 1, the packet is discarded.
- If the length is less than 24, the packet is discarded.
- If the Detect Mult field is 0, the packet is discarded.
- If any of the P, F, A, D or M flags are set, the packet is discarded.
- If the My Discriminator field is 0, the packet is discarded.
- If the Required Min Echo Rx Interval is not 0, the packet is discarded.
- If the Your Discriminator field is 0 and the state is not DOWN or ADMINDOWN, the packet is discarded.
- If the Your Discriminator field is not 0 and no corresponding session can be found based on MI_Local_Discr[], the packet is discarded.

If the checks pass, the packet is passed to the instance of the CCCV reception process whose MI_Local_Discr is equal to the Your Discriminator field. Packets received on the BFD-CC port from the OAM Demux process are passed on to the BFD-CC port in the CCCV reception process, and packets received on the BFD-CV port from the OAM Demux process are passed on to the BFD-CV port in the CCCV reception process.

Selection of the correct CCCV reception process when the Your Discriminator field is 0 is for further study.

8.8.2 Remote defect indication (RDI)

As described in [IETF RFC 6428], RDI is communicated by the BFD diagnostic field in CC messages, see clause 8.8.1.

8.8.3 On-demand CV processes

An overview of the on-demand CV processes is shown in Figure 8-7.

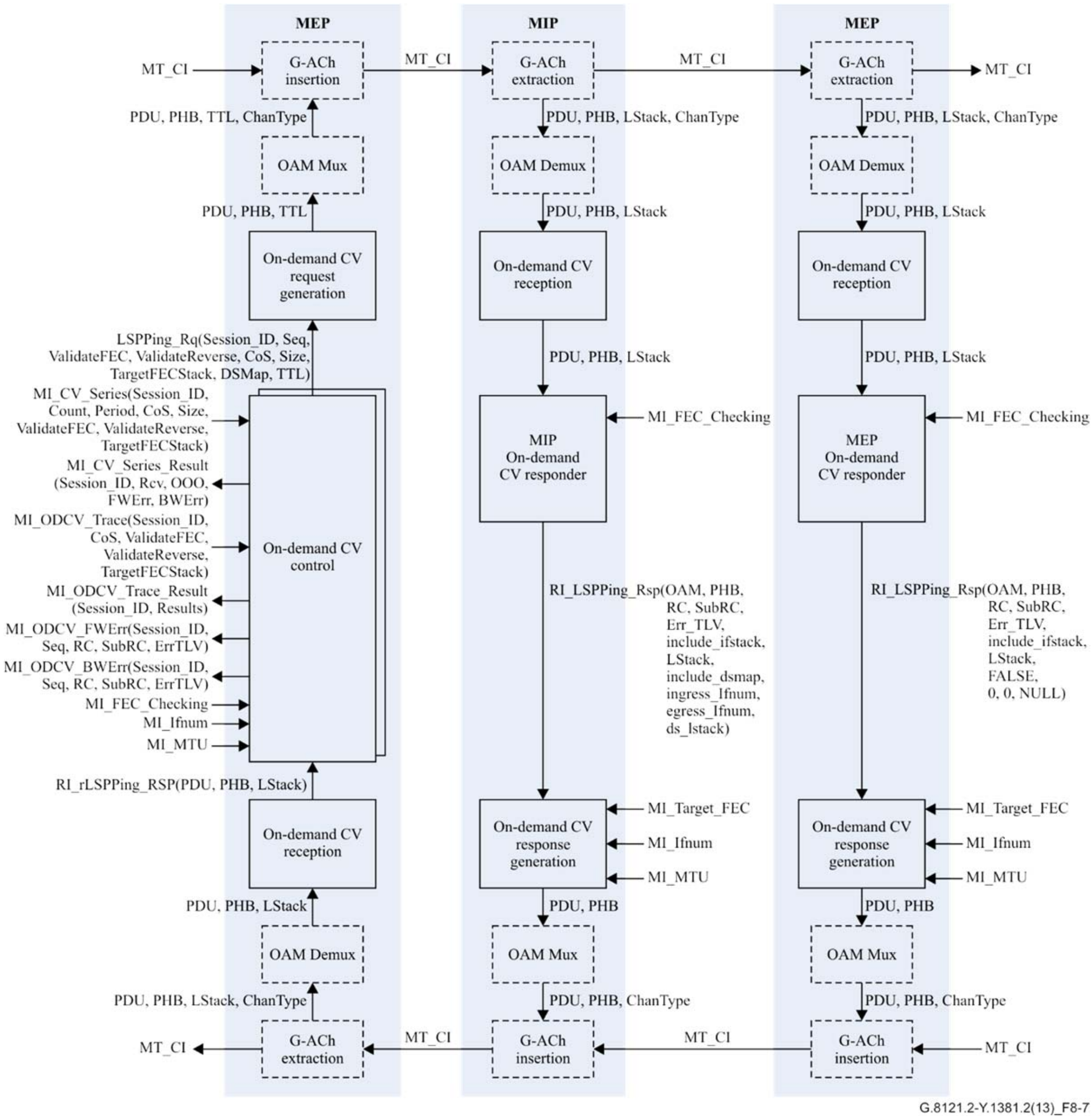

Figure 8-7 – Overview of the on-demand CV processes

The on-demand CV protocol is controlled by the on-demand CV control process. An on-demand session starts when the MI_CV_Series (Session_ID, Count, Period, CoS, Size, ValidateFEC, ValidateReverse, TargetFECStack) or MI_ODCV_Trace (Session_ID, CoS, ValidateFEC, ValidateReverse, TargetFECStack) signal is called. Multiple instances of the on-demand CV control process can be used to run multiple on-demand CV sessions concurrently, provided each instance has a different session ID.

The on-demand CV control process sends LSPPing request packets via the on-demand CV request generation process, and receives LSPPing responses via the on-demand CV reception process. Received responses may be checked for errors, if requested in the MI_CV_Series () or MI_ODCV_Trace () signal.

The on-demand CV control process reports errors in the forward direction via the MI_ODCV_FWErr(Session_ID, Seq, RC, SubRC, ErrTLV) signal, and in the backward direction via the MI_ODCV_BWErr (Session_ID, Seq, RC, SubRC, ErrTLV) signal. Results are reported via the MI_CV_Series_Result (Session_ID, Rcv, OOO, FWErr, BWErr) and MI_ODCV_Trace_Result (Session_ID, Result) signals.

The MEP on-demand CV responder and MIP on-demand CV responder processes are responsible for checking received LSPPing requests for errors, and sending responses via the on-demand CV response generation process.

The on-demand CV request generation and on-demand CV response generation processes generate LSPPing request and response packets in conformance with [IETF RFC 4379] and [IETF RFC 6426].

The MEP on-demand CV responder, MIP on-demand CV responder, and on-demand CV control processes all perform similar steps to check received packets for errors. This checking uses the copy of the original label stack that is carried as part of the MT_CI. This common validation is described further below, followed by descriptions of each of the on-demand CV processes.

8.8.3.1 Common validation

In the description below, label stacks and FEC stacks are denoted as arrays (Stack[]), where:

- Stack[1] is the bottom (innermost) label/FEC
- Stack[Count(Stack)] is the top (outermost) label/FEC
- Stack[0] is invalid

Count(Stack) returns the number of labels or FECs in the stack.

The validation is described by the following pseudocode. The values assigned to 'rc' are as described in [IETF RFC 4379].

```
ODCV_Validate (OAM, LStack_in[], FECStack[], MP_Type) {
    rc = 0
    sub_rc = 0
    err_TLV = NULL
    done = FALSE
    include_ifstack = FALSE
    include_dsmap = FALSE
    ldepth = 0
    LStack = LStack_in
    if (malformed(OAM)) {
        rc = 1
        done = TRUE
    } else if (OAM contains TLVs with types 4, 6, 8 or 10-32767) {
        rc = 2
        err_TLV = make_err_TLV(bad TLVs)
        done = TRUE
    } else {
        if (LStack[1] = GAL) {
            remove_GAL_from_LStack()
        }
        ldepth = count(LStack)
        while (!done && ldepth> 0) {
```

```
        if (!label_known(LStack[ldepth])) {
            rc = 11
            sub_rc = ldepth
            done = TRUE
        }
        ldepth--
    }
}
if (MP_Type = MEP) {
    if (!done) {
        FECdepth = 1
        L = IMPLICIT_NULL
        rc = 3
        sub_rc = 1
        if (DSMAP(OAM) != NULL && Ingress_Ifnum(DSMAP(OAM)) != 0) {
            if (DownstreamLabels(DSMAP(OAM)) != LStack) {
                rc = 5
                include_ifstack = TRUE
                done = TRUE
            }
        }
    }
    while (!done) {
        (FECstatus, FECrc) = checkFEC(FECStack[FECdepth], L)
        rc = FECrc
        sub_rc = FECdepth
        if (FECstatus = 1) {
            done = TRUE
        } else {
            FECdepth++
            if (FECdepth > count(FECStack)) {
                done = TRUE
            }
        }
        if (!done) {
            if (FECstatus = 0) {
                ldepth++
                if (ldepth > count(LStack)) {
                    done = TRUE
                } else {
                    L = LStack[ldepth]
                }
```

```
            }
        }
    }
} else {
    // MP_Type = MIP
    if (!done) {
        rc = 8
        sub_rc = 1
        if (DSMAP(OAM) != NULL) {
            if (Ingress_Ifnum(DSMAP(OAM)) = 0) {
                rc = 6
                include_ifstack = TRUE
            } else {
                if (DownstreamLabels(DSMAP(OAM)) != LStack) {
                    rc = 5
                    include_ifstack = TRUE
                    done = TRUE
                }
            }
        }
    }
    if (!done) {
        Egress_Ifnum = get_egress_interface()
        if (Egress_Ifnum = 0) {
            rc = 9
            done = TRUE
        }
    }
    if (!done) {
        if (DSMAP(OAM) != NULL) {
            include_dsmap = TRUE
        } else {
            done = TRUE
        }
    }
    if (!done) {
        if (V(OAM) == 0 && MI_FEC_Checking = 0) {
            done = TRUE
        }
    }
    if (!done) {
        FECdepth = 0
```

```
            i = 1
            while (i > 0) {
                FECdepth++
                if (DownstreamLabels(DSMAP(OAM))[FECdepth] != IMPLICIT_NULL) {
                   i--
                }
            }
            if (count(FECStack) >= FECdepth) {
                (FECstatus, FECrc) = checkFEC(FECStack[FECdepth], LStack[1])
                if (FECstatus = 2) {
                    rc = 10
                } else if (FECstatus = 1) {
                    rc = FECrc
                    sub_rc = FECdepth
                }
            }
        }
    }
    return(rc, sub_rc, err_TLV, include_ifstack, include_dsmap)
}
```

The utility functions used in the pseudocode above are described below:

- malformed(OAM) checks that the packet is in accordance with the format described in [IETF RFC 4379] and [IETF RFC 6426]. It also checks that:
  - If the packet is a request, it contains a target FEC stack TLV.
  - If the packet is a reply and the R flag is set, it contains a reverse target FEC stack TLV
  - The target FEC stack or reverse target FEC stack TLVs contain only subtypes 'Static LSP', 'Static Pseudowire' and 'Nil FEC'. Use of other subtypes are for further study.
  - If the packet contains a downstream mapping TLV, the address type is 'Non-IP'. Use of other address types is for further study.
- make_err_TLV(TLVs) creates an 'Errored TLVs' TLV according to [IETF RFC 4379] and copies the bad TLVs into it.
- remove_GAL_from_LStack removes the GAL from the bottom of the label stack, so that LStack[1] now refers to the label that immediately preceded the GAL.
- label_known(Label) checks whether the Label value is known and can be processed.
- checkFEC(FEC, Label) implements the FEC checking procedure described in section 4.4.1 of [IETF RFC 4379].
- get_egress_interface() returns MI_Ifnum if this is the egress interface, otherwise it uses forwarding information to find the egress interface and returns its interface number, or 0 if no egress interface was found or it is not MPLS-enabled.

8.8.3.2 On-demand CV control process

The on-demand CV control process operates the LSPPing on-demand CV protocol. An LSPPing session is started by the MI_CV_Series(Session_ID, Count, Period, CoS, Size, ValidateFEC, ValidateReverse, TargetFECStack) or MI_ODCV_Trace(Session_ID, CoS, ValidateFEC, ValidateReverse, TargetFECStack) signals. In either case, a session ID is supplied; multiple instances of the on-demand CV control process can be created, provided each has a unique session ID.

The target FEC stack to be checked by the peer device is specified in the MI_CV_Series() or MI_ODCV_Trace() signal. Other mechanisms for deriving the target FEC stack, for example if dynamic signalling protocols are in use, are for further study. The target FEC stack passed in the MI_CV_Series() or MI_ODCV_Trace() signals must only contain FECs with subtypes 'Static LSP', 'Static Pseudowire' or 'Nil FEC'.

Results are reported by the on-demand CV control process using the MI_CV_Series_Result (Session_ID, Rcv, OOO, FWErr, BWErr) or MI_ODCV_Trace_Result(Session_ID, Result) signals when the session ends. In addition, any errors detected while the session is running are reported by using the MI_ODCV_FWErr(Session_ID, Seq, RC, SubRC, ErrTLV) signal (for errors in the Control-to-Responder direction) or MI_ODCV_BWErr(Session_ID, Seq, RC, SubRC, ErrTLV) signal (for errors in the Responder-to-Control direction). Note that errors in the Responder-to-Control direction are only detected if ValidateReverse is set to TRUE in the MI_CV_Series() or MI_ODCV_Trace() signal.

The behaviour of the on-demand CV control process is shown in the figures below (Figures 8-8a, 8-8b, 8-8c and 8-8d). In PingRunning state, the process sends LSPPing requests periodically, and handles any received replies by counting them and checking for any errors. In TraceRunning state, an initial LSPPing request is sent with TTL 1, so that it is intercepted by the first MIP (or MEP) reached. When a response is received, it is first checked for any errors. Then, if the response was from an MIP (i.e., it contains a DSMap TLV), the TTL is incremented and a new LSPPing request is sent. Incrementing the TTL ensures the request is intercepted by the next MIP (or MEP). If no response is received, three attempts are made to resend the request, before giving up and reporting any results collected so far.

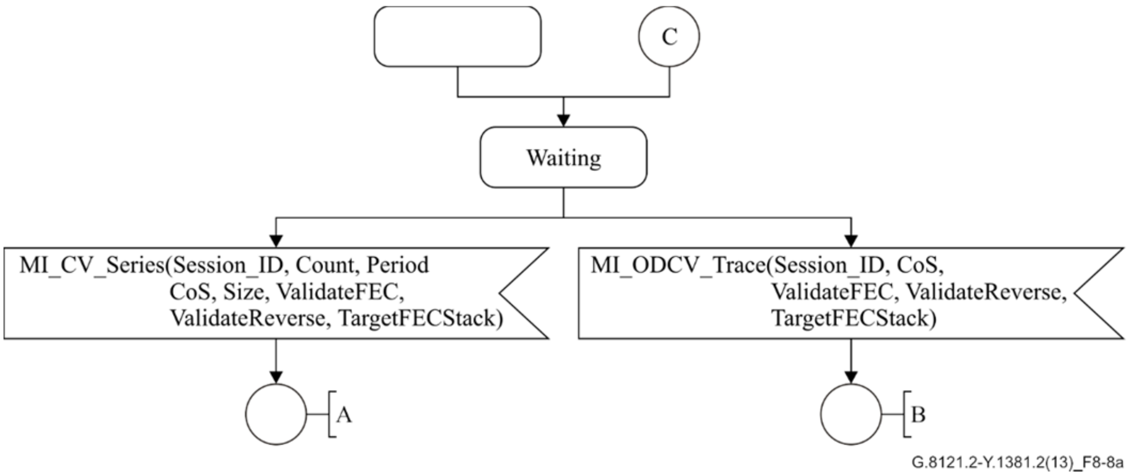

Figure 8-8a – On-demand CV control process

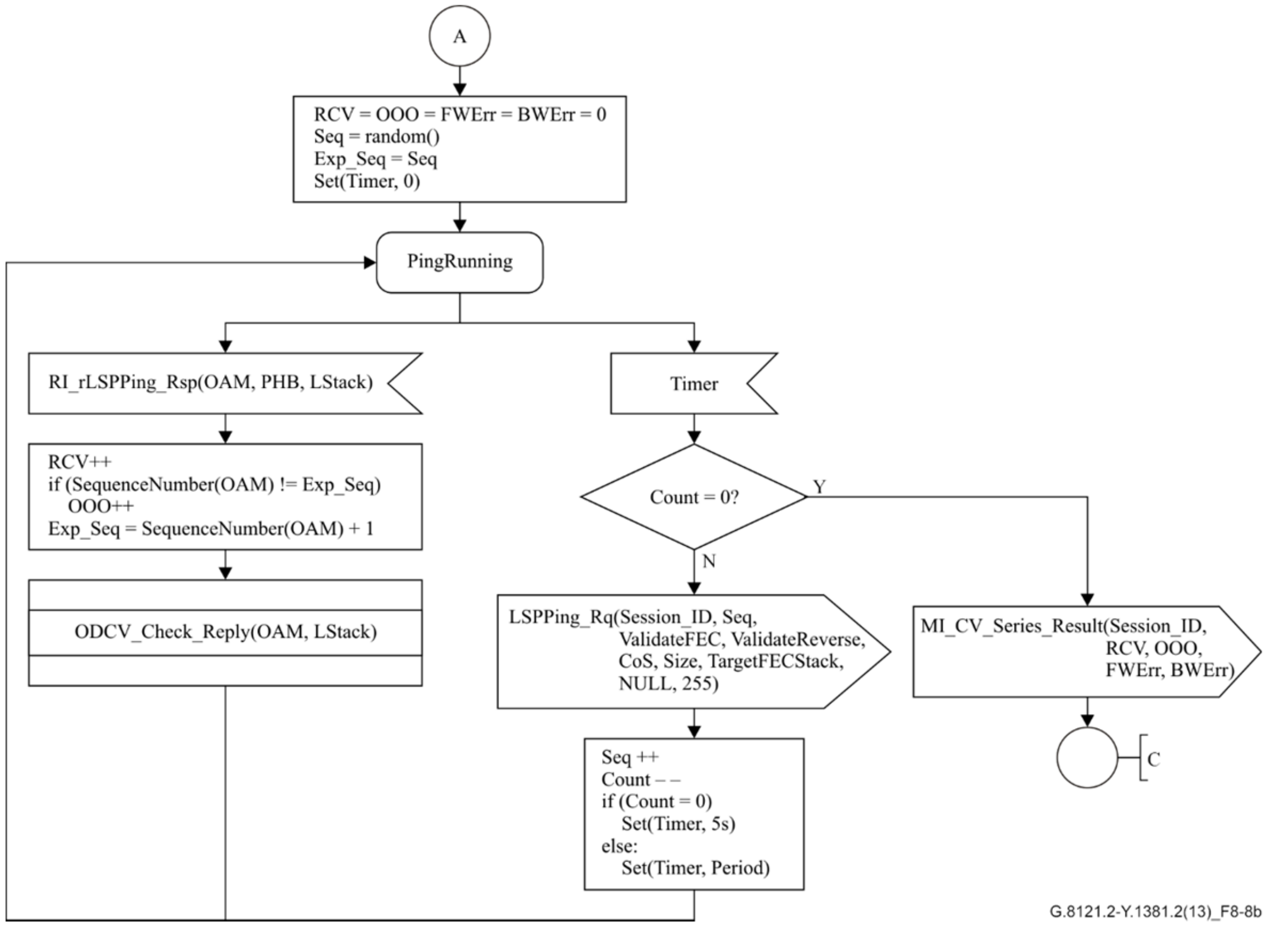

Figure 8-8b – On-demand CV control process

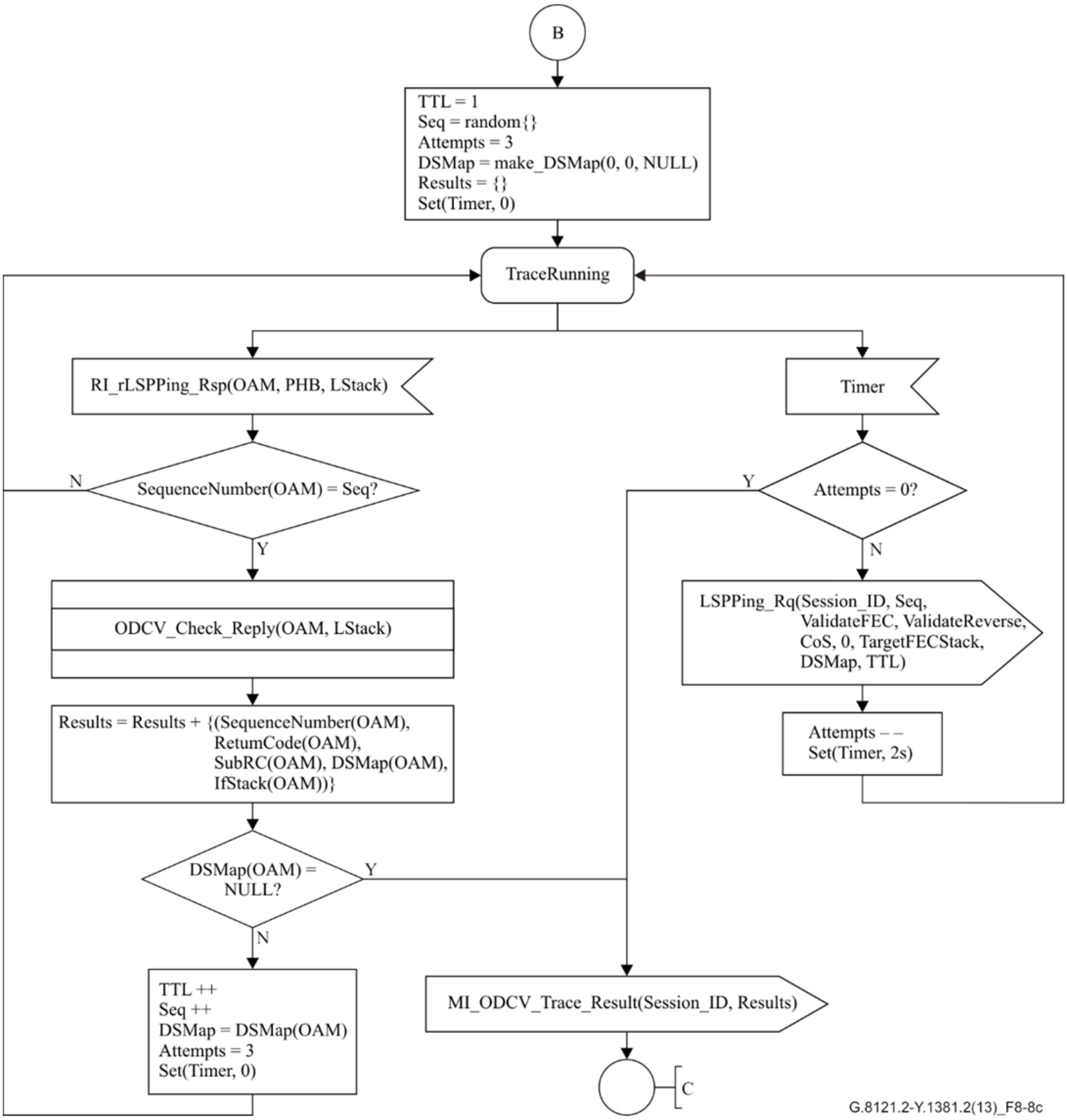

Figure 8-8c – On-demand CV control process

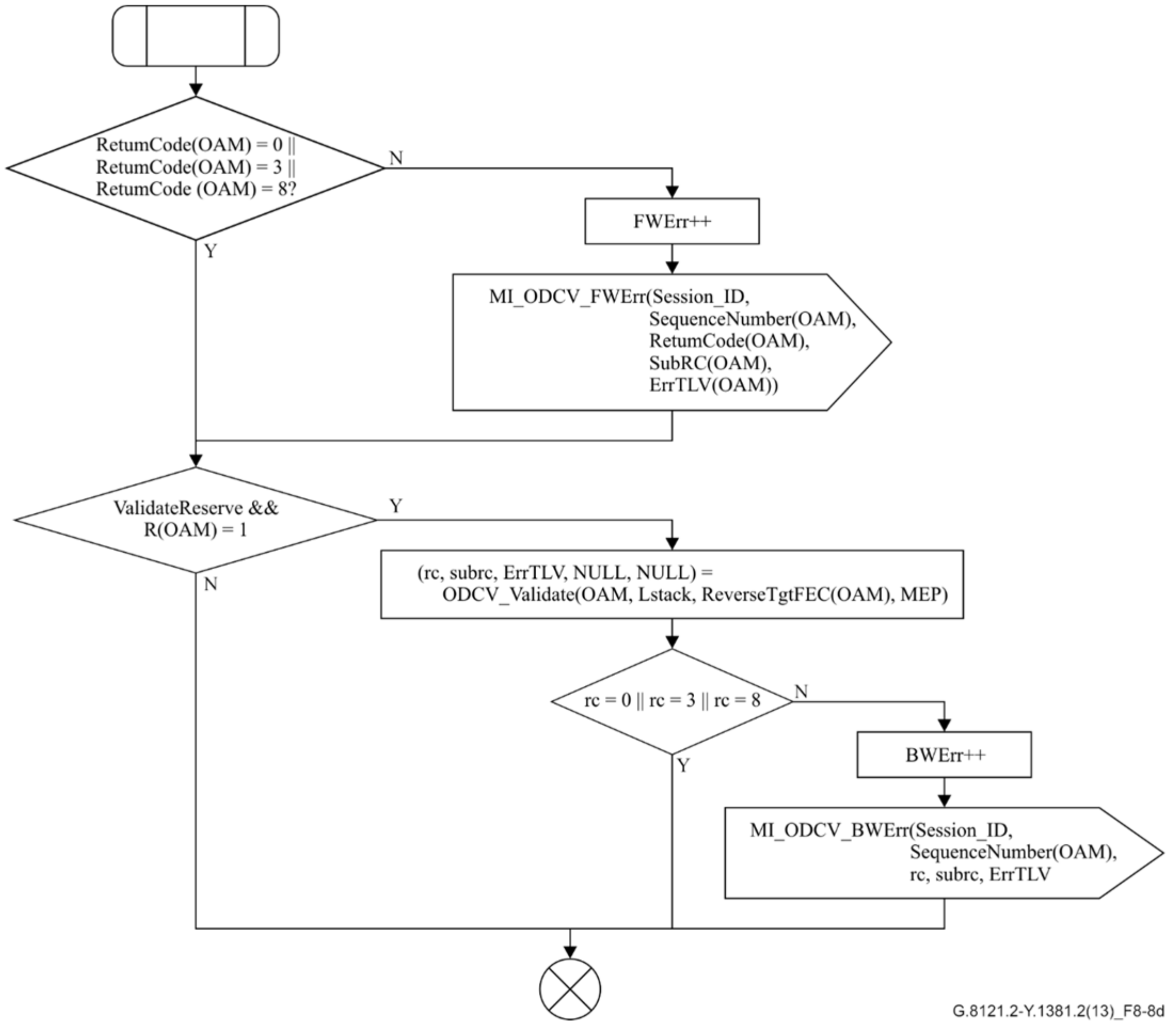

Figure 8-8d – On-demand CV control process

The make_DSMap(ingress_ifnum, egress_ifnum, ds_lstack) function creates a downstream mapping TLV according to [IETF RFC 4379] and [IETF RFC 6426]. The fields are filled in as follows:

- MTU: Set to MI_MTU.
- Address Type: set to 5 (Non IP). Use of other address types is for further study.
- DS Flags: The I flag is set to 1, all other flags are set to 0.
- Ingress Ifnum: set to ingress_ifnum.
- Egress Ifnum: set to egress_ifnum.
- Multipath Type: set to 0 (no Multipath).
- Depth Limit: set to 0.
- Multipath Length: set to 0.
- Downstream Labels: derived from ds_lstack as described in [IETF RFC 4379]. The protocol is set to 1 (Static). Use of other values is for further study.

8.8.3.3 On-demand CV request generation process

The on-demand CV request generation process is shown in Figure 8-9.

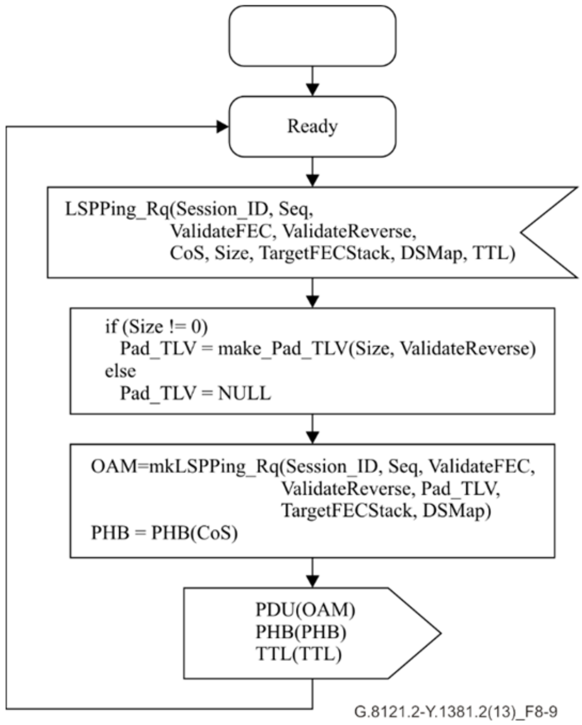

Figure 8-9 – On-demand CV request generation process

The make_Pad_TLV(Size) function creates a "Pad" TLV in accordance with [IETF RFC 4379]. The Length field is set to "Size". The first octet of the value field is set to 2 (Copy Pad TLV) if ValidateReverse is FALSE, and 1 (Drop Pad TLV) if ValidateReverse is TRUE.

The PHB(CoS) function returns the PHB with the lowest drop precedence within the class of service defined by the CoS input parameter.

NOTE – Size is only non-zero in Ping mode when no DSMap TLV is included. In this case, the responder will not add any additional TLVs (e.g., an interface and label stack TLV) to the reply unless the 'R' (ValidateReverse) flag is set, and so the pad TLV can be safely copied into the reply.

The mkLSPPing_Rq function creates an LSPPing echo request packet in accordance with [IETF RFC 4379] and [IETF RFC 6426]. The fields are filled in as follows:

- Version Number: set to 1.
- Global Flags: if ValidateFEC is TRUE, the V flag is set to 1; if ValidateReverse is TRUE, the R flag is set to 1; all other flags are set to 0.
- Message Type: set to MPLS echo request.
- Reply Mode: set to 4 (reply via application control channel).
- Return Code: set to 0.
- Return Subcode: set to 0.
- Sender's Handle: set to the value of Session_ID.
- Sequence Number: set to the value of Seq.
- Timestamp Sent: set to LocalTime.
- Timestamp Received: set to 0.

The following TLVs are added:

- A target FEC stack TLV is added containing the contents of TargetFECStack.
- If Pad_TLV is not NULL, a pad TLV is added containing the contents of Pad_TLV.

• If DSMap is not NULL, a downstream mapping TLV is added containing the contents of DSMap.

8.8.3.4 On-demand CV reception process

The on-demand CV reception process demultiplexes received LSPPing packets (formed of OAM, PHB, LStack signals) as follows:

• If the message type is MPLS echo request, the received OAM, PHB, and LStack signals are passed to the MIP on-demand CV responder or MEP on-demand CV responder process as appropriate.

• Otherwise, if this is an MIP the packet is discarded.

• If this is an MEP and the message type is MPLS echo reply, the on-demand CV reception process passes the received OAM, PHB, and LStack signals to the instance of the on-demand CV control process whose session ID is equal to the "Sender's handle" in the received packet, via RI_rLSPPing_Rsp(OAM, PHB, LStack). If there is no such instance of the on-demand CV control process, the packet is discarded.

8.8.3.5 MIP on-demand CV responder process

The MIP on-demand CV responder process is described in Figure 8-10.

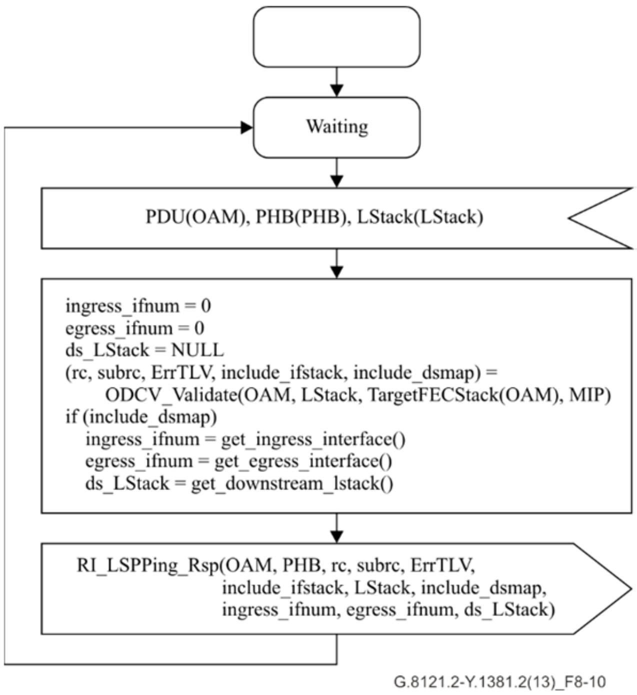

Figure 8-10 – MIP on-demand CV responder process

The get_egress_interface() function is described in clause 8.8.3.1.

The get_ingress_interface() function returns MI_Ifnum if this is the ingress interface, otherwise it returns the interface number of the interface where the packet arrived.

The get_downsteam_lstack() returns the label stack that would be attached to the packet if it were to be forwarded out of the egress interface, derived as described in [IETF RFC 4379].

8.8.3.6 MEP on-demand CV responder process

The MEP on-demand CV responder process is described in Figure 8-11.

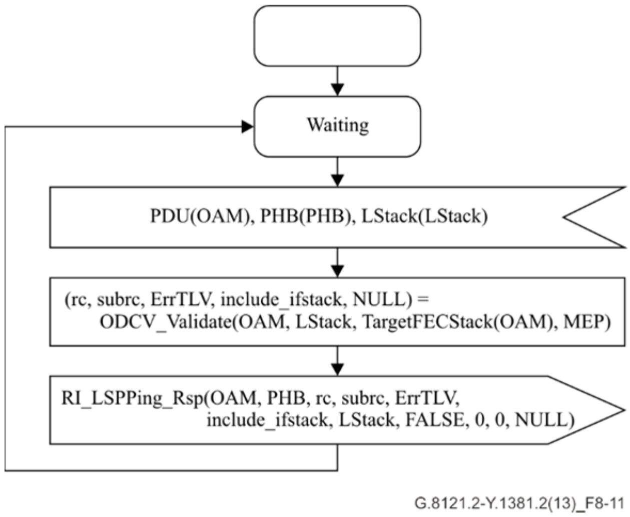

Figure 8-11 – MEP on-demand CV responder process

8.8.3.7 On-demand CV response generation process

The on-demand CV response generation process is shown in Figure 8-12.

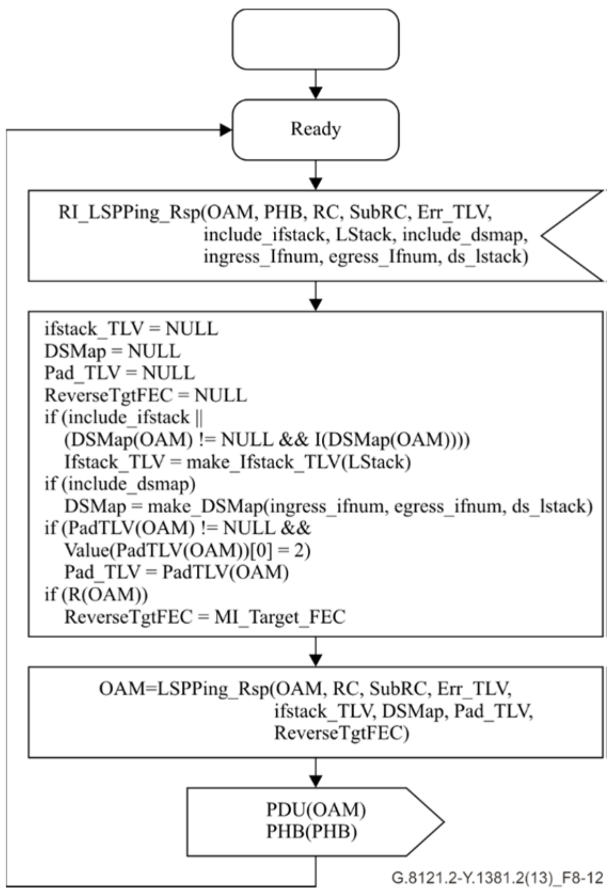

Figure 8-12 – On-demand CV response generation process

The make_Ifstack_TLV(LStack) function creates an interface and label stack TLV according to [IETF RFC 4379] and [IETF RFC 6426]. The fields are filled in as follows:

- Address Type: set to IPv4 Unnumbered.
- IP Address: set to 0.
- Interface: set to MI_Ifnum.

- Label Stack: Copied from LStack.

Use of other values in the interface and label stack TLV is for further study.

The make_DSMap(ingress_ifnum, egress_ifnum, ds_lstack) function is described in clause 8.8.3.2.

The LSPPing_Rsp function creates an LSPPing echo reply packet in accordance with [IETF RFC 4379] and [IETF RFC 6426]. The fields are filled in as follows:

- Version Number: set to 1.
- Global Flags: copied from the received echo request.
- Message Type: set to MPLS echo reply.
- Reply Mode: set to 0 (do not reply).
- Return Code: set to RC.
- Return Subcode: set to SubRC.
- Sender's Handle: copied from the received echo request.
- Sequence Number: copied from the received echo request.
- Timestamp Sent: copied from the received echo request.
- Timestamp Received: set to LocalTime.

If reverse FEC checking was requested in the LSPPing request (i.e., the R flag was set), a reverse target FEC stack is created based on MI_Target_FEC. Other mechanisms for deriving the FEC stack, for example, if dynamic signalling protocols are in use, are for further study.

The following TLVs are added:

- The TargetFECStack TLV is copied from the received packet.
- If Err_TLV is not NULL, an errored TLVs TLV is added containing the contents of Err_TLV.
- If Ifstack_TLV is not NULL, an interface and label stack TLV is added containing the contents of Ifstack_TLV.
- If DSMap is not NULL, a downstream mapping TLV is added containing the contents of DSMap.
- If Pad_TLV is not NULL, a pad TLV is added containing the contents of Pad_TLV.
- If ReverseTgtFEC is not NULL, a reverse-path target FEC stack TLV is added containing the contents of ReverseTgtFEC.

8.8.4 Proactive packet loss measurement (LMp)

As described in clauses 7.2.2.1 and 8.6 to 8.8 of [ITU-T G.8113.2], loss and delay measurements may be combined. The format for the combined measurement, referred to here as LMDM, is described in section 3.3 of [IETF RFC 6374]. In addition, the same LM and DM protocols can be used for both proactive and on-demand measurement.

An overview of the performance monitoring processes for a single proactive PM session is shown in Figure 8-13. The same processes are used for LM, DM or LMDM.

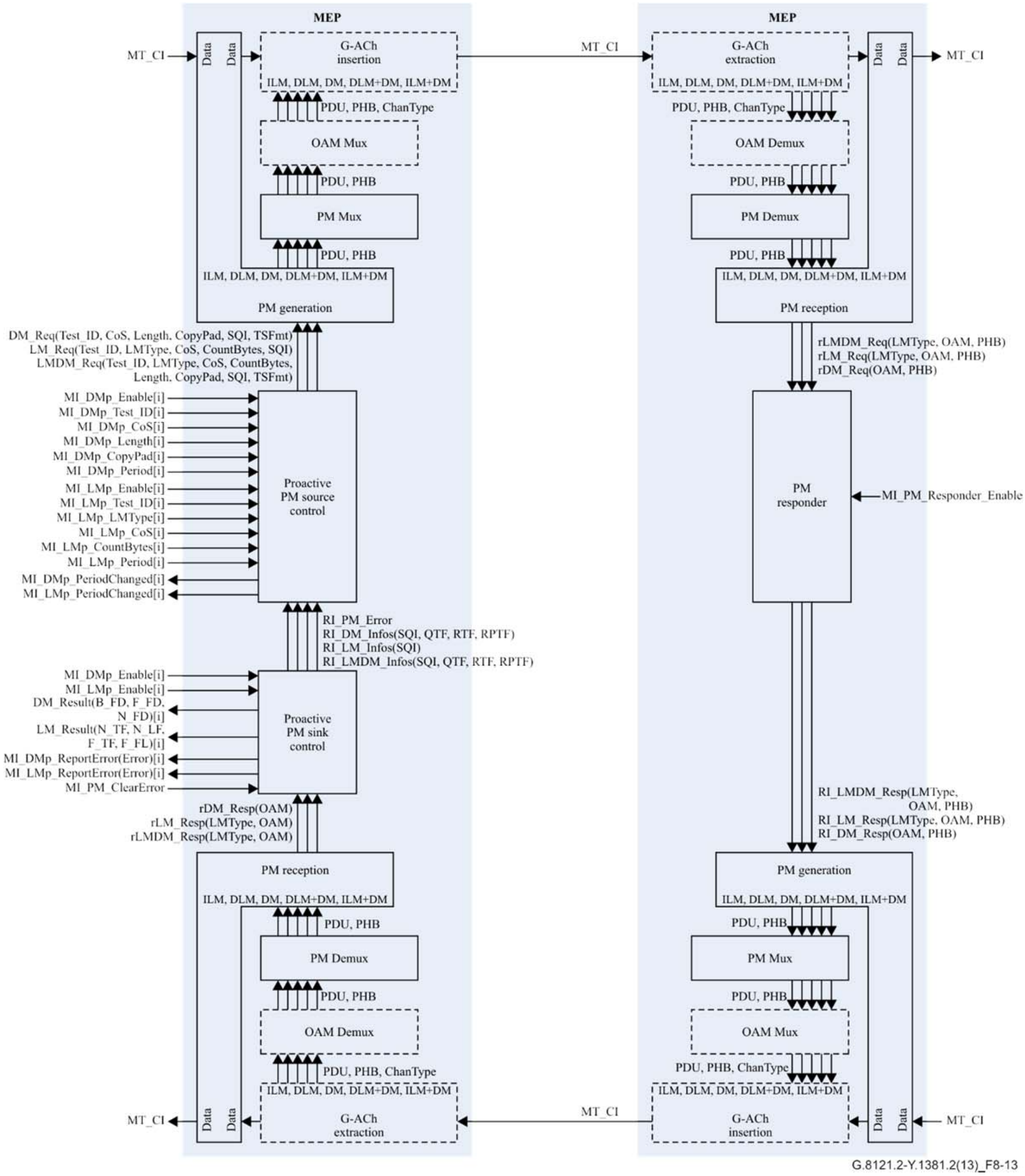

Figure 8-13 – Proactive PM processes

The proactive PM source control process controls the session, including scheduling request packets; the proactive PM sink control process handles processing responses to calculate performance metrics.

The PM generation process generates requests and responses for the five different types of PM PDUs: ILM, DLM, DM, ILM+DM and DLM+DM. It also counts data traffic (including test packets) and is responsible for writing the counters and/or timestamps into the outgoing PM PDUs. The location of the counter part is shown in the figure above for illustration only; the exact set of packets to be counted is implementation-specific, as described in [IETF RFC 6374].

The PM reception process handles received requests and responses. Like the PM generation process, it counts the appropriate packets and writes the counters and/or timestamps into the received PM PDUs. Again, the location of the counter part is shown in the figure above for illustration only; the exact set of packets to be counted is implementation-specific, as described in [IETF RFC 6374].

The PM responder is responsible for replying to received PM request packets.

Multiple PM sessions can be used simultaneously, by instantiating multiple instances of the PM source control, PM sink control, PM reception, PM generation and PM responder processes. Each instance of these processes supports a single PM session. Each PM session (proactive or on-demand) must have a unique test ID. For each test ID, a pair of control processes (i.e., source and sink) is associated with a corresponding instance of the PM reception and PM generation processes. Similarly, the responder process for a given session is associated with a corresponding instance of the PM reception and PM generation processes. The PM Mux process multiplexes PM packets for different sessions, while the PM Demux process demultiplexes them based on the test ID (session ID) and R (response) flag.

Note therefore that a given instance of the PM reception process is associated with exactly one other process to which it passes received packets. Depending on how it is instantiated, this could be the proactive PM sink control process, the on-demand PM control process (see clause 8.8.5), or the PM responder process.

8.8.4.1 Proactive PM source control process for LM

The proactive PM source control process includes LM, DM and LMDM. Each instance of the process operates a single proactive PM session. Multiple sessions can be supported by instantiating multiple instances of the process, along with corresponding instances of the PM sink control, PM generation and PM reception processes.

The proactive PM source control process performs delay measurements when MI_DMp_Enable is true and performs loss measurements when MI_LMp_Enable is true. If both are enabled, then where possible, the same PDUs are used to make both measurements (i.e., ILM+DM or DLM+DM PDUs). Otherwise, separate PDUs are used for loss (ILM or DLM) and delay (DM). The type of PDU used for loss is determined by MI_LMp_LMType, and can be "ILM" for inferred (synthetic) loss or "DLM" for direct (data traffic) loss measurement.

If an error is detected while the session is running, this is signalled via RI_PM_Error being set to true and the session is stopped until RI_PM_Error is set to false or the session is disabled.

The PM protocol includes a mechanism to negotiate the packet sending period with the responder. If the period is changed from that specified by the management information (MI_DMp_Period or MI_LMp_Period), this is signalled via MI_DMp_PeriodChanged or MI_LMp_PeriodChanged.

MI_LMp_CoS and MI_DMp_CoS specify the CoS (traffic class) to use for the measurement. In the case of MI_LMp_CoS, this can either be a specific value, or the special value "ALL" indicating that loss across all traffic classes should be measured.

The proactive PM control process is described in Figures 8-14a, 8-14b and 8-14c. These figures include LM, DM and LMDM.

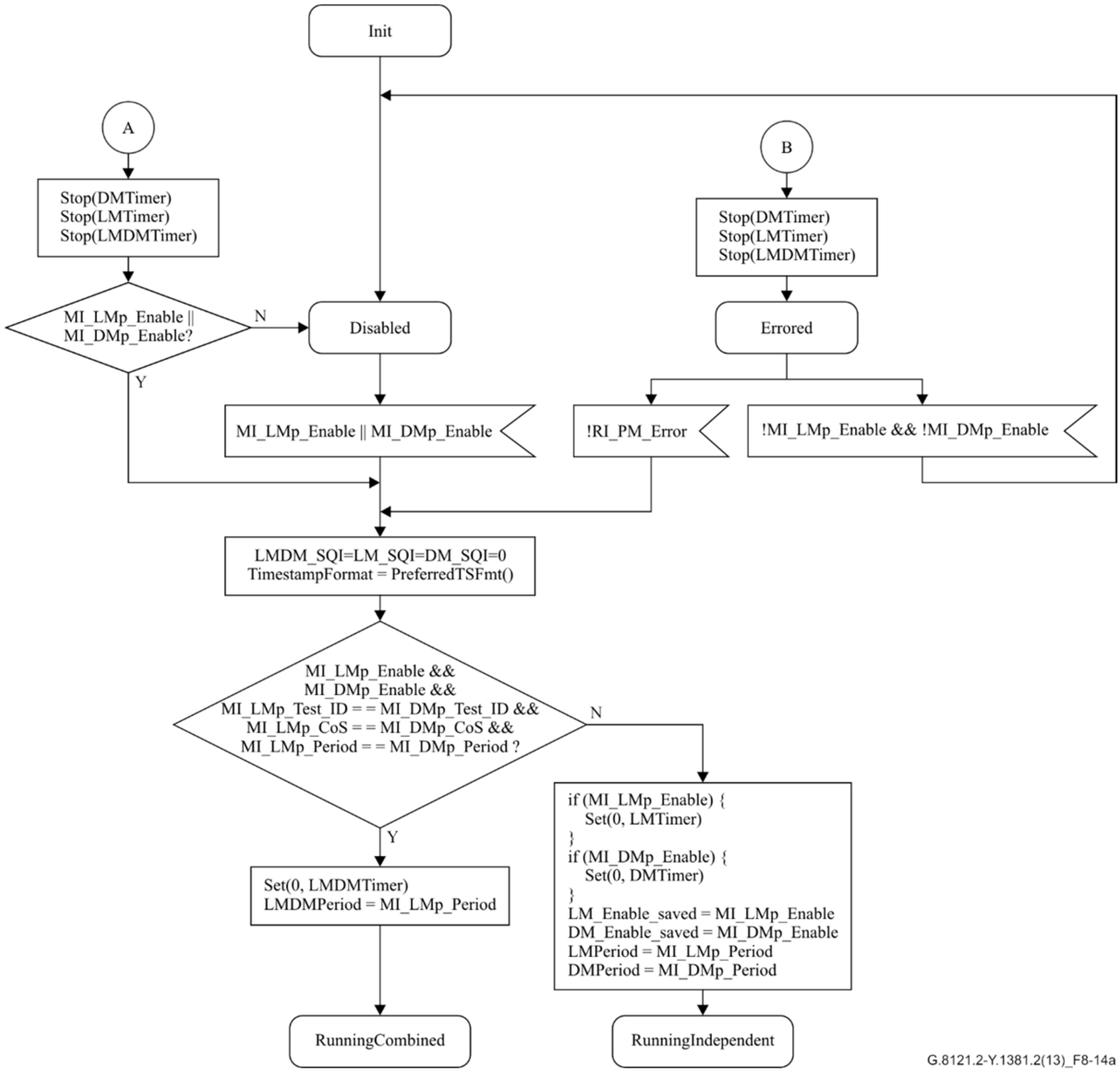

Figure 8-14a – Proactive PM source control process

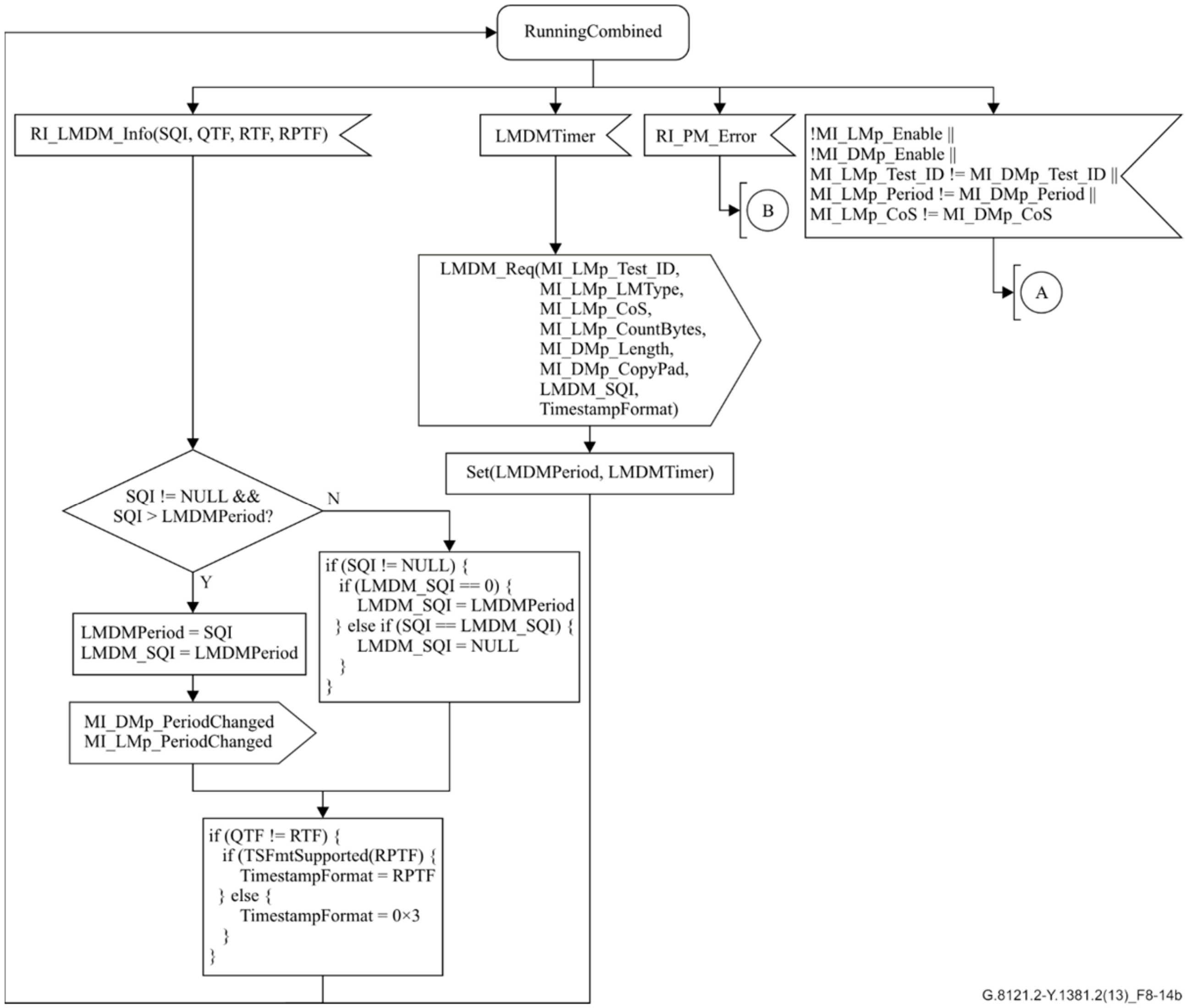

Figure 8-14b – Proactive PM source control process

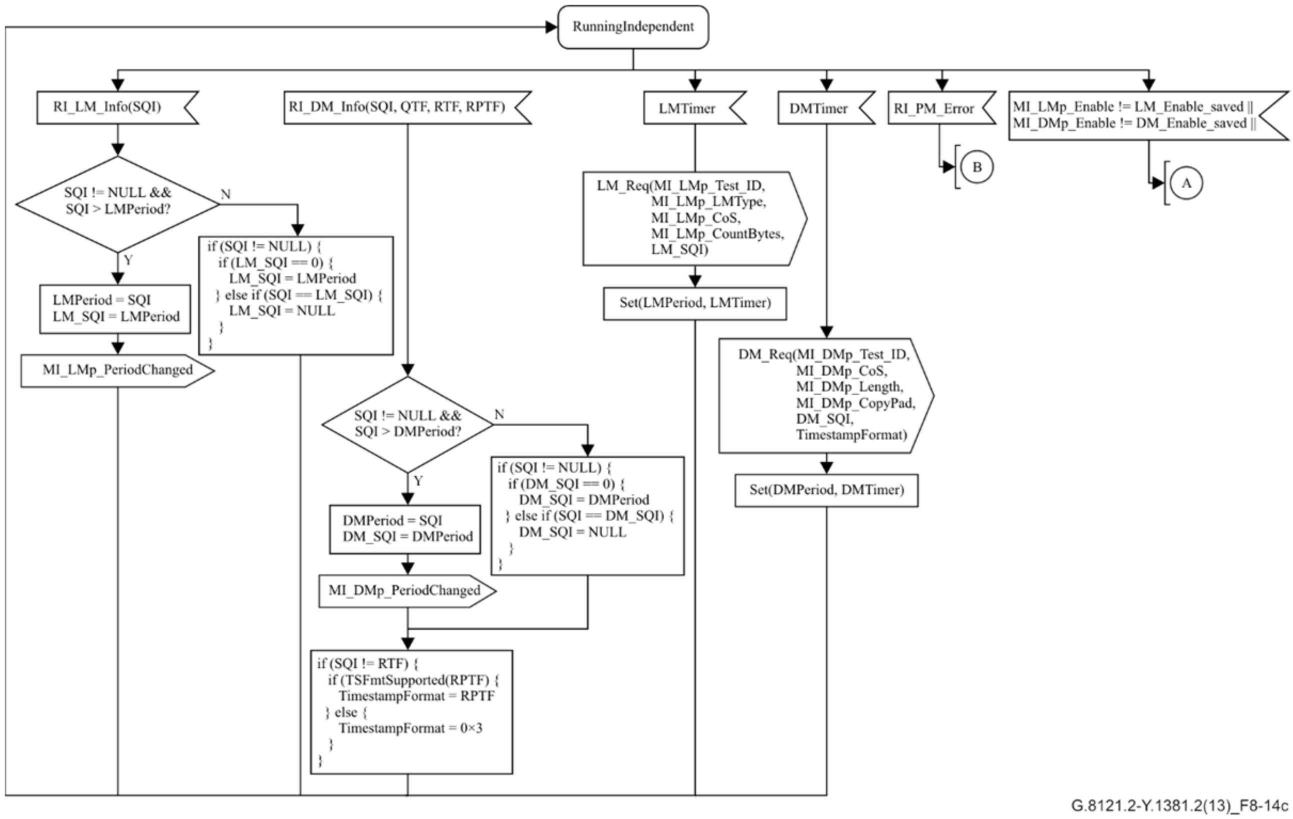

Figure 8-14c – Proactive PM source control process

The TSFmtSupported() function determines whether the specified timestamp format, from [IETF RFC 6374], is supported by the implementation, while the PreferredTSFmt() function returns the timestamp format that is preferred by the implementation, as described in [IETF RFC 6374].

Note that both the period and the timestamp format are negotiated with the responder. The period is negotiated by setting the SQI appropriately, while the timestamp format is negotiated via the QTF, RTF and RPTF fields. Initially, the implementation's preferred timestamp is used. If the responder does not respond to the first request using the same timestamp format, then the responder's preferred timestamp format is used if it is supported, otherwise the IEEE 1588v1 format is used as described in [IETF RFC 6374]. Note that support for this format is mandatory.

8.8.4.2 Proactive PM sink control process for LM

The proactive PM sink control process includes LM, DM and LMDM. Each instance of the process operates a single proactive PM session. Multiple sessions can be supported by instantiating multiple instances of the process, along with corresponding instances of the PM source control, PM generation and PM reception processes.

As for the source control process, the proactive PM sink control process performs delay measurements when MI_DMp_Enable is true and performs loss measurements when MI_LMp_Enable is true. If both are enabled, then where possible, the same PDUs are used to make both measurements (i.e., ILM+DM or DLM+DM PDUs). Otherwise, separate PDUs are used for loss (ILM or DLM) and delay (DM).

If an error is detected while the session is running, this is reported via MI_DMp_ReportError or MI_LMp_ReportError, and the session is stopped until MI_PM_ClearError is set or the session is disabled.

The proactive PM sink control process is described in Figures 8-15a, 8-15b and 8-15c. These figures include LM, DM and LMDM.

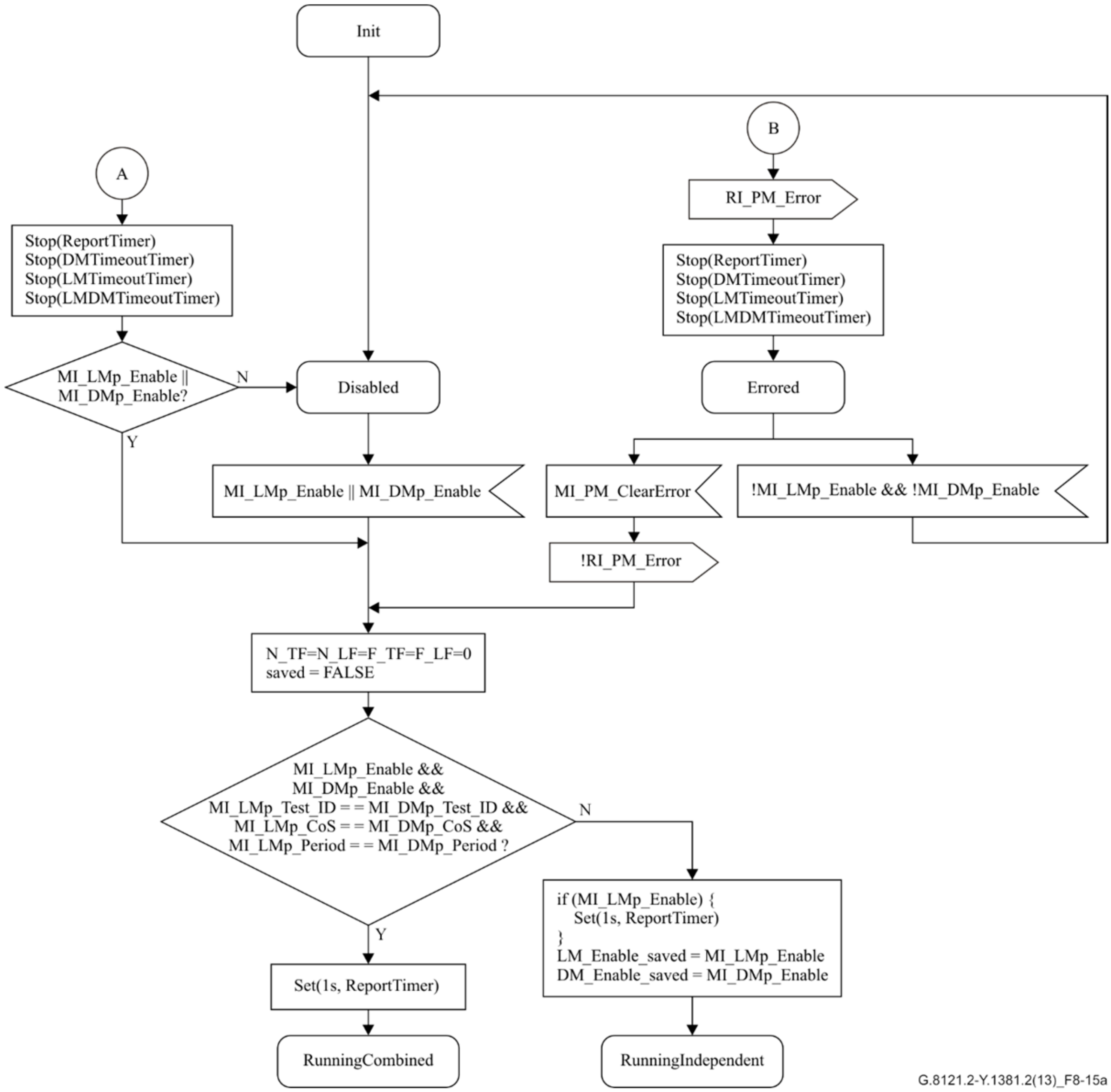

Figure 8-15a – Proactive PM sink control process

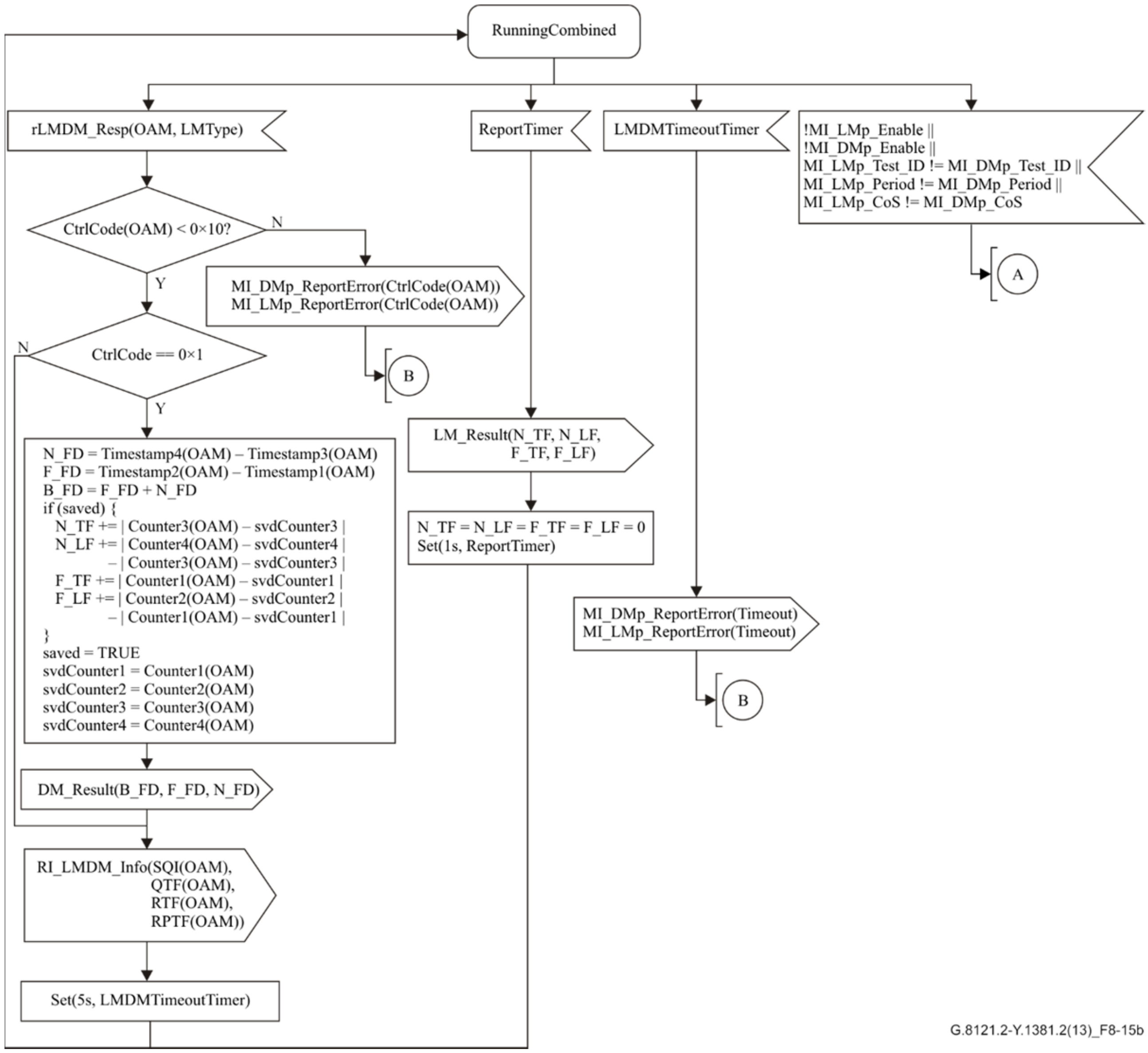

Figure 8-15b – Proactive PM sink control process

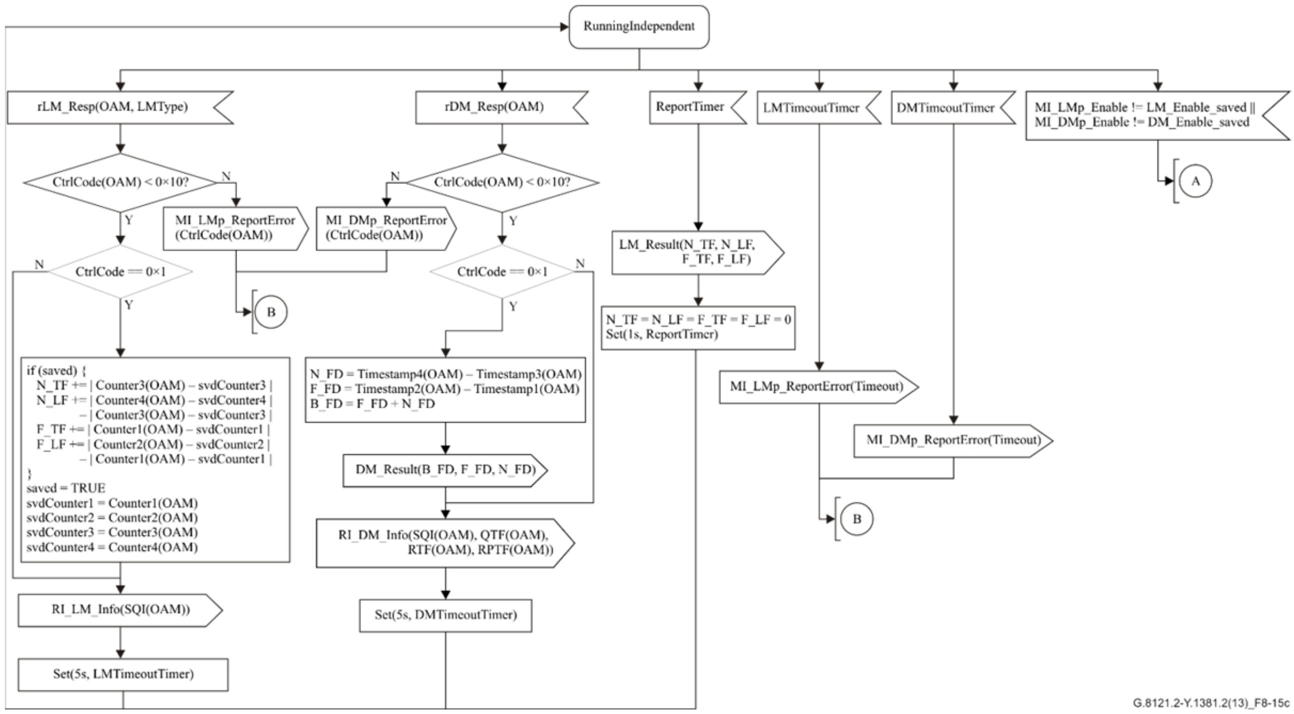

Figure 8-15c – Proactive PM sink control process

8.8.4.3 PM generation process for proactive LM

The PM generation process includes LM, DM and LMDM. It generates PM requests when it receives the DM_Req, LM_Req or LMDM_Req signals from the corresponding proactive source or on-demand control process, and generates PM responses when it receives the RI_DM_Resp, RI_LM_Resp or RI_LMDM_Resp signals from the corresponding PM responder process.

For delay measurement, it writes the packet send time into the PDU, using the requested timestamp format.

For loss measurement, it counts the appropriate traffic depending on the type of loss measurement, and writes the counters into the transmitted PM PDUs. The packets to count are dependent on the LMType (ILM or DLM) and the CoS (which may be a particular value, or the special value "ALL"). If the CountBytes parameter is set, the number of bytes in each matching packet is counted, otherwise the count is simply incremented for each matching packet.

In the DM_Req, LM_Req and LMDM_Req signals, the SQI parameter specifies the value to place in the SQI TLV. If it is set to NULL, no SQI TLV is included. The TSFmt parameter specifies the timestamp format to use when writing timestamps. The Length parameter specifies the length of padding to include in the PDU. If set to 0, no padding TLV is included.

The PM generation process is described in Figure 8-16.

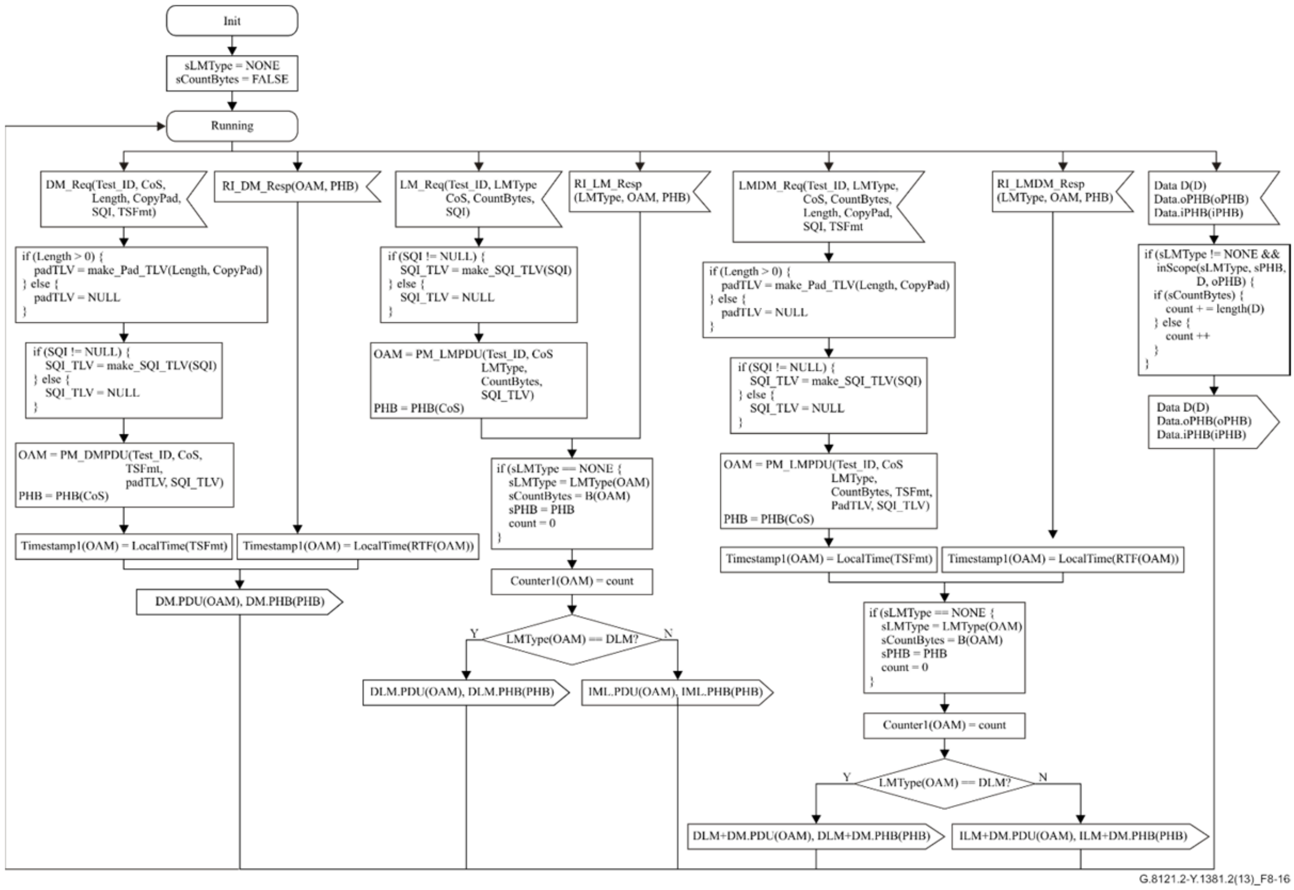

Figure 8-16 – PM generation process

The make_Pad_TLV(Length, CopyPad) function creates a padding TLV as specified in [IETF RFC 6374], as follows:

- If CopyPad is set, the type is set to 0, otherwise it is set to 128.
- The Length field is set to Length
- The Value field is set to all 0s.

The make_SQI_TLV(SQI) function creates an SQI TLV as specified in [IETF RFC 6374], as follows:

- The type is set to 2.
- The Length field is set to 4.
- The Value field is set to SQI.

The PM_DMPDU(Test_ID, TSFmt, CoS, padTLV, SQI_TLV) function creates a DM PDU as specified in [IETF RFC 6374], as follows:

- The version is set to 0.
- The R flag is unset; the T flag is set; and the rest of the flags field is set to 0.
- The control code is set to 0. Other values for the control code are for further study.
- The message length is set to the total length of the PDU.
- The QTF field is set to TSFmt, the RTF and RPTF fields are set to 0.
- The reserved field is set to 0.
- The session ID and DS fields are set to Test_ID and CoS respectively.
- The timestamp fields are all set to 0.

- The pad TLV and SQI TLV, if not NULL, are appended to the message. The use of other TLVs is for further study.

The PM_LMPDU() function creates an ILM or DLM PDU as specified in [IETF RFC 6374], as follows:

- The version is set to 0.
- The R flag is unset; the T flag is set if a specific CoS value has been specified and is unset if the CoS is set to "ALL"; and the rest of the flags field is set to 0.
- The control code is set to 0. Other values for the control code are for further study.
- The message length is set to the total length of the PDU.
- In the Dflag field, the X flag is set appropriately depending on whether the implementation writes 32 or 64 bit counters; the B flag is set if CountBytes is set, and is unset otherwise; and the rest of the field is set to 0.
- The OTF field is set to the implementations preferred timestamp format.
- The reserved field is set to 0.
- The session ID and DS fields are set to Test_ID and CoS respectively. If the CoS is "ALL", the DS field is set to 0.
- The origin timestamp field is set to the local time-of-day, using the format specified in the OTF field.
- The counter fields are all set to 0.
- The SQI TLV, if not NULL, is appended to the message. The use of other TLVs is for further study.

The PHB(CoS) function returns the PHB with the lowest drop precedence within the class of service defined by the CoS input parameter.

The PM_LMDMPDU() function creates an ILM+DM or DLM_DM PDU as specified in [IETF RFC 6374], in a similar way to the DM and LM cases described above.

The InScope() function determines whether a given data packet should be counted, depending on the LM Type (ILM or DLM) and the CoS/PHB (a specific TC value or "ALL").

The LocalTime(TSFmt) function returns the local time-of-day, in the format specified.

8.8.4.4 PM reception process for proactive LM

The PM reception process receives PM messages for a given Test ID, and passes them to the corresponding proactive or on-demand control process or PM responder process.

For delay measurement, it writes the packet receive time into the PDU. For loss measurement, it counts the appropriate traffic depending on the type of loss measurement, and writes the counters into the received PM PDUs. The packets to count are dependent on the LMType (ILM or DLM) and the CoS (which may be a particular value, or the special value "ALL"). If the CountBytes bit is set, the number of bytes in each matching packet is counted, otherwise the count is simply incremented for each matching packet.

The PM reception process is described in Figure 8-17.

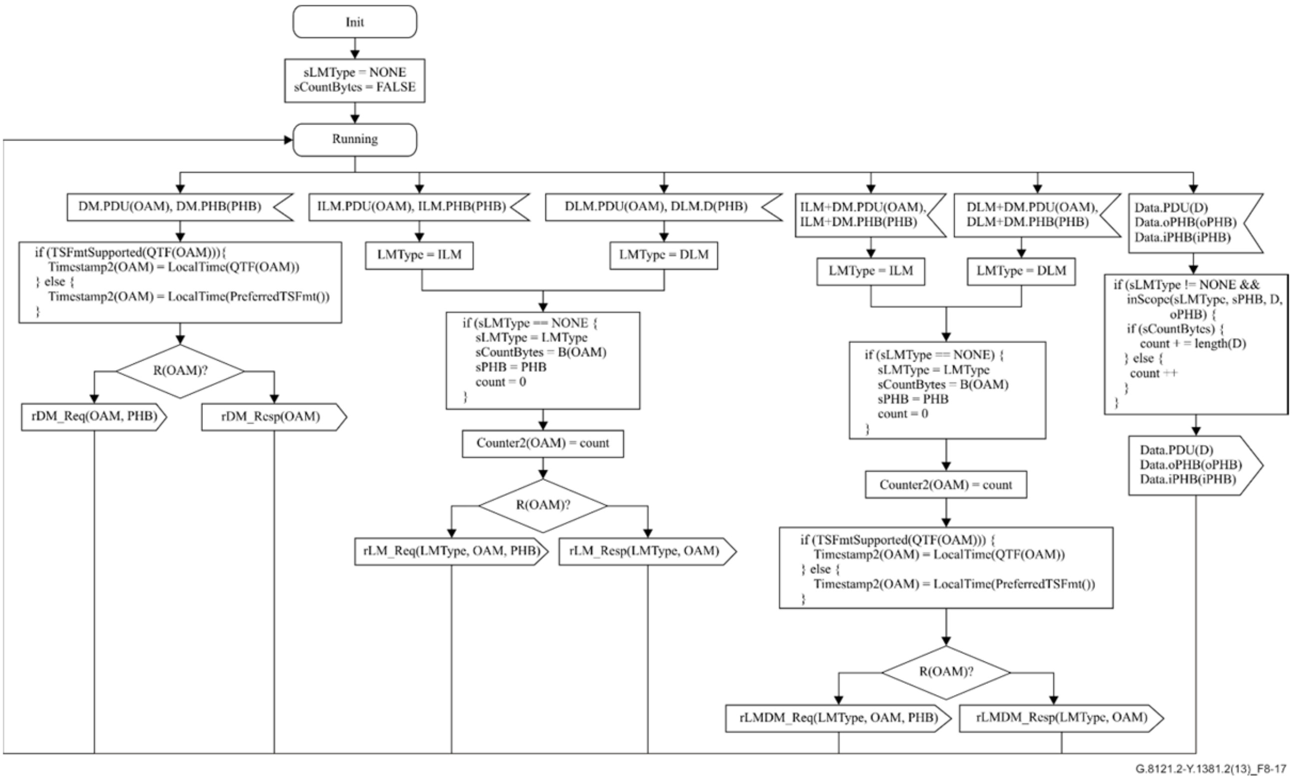

Figure 8-17 – PM reception process

8.8.4.5 PM responder process for proactive LM

The PM responder process responds to PM messages for a single PM session. Multiple sessions can be supported by instantiating multiple instances of the process, along with corresponding instances of the PM generation and PM reception processes.

The PM responder process is described in Figure 8-18.

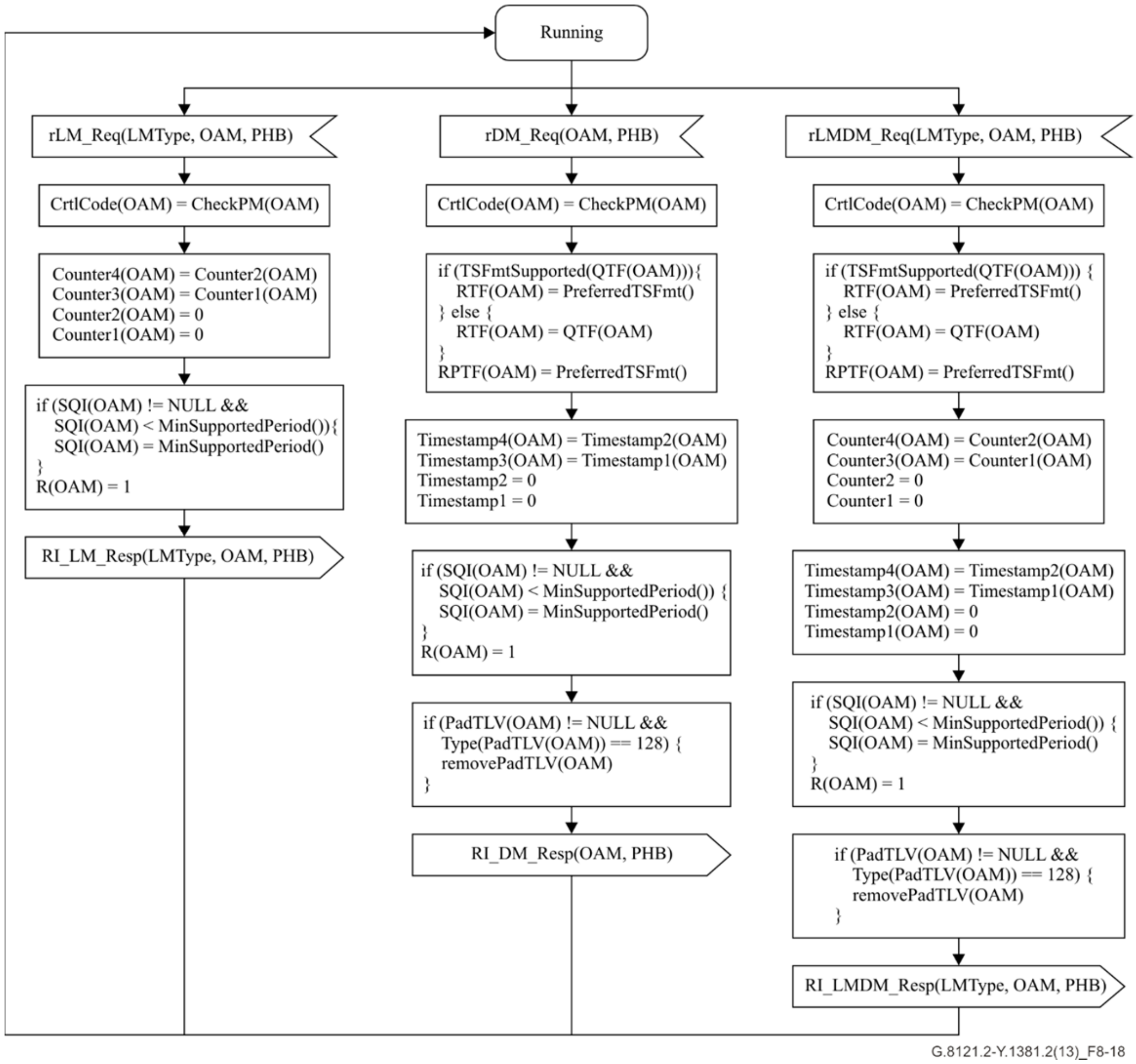

Figure 8-18 – PM responder process

The CheckPM() function checks the received PDU and returns an appropriate control code, as described in [IETF RFC 6374]. In particular, it returns 0x19 (Administrative Block) if MI_PM_Responder_Enable is not set, and 0x2 (Data Format Invalid) if the QTF in a DM, ILM+DM or DLM_DM message is not supported.

Note that when MI_PM_Responder_Enable is not set, responses are still sent, with the above error.

The PM responder process also unsets the X flag in LM messages if the implementation does not support 64 bit counters.

8.8.5 On-demand packet loss measurement (LMo)

As described in clauses 7.2.2.1 and 8.6 to 8.8 of [ITU-T G.8113.2], loss and delay measurements may be combined. The format for the combined measurement, referred to here as LMDM, is described in section 3.3 of [IETF RFC 6374]. In addition, the same LM and DM protocols can be used for both proactive and on-demand measurement.

An overview of the performance monitoring processes for a single on-demand PM session is shown in Figure 8-19. The same processes are used for LM, DM or LMDM.

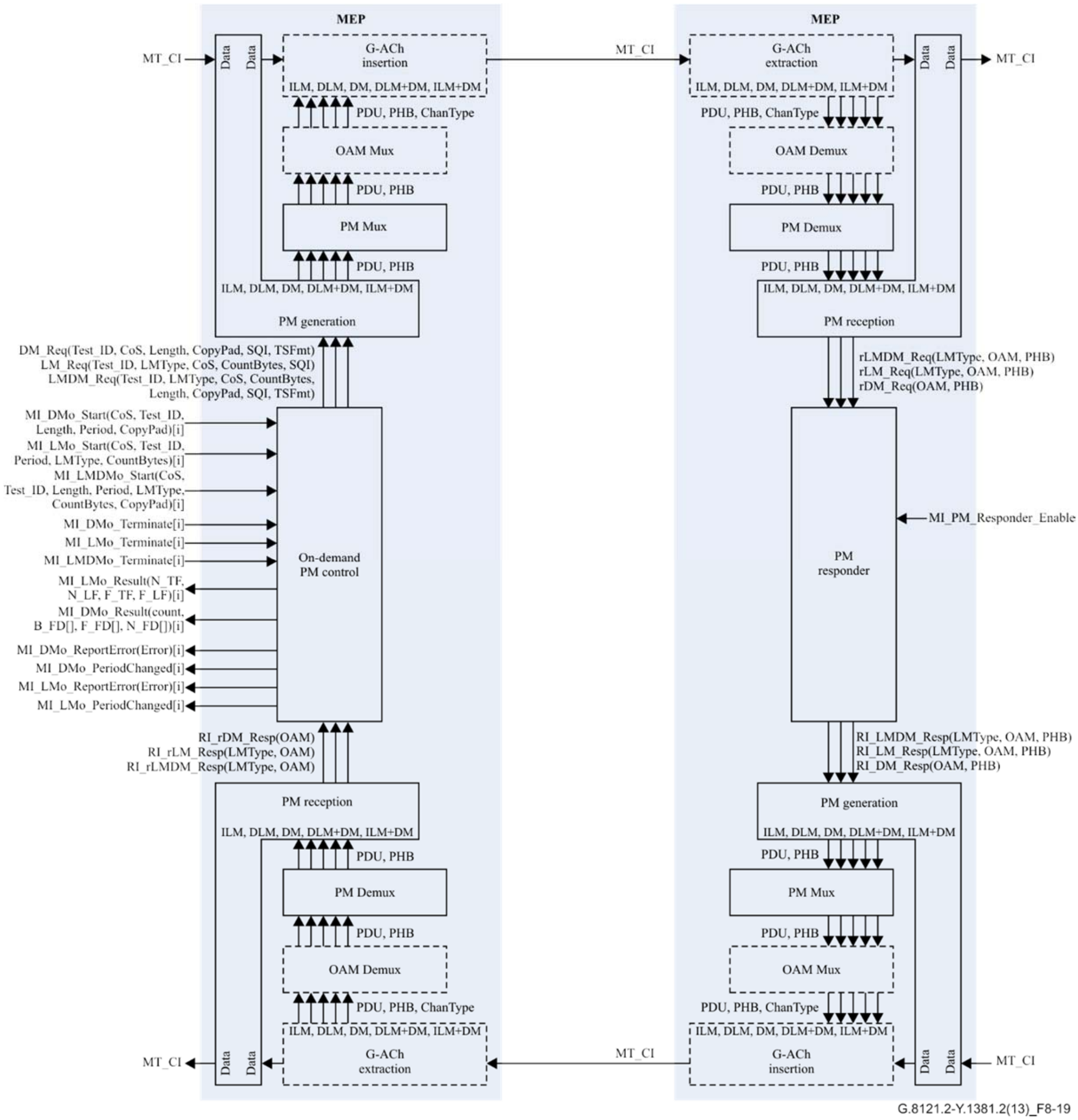

Figure 8-19 – On-demand PM processes

The on-demand PM control process controls the session, including scheduling request packets, and processing responses to calculate performance metrics. The other processes shown are the same as those used for proactive LM, as described in clause 8.8.4. As in the case of proactive LM, the location of the counter part in the PM generation and PM reception processes is shown in the figure above for illustration only; the exact set of packets to be counted is implementation-specific, as described in [IETF RFC 6374].

8.8.5.1 On-demand control process for LM

The on-demand PM control process includes LM, DM and LMDM. Each instance of the process operates a single on-demand PM session. Multiple sessions can be supported by instantiating multiple instances of the process, along with corresponding instances of the PM generation and PM reception processes.

The on-demand PM control process performs either delay measurement (via MI_DMo_Start/MI_DMo_Terminate), loss measurement (via MI_LMo_Start/MI_LMo_Terminate) or both simultaneously (via MI_LMDMo_Start/MI_LMDMo_Terminate). The type of loss measurement to perform is specified by the LMType parameter, and can be "ILM" for inferred (synthetic) loss or "DLM" for direct (data traffic) loss.

Results are reported via MI_DMo_Result and MI_LMo_Result.

If an error is detected while the session is running, this is reported via MI_DMo_ReportError or MI_LMo_ReportError, and the session is terminated automatically. The results collected up to that point are reported.

The PM protocol includes a mechanism to negotiate the packet sending period with the responder. If the period is changed from that specified when the session was started, this is signalled via MI_DMo_PeriodChanged or MI_LMo_PeriodChanged.

The CoS parameter of MI_LMo_Start, MI_DMo_Start or MI_LMDMo_Start specifies the CoS (traffic class) to use for the measurement. In the case of MI_LMo_Start, this can either be a specific value, or the special value "ALL" indicating that loss across all traffic classes should be measured.

The on-demand PM control process is described in Figures 8-20a, 8-20b, 8-20c and 8-20d.

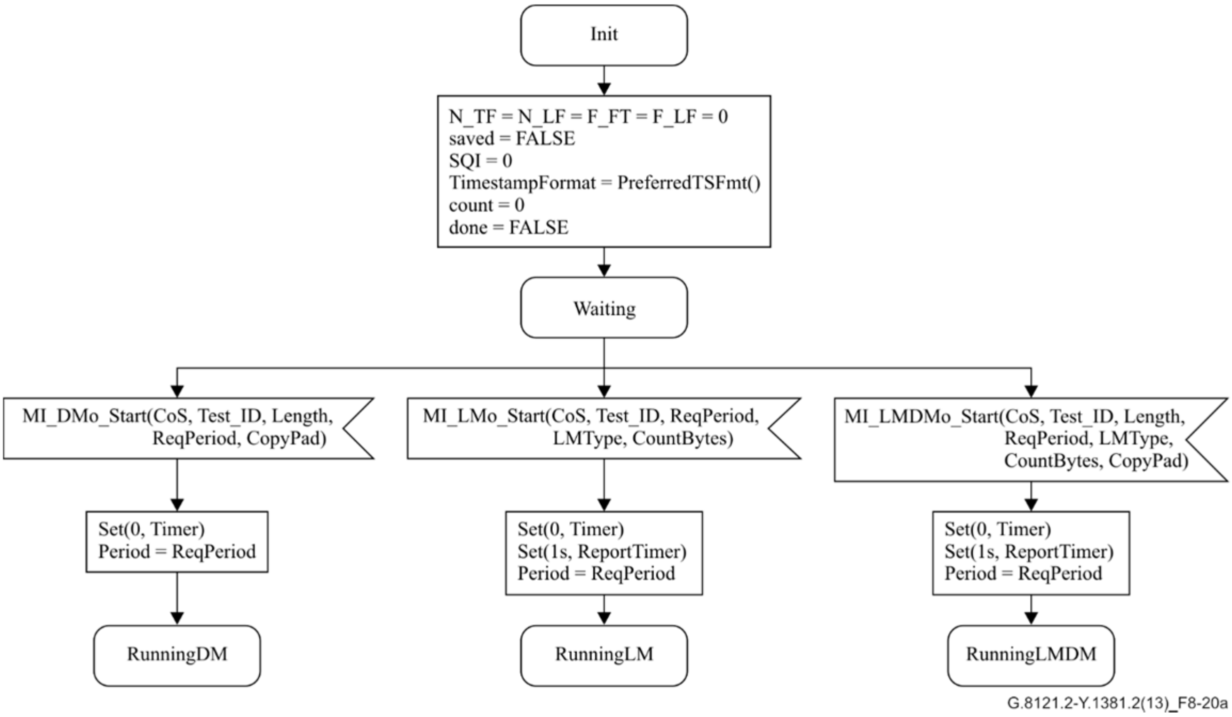

Figure 8-20a – On-demand PM control process

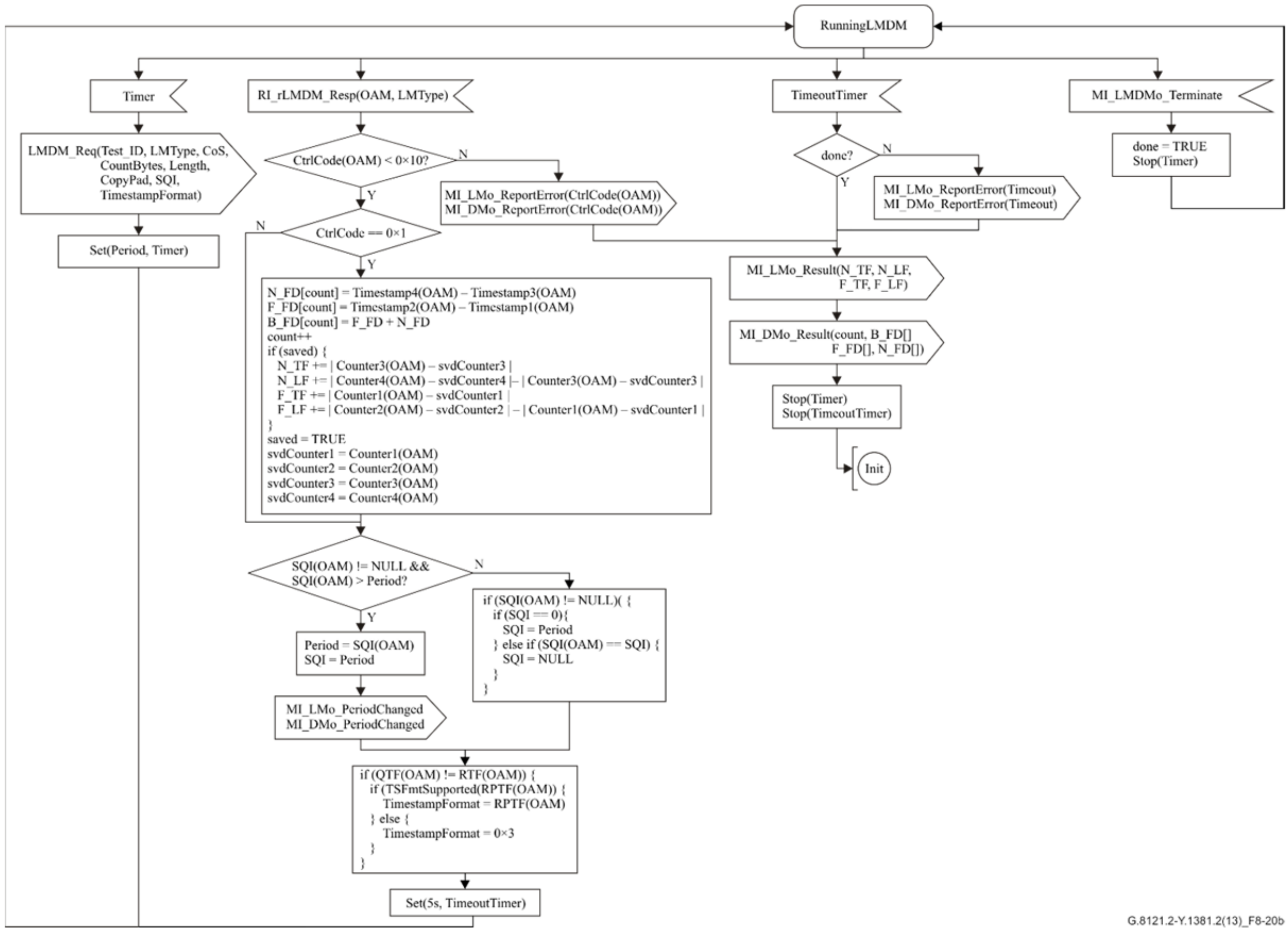

Figure 8-20b – On-demand PM control process

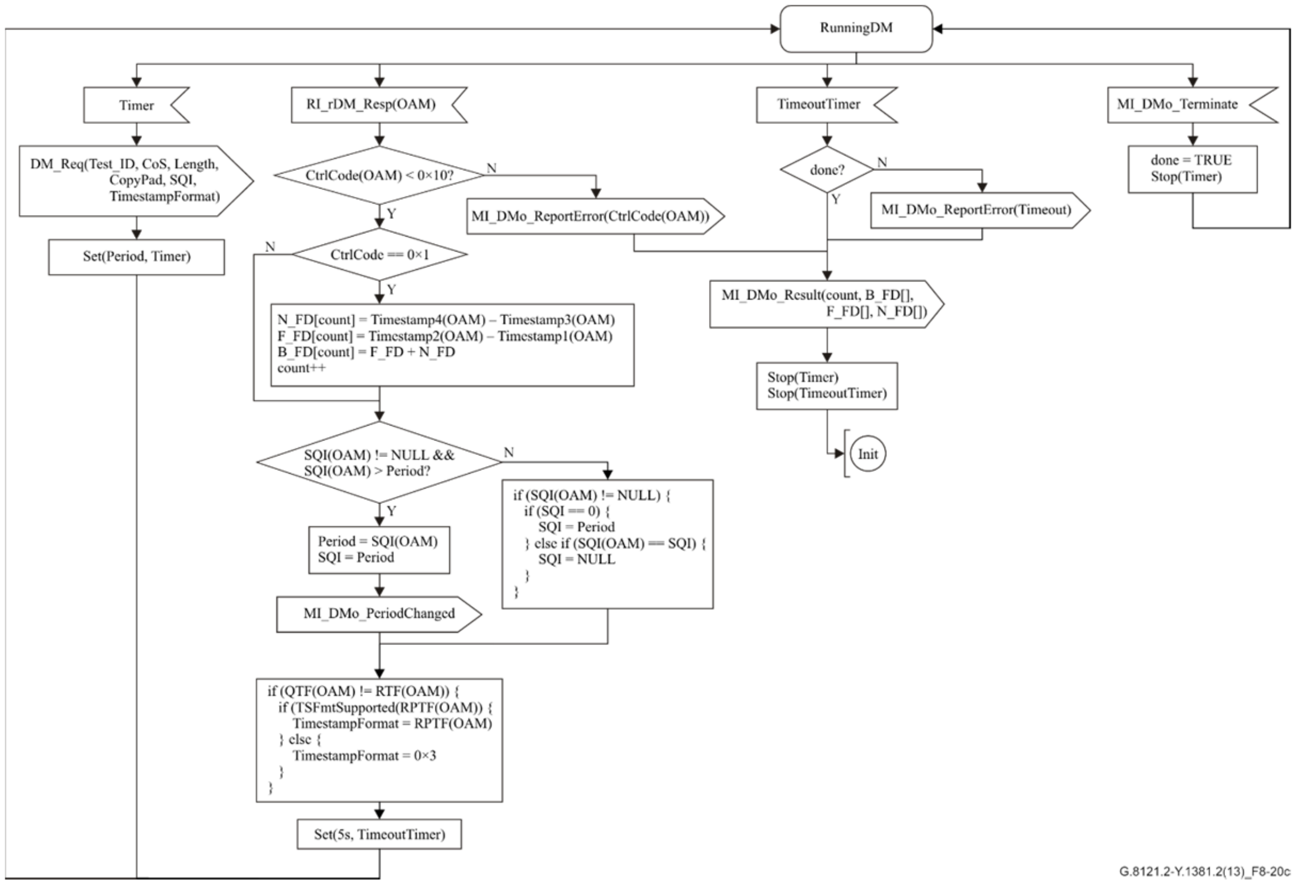

Figure 8-20c – On-demand PM control process

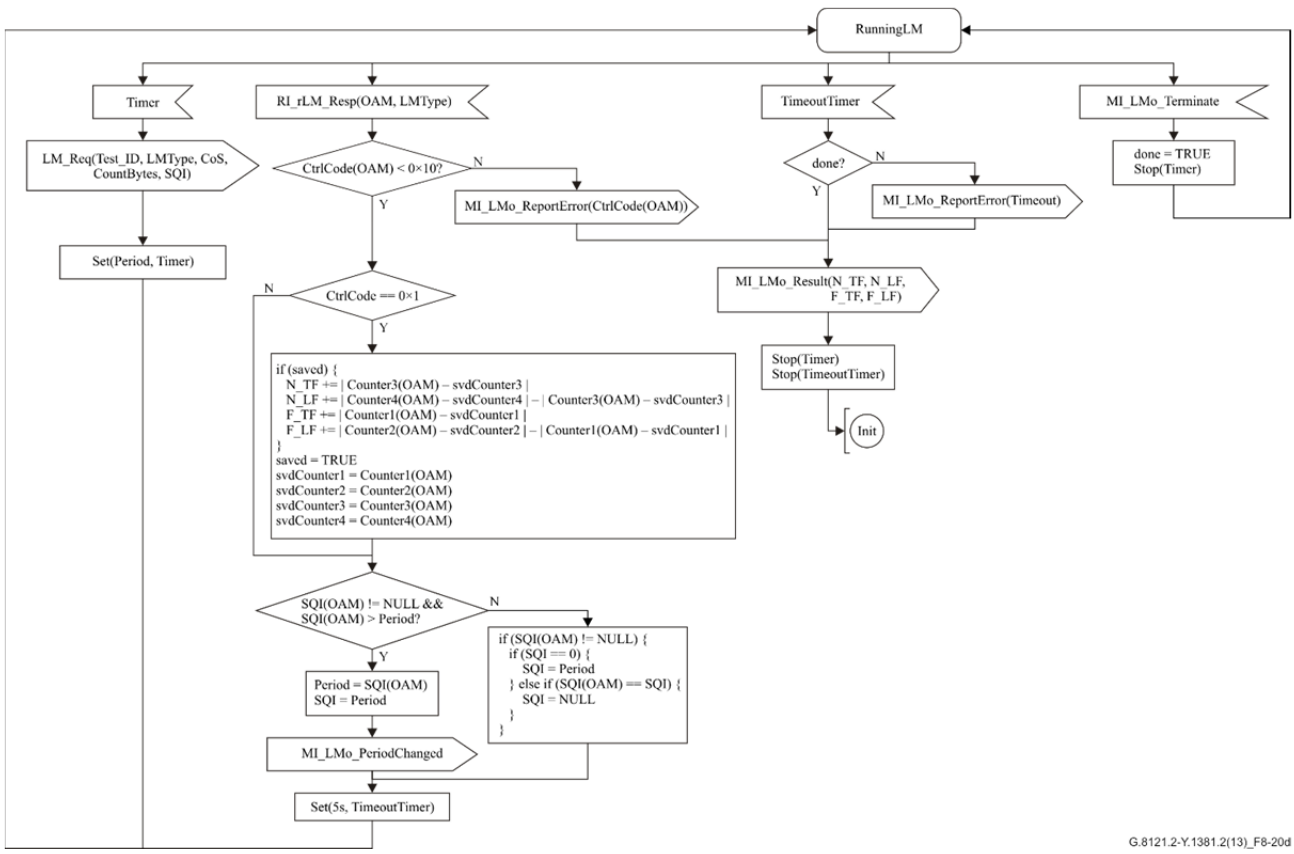

Figure 8-20d – On-demand PM control process

As in the proactive PM control process, the period and timestamp format are negotiated with the responder, as described in clause 8.8.4.1.

8.8.5.2 PM generation process for on-demand LM

The PM generation process for on-demand LM is identical to that of proactive LM, and is described in clause 8.8.4.3.

8.8.5.3 PM reception process for on-demand LM

The PM reception process for on-demand LM is identical to that of proactive LM, and is described in clause 8.8.4.4.

8.8.5.4 PM responder process for on-demand LM

The PM responder process for on-demand LM is identical to that of proactive LM, and is described in clause 8.8.4.5.

8.8.6 Proactive packet delay measurement (DMp)

As described in clauses 7.2.2.1 and 8.6 to 8.8 of [ITU-T G.8113.2], loss and delay measurements may be combined. The format for the combined measurement, referred to here as LMDM, is described in section 3.3 of [IETF RFC 6374]. In addition, the same LM and DM protocols can be used for both proactive and on-demand measurement.

The processes for proactive delay measurement are described in clause 8.8.4.

8.8.7 On-demand packet delay measurement (DMo)

As described in clauses 7.2.2.1 and 8.6 to 8.8 of [ITU-T G.8113.2], loss and delay measurements may be combined. The format for the combined measurement, referred to here as LMDM, is described in section 3.3 of [IETF RFC 6374]. In addition, the same LM and DM protocols can be used for both proactive and on-demand measurement.

The processes for on-demand delay measurement are described in clause 8.8.5.

8.8.8 Throughput test

For further study.

8.8.9 Route tracing (RT)

The processes for route tracing are described in clause 8.8.3.

8.8.10 LCK/AIS reception

The LCK/AIS reception process handles received LKR and AIS packets, and signals the LCK, AIS and SSF defects. The behaviour is shown in Figure 8-21.

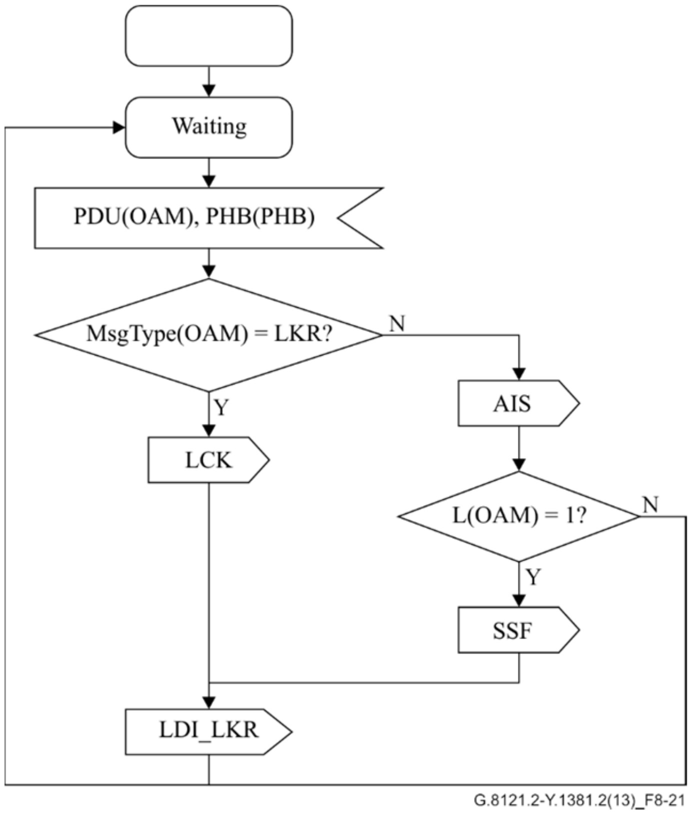

Figure 8-21 – LCR/AIS reception behaviour

8.8.11 Lock instruct processes

An overview of the processes relating to the lock instruct (LI) mechanism is shown in Figure 8-22. Note that [ITU-T G.8121] uses the abbreviation LKI for the same mechanism.

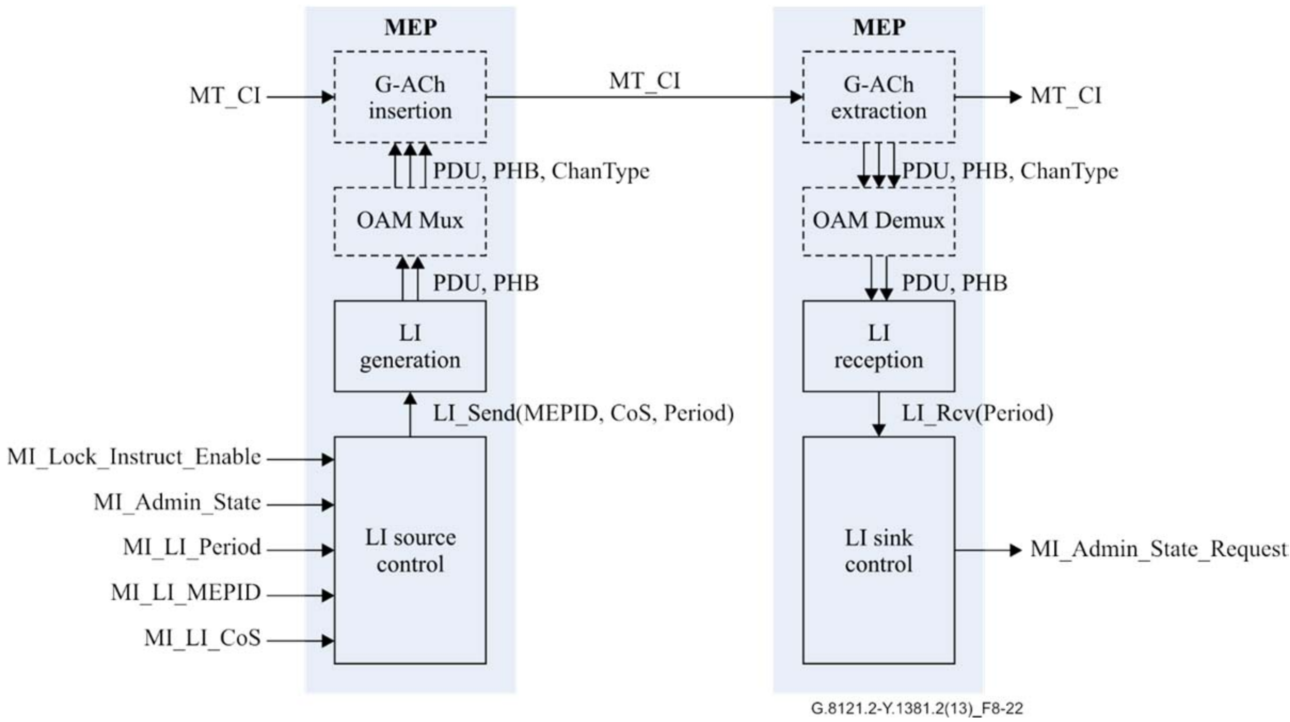

Figure 8-22 – Overview of lock instruct mechanism

The LI source control process controls sending LI messages when the admin state is "Locked" and MI_Lock_Instruct_Enable is set. The period at which to send is determined by MI_LI_Period, and the source MEP ID value is set by MI_LI_MEPID to one of the three values described in [IETF RFC 6435].

The LI generation process formats LI messages and passes them to the OAM Mux process and hence to the G-Ach insertion process.

The LI reception process handles received LI messages and checks them for correctness.

The LI sink control process monitors received LI messages to determine whether a lock instruct condition exists, and signals this to the EMF via MI_Admin_State_Request.

8.8.11.1 LI source control process

The LI source control process is described in Figure 8-23.

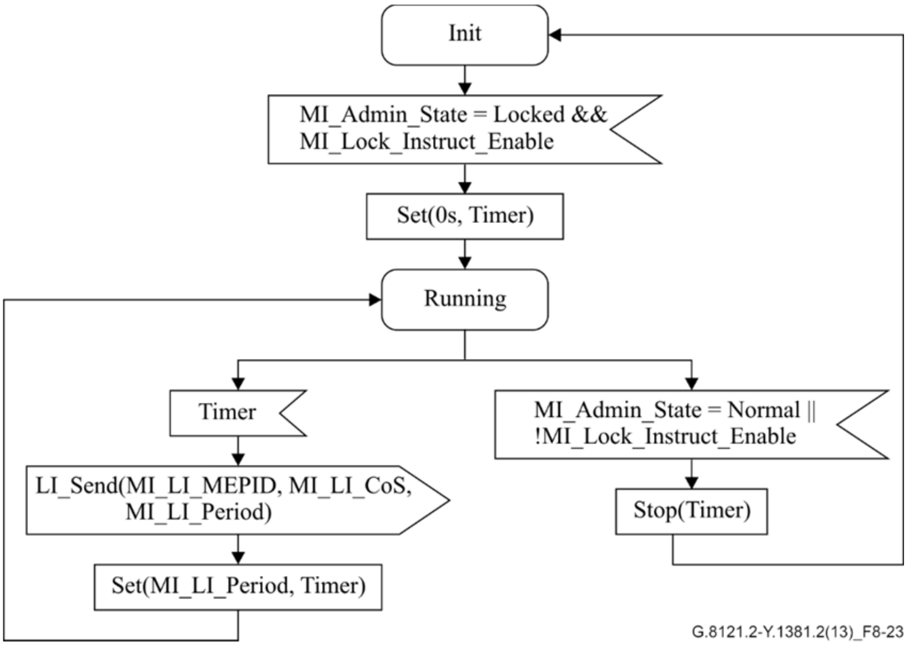

Figure 8-23 – LI source control process

8.8.11.2 LI generation process

The LI generation process is described in Figure 8-24.

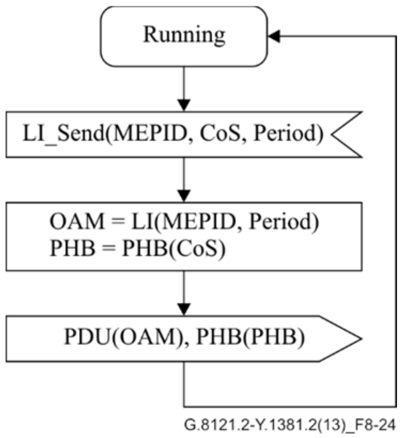

Figure 8-24 – LI generation process

The LI(MEPID, Period) function formats an LI PDU according to [IETF RFC 6435], as follows:

- The version is set to 1.
- The reserved field is set to 0.
- The refresh timer field is set to the 'Period'.
- The MEPID is copied into the MEP source ID TLV.

The PHB(CoS) function returns the PHB with the lowest drop precedence within the class of service defined by the CoS input parameter.

8.8.11.3 LI reception process

The LI reception process is described in Figure 8-25.

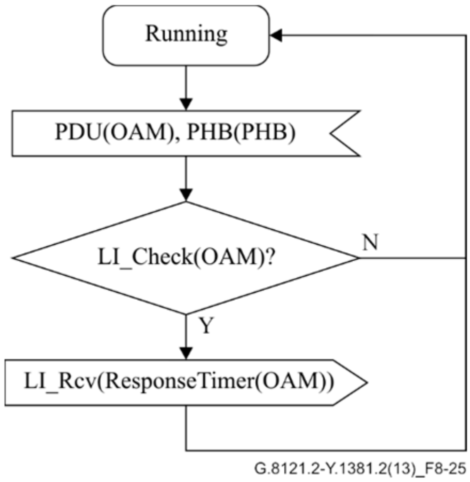

Figure 8-25 – LI reception process

The LI_Check(OAM) function performs implementation-specific checks, including those described in [IETF RFC 6435], and returns true if the OAM is valid and false otherwise.

8.8.11.4 LI sink control process

The LI sink control process is described in Figure 8-26.

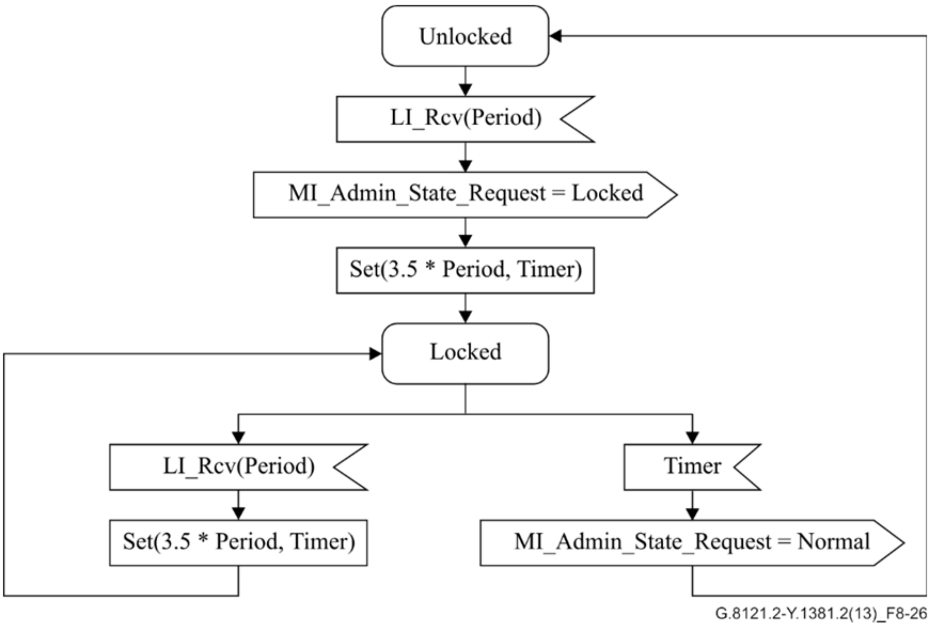

Figure 8-26 – LI sink control process

8.9 Data-plane loopback processes

See clause 8.9 of [ITU-T G.8121].

9 MPLS-TP layer functions

9.1 Connection functions (MT_C)

Connection functions are described in [ITU-T G.8121].

9.2 Termination functions

9.2.1 MPLS-TP trail termination function (MT_TT)

The bidirectional MPLS-TP trail termination (MT_TT) function terminates the MPLS-TP OAM to determine the status of the MPLS-TP (sub)layer trail. The MT_TT function is performed by a co-located pair of the MPLS-TP trail termination source (MT_TT_So) and sink (MT_TT_Sk) functions as shown in Figure 9-1.

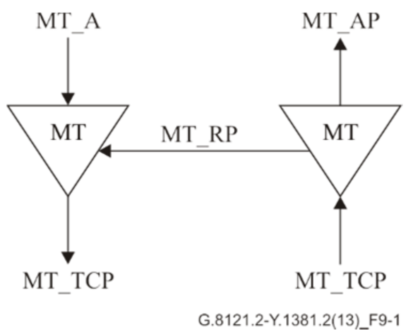

Figure 9-1 – MT_TT

9.2.1.1 MPLS-TP trail termination source function (MT_TT_So)

The MT_TT_So function determines and inserts the TTL value in the shim header TTL field and adds MPLS-TP OAM to the MT_AI signal at its MT_AP.

The information flow and processing of the MT_TT_So function is defined with reference to Figure 9-2.

Symbol

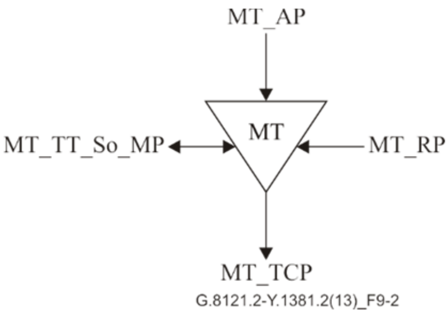

Figure 9-2 – MT_TT_So symbol

Interfaces

Table 9-1 – MT_TT_So inputs and outputs

Input(s)	Output(s)
MT_AP: MT_AI_D MT_AI_PHB **MT_RP**: MT_RI_CCCV_Params(State, Diag, TxIntl, rDiscr) MT_RI_CC_Blk MT_RI_DM_Resp(OAM, PHB) MT_RI_LM_Resp(LMType, OAM, PHB) MT_RI_LMDM_Resp (LMType, OAM, PHB) MT_RI_PM_Error MT_RI_DM_Info(SQI, QTF, RTF, RPTF) MT_RI_LM_Info(SQI) MT_RI_LMDM_Info(SQI, QTF, RTF, RPTF)	**MT_CP**: MT_CI_D MT_CI_oPHB MT_CI_iPHB **MT_TT_So_MP**: MT_TT_So_MI_DMp_PeriodChanged[1...M_{DMp}] MT_TT_So_MI_LMp_PeriodChanged[1...M_{LMp}]

Table 9-1 – MT_TT_So inputs and outputs

Input(s)	Output(s)
MT_TT_So_MP: MT_TT_So_MI_GAL_Enable MT_TT_So_MI_TTLValue MT_TT_So_MI_CCCV_Mode[1…M_{CCCV}] MT_TT_So_MI_MEPID[1…M_{CCCV}] MT_TT_So_MI_Local_Discr[1…M_{CCCV}] MT_TT_So_MI_CC_Period MT_TT_So_MI_CC_CoS[1…M_{CCCV}] MT_TT_So_MI_DMp_Enable[1...M_{DMp}] MT_TT_So_MI_DMp_Test_ID[1...M_{DMp}] MT_TT_So_MI_DMp_CoS[1...M_{DMp}] MT_TT_So_MI_DMp_Length[1...M_{DMp}] MT_TT_So_MI_DMp_CopyPad[1...M_{DMp}] MT_TT_So_MI_DMp_Period[1...M_{DMp}] MT_TT_So_MI_LMp_Enable[1...M_{LMp}] MT_TT_So_MI_LMp_Test_ID[1...M_{LMp}] MT_TT_So_MI_LMp_LMType[1...M_{LMp}] MT_TT_So_MI_LMp_CoS[1...M_{LMp}] MT_TT_So_MI_LMp_CountBytes[1...M_{LMp}] MT_TT_So_MI_LMp_Period[1...M_{LMp}]	

Processes

The processes associated with the MT_TT_So function are depicted in Figure 9-3. The subprocesses within each process, described in clause 8.8, are not shown separately.

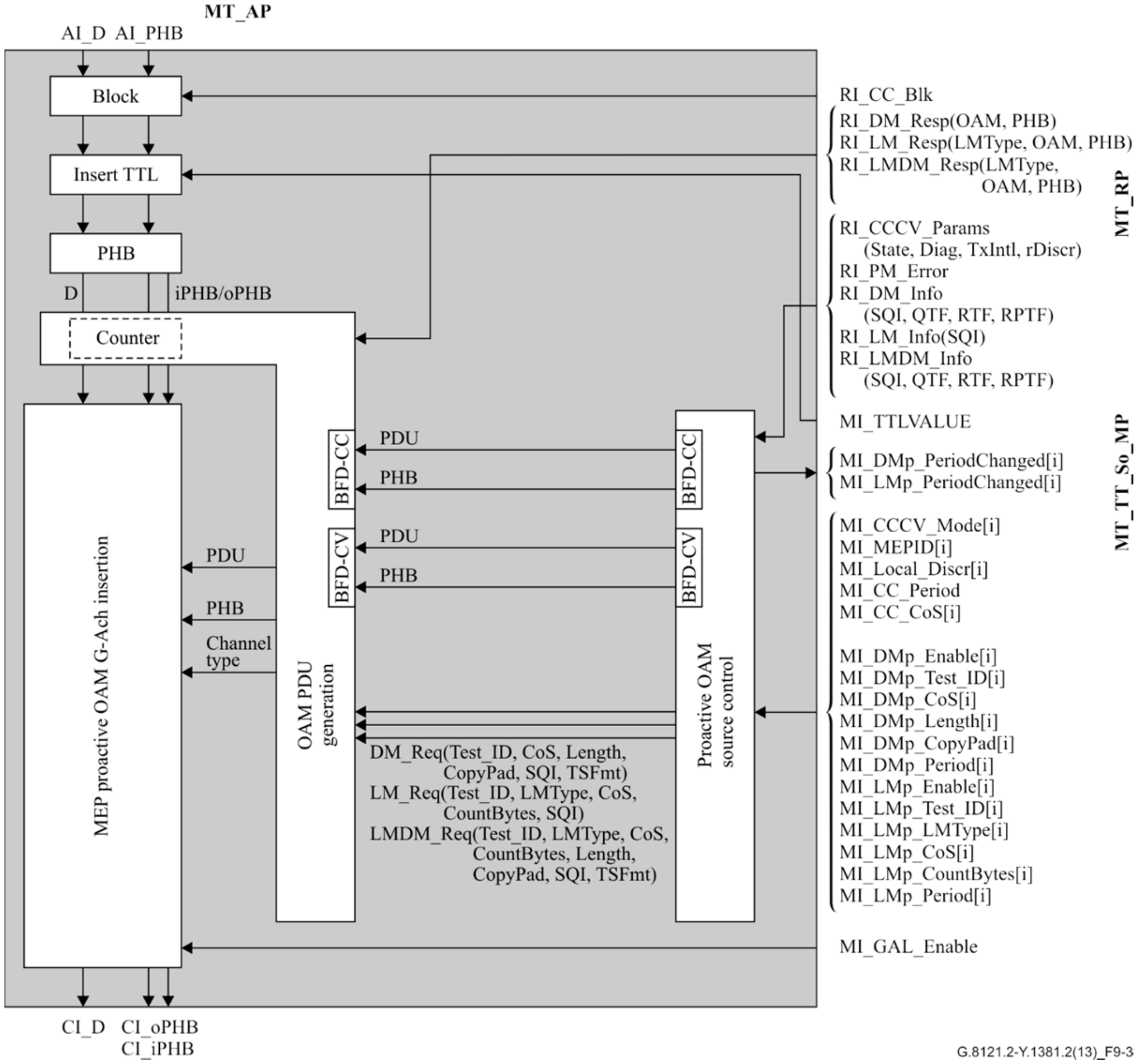

Figure 9-3 – MT_TT_So process

PHB: See clause 9.2.1.1 of [ITU-T G.8121].The AI_PHB signal is assigned to both the CI_iPHB and CI_oPHB signals at the MT_TCP reference point.

Insert TTL: See clause 9.2.1.1 of [ITU-T G.8121].

Block: See clause 9.2.1.1 of [ITU-T G.8121].

MEP proactive OAM G-ACh insertion: See clause 8.1.2 of [ITU-T G.8121].

OAM PDU generation: This contains the following subprocesses as described in clause 8.8: PM generation; OAM Mux; PM Mux.

Proactive OAM source control: This contains the following subprocesses as described in clause 8.8: CCCV generation; proactive PM source control.

The location of the counter part of the OAM PDU generation process is shown for illustration only. The exact set of packets to be counted is implementation-specific, as described in [IETF RFC 6374].

Defects

None.

Consequent actions

None.

Defect correlations

None.

Performance monitoring

None.

9.2.1.2 MPLS-TP trail termination sink function (MT_TT_Sk)

The MT_TT_Sk function reports the state of the MPLS-TP trail (network connection). It extracts the MPLS-TP trail OAM from the MPLS-TP signal at its MT_TCP, detects defects, counts during 1-second periods errors and defects to feed performance monitoring when connected and forwards the defect information as backward indications to the companion MT_TT_So function.

NOTE – The MT_TT_Sk function extracts and processes one level of MPLS-TP OAM irrespective of the presence of more levels.

The information flow and processing of the MT_TT_Sk function is defined with reference to Figure 9-4.

Symbol

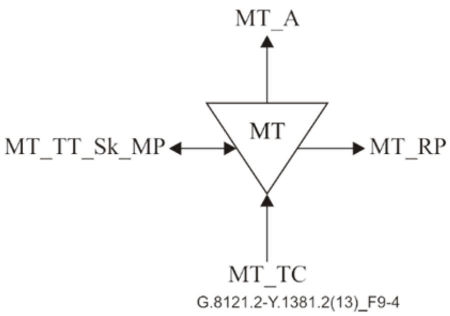

Figure 9-4 – MT_TT_Sk function symbol

Interfaces

Table 9-2 – MT_TT_Sk inputs and outputs

Input(s)	Output(s)
MT_TCP: MT_CI_D MT_CI_iPHB MT_CI_oPHB MT_CI_SSF MT_CI_LStack **MT_TT_Sk_MP**: MT_TT_Sk_MI_GAL_Enable MT_TT_Sk_MI_CC_Enable[1…M_{CCCV}] MT_TT_Sk_MI_CCCV_Mode[1…M_{CCCV}] MT_TT_Sk_MI_CC_Period	**MT_AP**: MT_AI_D MT_AI_PHB MT_AI_TSF MT_AI_TSD MT_AI_AIS MT_AI_LStack **MT_RP**: MT_RI_CCCV_Params(State, Diag, TxIntl, rDiscr) MT_RI_CC_Blk MT_RI_DM_Resp(OAM, PHB)

Table 9-2 – MT_TT_Sk inputs and outputs

Input(s)	Output(s)
MT_TT_Sk_MI_PeerMEPID[1…M_{CCCV}] MT_TT_Sk_MI_Remote_Discr[1…M_{CCCV}] MT_TT_Sk_MI_CC_CoS[] MT_TT_Sk_MI_GetSvdCC[1…M_{CCCV}] MT_TT_Sk_MI_DMp_Enable[1...M_{DMp}] MT_TT_Sk_MI_LMp_Enable[1...M_{LMp}] MT_TT_Sk_MI_PM_ClearError MT_TT_Sk_MI_PM_Responder_Enable	MT_RI_LM_Resp(LMType, OAM, PHB) MT_RI_LMDM_Resp(LMType, OAM, PHB) MT_RI_PM_Error MT_RI_DM_Info(SQI, QTF, RTF, RPTF) MT_RI_LM_Info(SQI) MT_RI_LMDM_Info(SQI, QTF, RTF, RPTF) **MT_TT_Sk_MP**: MT_TT_Sk_MI_SvdCC MT_TT_Sk_MI_cSSF MT_TT_Sk_MI_cLCK MT_TT_Sk_MI_cLOC[] MT_TT_Sk_MI_cMMG MT_TT_Sk_MI_cUNM MT_TT_Sk_MI_cUNC MT_TT_Sk_MI_cDEG MT_TT_Sk_MI_cRDI MT_TT_Sk_MI_DMp_ReportError(Error)[1...M_{DMp}] MT_TT_Sk_MI_LMp_ReportError(Error)[1...M_{LMp}] MT_TT_Sk_MI_pN_LF[1...P] MT_TT_Sk_MI_pN_TF[1...P] MT_TT_Sk_MI_pF_LF[1...P] MT_TT_Sk_MI_pF_TF[1...P] MT_TT_Sk_MI_pF_DS MT_TT_Sk_MI_pN_DS MT_TT_Sk_MI_pB_FD[1...P] MT_TT_Sk_MI_pB_FDV[1...P] MT_TT_Sk_MI_pN_FD[1...P] MT_TT_Sk_MI_pN_FDV[1...P] MT_TT_Sk_MI_pF_FD[1...P] MT_TT_Sk_MI_pF_FDV[1...P]

Processes

The processes associated with the MT_TT_Sk function are depicted in Figure 9-5.

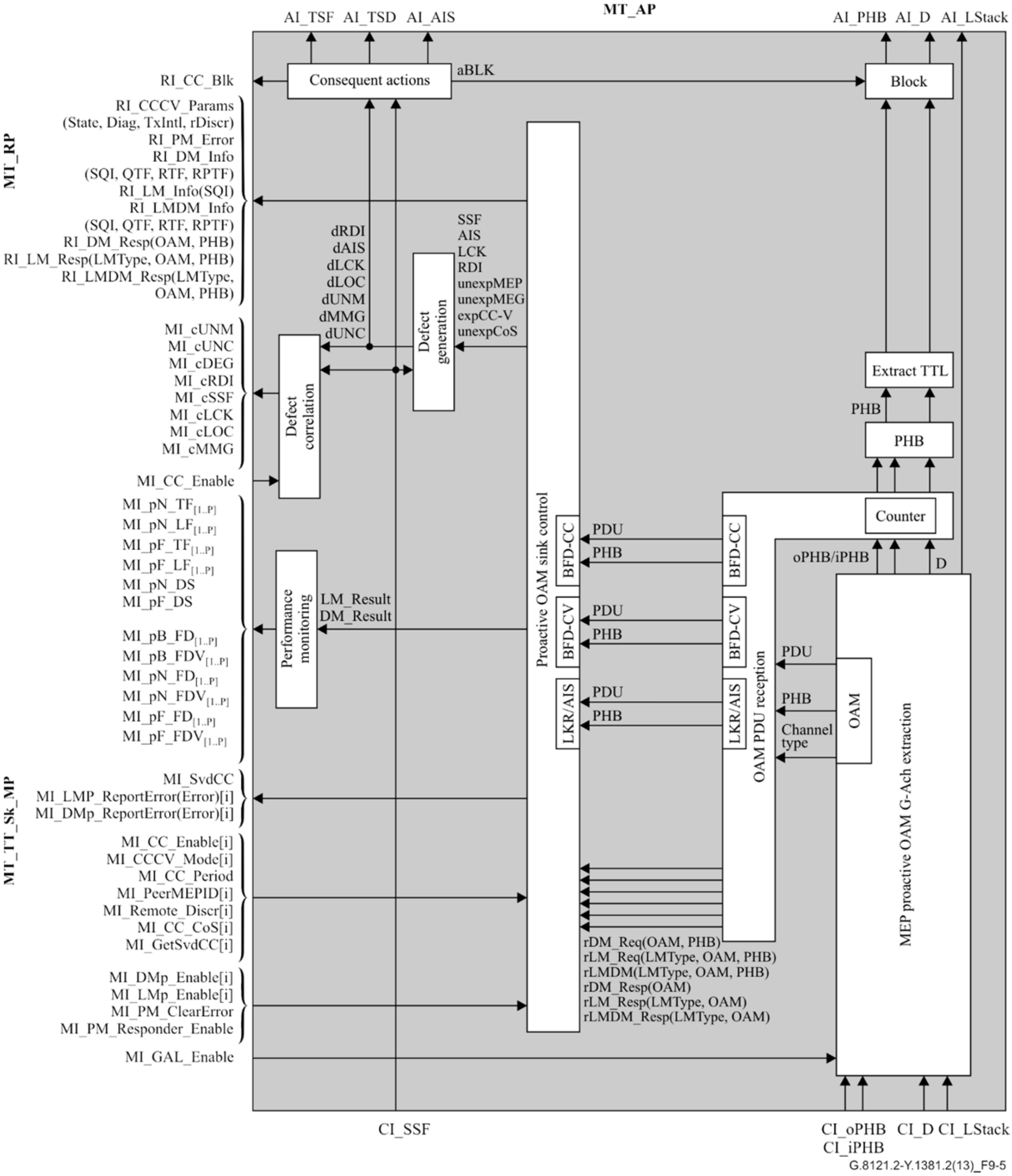

Figure 9-5 – MT_TT_Sk process

PHB: See clause 9.2.1.2 of [ITU-T G.8121].

Extract TTL: See clause 9.2.1.2 of [ITU-T G.8121].

Block: See clause 9.2.1.2 of [ITU-T G.8121].

MEP proactive OAM G-ACh extraction: See clause 8.1.3 of [ITU-T G.8121].

OAM PDU reception: This contains the following subprocesses as described in clause 8.8: PM reception; OAM Demux; session Demux; PM Demux.

Proactive OAM sink control: This contains the following subprocesses as described in clause 8.8: CCCV reception; LCK/AIS reception; proactive PM sink control; PM responder.

Performance counter: See clause 9.2.1.2 of [ITU-T G.8121].

Defect generation: See clause 9.2.1.2 of [ITU-T G.8121].

The location of the counter part of the OAM PDU reception process is shown for illustration only. The exact set of packets to be counted is implementation-specific, as described in [IETF RFC 6374].

Defects

See [ITU-T G.8121].

Consequent actions

See [ITU-T G.8121].

Defect correlations

See [ITU-T G.8121].

Performance monitoring

See [ITU-T G.8121].

9.3 Adaptation functions

9.3.1 MPLS-TP to MPLS-TP adaptation function (MT/MT_A)

This atomic function is defined in clause 9.3.1 of [ITU-T G.8121]. They use the OAM protocol-specific AIS insert process and LCK generation process as defined in clauses 8.6.2 and 8.6.3. For the MT/MT_A_Sk function, in addition to the MI shown in Table 9-5 of [ITU-T G.8121] and Figure 9-11 of [ITU-T G.8121], there is an additional protocol-specific MI used by the AIS insert process defined in this document: MI_Local_Defect[1..M].

9.4 MT diagnostic functions

9.4.1 Diagnostic functions for MEPs

9.4.1.1 MT diagnostic trail termination functions for MEPs (MTDe_TT)

The bidirectional MTDe trail termination (MTDe_TT) function is performed by a co-located pair of MTDe trail termination source (MTDe_TT_So) and sink (MTDe_TT_Sk) functions as shown in Figure 9-6.

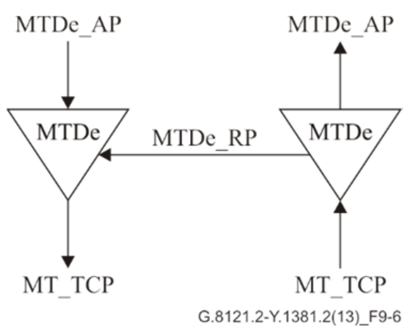

Figure 9-6 – MTDe_TT

9.4.1.1.1 MT diagnostic trail termination source function for MEPs (MTDe_TT_So)

The MTDe_TT_So process diagram is shown in Figure 9-8.

Symbol

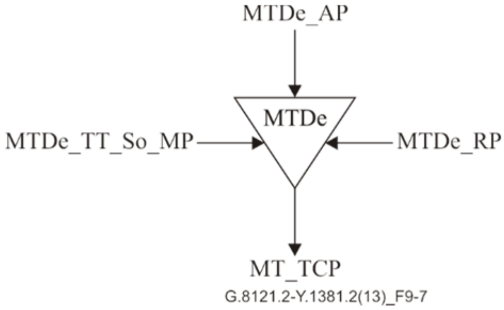

Figure 9-7 – MTDe_TT_So_Symbol

Interfaces

Table 9-3 – MTDe_TT_So interfaces

Input(s)	Output(s)
MT_AP: MTDe_AI_D MTDe_AI_iPHB MTDe_AI_oPHB **MTDe_RP**: MTDe_RI_LSPPing_Rsp(OAM, PHB, RC, SubRC,Err_TLV, include_ifstack, LStack Include_dsmap, ingress_Ifnum, egress_Ifnum, ds_lstack) MTDe_RI_DM_Resp(OAM, PHB) MTDe_RI_LM_Resp(LMType, OAM, PHB) MTDe_RI_LMDM_Resp(LMType, OAM, PHB) MTDe_RI_rLSPPing_Rsp(OAM, PHB, LStack) MTDe_RI_rDM_Resp(OAM) MTDe_RI_rLM_Resp(LMtype, OAM) MTDe_RI_rLMDM_Resp(LMtype, OAM) MTDe_RI_CI (D, iPHB, oPHB) **MTDe_TT_So_MP**: MTDe_TT_So_MI_GAL_Enable MTDe_TT_So_MI_CV_Series (Session_ID, Count, Period, CoS, Size, ValidateFEC, ValidateReverse, TargetFECStack) MTDe_TT_So_MI_ODCV_Trace (Session_ID, CoS, ValidateFEC, ValidateReverse, TargetFECStack) MTDe_TT_So_MI_FEC_Checking MTDe_TT_So_MI_Target_FEC MTDe_TT_So_MI_Ifnum MTDe_TT_So_MI_MTU MTDe_TT_So_MI_DMo_Start(CoS, Test_ID,	**MT_CP**: MT_CI_D MT_CI_oPHB MT_CI_iPHB **MTDe_RP**: **MTDe_TT_So_MP**: MTDe_TT_So_MI_DMo_ReportError(Error) [1...M_{DMo}] MTDe_TT_So_MI_DMo_PeriodChanged [1...M_{DMo}] MTDe_TT_So_MI_LMo_ReportError(Error) [1...M_{LMo}] MTDe_TT_So_MI_LMo_PeriodChanged [1...M_{LMo}] MTDe_TT_So_MI_DMo_Result(count, B_FD[], F_FD[], N_FD[])[1...M_{DMo}] MTDe_TT_So_MI_LMo_Result(N_TF, N_LF, F_TF, F_LF)[1...M_{LMo}] MTDe_TT_So_MI_CV_Series_Result (Session_ID, Rcv, OOO, FWErr, BWErr) MTDe_TT_So_MI_ODCV_Trace_Result (Session_ID, Result) MTDe_TT_So_MI_ODCV_FWErr(Session_ID, Seq, RC, SubRC, ErrTLV) MTDe_TT_So_MI_ODCV_BWErr(Session_ID, Seq, RC, SubRC, ErrTLV)

Table 9-3 – MTDe_TT_So interfaces

Input(s)	Output(s)
Length, Period, CopyPad)[1...M_{DMo}] MTDe_TT_So_MI_LMo_Start(CoS, Test_ID, Period, LMType, CountBytes)[1...M_{LMo}] MTDe_TT_So_MI_LMDMo_Start(CoS, Test_ID, Length, Period, LMType, CountBytes, CopyPad)[1...M_{LMDMo}] MTDe_TT_So_MI_DMo_Terminate [1...M_{DMo}] MTDe_TT_So_MI_LMo_Terminate[1...M_{LMo}] MTDe_TT_So_MI_LMDMo_Terminate [1...M_{LMDMo}] MTDe_TT_So_MI_Lock_Instruct_Enable MTDe_TT_So_MI_Admin_State MTDe_TT_So_MI_LI_Period MTDe_TT_So_MI_LI_MEPID MTDe_TT_So_MI_LI_CoS MTDe_TT_So_MI_DP_Loopback_Enable	

Processes

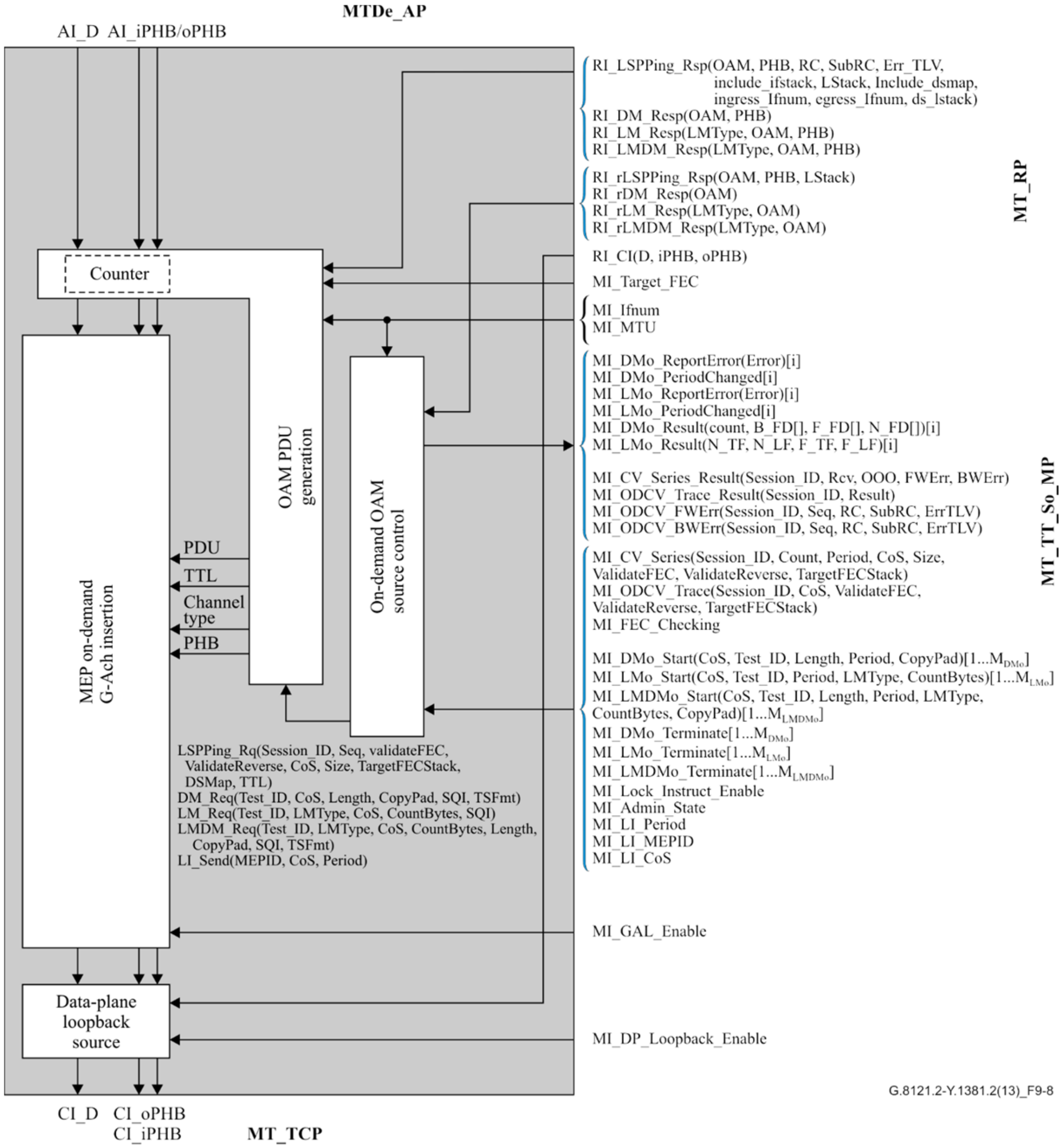

Figure 9-8 – MTDe_TT_So process

MEP on-demand OAM G-ACh insertion: See clause 8.1.2 of [ITU-T G.8121].

OAM PDU generation: This contains the following subprocesses as described in clause 8.8: on-demand CV request generation; on-demand CV response generation; LI generation; PM generation; OAM Mux; PM Mux.

On-demand OAM source control: This contains the following subprocesses as described in clause 8.8: LI source control; on-demand PM control; on-demand CV control.

The location of the counter part of the OAM PDU generation process is shown for illustration only. The exact set of packets to be counted is implementation-specific, as described in [IETF RFC 6374].

Data-plane loopback source process: See clause 8.9.1 of [ITU-T G.8121].

Defects

None.

Consequent actions

None.

Defect correlations

None.

Performance monitoring

None.

9.4.1.1.2 MT diagnostic trail termination sink function for MEPs (MTDe_TT_Sk)

The MTDe_TT_Sk process diagram is shown in Figure 9-10.

Symbol

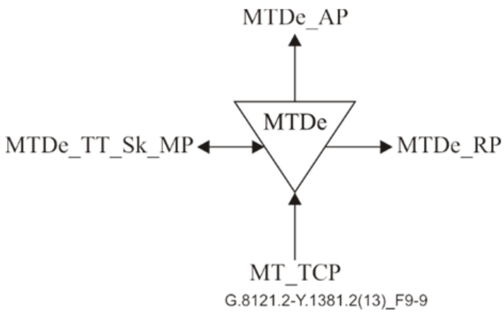

Figure 9-9 – MTDe_TT_Sk_Symbol

Interfaces

Table 9-4 – MTDe_TT_Sk interfaces

Input(s)	Output(s)
MT_CP: MT_CI_D MT_CI_iPHB MT_CI_oPHB MT_CI_LStack **MT_RP**: MTDe_TT_Sk_MP: MTDe_TT_Sk_MI_GAL_Enable MTDe_TT_Sk_MI_FEC_Checking MTDe_TT_Sk_MI_PM_Responder_Enable MTDe_TT_Sk_MI_DP_Loopback_Enable	**MTDe_AP**: MTDe_AI_D MTDe_AI_iPHB MTDe_AI_oPHB MTDe_AI_LStack **MTDe_RP**: MTDe_RI_rLSPPing_Rsp (OAM, PHB, LStack) MTDe_RI_LSPPing_Rsp(OAM, PHB, RC, SubRC,Err_TLV, include_ifstack, LStack, Include_dsmap, ingress_Ifnum, egress_Ifnum, ds_lstack) MTDe_RI_rDM_Resp(OAM) MTDe_RI_rLM_Resp (LMtype,OAM)

Table 9-4 – MTDe_TT_Sk interfaces

Input(s)	Output(s)
	MTDe_RI_rLMDM_Resp (LMtype,OAM) MTDe_RI_DM_Resp (OAM, PHB) MTDe_RI_LM_Resp (LMType, OAM, PHB) MTDe_RI_LMDM_Resp (LMType, OAM, PHB) MTDe_RI_CI(D, iPHB, oPHB) **MTDe_FT_Sk_MP**: MTDe_TT_Sk_MI_Admin_State_Request

Processes

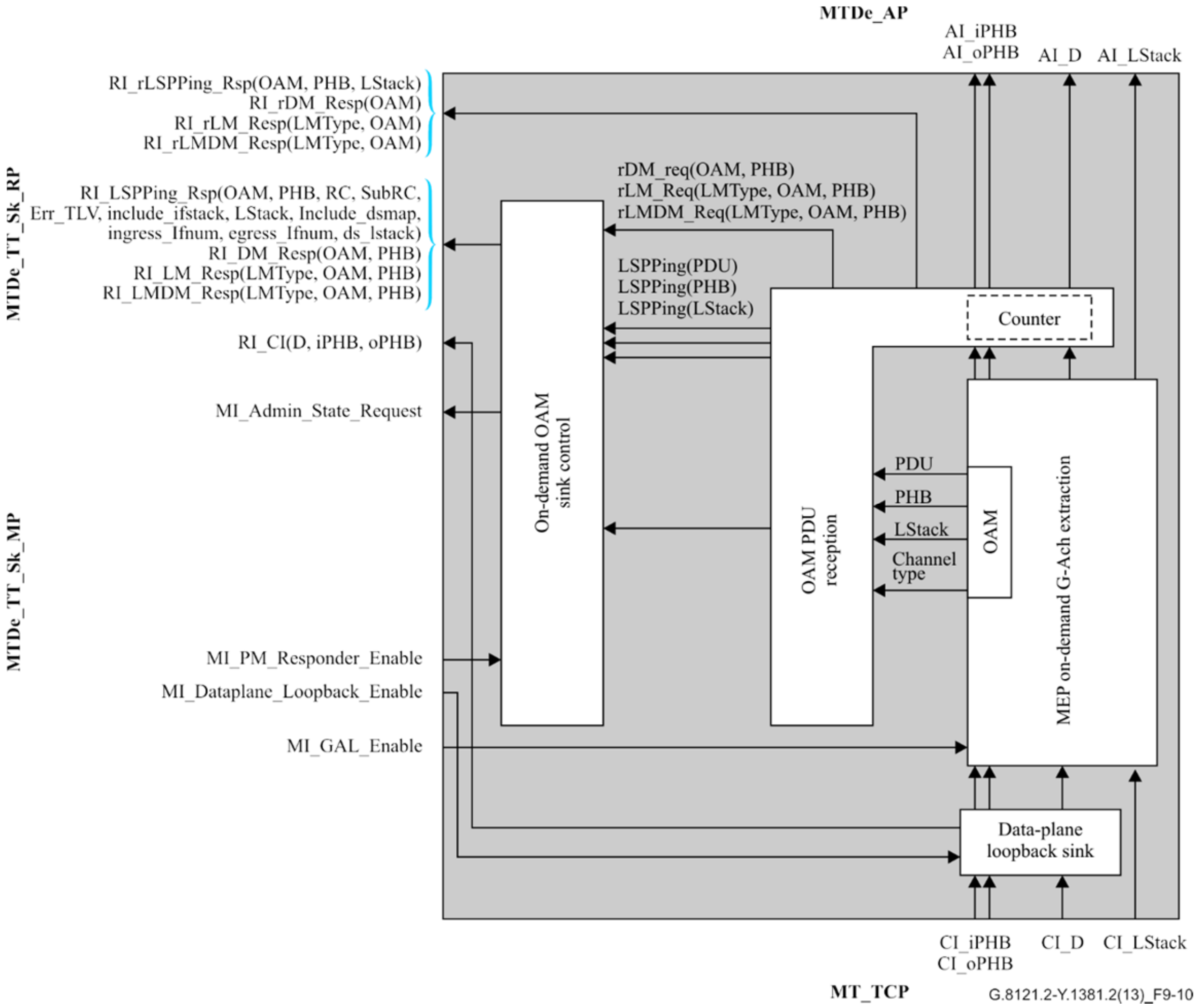

Figure 9-10 – MTDe_TT_Sk process

MEP on-demand G-Ach extraction: See clause 8.1.3 of [ITU-T G.8121].

OAM PDU reception: This contains the following subprocesses as described in clause 8.8: on-demand CV reception; LI reception; PM reception; OAM Demux; PM Demux.

On-demand OAM sink control: This contains the following subprocesses as described in clause 8.8: MEP on-demand CV responder; LI sink control; PM responder.

The location of the counter part of the OAM PDU reception process is shown for illustration only. The exact set of packets to be counted is implementation-specific, as described in [IETF RFC 6374].

Data-plane loopback sink: See clause 8.9.1 of [ITU-T G.8121].

Defects

None.

Consequent actions

None.

Defect correlations

None.

Performance monitoring

None.

9.4.1.2 MTDe to MT adaptation functions (MTDe/MT_A)

The MPLS-TP MEP diagnostic adaptation function (MTDe/MT_A) is described in clause 9.4.1.2 of [ITU-T G.8121].

9.4.2 Diagnostic functions for MIPs

9.4.2.1 MPLS-TP MIP diagnostic trail termination function (MTDi_TT)

The bidirectional MPLS-TP MIP diagnostic trail termination (MTDi_TT) function is performed by a co-located pair of the MPLS-TP trail termination source (MTDi_TT_So) and sink (MTDi_TT_Sk) functions as shown in Figure 9-11.

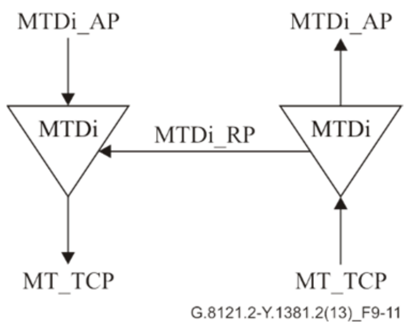

Figure 9-11 – MTDi_TT

9.4.2.1.1 MPLS-TP MIP diagnostic trail termination source function (MTDi_TT_So)

The MTDi_TT_So function adds MPLS-TP OAM to the MT_AI signal at its MT_AP.

The information flow and processing of the MTDi_TT_So function is defined with reference to Figure 9-12.

Symbol

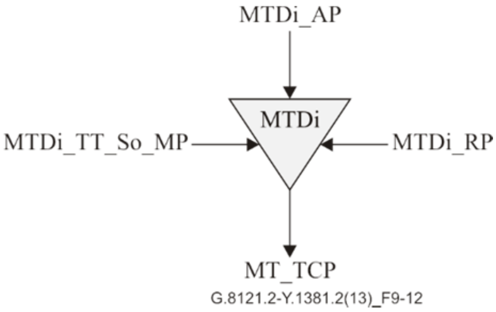

Figure 9-12 – MTDi_TT_So symbol

Interfaces

Table 9-5 – MTDi_TT_So inputs and outputs

Input(s)	Output(s)
MTDi_AP: MTDi_AI_D MTDi_AI_iPHB MTDi_AI_oPHB **MTDi_RP**: MTDi_RI_LSPPing_Rsp MTDi_RI_CI **MTDi_TT_So_MP**: MTDi_TT_So_MI_Target_FEC MTDi_TT_So_MI_Ifnum MTDi_TT_So_MI_MTU MTDi_TT_Sk_MI_GAL_Enable MTDi_TT_So_MI_DP_Loopback_Enable	**MT_CP**: MT_CI_D MT_CI_oPHB MT_CI_iPHB

Processes

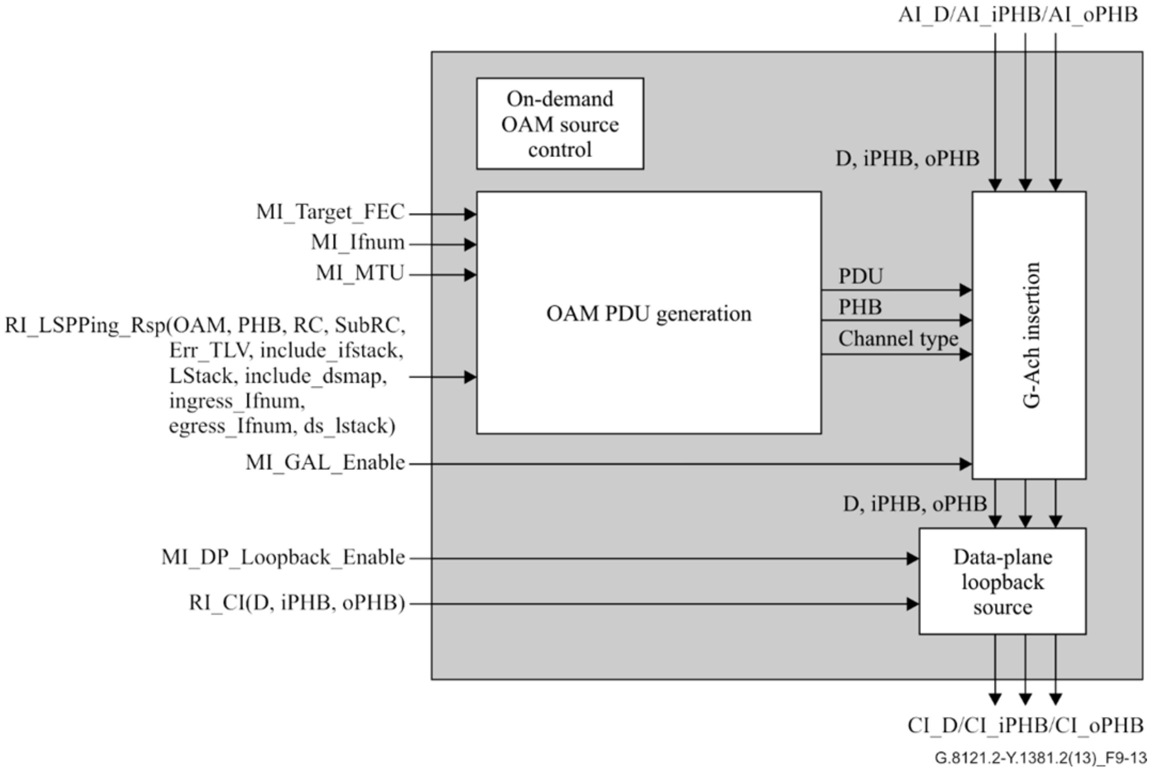

Figure 9-13 – MTDi_TT_So process

The processes associated with the MTDi_TT_So function are depicted in the figure above.

G-ACh insertion: See clause 8.1.2 of [ITU-T G.8121].

OAM PDU generation: This contains the following subprocesses as described in clause 8.8: on-demand CV response generation; OAM Mux.

On-demand OAM source control: This process performs no operations.

Data-plane loopback source: See clause 8.9.1 of [ITU-T G.8121].

Defects

None.

Consequent actions

None.

Defect correlations

None.

Performance monitoring

None.

9.4.2.1.2 MPLS-TP MIP diagnostic trail termination sink function (MTDi_TT_Sk)

The information flow and processing of the MTDi_TT_Sk function is defined with reference to Figure 9-14.

Symbol

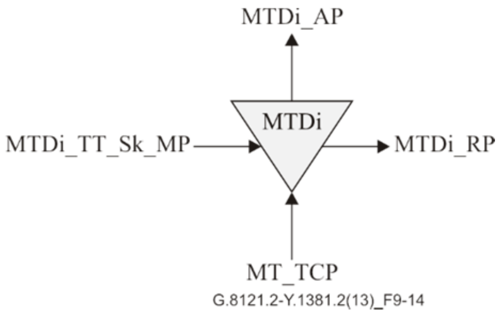

Figure 9-14 – MTDi_TT_Sk symbol

Interfaces

Table 9-6 – MTDi_TT_Sk inputs and outputs

Input(s)	Output(s)
MT_TCP: MT_CI_D MT_CI_iPHB MT_CI_oPHB MT_CI_LStack **MTDi_TT_Sk_MP**: MTDi_TT_Sk_MI_FEC_Checking MTDi_TT_Sk_MI_GAL_Enable MTDi_TT_Sk_MI_DP_Loopback_Enable	**MTDi_AP**: MTDi_AI_D MTDi_AI_iPHB MTDi_AI_oPHB MTDi_AI_LStack **MTDi_RP**: MTDi_RI_LSPPing_Rsp MTDi_RI_CI

Processes

The processes associated with the MTDi_TT_Sk function are depicted in Figure 9-15.

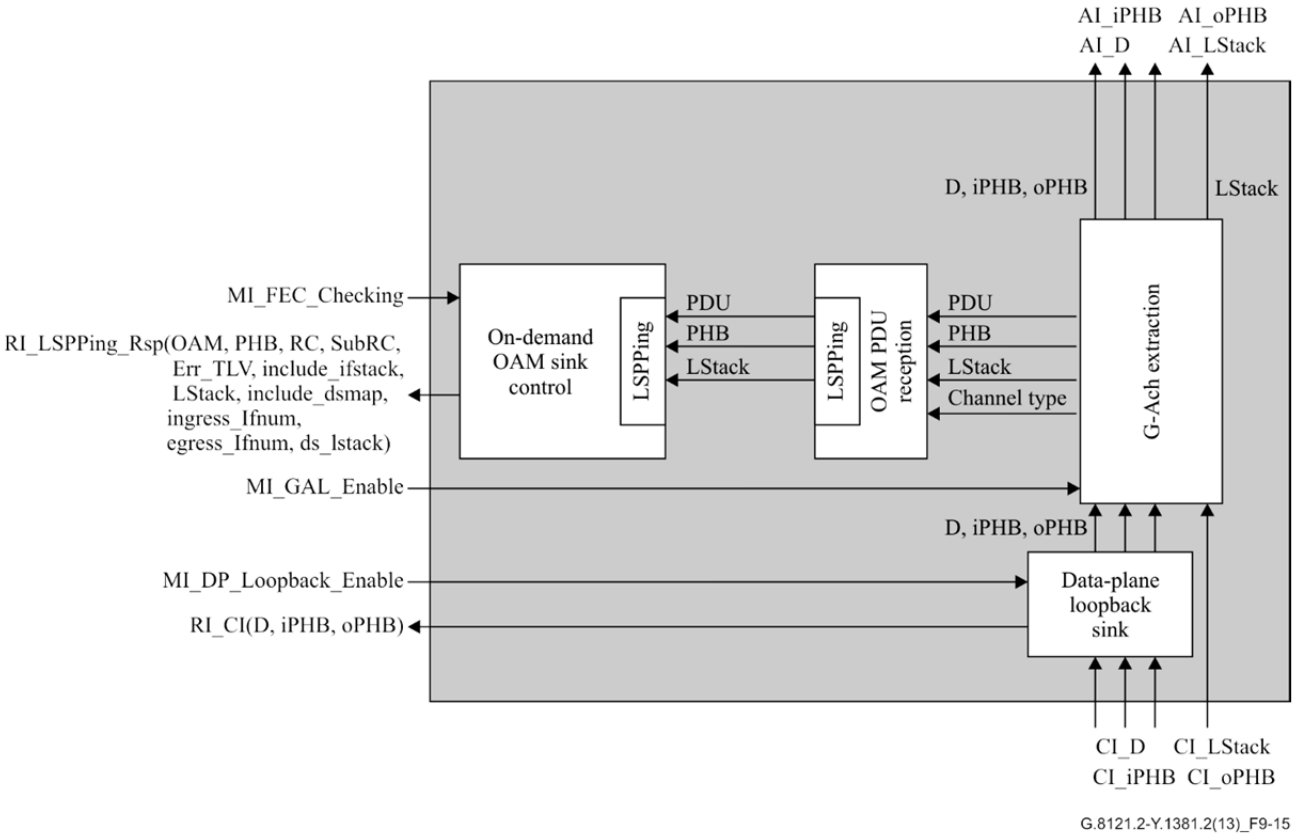

Figure 9-15 – MTDi_TT_Sk process

G-ACh extraction: See clause 8.1.3 of [ITU-T G.8121].

OAM PDU reception: This contains the following subprocesses as described in clause 8.8: on-demand CV reception; OAM Demux.

On-demand OAM sink control: This contains the following subprocesses as described in clause 8.8: MIP on-demand CV responder.

Data-plane loopback sink: See clause 8.9.1 of [ITU-T G.8121].

Defects

None.

Consequent actions

None.

Defect correlations

None.

Performance monitoring

None.

9.4.2.2 MPLS-TP MIP diagnostic adaptation function (MTDi/MT_A)

The MPLS-TP MIP diagnostic adaptation function (MTDi/MT_A) is described in clause 9.4.2.2 of [ITU-T G.8121].

10 MPLS-TP to non-MPLS-TP client adaptation functions

These atomic functions are defined in clause 10 of [ITU-T G.8121].

11 Non-MPLS-TP server to MPLS-TP adaptation functions

These atomic functions are defined in clause 11 of [ITU-T G.8121]. They use the OAM protocol-specific AIS insert process and LCK generation process as defined in clauses 8.6.2 and 8.6.3.

Appendix I

OAM process and subprocesses

(This appendix does not form an integral part of this Recommendation.)

Table I.1 indicates the relationship between processes and subprocesses and where these (sub)processes are implemented to the termination functions (MT_TT, MTDe_TT and MTDi_TT).

Table I.1 – OAM process and subprocesses

Process	Subprocesses	MT_TT	MTDe_TT	MTDi_TT
Proactive OAM Source control	CCCV generation	Yes		
	Proactive PM source control	Yes		
Proactive OAM sink control	CCCV reception	Yes		
	LCK/AIS reception	Yes		
	Proactive PM sink control	Yes		
	PM responder	Yes		
On-demand OAM source control	On-demand CV control		Yes	
	LI source control		Yes	
	On-demand PM control		Yes	
On-demand OAM sink control	MIP on-demand CV responder			Yes
	MEP on-demand CV responder		Yes	
	LI sink control		Yes	
	PM responder		Yes	
OAM PDU generation	On-demand CV request generation		Yes	
	On-demand CV response generation		Yes	Yes
	OAM Mux	Yes	Yes	Yes
	LI generation		Yes	
	PM Mux	Yes	Yes	
	PM generation	Yes	Yes	
OAM PDU reception	Session Demux	Yes		
	On-demand CV reception		Yes	Yes
	OAM Demux	Yes	Yes	Yes
	LI reception		Yes	
	PM Demux	Yes	Yes	
	PM reception	Yes	Yes	

ITU-T Y-SERIES RECOMMENDATIONS

GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS AND NEXT-GENERATION NETWORKS

GLOBAL INFORMATION INFRASTRUCTURE	
General	Y.100–Y.199
Services, applications and middleware	Y.200–Y.299
Network aspects	Y.300–Y.399
Interfaces and protocols	Y.400–Y.499
Numbering, addressing and naming	Y.500–Y.599
Operation, administration and maintenance	Y.600–Y.699
Security	Y.700–Y.799
Performances	Y.800–Y.899
INTERNET PROTOCOL ASPECTS	
General	Y.1000–Y.1099
Services and applications	Y.1100–Y.1199
Architecture, access, network capabilities and resource management	Y.1200–Y.1299
Transport	**Y.1300–Y.1399**
Interworking	Y.1400–Y.1499
Quality of service and network performance	Y.1500–Y.1599
Signalling	Y.1600–Y.1699
Operation, administration and maintenance	Y.1700–Y.1799
Charging	Y.1800–Y.1899
IPTV over NGN	Y.1900–Y.1999
NEXT GENERATION NETWORKS	
Frameworks and functional architecture models	Y.2000–Y.2099
Quality of Service and performance	Y.2100–Y.2199
Service aspects: Service capabilities and service architecture	Y.2200–Y.2249
Service aspects: Interoperability of services and networks in NGN	Y.2250–Y.2299
Enhancements to NGN	Y.2300–Y.2399
Network management	Y.2400–Y.2499
Network control architectures and protocols	Y.2500–Y.2599
Packet-based Networks	Y.2600–Y.2699
Security	Y.2700–Y.2799
Generalized mobility	Y.2800–Y.2899
Carrier grade open environment	Y.2900–Y.2999
FUTURE NETWORKS	Y.3000–Y.3499
CLOUD COMPUTING	Y.3500–Y.3999

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

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, 2014