

---

**Question(s):** 10, 14/15

15-26 February 2016

**TD**

**Source:** Editor G.8121.2/Y.1381.2

**Title:** Draft revised Recommendation ITU-T G.8121.2/Y.1381.2 (for Consent, 26 February 2016)

---

**Abstract**

This document provides the draft revised G.8121.2/Y.1381.2. (for consent, 26 February 2016)

**Summary of major updates from version 1 (G.8121.2 (2013))**

- Fixed some behaviors in OAM related processes in clause 8.8, e.g. adding OAM\_Tool and intermediate request to on-demand loss measurement
- Rename ODCV by CV
- Update clause 9.2.1 and 9.4.1 in align with clause 8.8
- Fixed some texts and behaviors in OAM related processes in clause 8.8.
- Updates some diagrams for MT\_TT\_(So and Sk), MTDe\_TT\_(So and Sk), and MTDi\_TT\_So
- Fixed some incorrect diagrams (including SDL diagrams)

---

**Contact:** Yuji Tochio  
Fujitsu  
Japan

Tel: +81-44-754-8829  
Email: [tochio@jp.fujitsu.com](mailto:tochio@jp.fujitsu.com)

**Attention:** This is not a publication made available to the public, but **an internal ITU-T Document** intended only for use by the Member States of ITU, by ITU-T Sector Members and Associates, and their respective staff and collaborators in their ITU related work. It shall not be made available to, and used by, any other persons or entities without the prior written consent of ITU-T.

## **Draft revised Recommendation ITU-T G.8121.2/Y.1381.2**

### **Characteristics of MPLS-TP equipment functional blocks supporting ITU-T G.8113.2/Y.1372~~3~~.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 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.~~1373~~1372.2.

#### **Keywords**

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

## Draft revised Recommendation ITU-T G.8121.2/Y.1381.2

### Characteristics of MPLS-TP equipment functional blocks supporting ITU-T G.8113.2/Y.~~1373~~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 as such.

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)<sup>1</sup>.

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), <i>Generic functional architecture of transport networks</i> .                                      |
| [ITU-T G.806] | Recommendation ITU-T G.806 (2012), <i>Characteristics of transport equipment – Description methodology and generic functionality</i> . |

---

<sup>1</sup> This ITU-T Recommendation is intended to be aligned with the IETF MPLS RFCs normatively referenced by this Recommendation.

- [ITU-T G.8101] Recommendation ITU-T G.8101/[Y.1355](#) (2015), *Terms and definitions for MPLS transport profile*.
- [ITU-T G.8110.1] Recommendation ITU-T G.8110.1/Y.1370.1 (2011), *Architecture of MPLS Transport Profile (MPLS-TP) layer networks*.
- [ITU-T G.8113.2] Recommendation ITU-T G.8113.2/[Y.1372.2](#) (2013~~5~~), *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~~5~~), *Characteristics of MPLS-TP network 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)*.
- [\[IETF RFC 6370\]](#) [IETF RFC 6370 \(2011\), MPLS Transport Profile \(MPLS-TP\) Identifiers](#)<sup>[YT1]</sup>
- [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*.
- [IETF RFC 6435] IETF RFC 6435 (2011), *MPLS Transport Profile Lock Instruct and Loopback Functions*, plus Errata 3429 (2013) and 4017 (2014)

### 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** characteristic information: [ITU-T G.805]
- 3.1.4** client/server relationship: [ITU-T G.805]
- 3.1.5** connection: [ITU-T G.805]
- 3.1.6** connection point: [ITU-T G.805]
- 3.1.7** layer network : [ITU-T G.805]
- 3.1.8** matrix: [ITU-T G.805]

- 3.1.9 network: [ITU-T G.805]
- 3.1.10 network connection: [ITU-T G.805]
- 3.1.11 reference point: [ITU-T G.805]
- 3.1.12 subnetwork: [ITU-T G.805]
- 3.1.13 subnetwork connection: [ITU-T G.805]
- 3.1.14 termination connection point : [ITU-T G.805]
- 3.1.15 trail: [ITU-T G.805]
- 3.1.16 trail termination: [ITU-T G.805]
- 3.1.17 transport: [ITU-T G.805]
- 3.1.18 transport entity: [ITU-T G.805]
- 3.1.19 transport processing function: [ITU-T G.805]
- 3.1.20 unidirectional connection: [ITU-T G.805]
- 3.1.21 unidirectional trail: [ITU-T G.805]
- 3.1.22 label: [ITU-T G.8101]
- 3.1.23 label stack: [ITU-T G.8101]
- 3.1.2 MPLS label stack: [ITU-T G.8101]
- 3.1.24 label switched path: [ITU-T G.8101]
- 3.1.25 Bottom of Stack: [ITU-T G.8101]
- 3.1.26 Time To Live: [ITU-T G.8101]
- 3.1.27 Label value: [ITU-T G.8101]
- 3.1.28 Per-Hop Behaviour: [ITU-T G.8101]
- 3.1.29 Associated Channel Header: [ITU-T G.8101]
- 3.1.30 Generic Associated Channel: [ITU-T G.8101]
- 3.1.31 G-ACh Label: [ITU-T G.8101]
- 3.1.32 traffic class: [ITU-T G.8101]
- 3.1.33 Explicitly TC-encoded-PSC LSP: [ITU-T G.8101]
- 3.1.34 label inferred PHB scheduling class LSP: [ITU-T G.8101]

## **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
CCCV	Continuity Check and Connectivity Verification

CC/CV	Continuity Check or 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 <del>ODCV</del> <del>On-Demand Connectivity</del> <del>Verification</del>
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
TC	Traffic Class
TS	Timestamp
TSFmt	Timestamp Format
TLV	Type Length Value
TTL	Time-To-Live

[Note following abbreviations will be included in align with in forced version]

<del>AIS</del>	<del>Alarm Indication Signal</del>
<del>APS</del>	<del>Automatic Protection Switching</del>
<del>CoS</del>	<del>Class of Service</del>
<del>CSF</del>	<del>Client Signal Fail</del>
<del>CV</del>	<del>Connectivity Verification</del>
<del>DM</del>	<del>Delay Measurement</del>
<del>EMF</del>	<del>Equipment Management Function</del>
<del>G-ACh</del>	<del>Generic Associated Channel</del>
<del>GAL</del>	<del>G-ACh Label</del>
<del>GFP</del>	<del>Generic Framing Procedure</del>
<del>LCK</del>	<del>Locked</del>
<del>LM</del>	<del>Loss Measurement</del>
<del>LStack</del>	<del>Label Stack</del>
<del>MCC</del>	<del>Maintenance Communication Channel</del>
<del>MEP</del>	<del>Maintenance entity group (MEG) End Point</del>
<del>MIP</del>	<del>Maintenance entity group (MEG) Intermediate Point</del>
<del>MPLS</del>	<del>Multi-Protocol Label Switching</del>
<del>MPLS-TP</del>	<del>Multi-Protocol Label Switching—Transport Profile</del>
<del>MT</del>	<del>Multi-Protocol Label Switching—Transport Profile</del>
<del>MTDe</del>	<del>MPLS-TP MEP Diagnostic function</del>
<del>OAM</del>	<del>Operation, Administration and Maintenance</del>
<del>PDU</del>	<del>Protocol Data Unit</del>
<del>PHB</del>	<del>Per Hop Behaviour</del>
<del>PM</del>	<del>Performance Monitoring</del>
<del>RDI</del>	<del>Remote Detect Indication</del>
<del>SCC</del>	<del>Signalling Communication Channel</del>
<del>SSF</del>	<del>Server Signal Fail</del>
<del>TC</del>	<del>Traffic Class</del>
<del>TLV</del>	<del>Type Length Value</del>
<del>TTL</del>	<del>Time-To-Live</del>

## **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 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 the clause 8.2 in [ITU-T G.8121]*

### **8.3 Queuing process**

*See the clause 8.3 in [ITU-T G.8121]*

### **8.4 MPLS-TP-specific GFP-F processes**

*See the clause 8.4 in [ITU-T G.8121]*

### **8.5 Control Word (CW) processes**

*See the clause 8.5 in [ITU-T G.8121]*

### **8.6 OAM related Processes used by Server adaptation functions**

#### **8.6.1 Selector Process**

*See the clause 8.6.1 in [ITU-T G.8121]*

#### **8.6.2 AIS Insert Process**

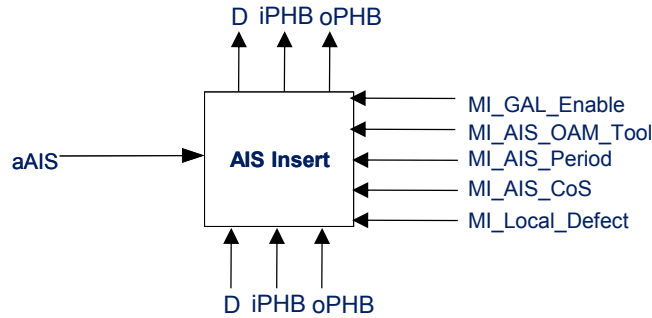
The AIS Insert Process generates MT\_CI traffic units containing the AIS signal. MI\_AIS\_Period specifies the period between successive AIS messages, in seconds between 1 and 20. MI\_AIS\_CoS specifies the priority for AIS messages. MI\_Local\_Defect specifies whether an alternative path is



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.

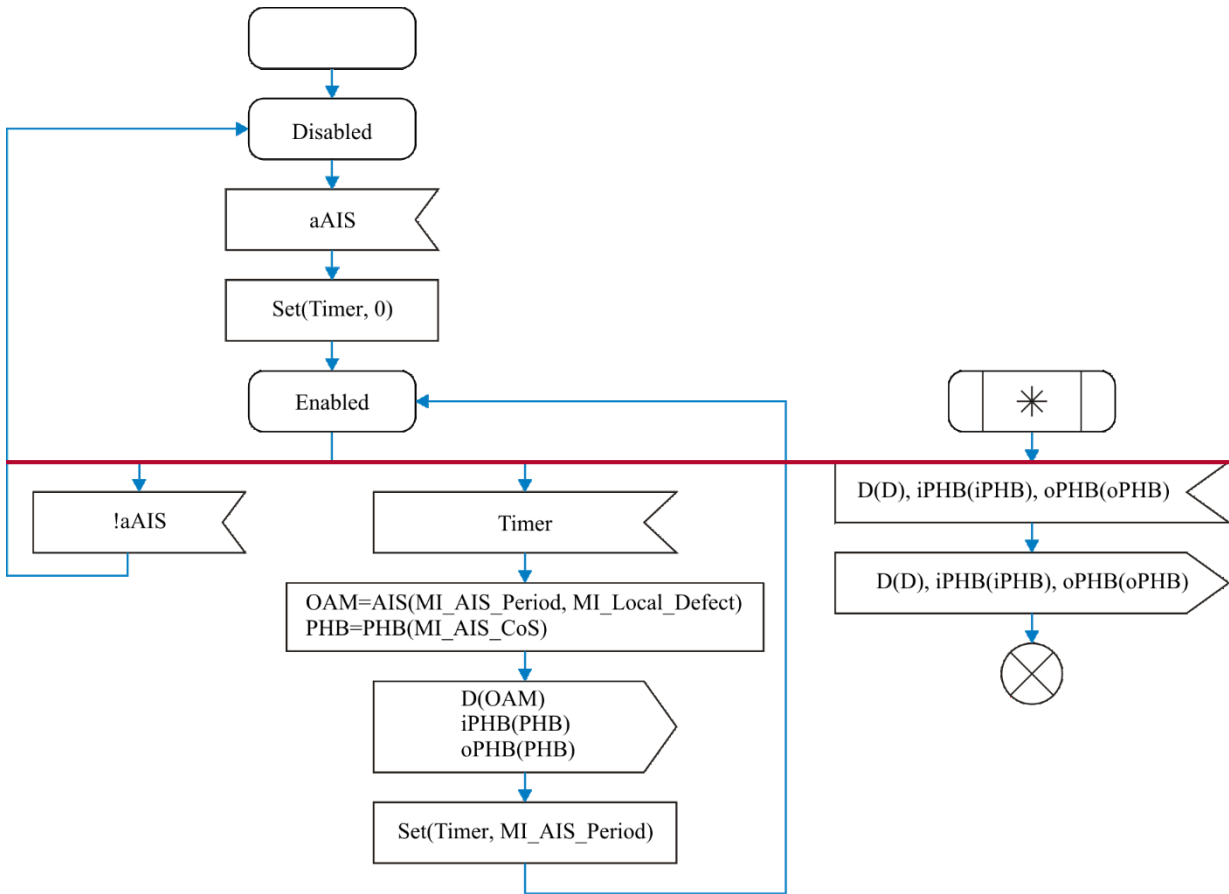
Note: it is expected that MI\_Local\_Defect can be set correctly by the EMF without explicit interaction by the end user. The value can be precomputed as described in [IETF RFC6427].

The AIS Insert Process is described in Clause 8.6.2 in [ITU-T G.8121], and is shown in Figure 8-1

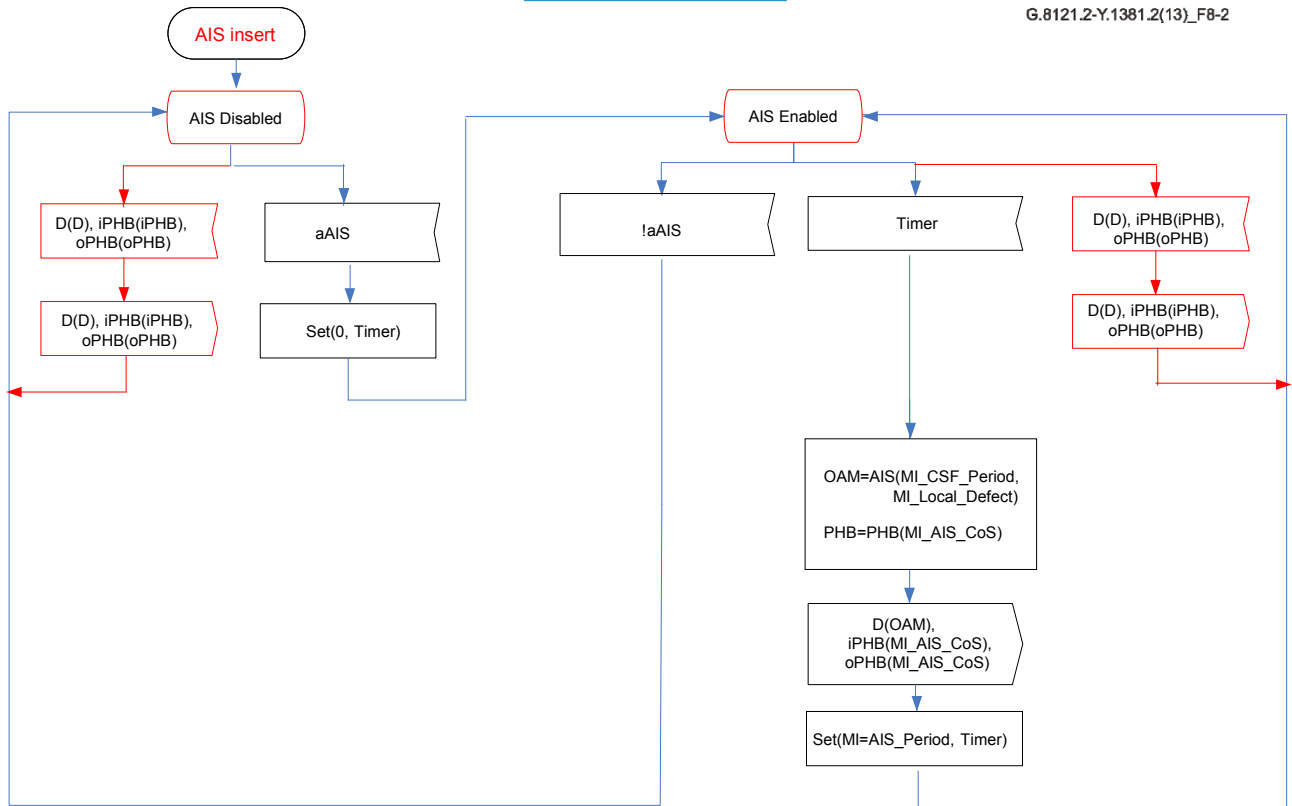


**Figure 8-1/G.8121.2/Y.1381.2 AIS Insert process**

Figure 8-2 defines the behaviour of the AIS Insert Process:



G.8121.2-Y.1381.2(13)\_F8-2



### **Figure 8-2/G.8121.2/Y.1381.2 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 [RFC 6427]. The fields are filled in 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.

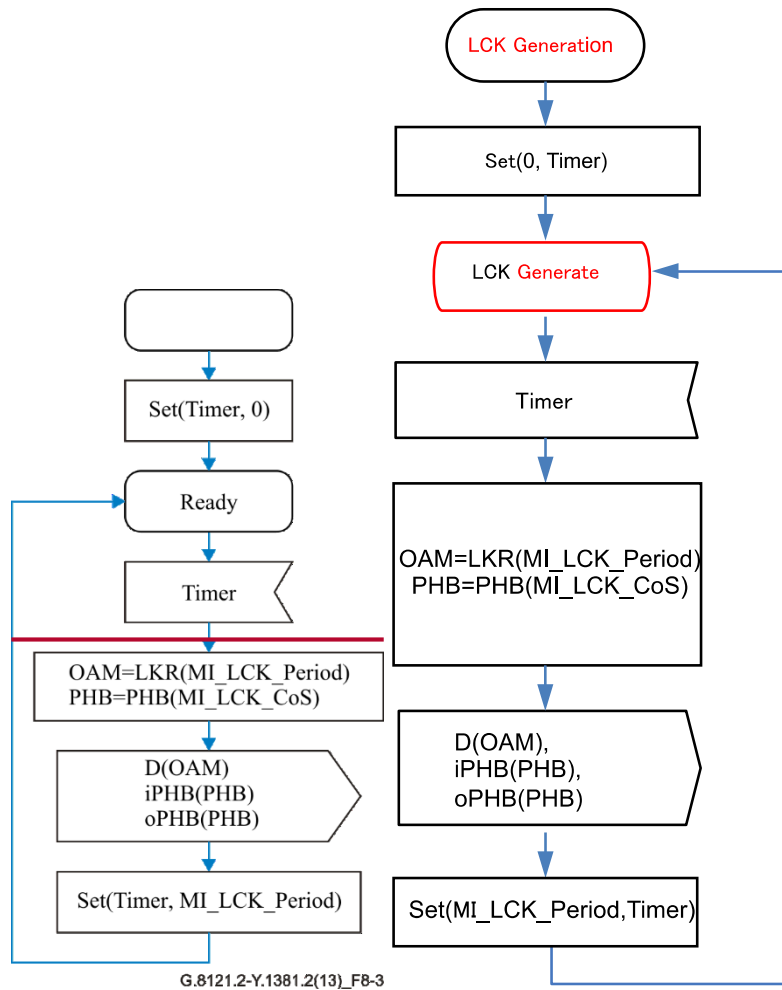
The PHB(CoS) function returns the PHB with the lowest drop precedence within the Class of Service defined by the CoS input parameter.

#### **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/G.8121.2/Y.1381.2 - 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 [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 the clause 8.7.1 in [ITU-T G.8121]*

### **8.7.2 APS Insert and ExtractProcess**

*See the clause 8.7.2 in [ITU-T G.8121]*

### **8.7.3 CSF Insert and Extract Process**

*See the clause 8.7.3 in [ITU-T G.8121]*

## **8.8 Pro-active and on-demand OAM related Processes**

As described in Clause 8.8 of [ITU-T G.8121], there are 6 processes for pro-active 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

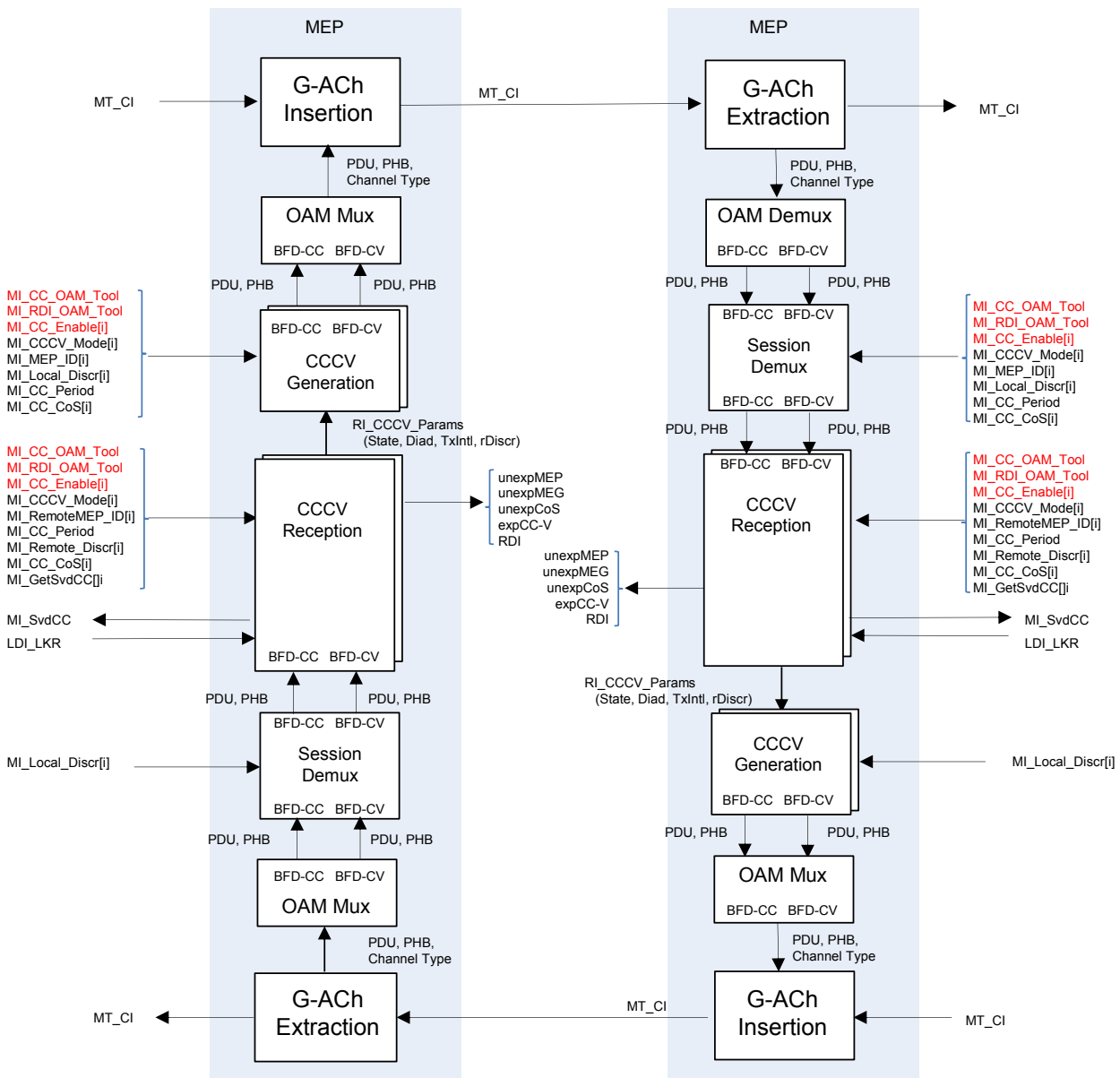
Each of these consists of a number of protocol-specific sub-processes, as described in [ITU-T G.8121]. Appendix I provides the table that indicates the relationship between processes and sub-processes and indicates where these (sub-)processes are implemented in the termination functions (MT\_TT, MTDe\_TT,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

The OAM Mux Sub-process is responsible for multiplexing together (PDU, TTL, PHB) signals from other sub-processes, and passing them to the G-ACh Insertion process along with the appropriate Channel Type. Similarly, the OAM Demux sub-process receives (PDU, PHB, LStack, Channel Type) signals from the G-ACh Extraction process, and passes on the (PDU, PHB, LStack) signals to the other sub-processes as appropriate depending on the Channel Type.

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

### **8.8.1. CC/CV Processes**

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



**Figure 8-4/G.8121.2/Y.1381.2 – Overview of CC/CV processes**

*[Add MI\_CC\_OAM\_Tool, MI\_RDI\_OAM\_Tool]*

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- [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 [RFC 6428].

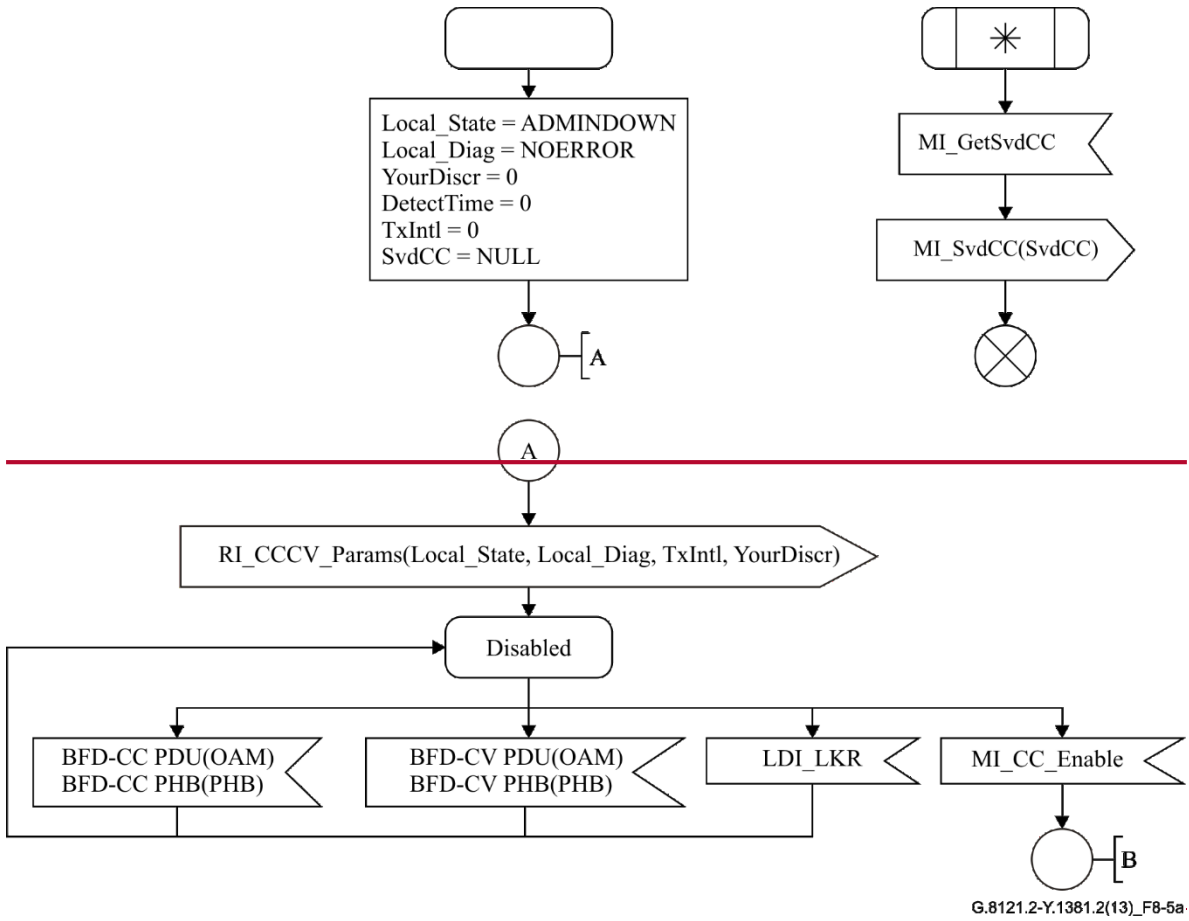
The CCCV generation process sends periodic BFD-CC and BFD-CV messages, when MI\_CC\_Enable is TRUE. There is a separate instance of the process for each corresponding instance of the CCCV reception process. MI\_MEP\_ID and MI\_Local\_Discr specify the local MEP ID (in one of the formats described in [RFC 6428]) 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 Figure 8-5a, Figure 8-5b, and Figure 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.





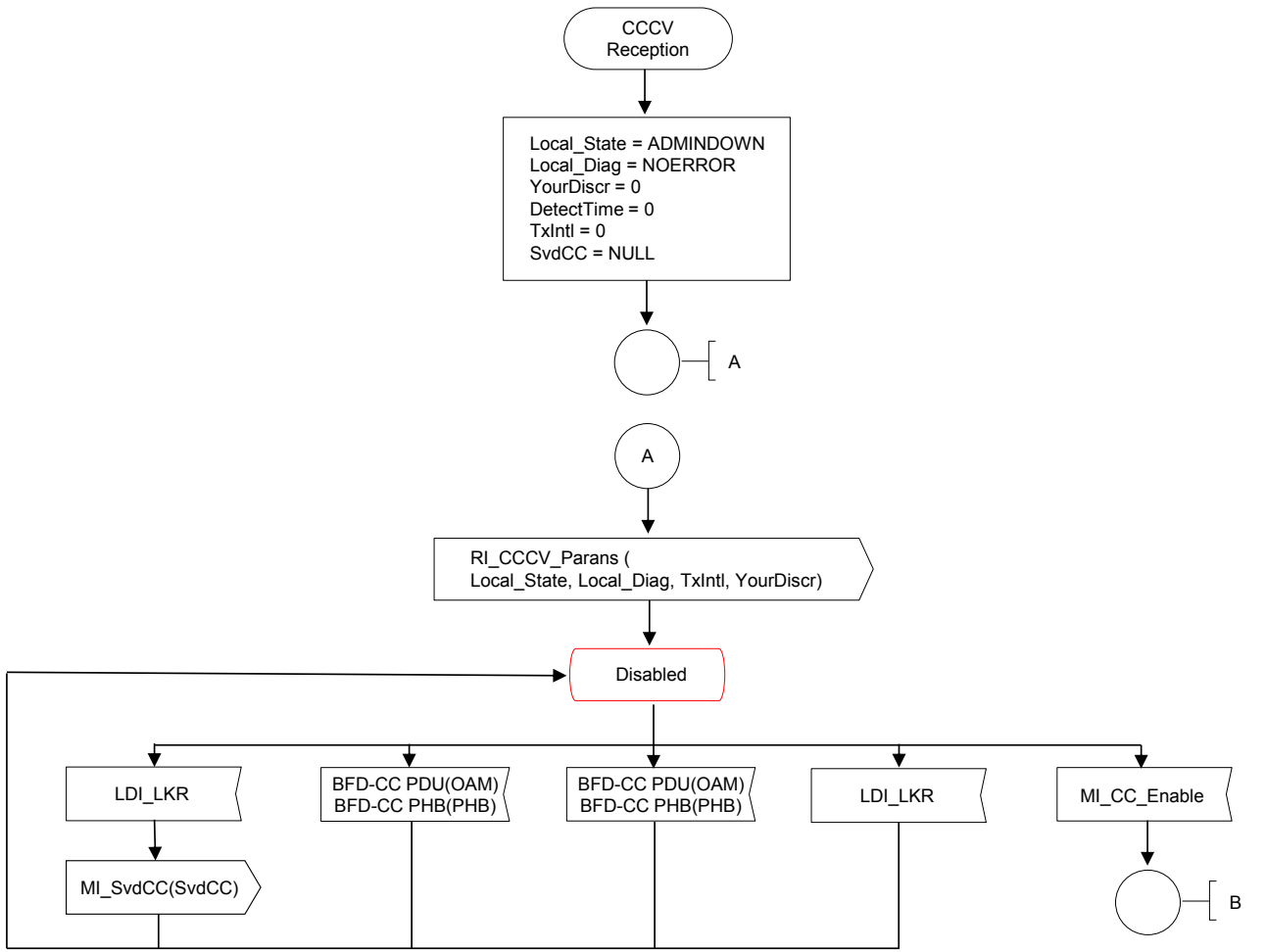
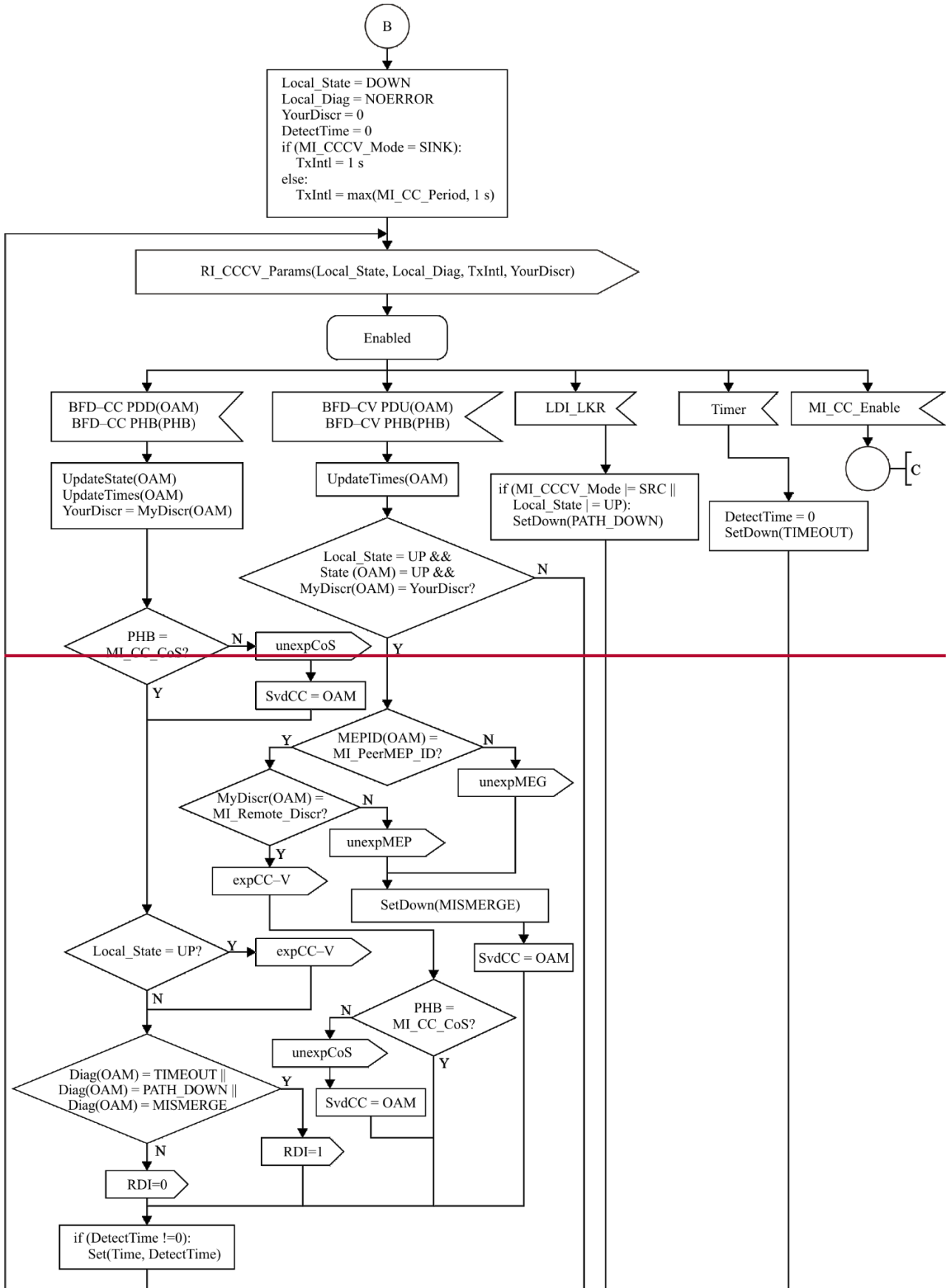


Figure 8-5a/G.8121.2/Y.1381.2 - CCCV Reception Process



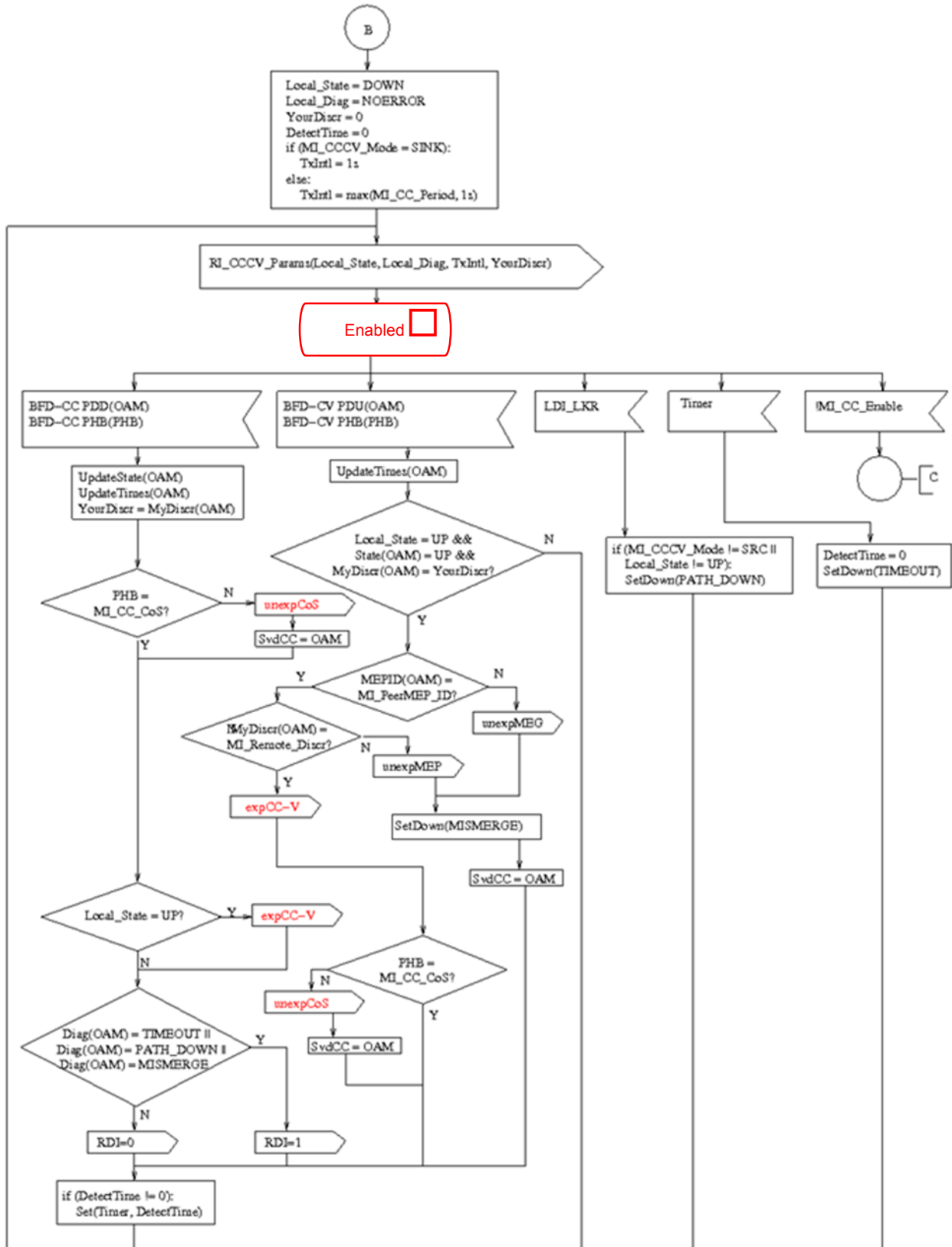
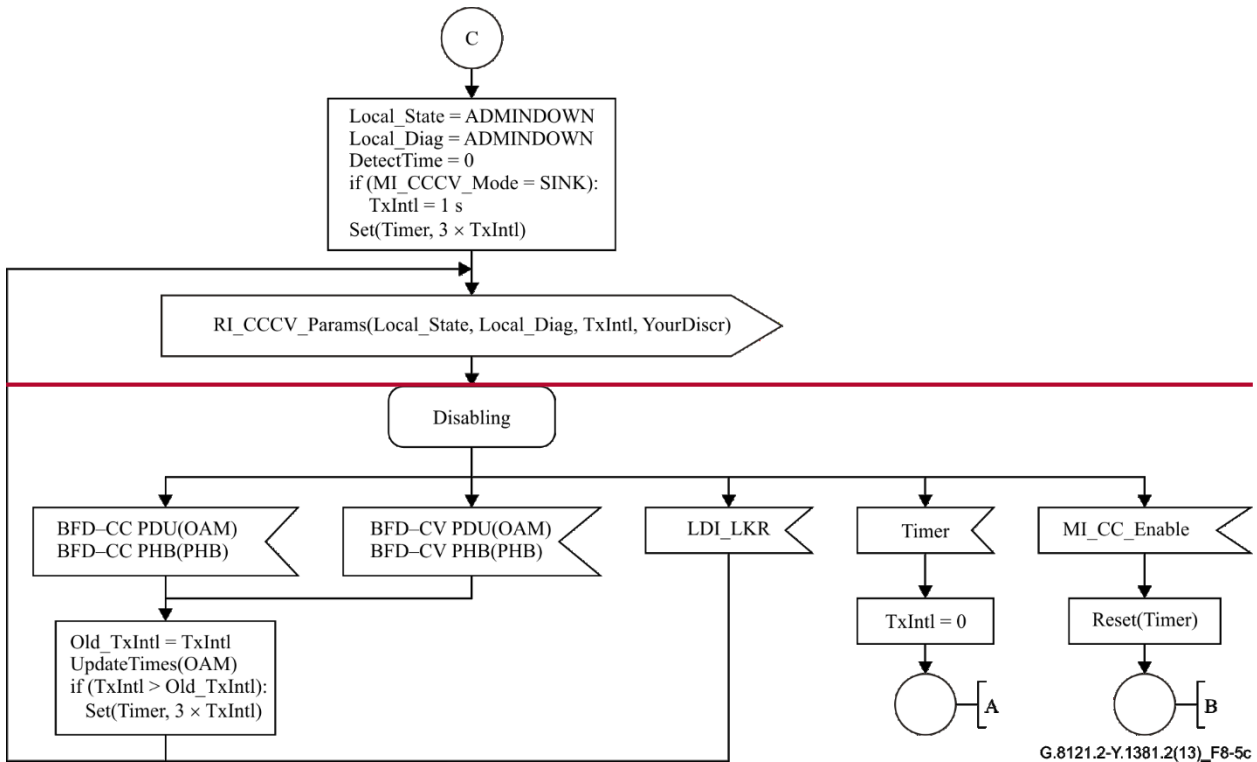
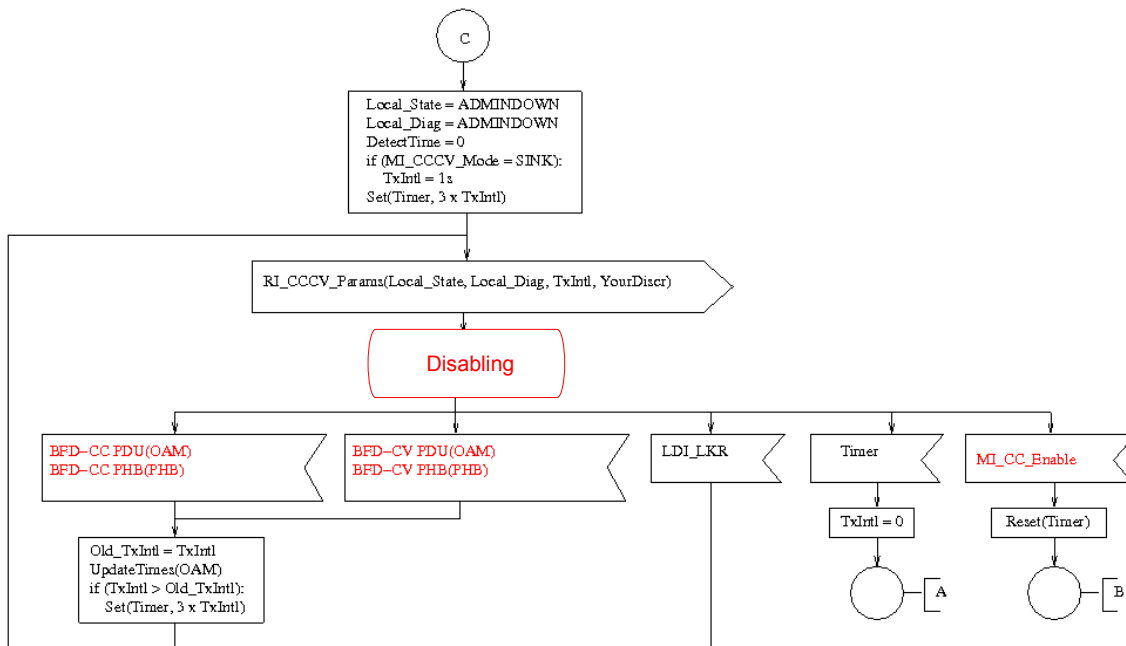


Figure 8-5b/G.8121.2/Y.1381.2 - CCCV Reception Process



G.8121.2-Y.1381.2(13)\_F8-5c



**Figure 8-5c/G.8121.2/Y.1381.2 - CCCV Reception Process**

The values of State and Diag correspond with those in [IETF RFC5880] and [IETF RFC6428]. 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))  
}  
}
```

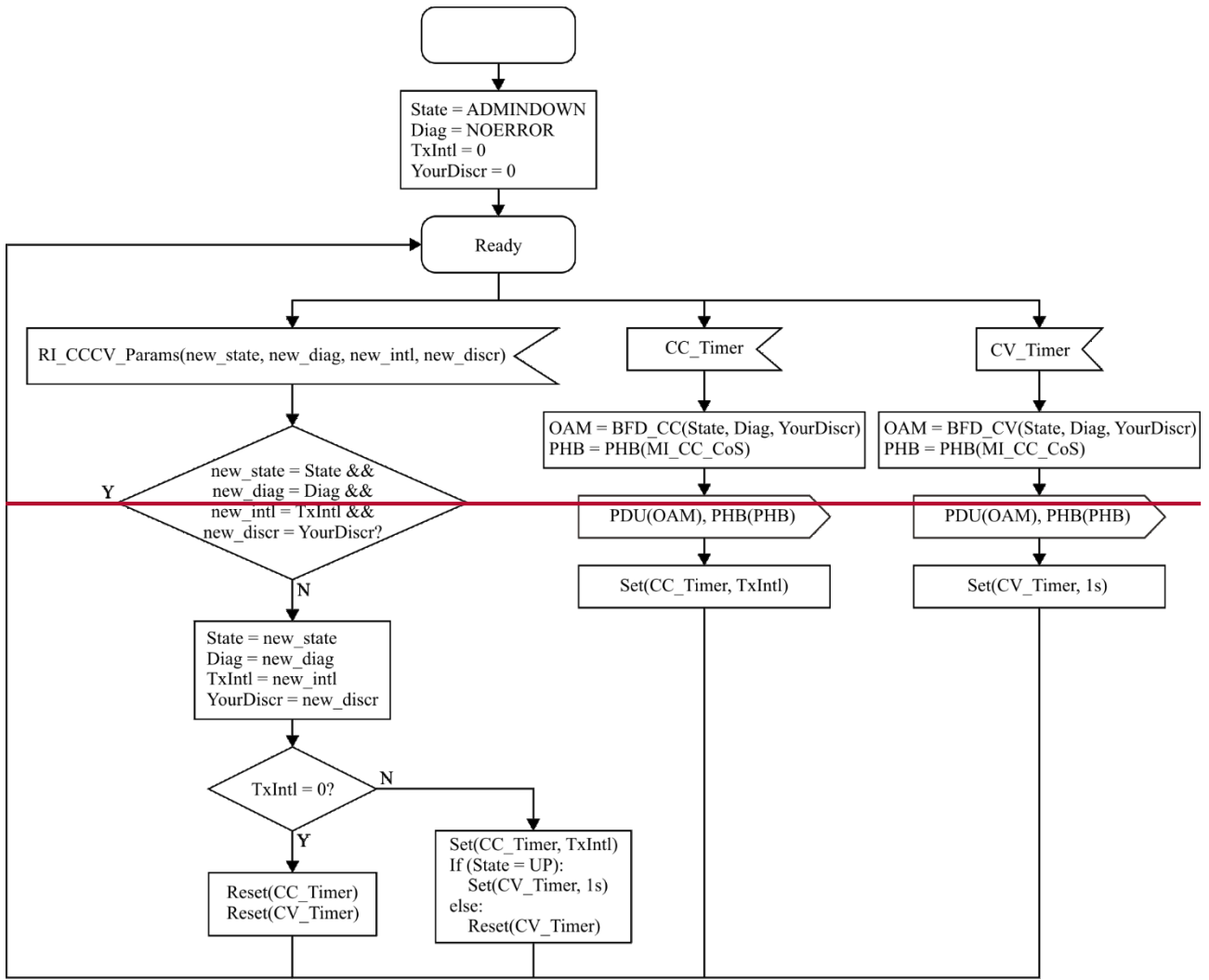
Use of authentication for CC/CV is for further study.

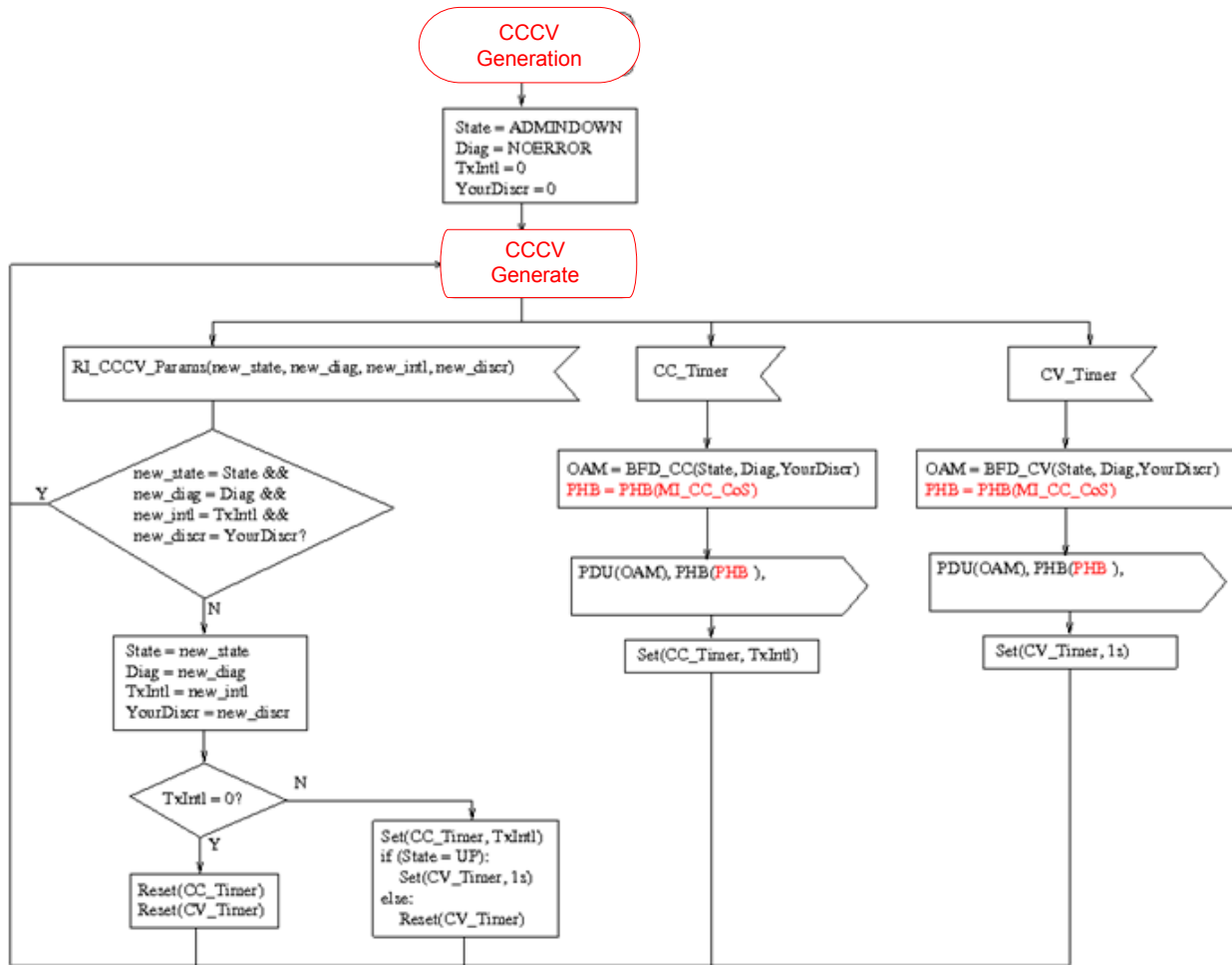
Use of the BFD Poll/Final mechanism for changing the value of TxIntl is for further study.

#### **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 1s.

The CCCV Generation Process is described in Figure 8-6.





**Figure 8-6/G.8121.2/Y.1381.2 - 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 dependent 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 a 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 on Figure 8-7.

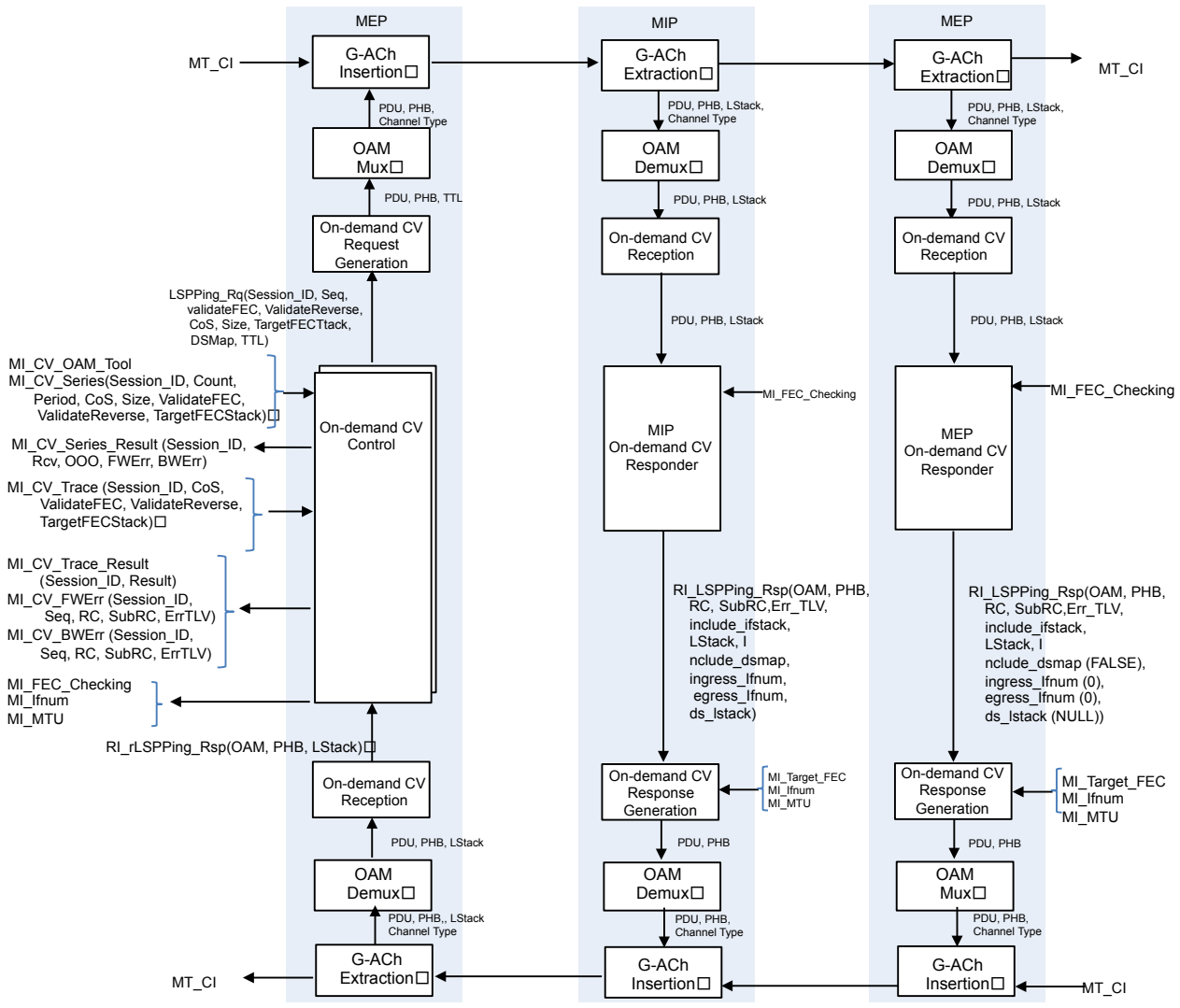


Figure 8-7/G.8121.2/Y.1381.2 - overview of the On-demand CV Processes  
~~Remove "OD" per C839~~

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\_⊖CV\_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 RFC4379] and [IETF RFC6426].

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].

```
⊖CV_Validate (OAM, LStack_in[], FECStack[], MP_Type) {  
    rc = 0  
    sub_rc = 0  
    err_TLV = NULL  
    done = FALSE  
    include_ifstack = FALSE  
    include_dsmmap = 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_dsmmap = 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 sub-types ‘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 [IETF RFC4379] section 4.4.1.
- `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.~~ the `IF_NUM` (as defined in [IETF RFC 6370]) of the interface over which the packet would be forwarded, if the TTL was high enough, or 0 if no such interface was found or it is not MPLS-enabled. [\[C.1074#7\]](#)

### 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.

The parameters are as follows:

- Session\_ID: The session ID passed to MI\_CV\_Series() or MI\_CV\_Trace().
- Rcv: Number of responses received
- OOO: Number of responses that were out of order
- FWErr: Number of responses in which forward errors (i.e. from Control to Responder) were indicated.
- BWErr: Number of responses in which backward errors (i.e. from Responder to Control) were indicated.
- Result: Array of results corresponding to received responses; each element contains the Sequence Number, Return Code, Return Subcode, Downstream Mapping TLV (if any) and Interface and Label Stack TLV (if any) extracted from the corresponding response.

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 on the Responder-to-Control direction). These signals include the session ID passed to MI\_CV\_Series() or MI\_ODCV\_Trace(), and the Sequence Number, Return Code, Return Subcode, and Errored TLVs extracted from the response that contained the error. 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 source and sink processes is shown in the figures below (Figure 8-8a is for source process, and Figures 8b, 8c and 8d are for sink process, and 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 a MIP (ie 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, 3 attempts are made to resend the request, before giving up and reporting any results collected so far.

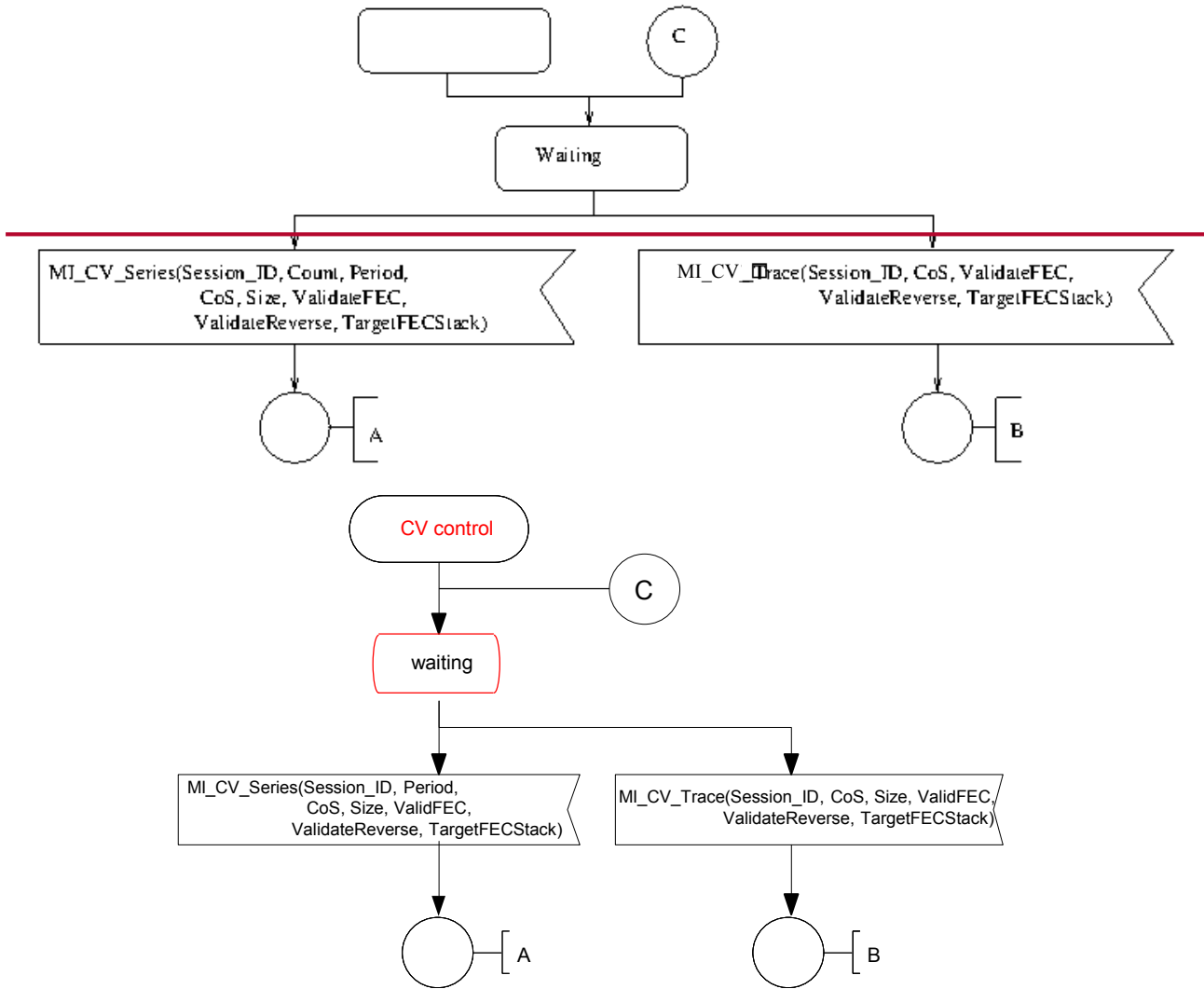


Figure 8-8a/G.8121.2/Y.1381.2 - On-demand CV Control Source Process

[Remove "OD" per C839 for all of Fig8-8s]



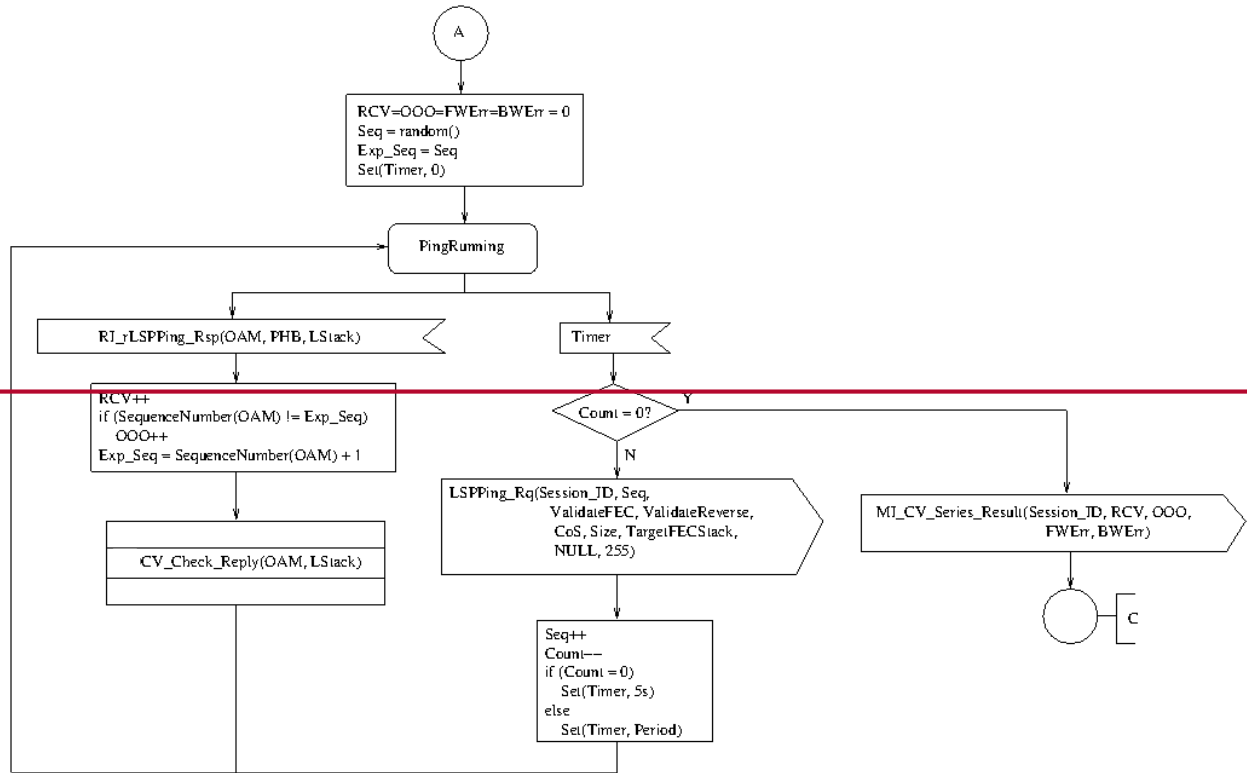
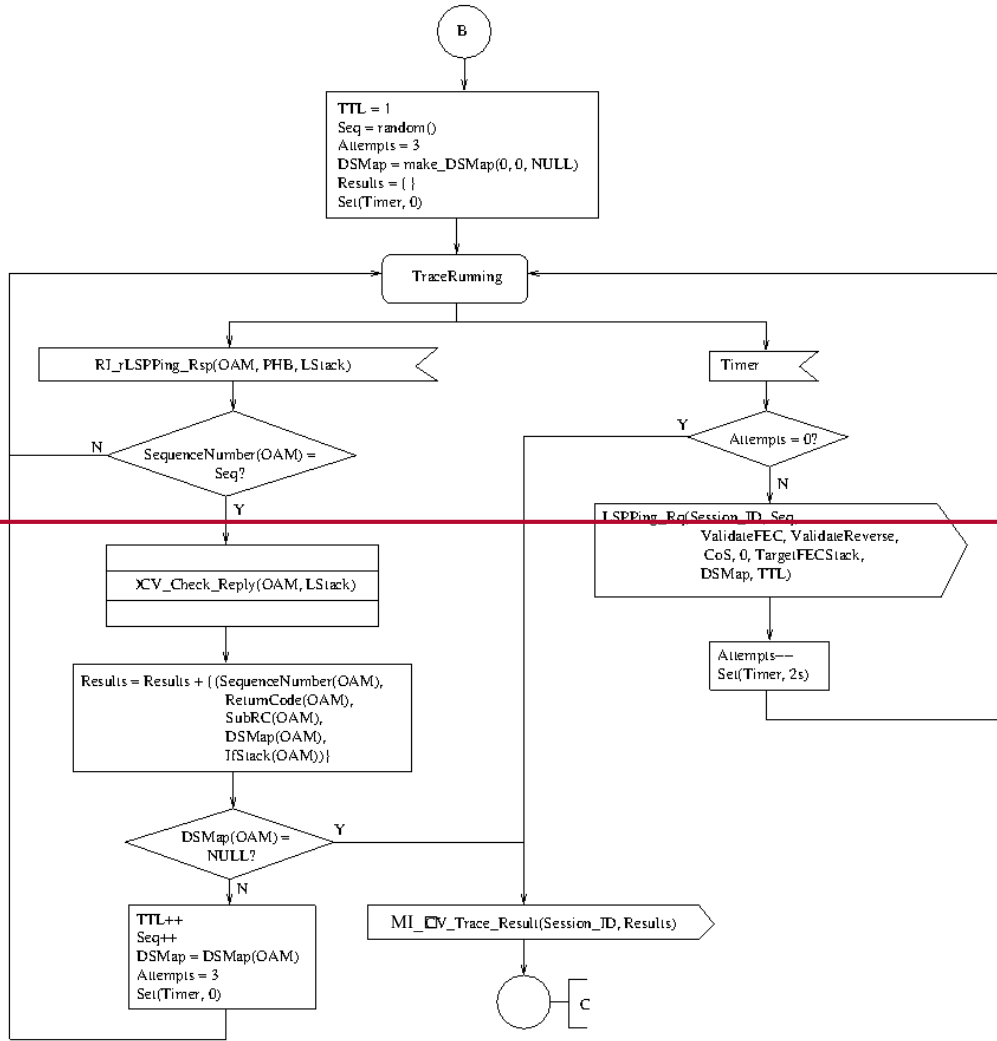


Figure 8-8b/G.8121.2/Y.1381.2 - On-demand CV Control Sink Process



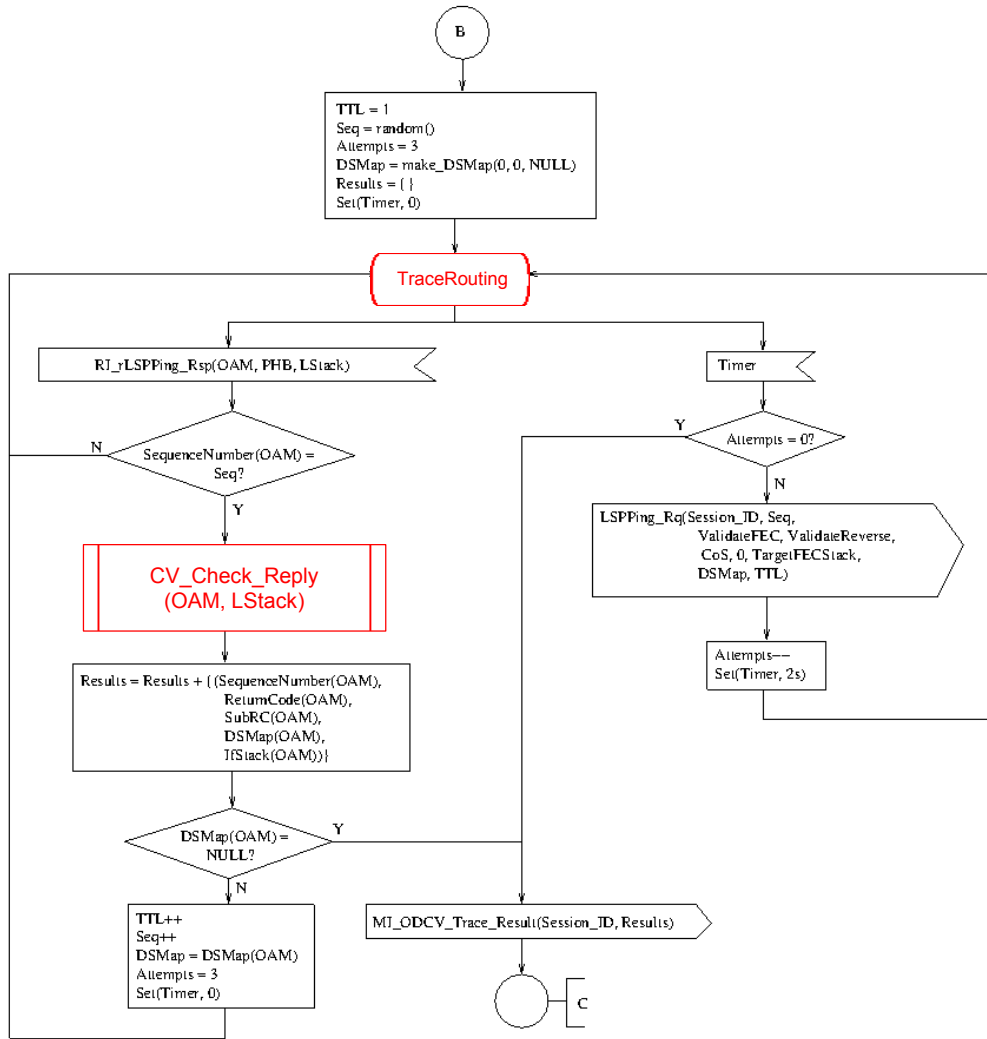
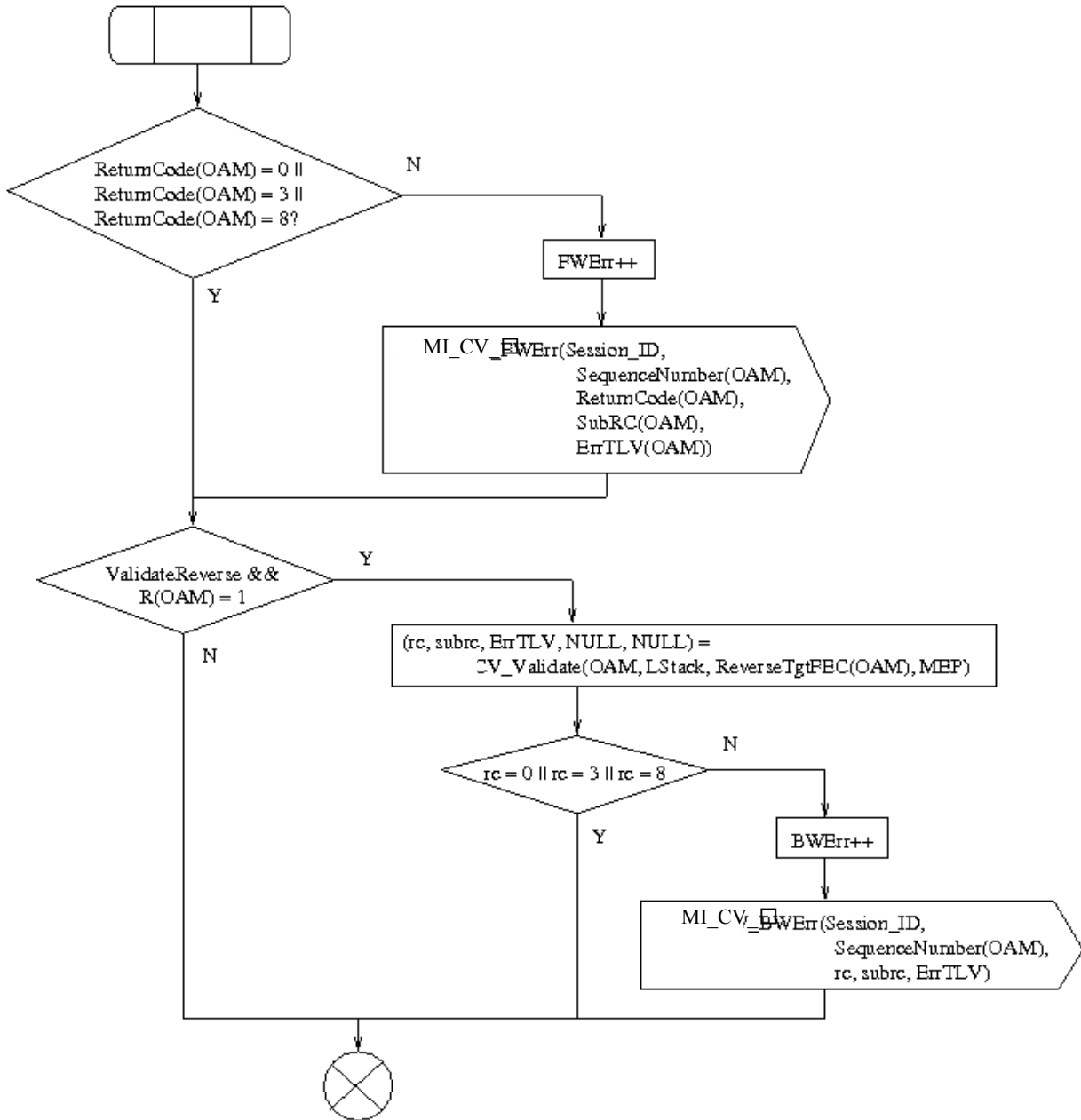
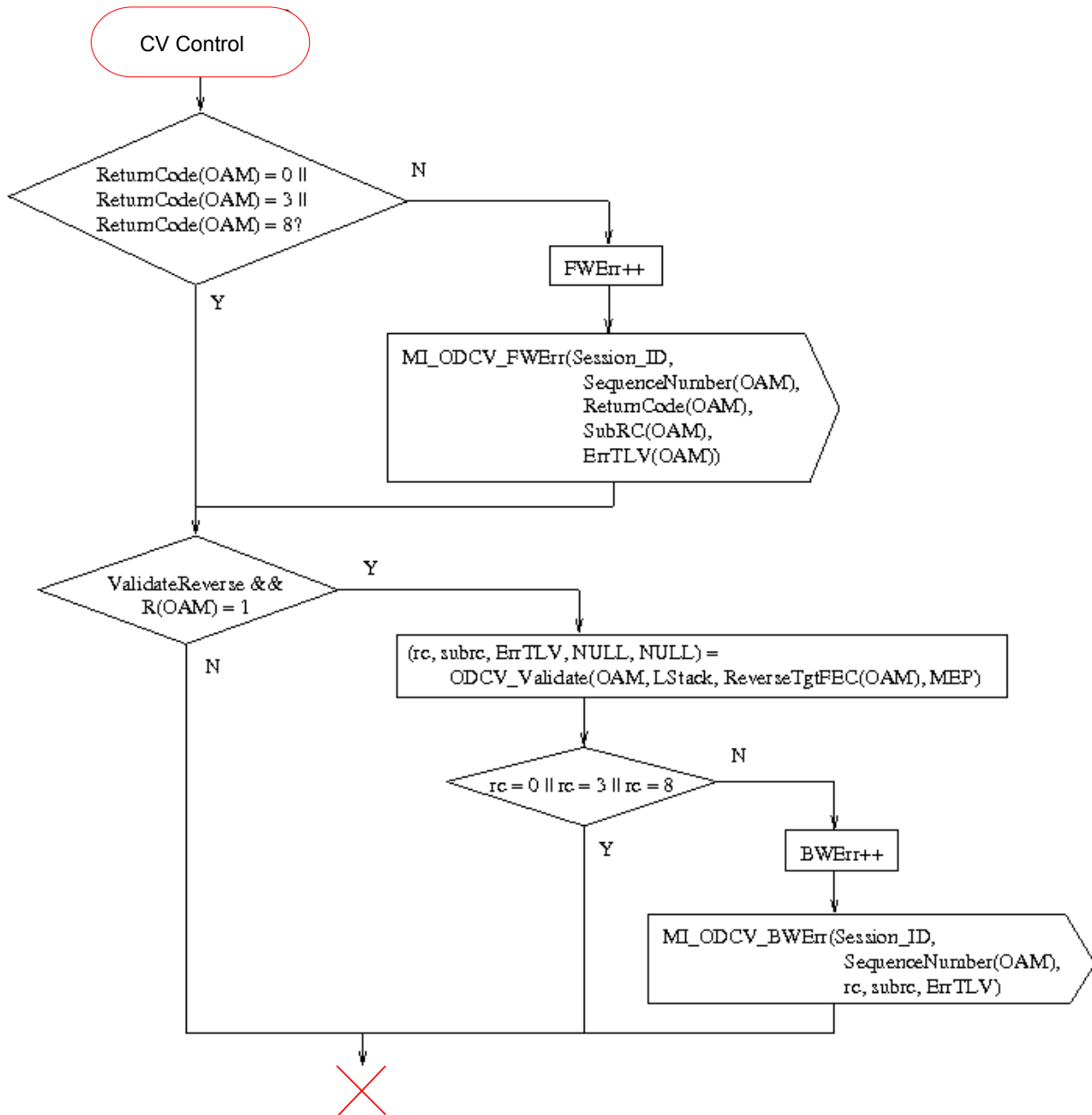


Figure 8-8c/G.8121.2/Y.1381.2 - On-demand CV Control Sink Process

Process CV\_Check\_Reply(OAM, LStack)





**Figure 8-8d/G.8121.2/Y.1381.2 - On-demand CV Control Sink 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 Set to the size of the largest MPLS frame (including label stack) that fits on the egress interface [C.1074#7]
- 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 RFC4379]. 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 (Source control)

The On-demand CV Request Generation process is shown in Figure 8-9:

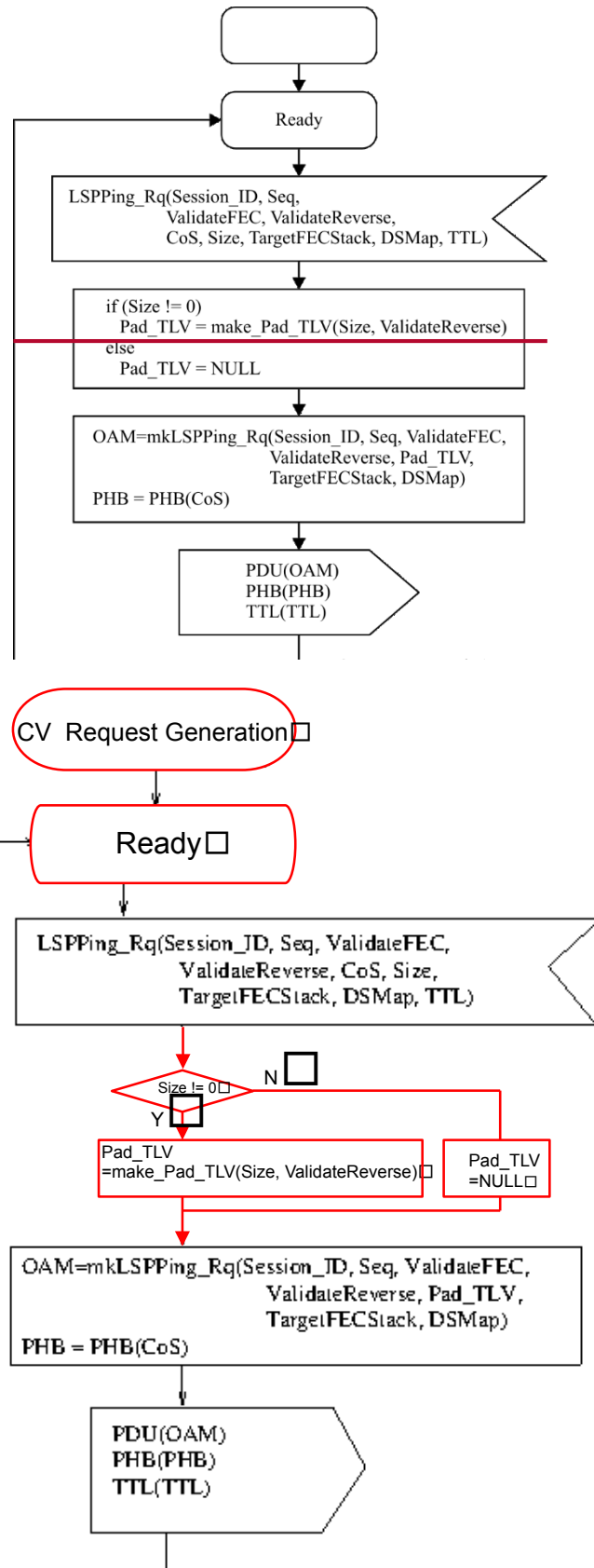


Figure 8-9/G.8121.2/Y.1381.2 On-demand CV Request Generation Process

The `make_Pad_TLV(Size)` function creates a Pad TLV in accordance with [IETF RFC4379]. 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 (eg 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**

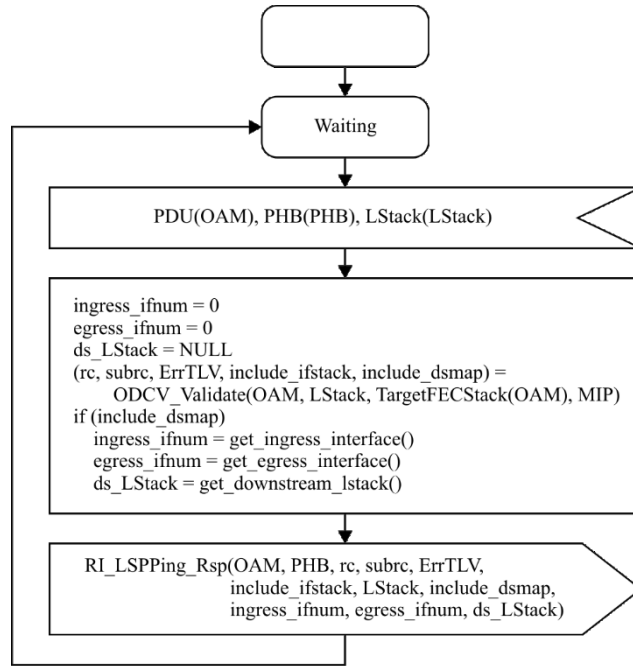
The On-demand CV Reception process demultiplexes received LSPPing packets (formed of OAM, PHB, LStack signals) as follows:

- 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 a MIP the packet is discarded.
- If this is a 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.



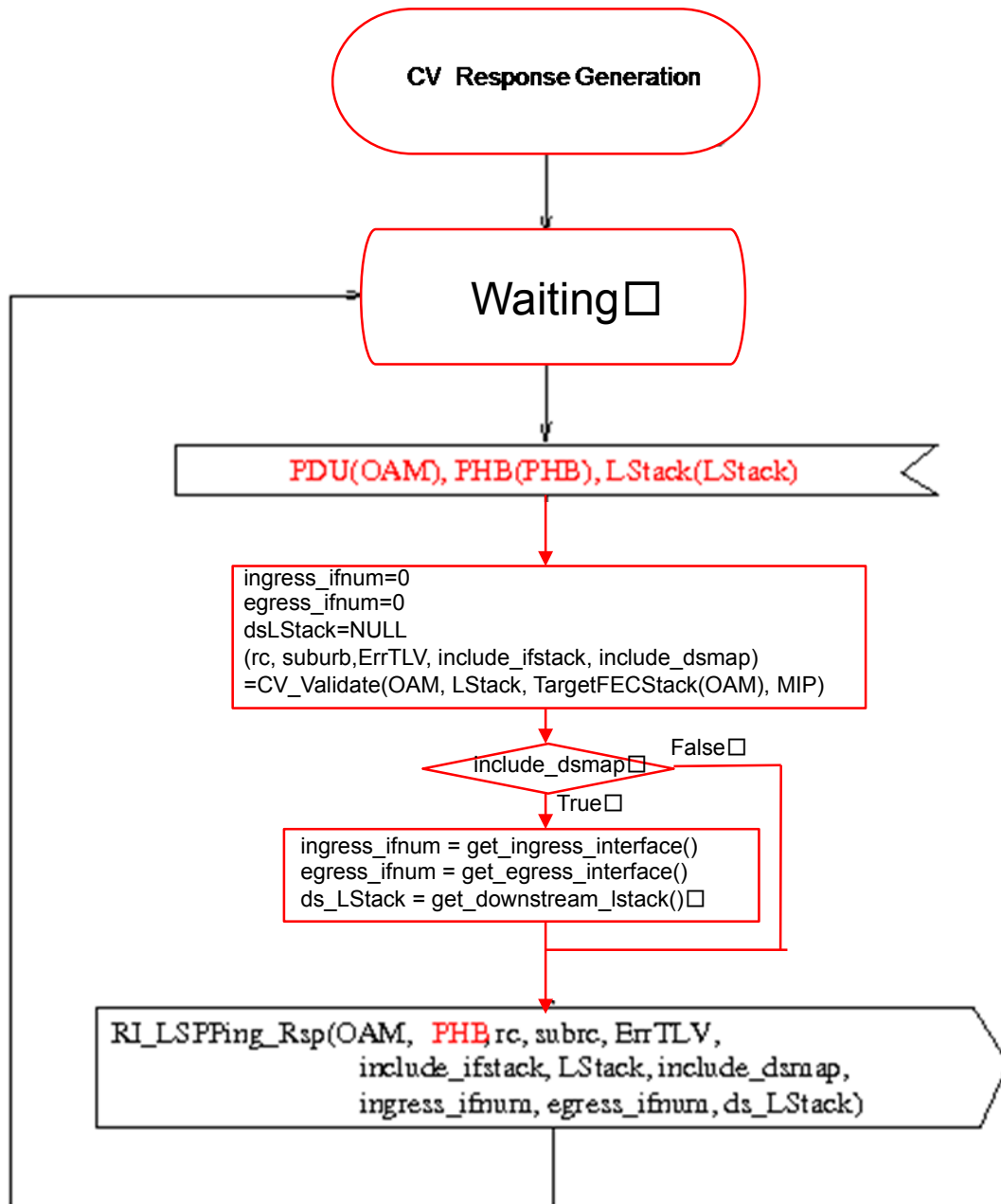


Figure 8-10/G.8121.2/Y.1381.2 MIP On-demand CV Responder Process  
[Remove "OD" per C839]

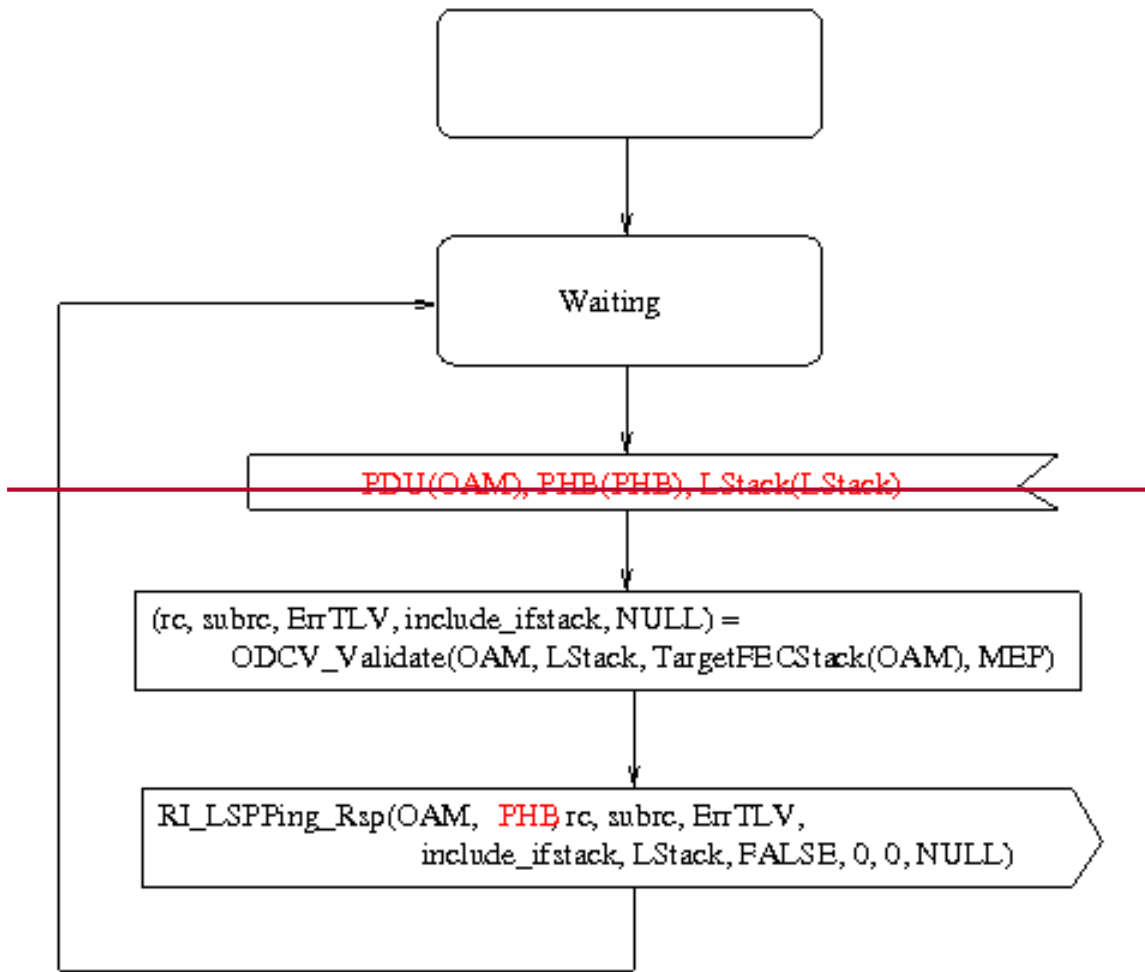
The get\_egress\_interface() function is described in section 8.8.3.1 above. [C.1074#7]

~~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\_ingress\_interface() function returns the IF\_NUM (as defined in [IETF RFC 6370]) of the interface where the packet was received, or 0 if it was generated locally.~~

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 RFC4379].

#### **8.8.3.6. MEP On-demand CV Responder process**

The MEP On-demand CV Responder process is described in Figure 8-11.



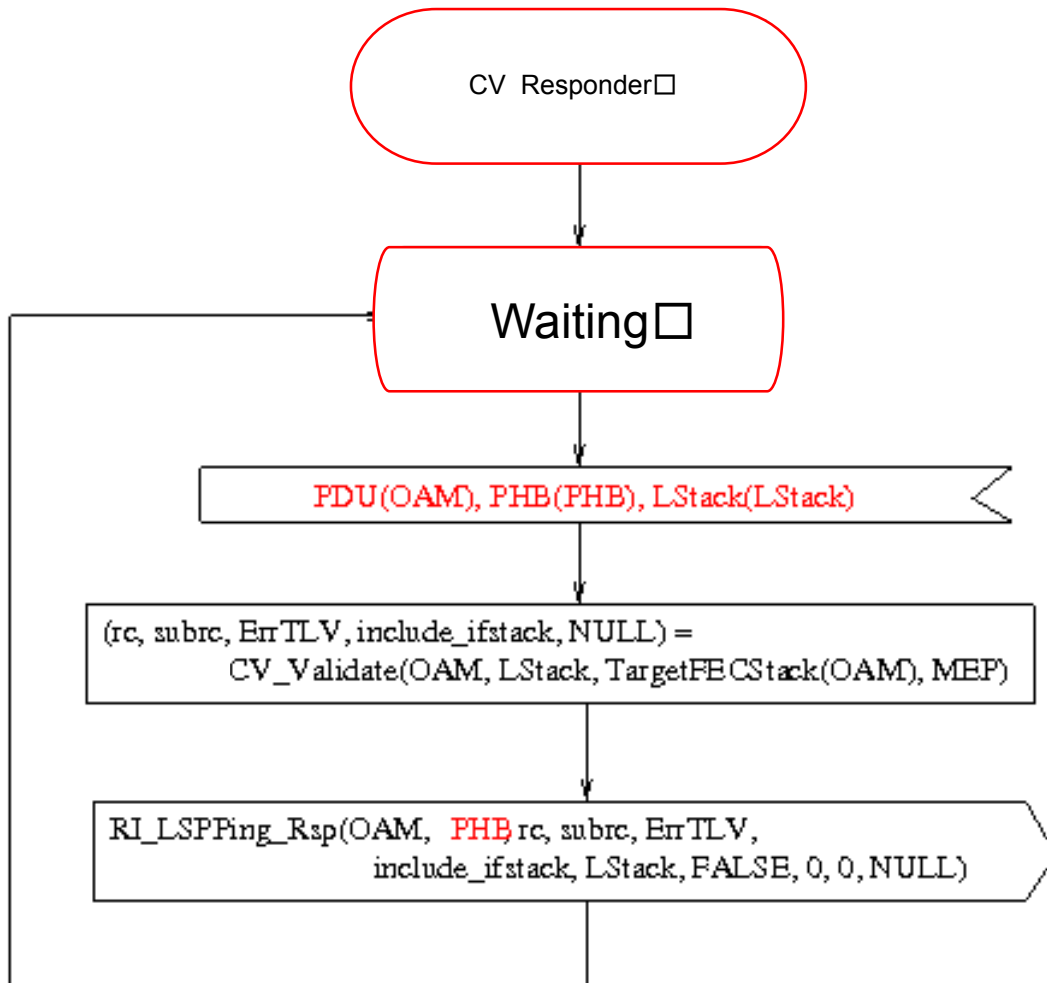
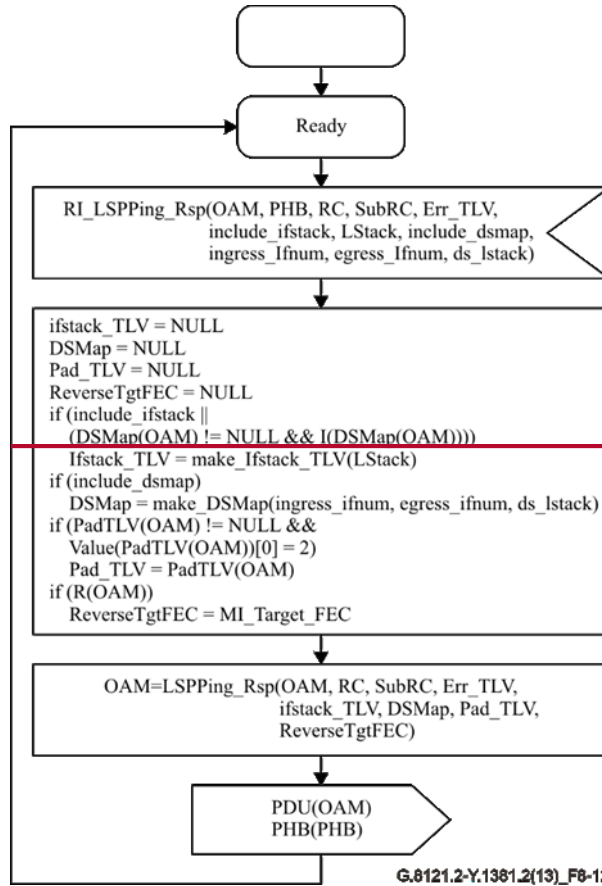
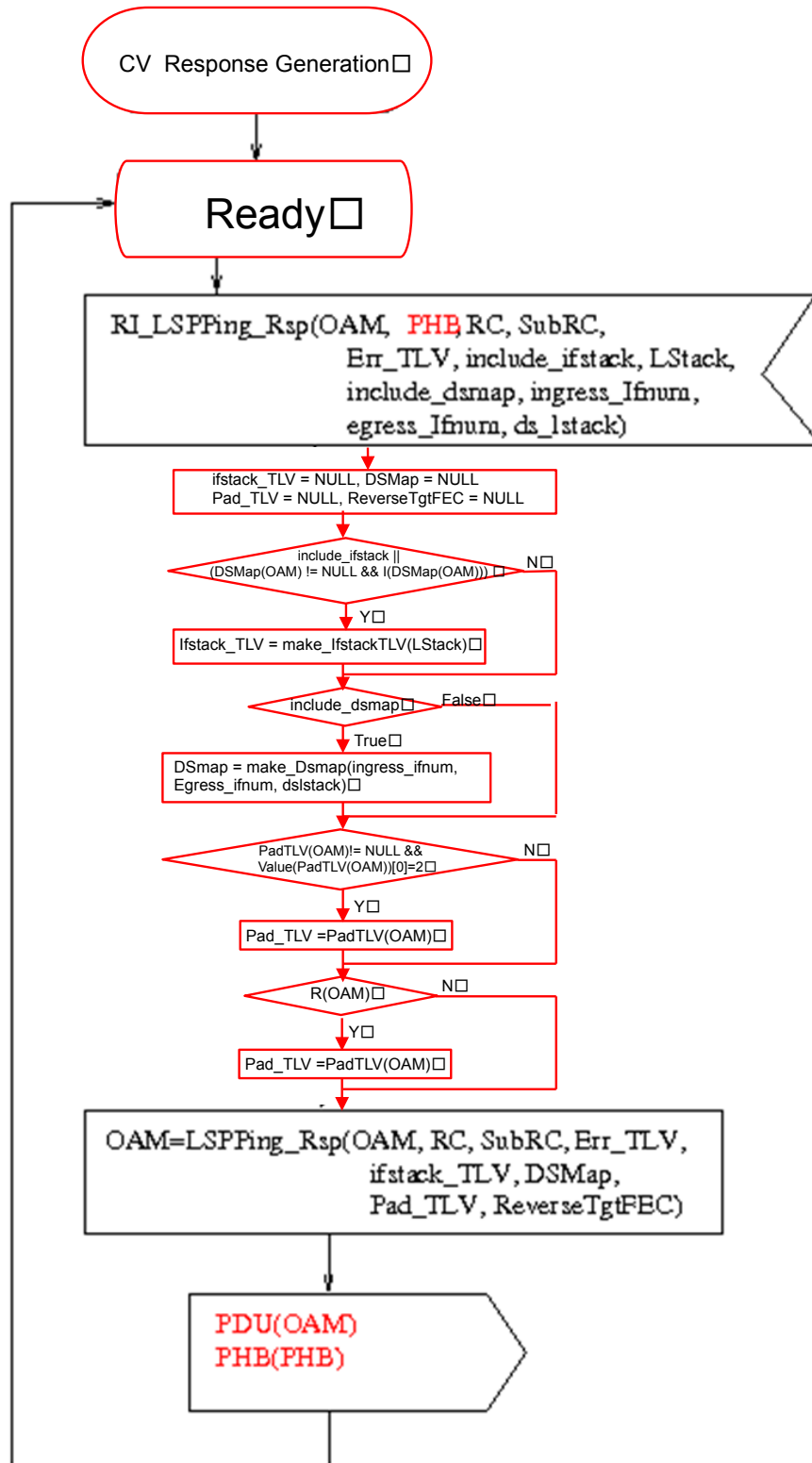


Figure 8-11/G.8121.2/Y.1381.2 MEP On-demand CV Responder Process

### 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/G.8121.2/Y.1381.2 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 the result of get\_ingress\_interface() (see 8.8.3.5)~~set to MI>Ifnum~~  
[C.1074#7]
- 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 section 8.8.3.2 above.

The LSPPing\_Rsp function creates an LSPPing Echo Reply packet in accordance with [IETF RFC4379] and [IETF RFC6426]. 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 (ie, 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.

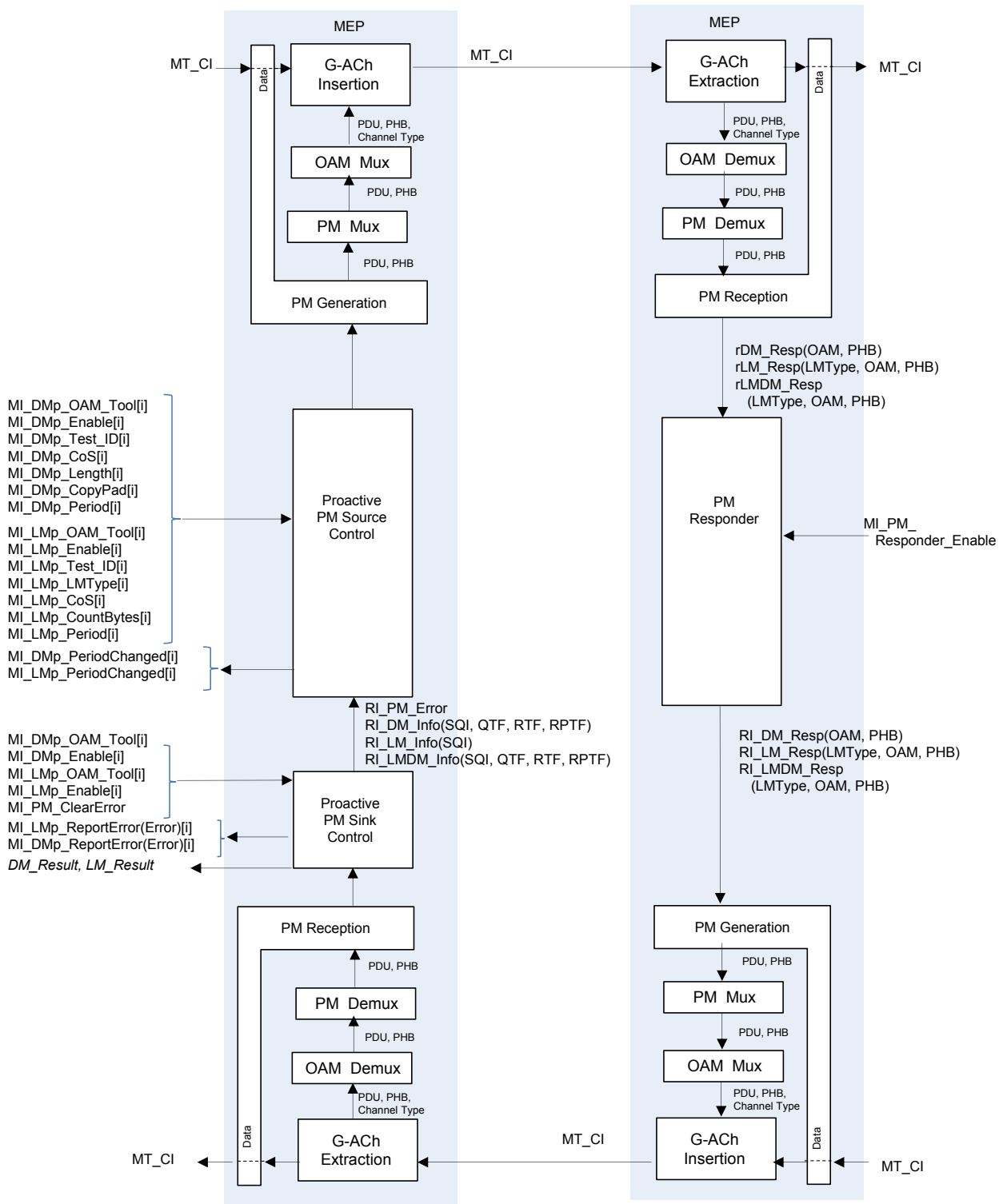


Figure 8-13/G.8121.2/Y.1381.2 Proactive PM processes

The Proactive PM Source Data 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 on 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 on 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 (ie, source and sink) are 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 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.

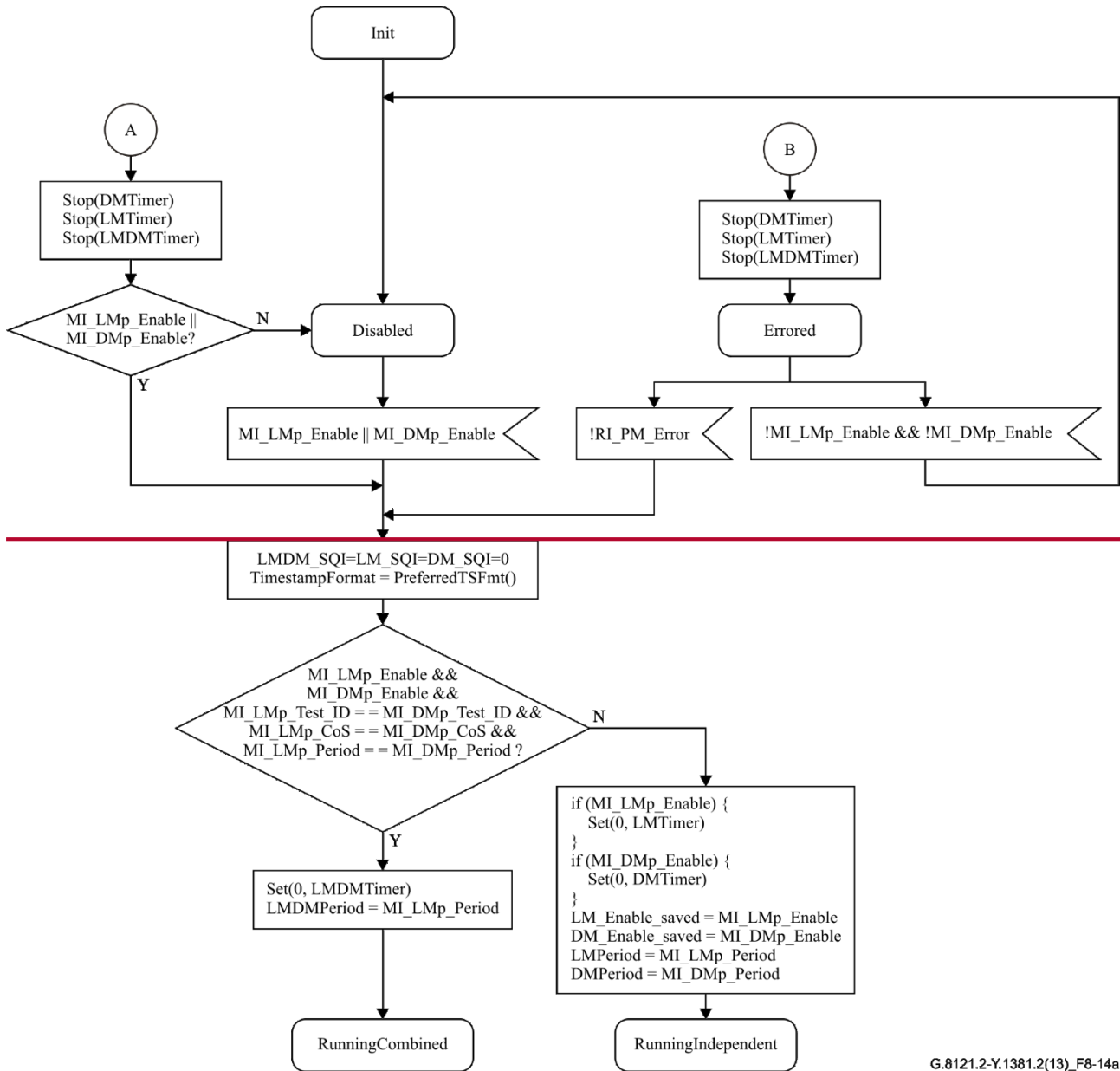
The Proactive PM Source control process performs delay measurements when MI\_DMp\_Enable is true and performs loss measurements when MI\_LMp\_Enable is true. If both are enabled, then where possible, the same PDUs are used to make both measurements (ie ILM+DM or DLM+DM PDUs). Otherwise, separate PDUs 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. For loss measurement, MI\_LMp\_CountBytes specifies whether to use octet counters (if set) or packet counters (if unset). **TC1074 #17**

The proactive PM control process is described in Figure 8-14, 8-15, and 8-16. These figures include LM, DM and LMDM.



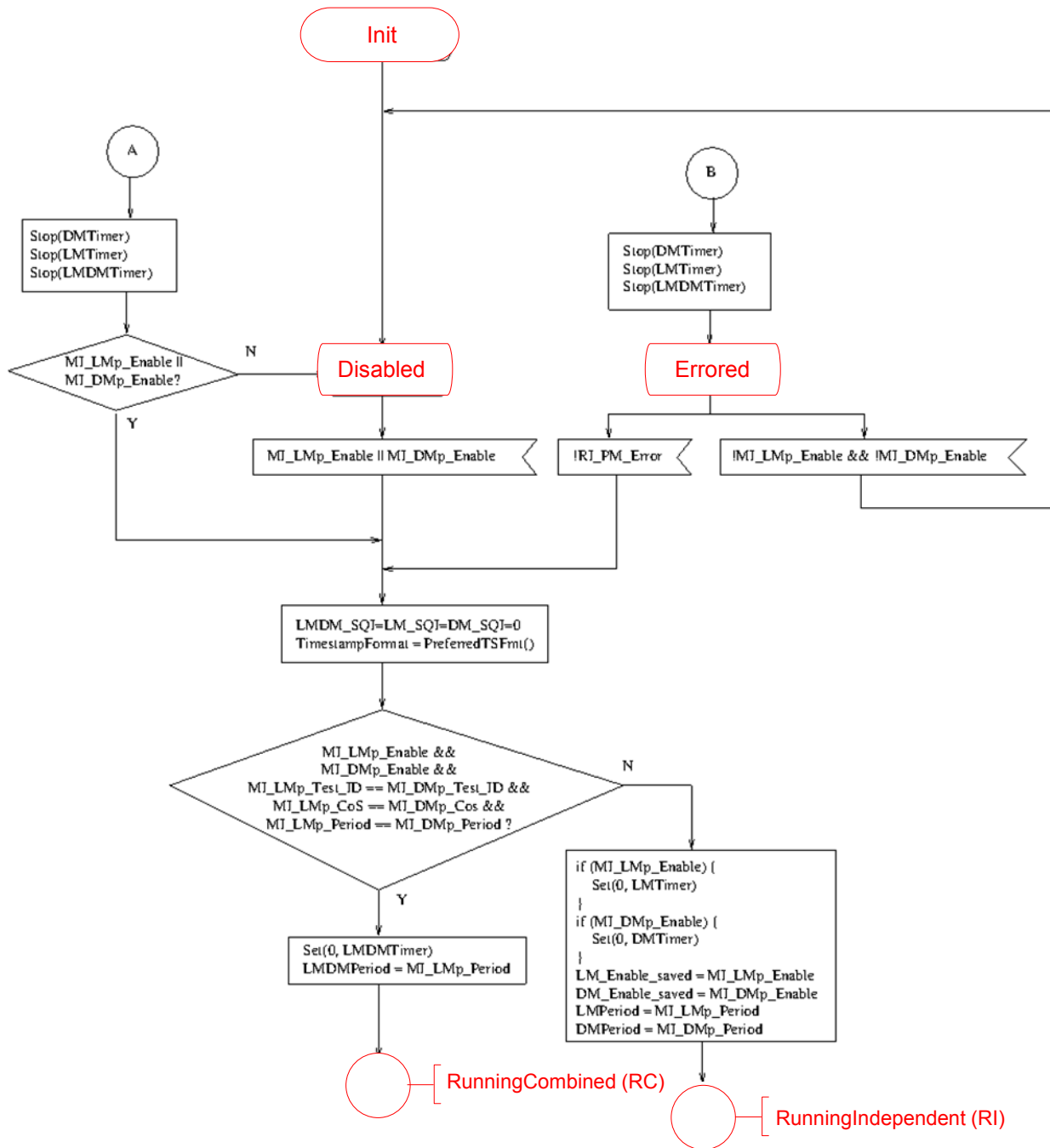
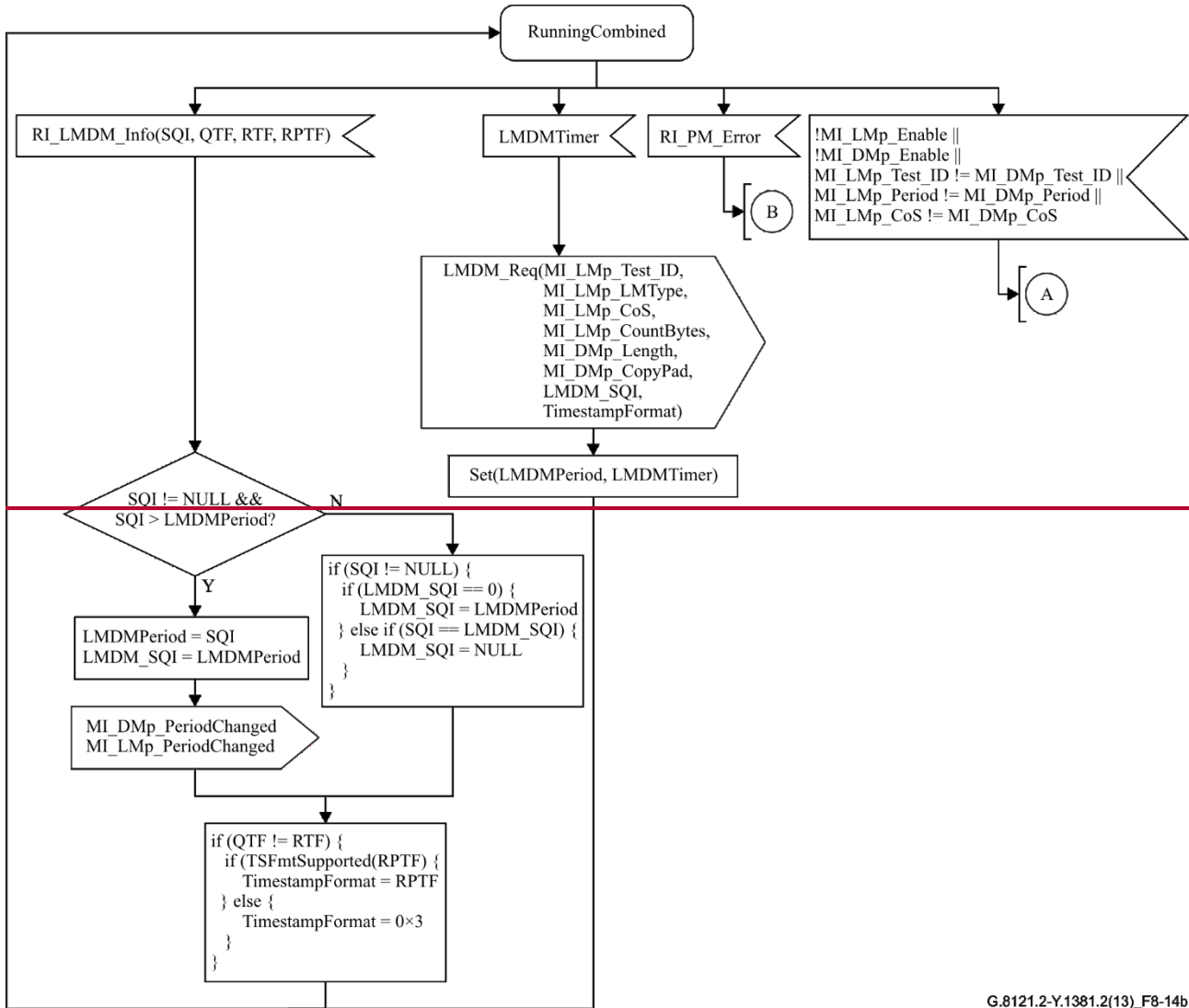


Figure 8-14a/G.8121.2/Y.1381.2 proactive PM Source control process





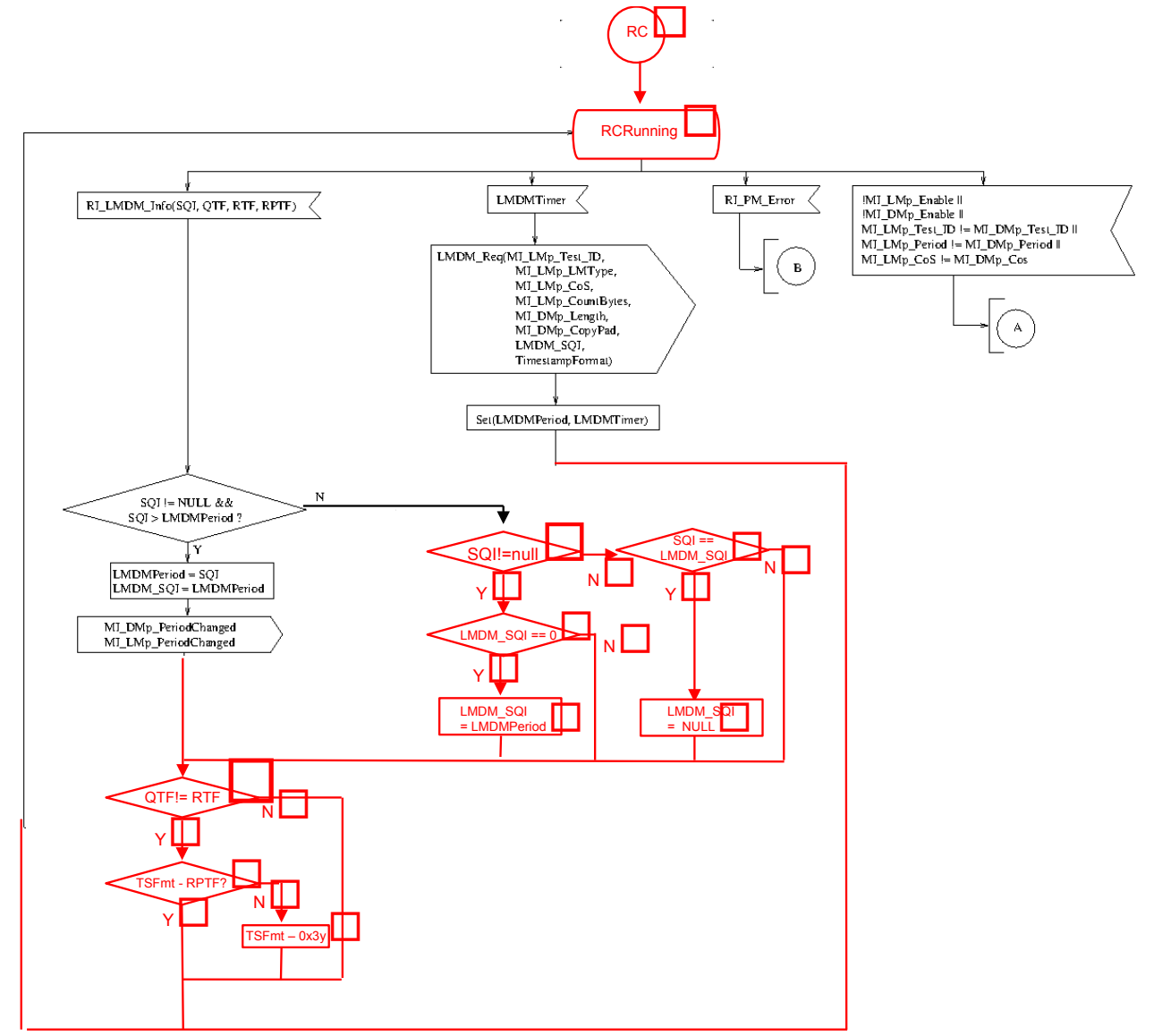
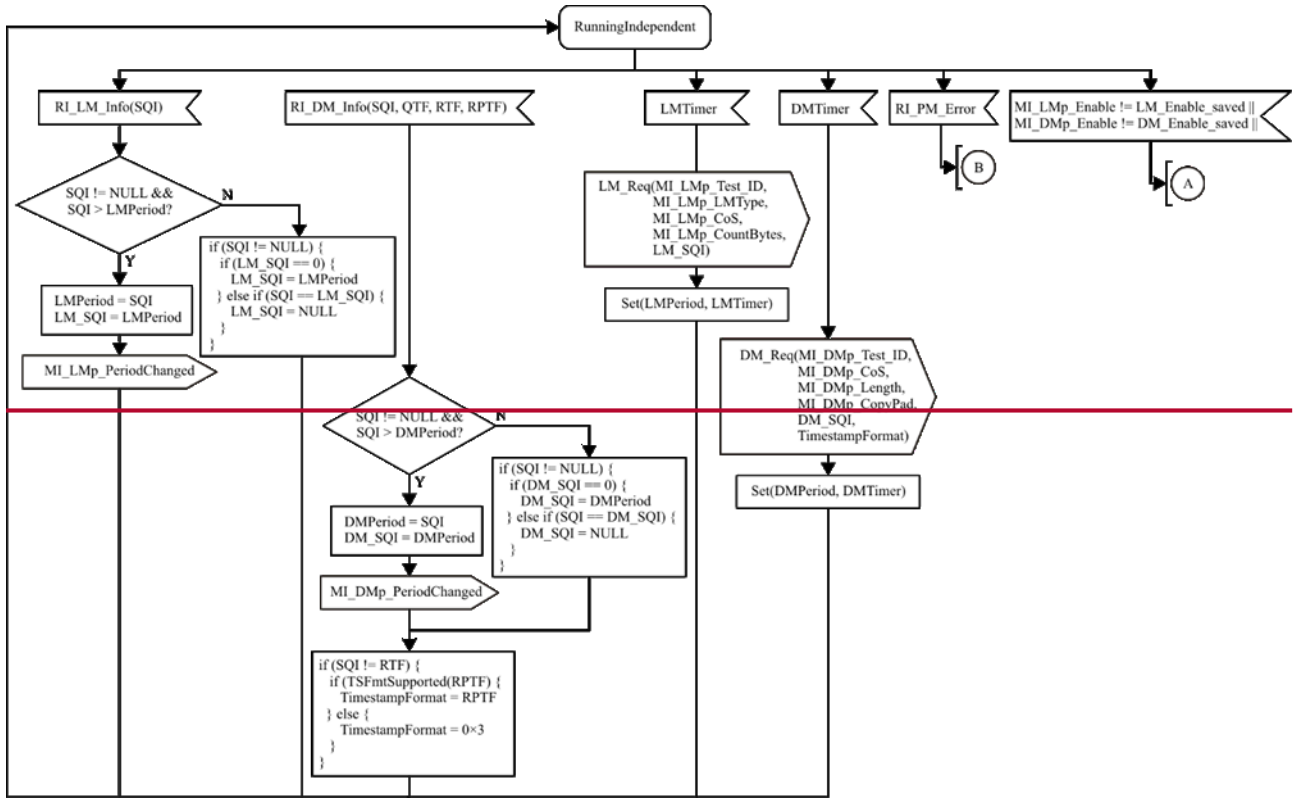
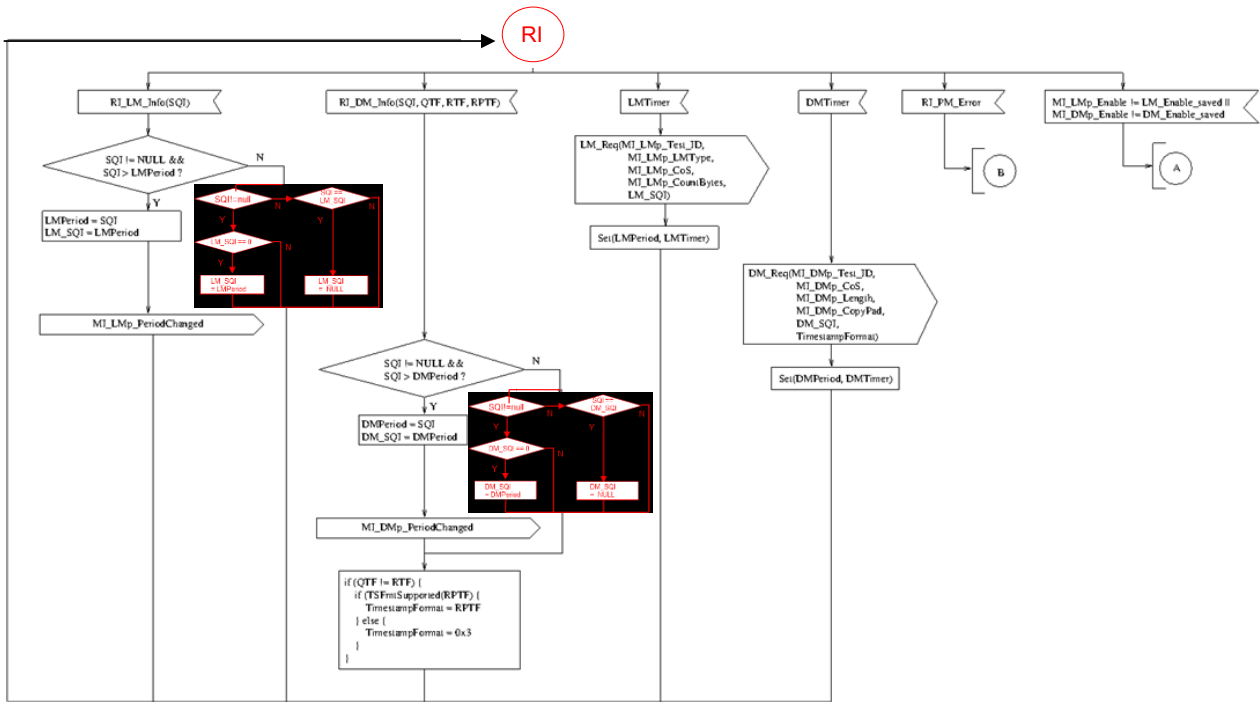
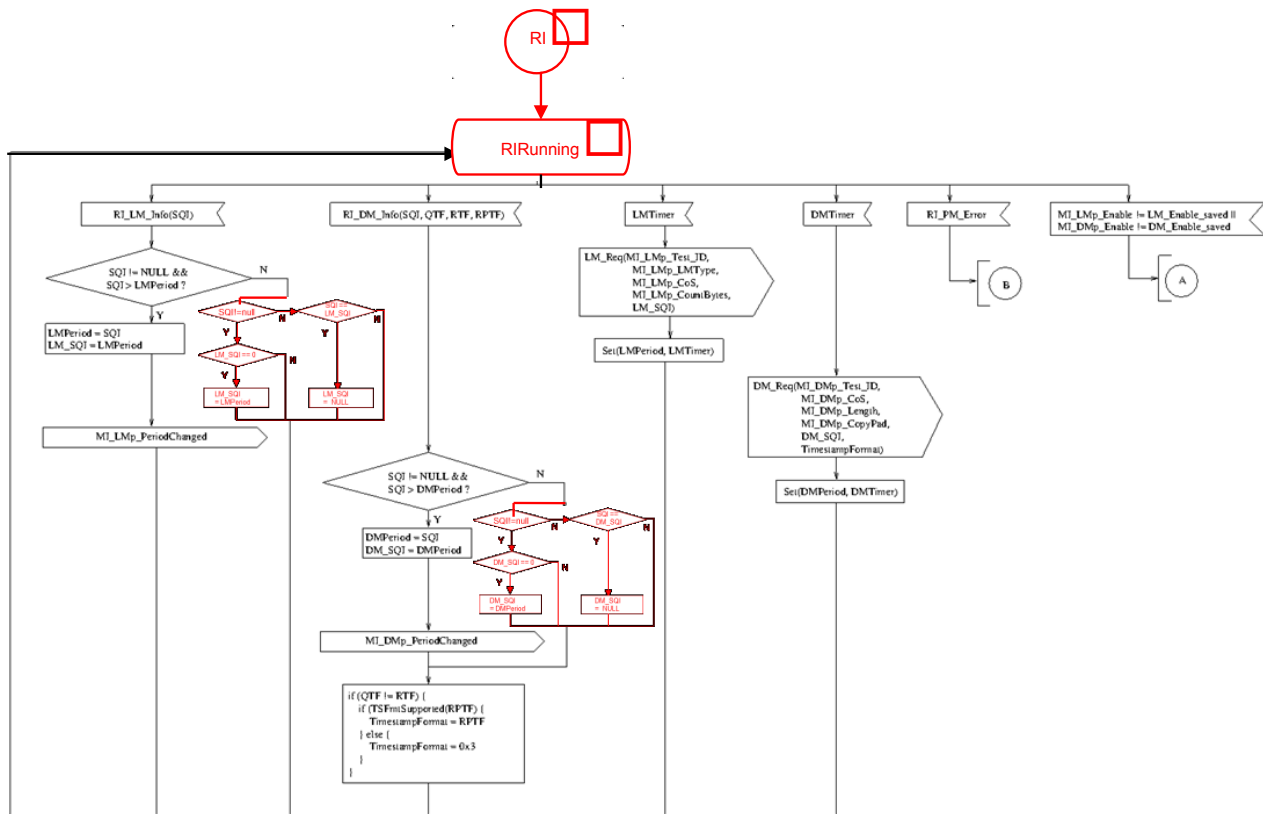


Figure 8-14b/G.8121.2/Y.1381.2 proactive PM Source control process



G.8121.2-Y.1381.2(13)\_FB-14c





**Figure 8-14c/G.8121.2/Y.1381.2 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 RFC6374].

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.

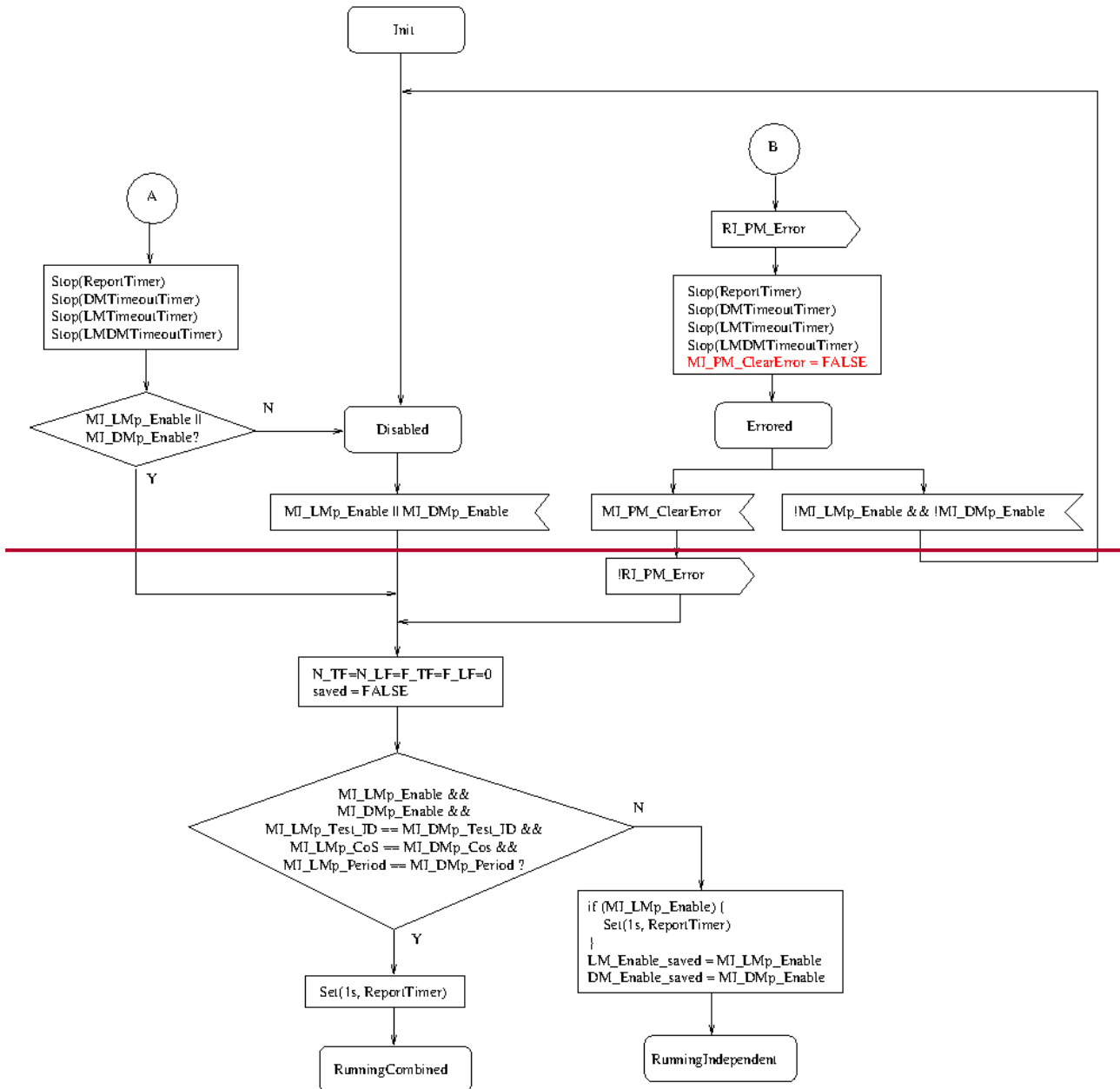
As for the source control process, the Proactive PM Sink control process performs delay measurements when MI\_DMp\_Enable is true and performs loss measurements when MI\_LMp\_Enable is true. If both are enabled, then where possible, the same PDUs are used to make both measurements (ie ILM+DM or DLM+DM PDUs). Otherwise, separate PDUs are used for loss (ILM or DLM) and delay (DM).

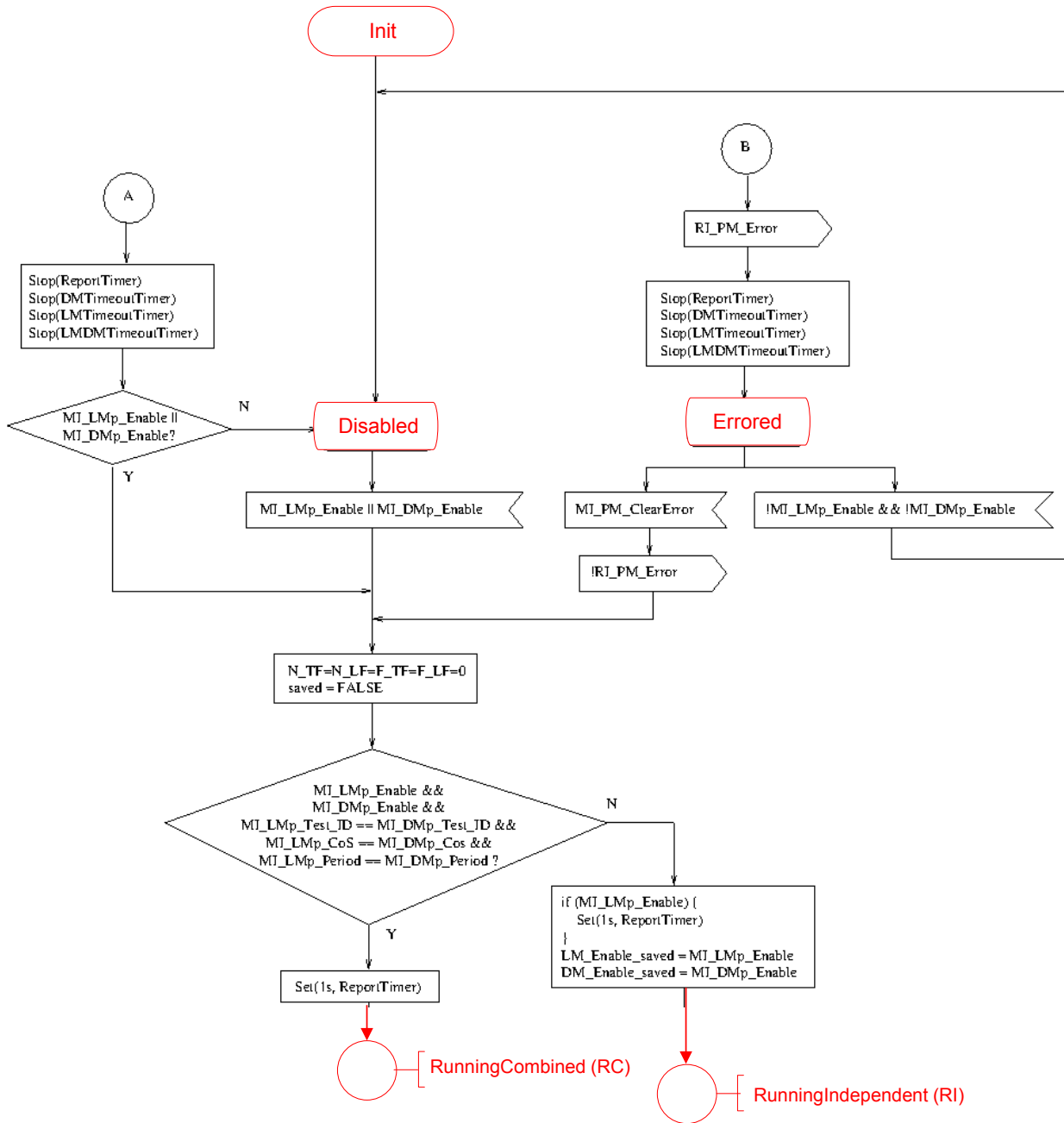
If an error is detected while the session is running, this is reported via MI\_DMp\_ReportError or MI\_LMp\_ReportError, and the session is stopped until MI\_PM\_ClearError is set or the session is

disabled. The reported error is either one of the control codes that indicates an error as described in section 3.1 of [IETF RFC 6374], or the special value 'Timeout' indicating that no response was received to a PM query within the timeout time

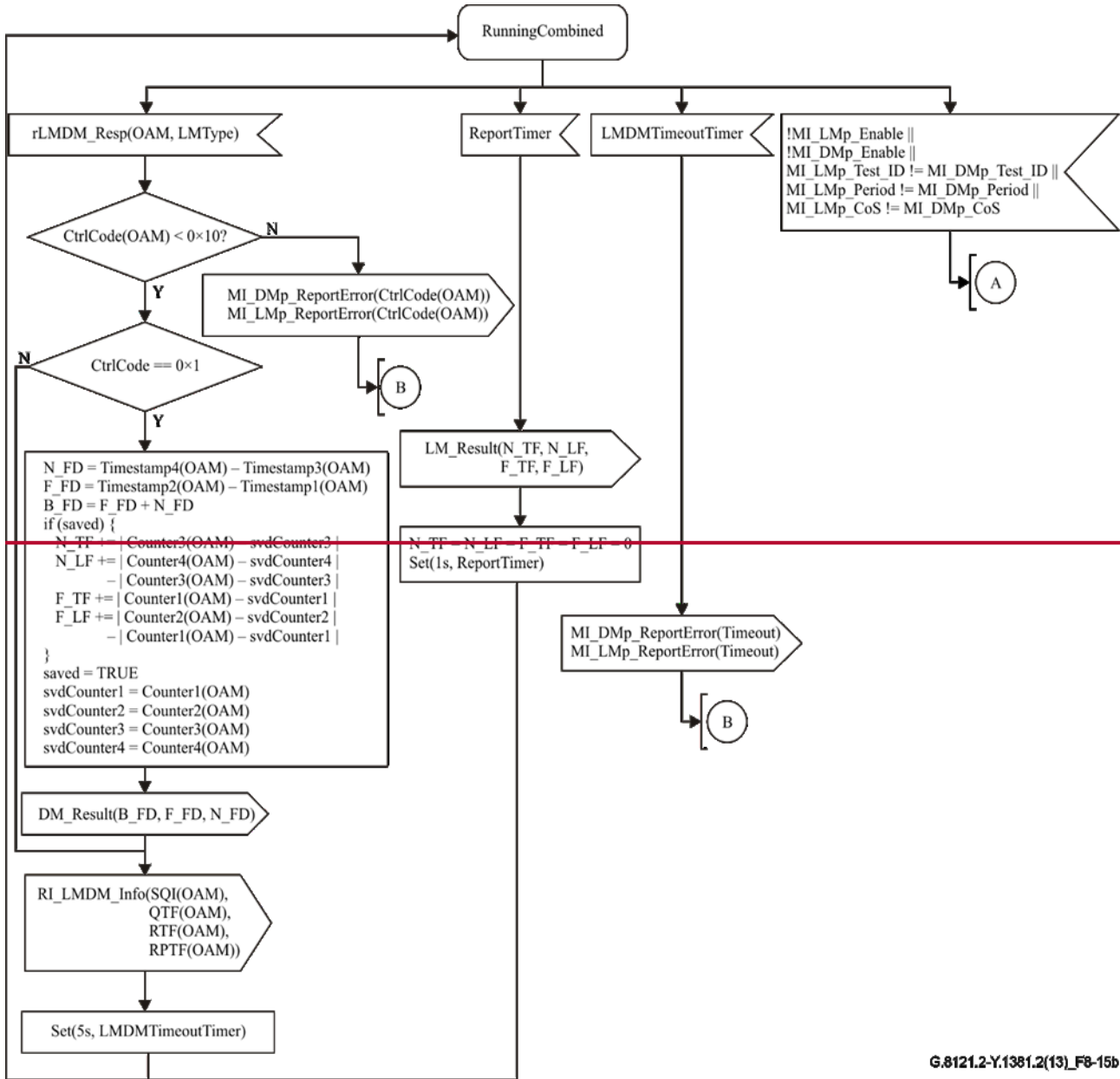
Note: MI\_PM\_ClearError is used as a trigger to resume a measurement process that has been stopped due to an error. It is expected to be set by the EMF once the cause of the error has been investigated and resolved. ~~[CI074 #2, #5]~~

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





**Figure 8-15a/G.8121.2/Y.1381.2 proactive PM Sink control process**  
 [to reset MI\_PM\_ClearError to false before entering the Errored state.] [C1074 #2]



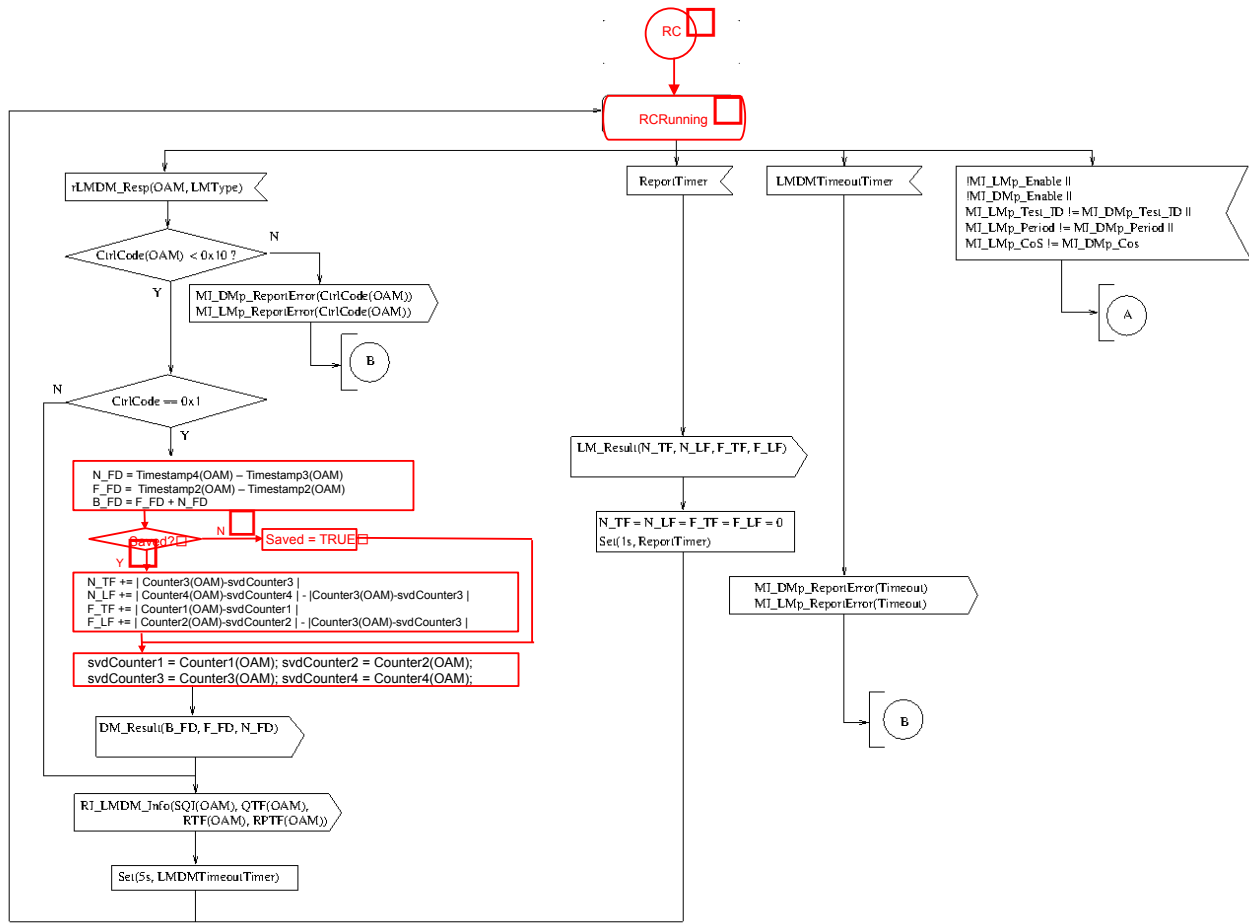
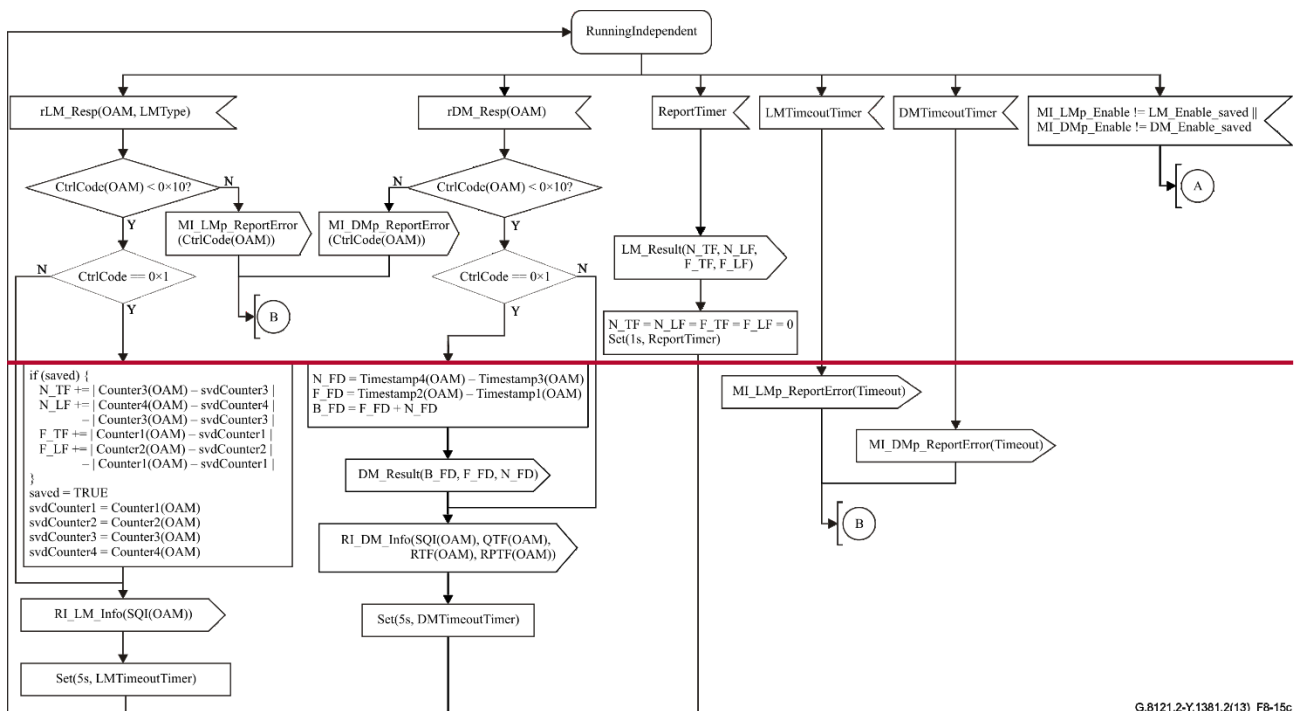


Figure 8-15b/G.8121.2/Y.1381.2 proactive PM Sink control process



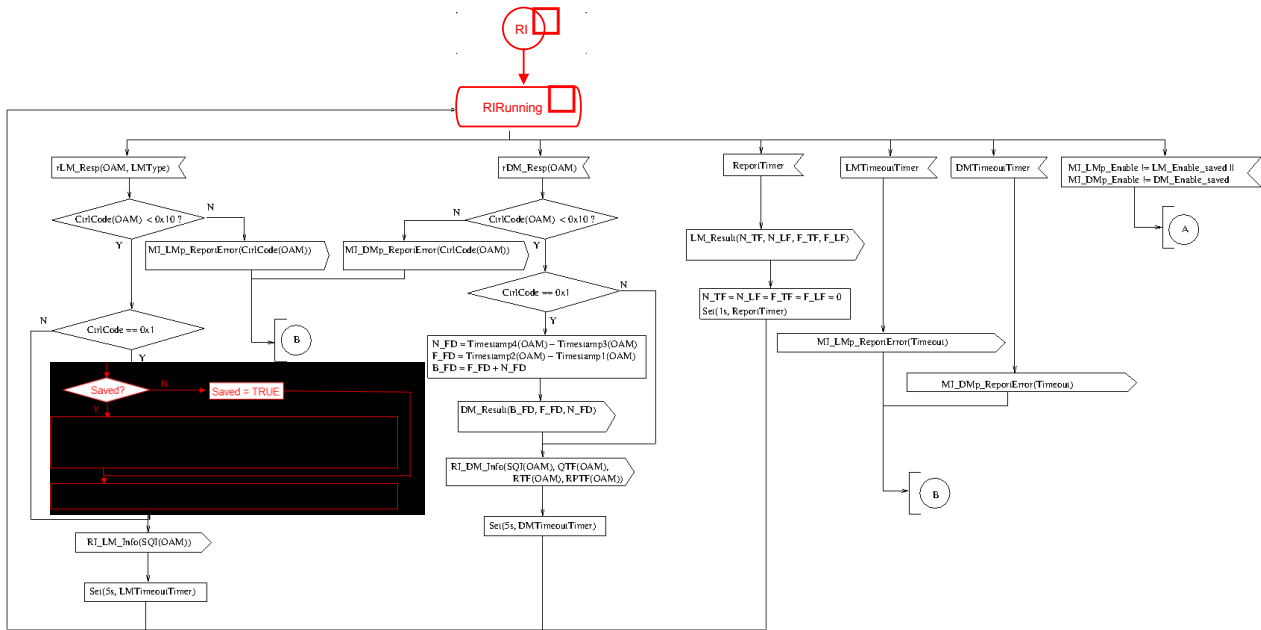


Figure 8-15c/G.8121.2/Y.1381.2 proactive PM Sink control process

### 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.

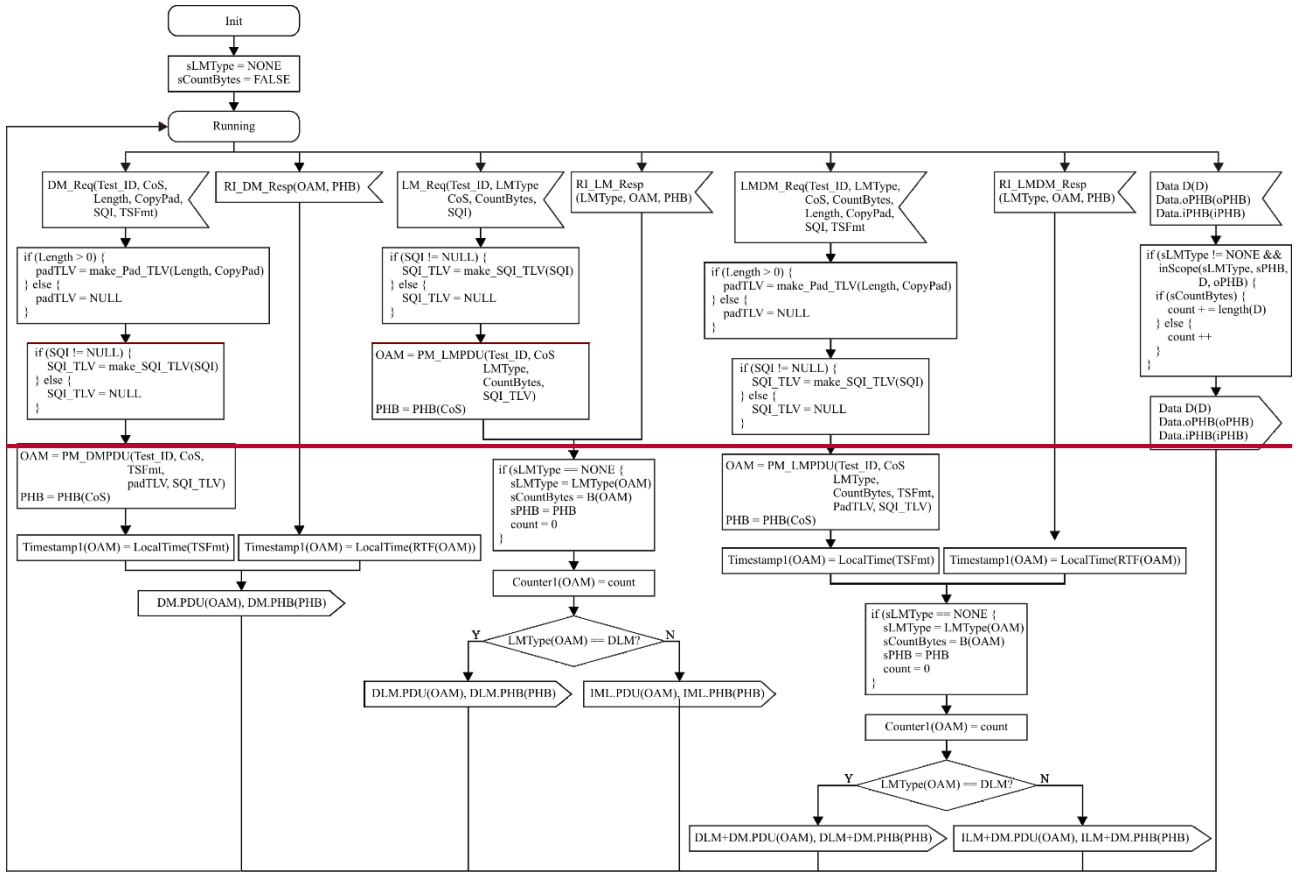
For delay measurement, it writes the packet send time into the PDU, using the requested timestamp format.

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:





G.8121.2-Y.1381.2(13)\_F8-16

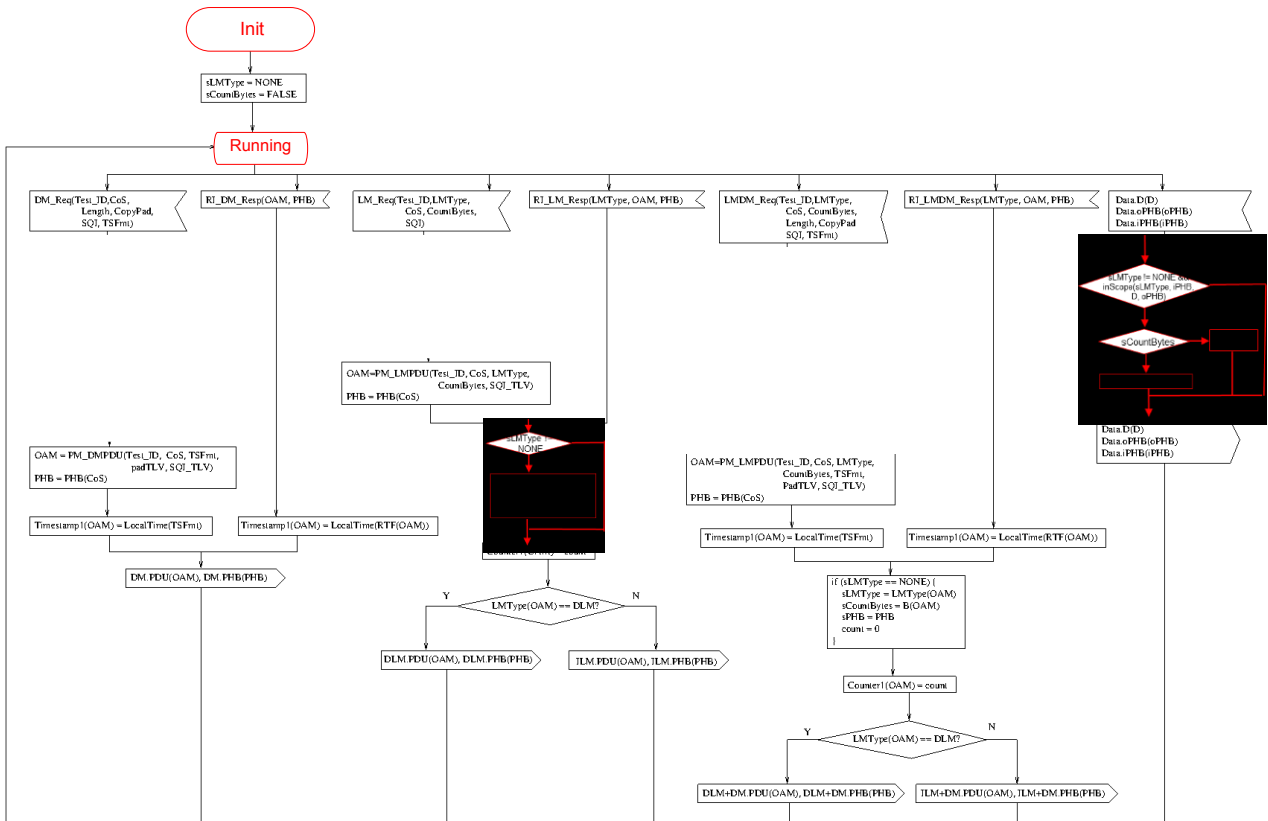


Figure 8-16/G.8121.2/Y.1381.2 - PM Generation Process

The `make_Pad_TLV(Length, CopyPad)` function creates a Padding TLV as specified in [IETF RFC6374], 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 RFC6374], 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 RFC6374], 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 RFC6374], 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 appropriate depending 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”).

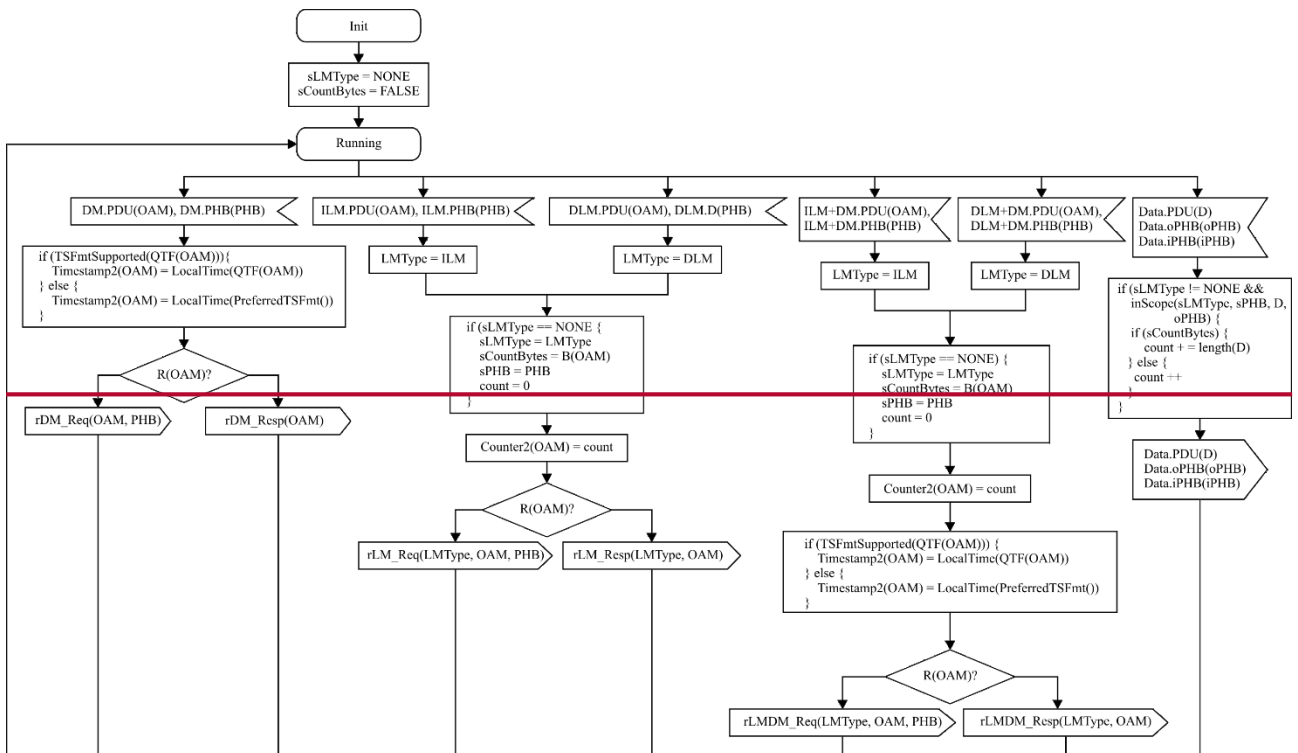
The LocalTime(TSFmt) function returns the local time-of-day, in the format specified.

#### 8.8.4.4 PM reception process for Proactive LM

The PM Reception process receives PM message 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:



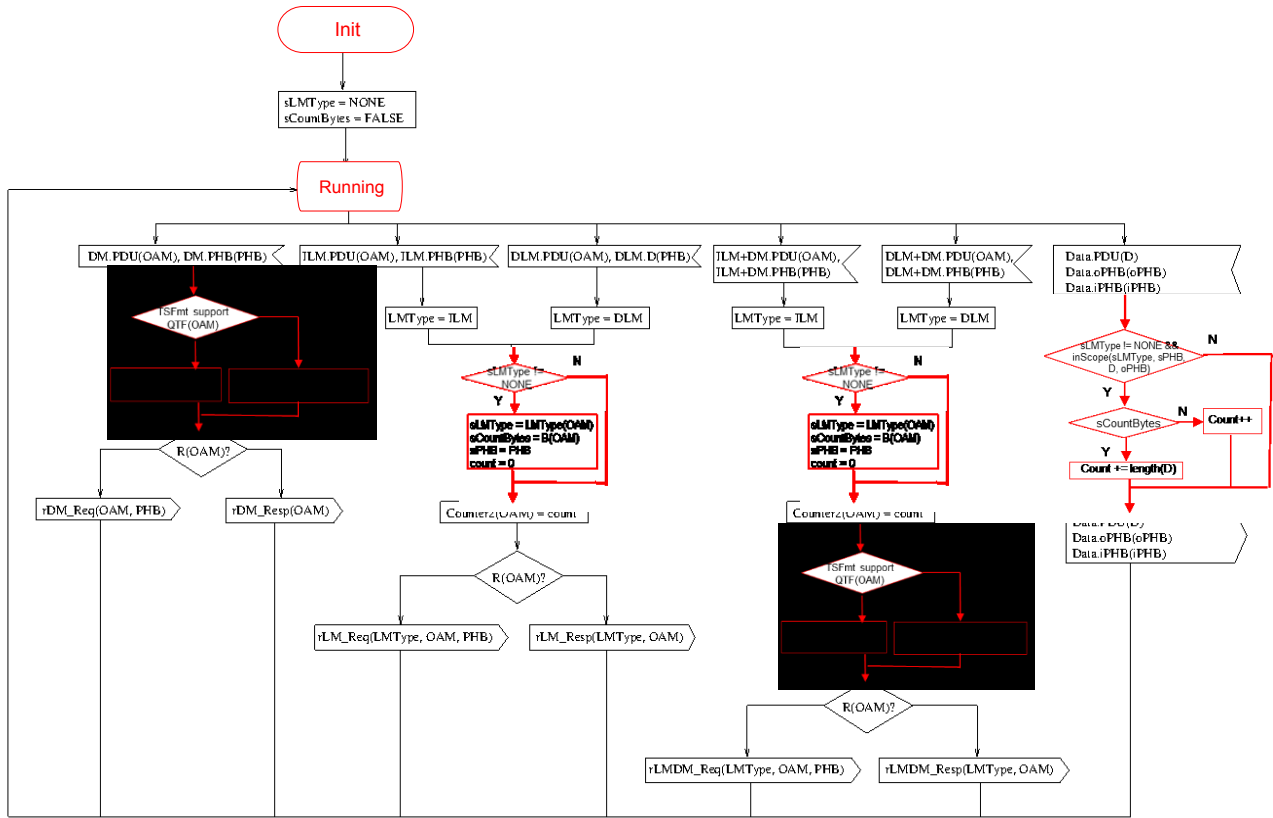
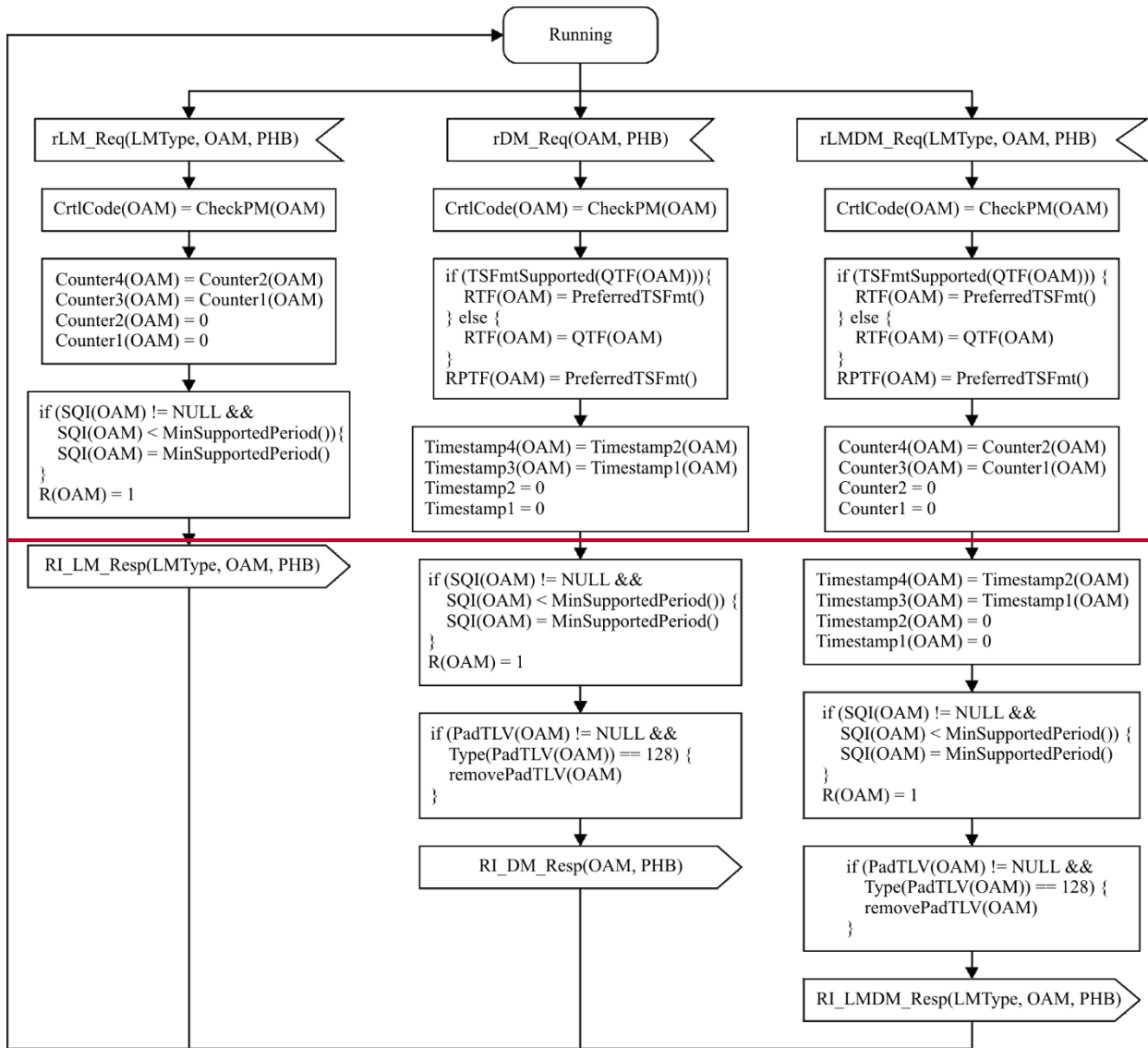


Figure 8-17/G.8121.2/Y.1381.2 - PM reception process

#### 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:



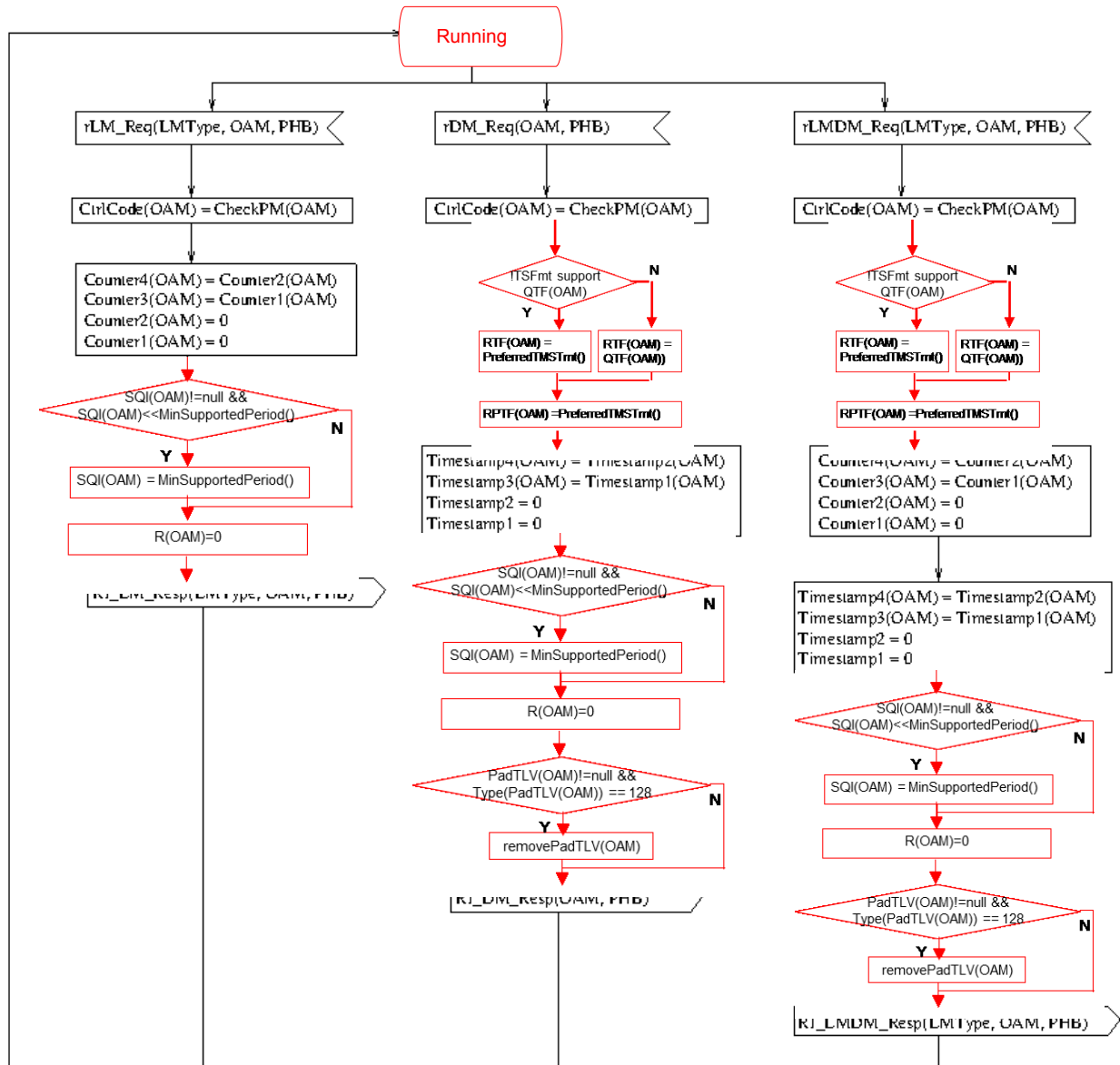


Figure 8-18/G.8121.2/Y.1381.2 PM responder process

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.

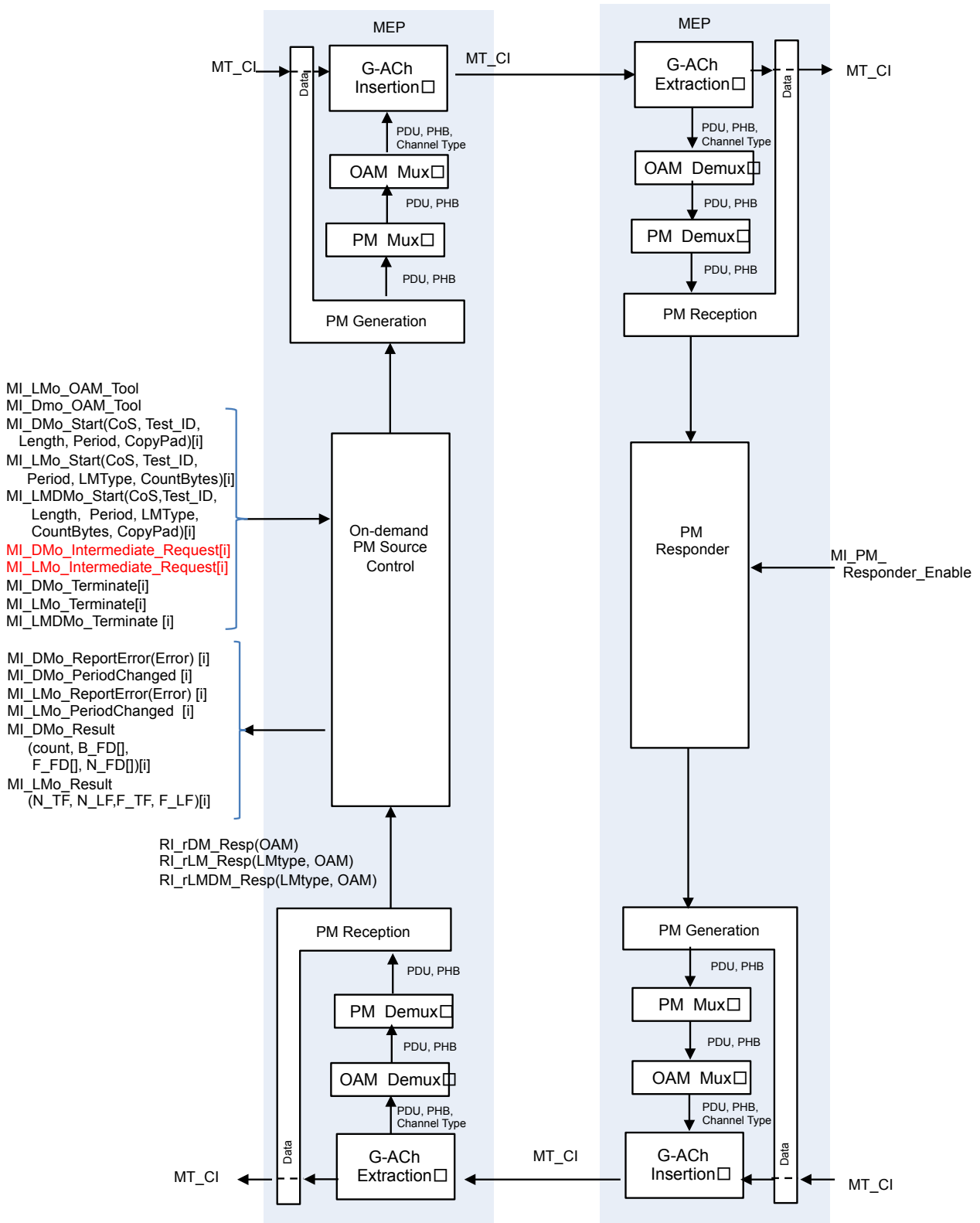


Figure 8-19/G.8121.2/Y.1381.2 On-demand PM processes

added intermediate request

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 on 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.

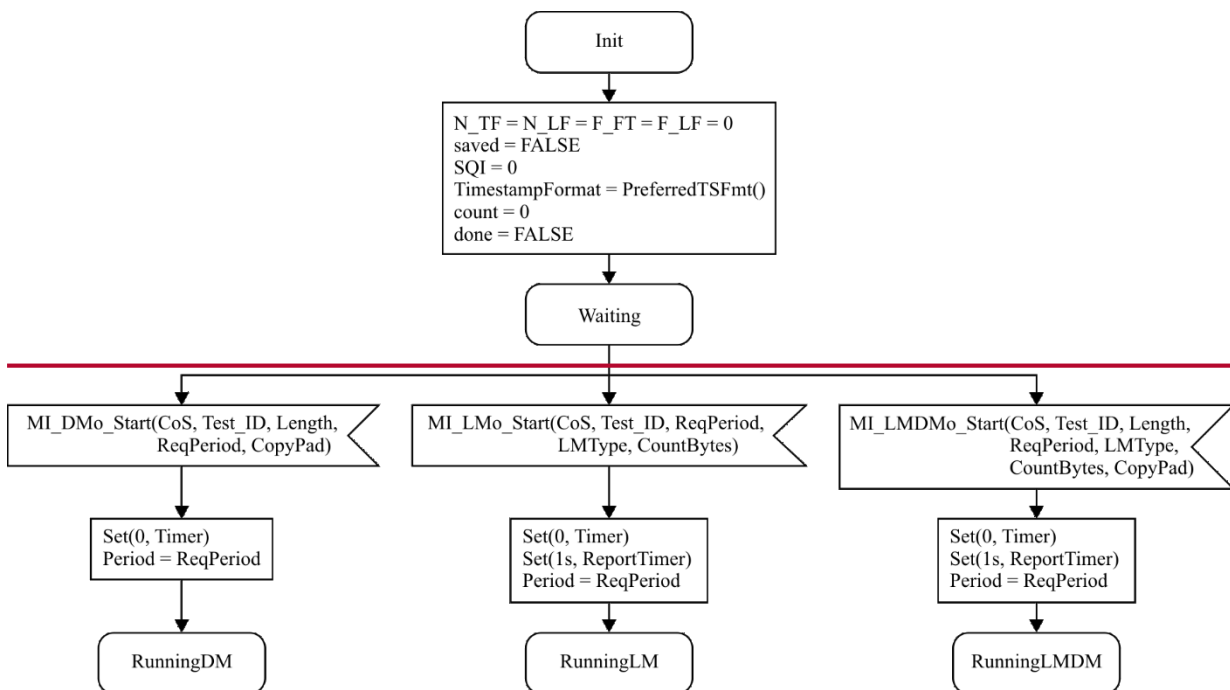
Results are reported via MI\_DMo\_Result and MI\_LMo\_Result.

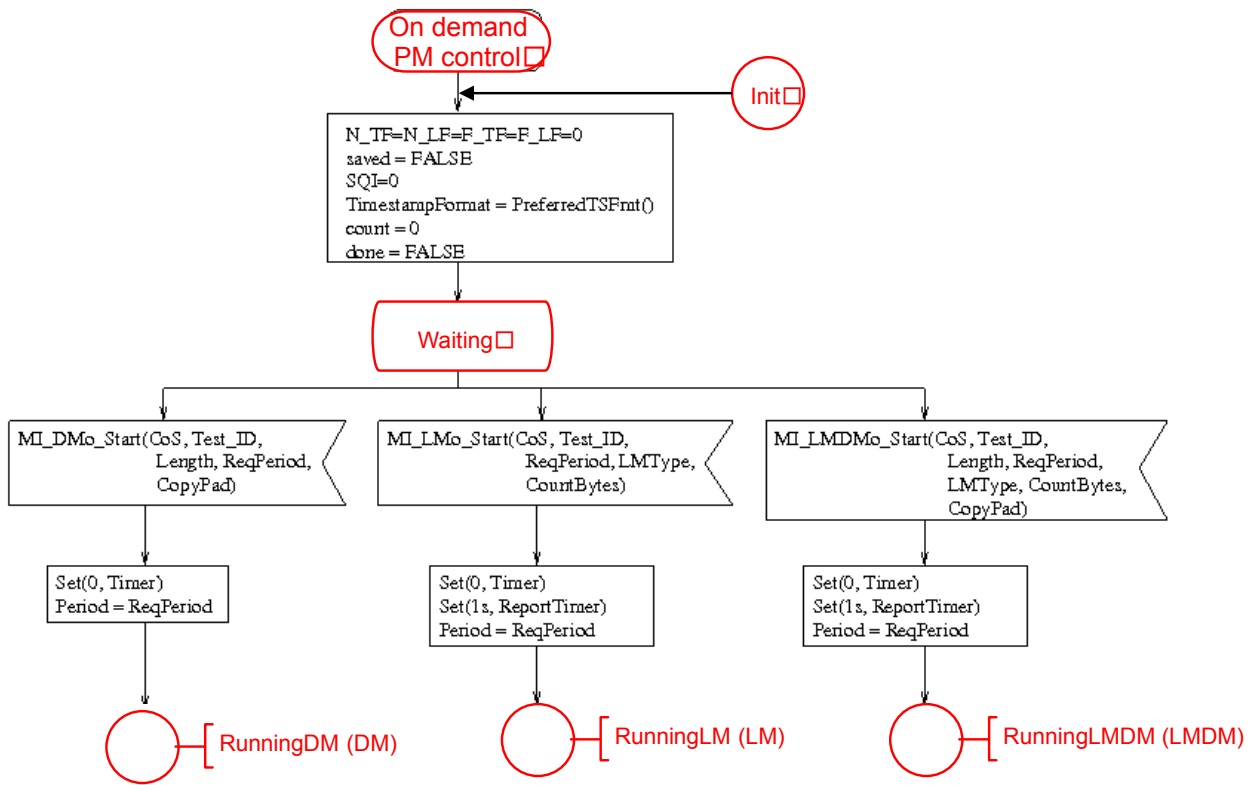
If an error is detected while the session is running, this is reported via MI\_DMo\_ReportError or MI\_LMo\_ReportError, and the session is terminated automatically. The results collected up to that point are reported.

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.

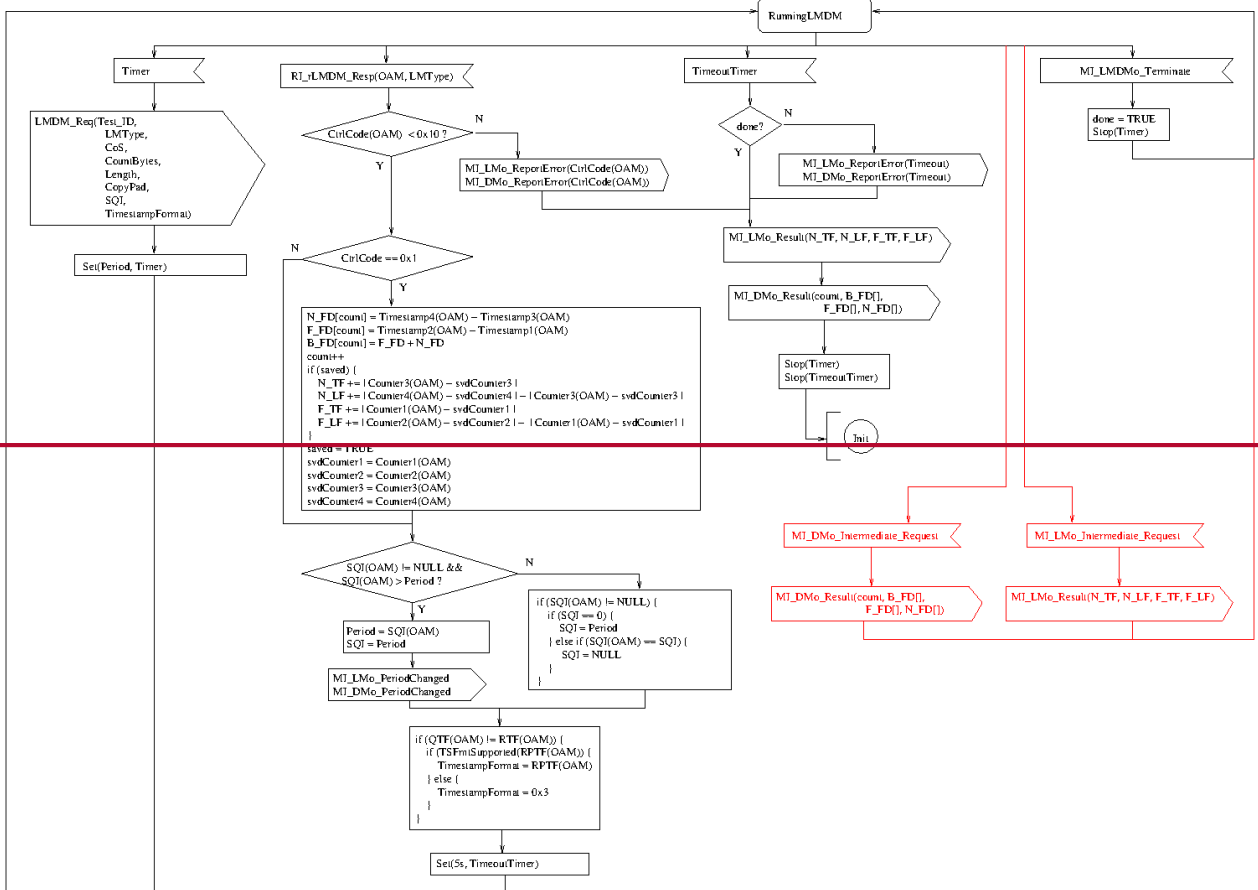
The CoS parameter of MI\_LMo\_Start, MI\_DMo\_Start or MI\_LMDMo\_Start specifies the CoS (traffic class) to use for the measurement. In the case of MI\_LMo\_Start, this can either be a specific value, or the special value “ALL” indicating that loss across all traffic classes should be measured. For loss measurement, the CountBytes parameter of MI\_LMo\_Start() or MI\_LMDMo\_Start() specifies whether to use octet counters (if set) or packet counters (if unset). **[C1074 #1]**

The on-demand PM control process is described in Figure 8-20a, 8-20b, 8-20c, and 8-21d.





**Figure 8-20a/G.8121.2/Y.1381.2 On-demand PM control process**



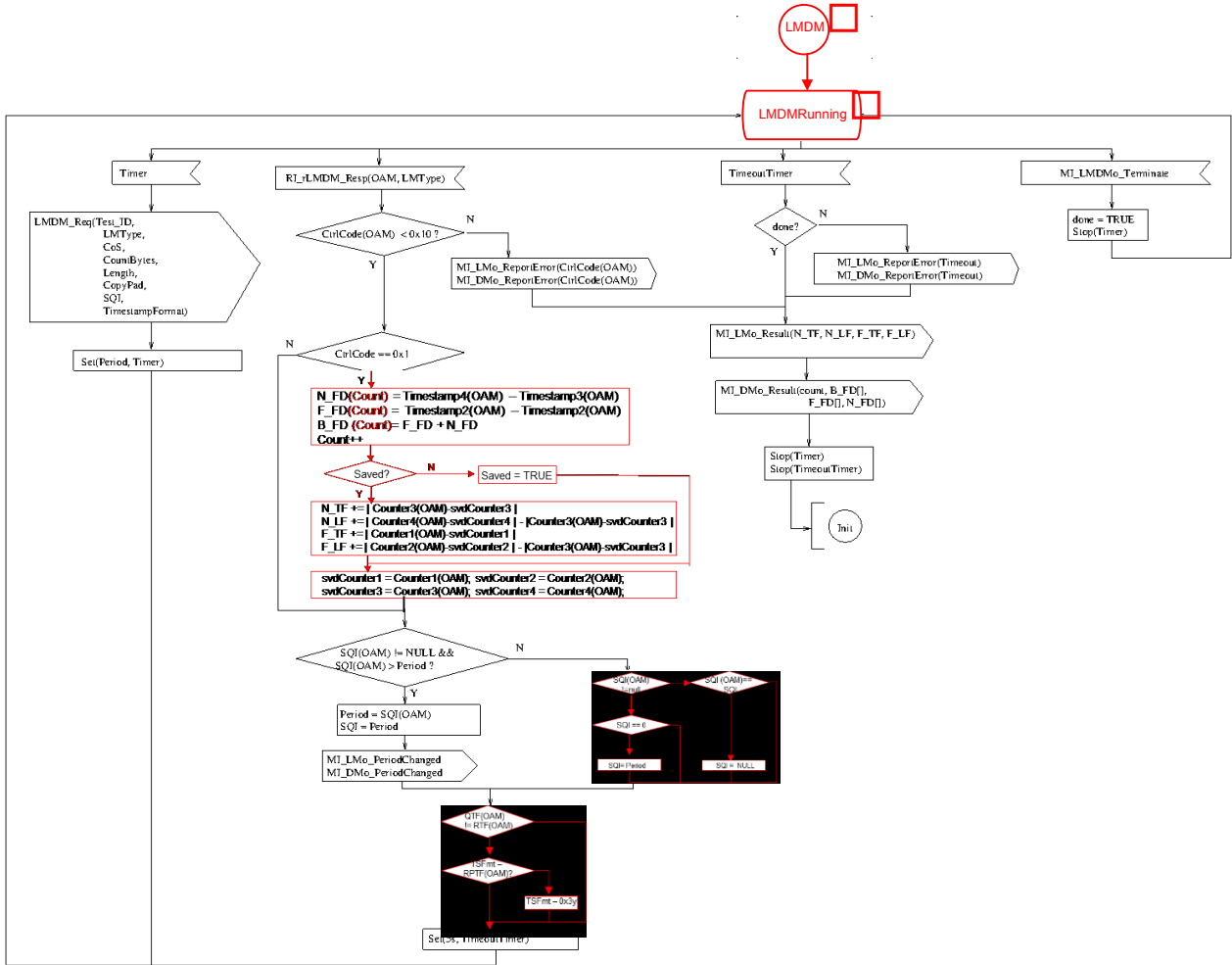


Figure 8-20b/G.8121.2/Y.1381.2 On-demand PM control process

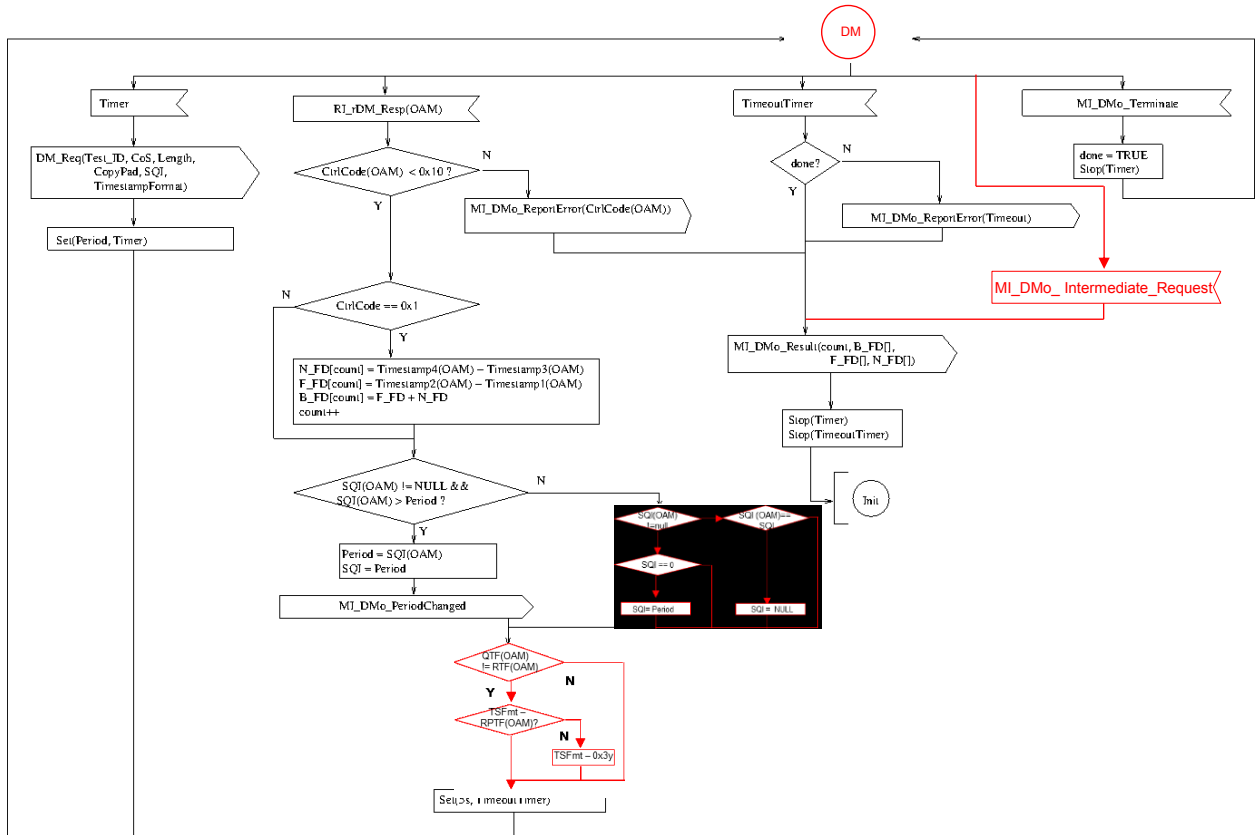
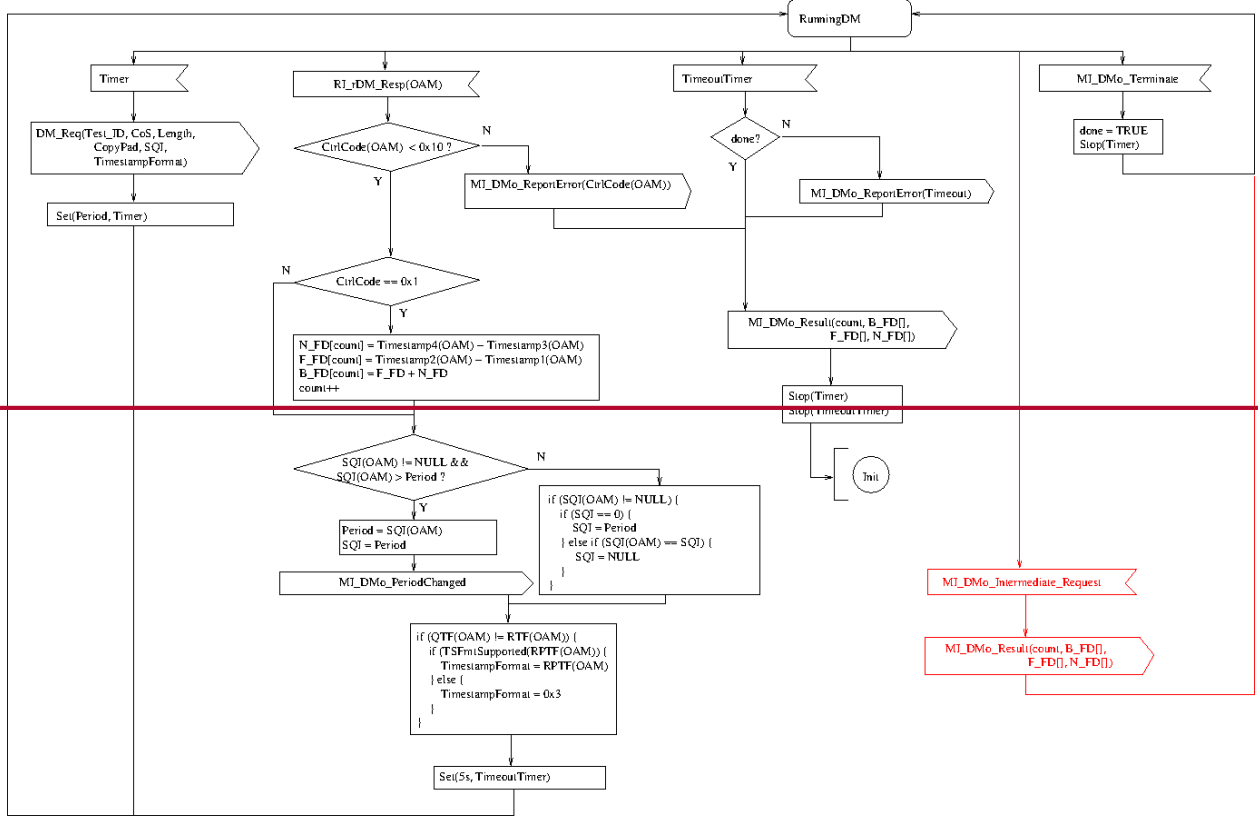


Figure 8-20c/G.8121.2/Y.1381.2 On-demand PM control process

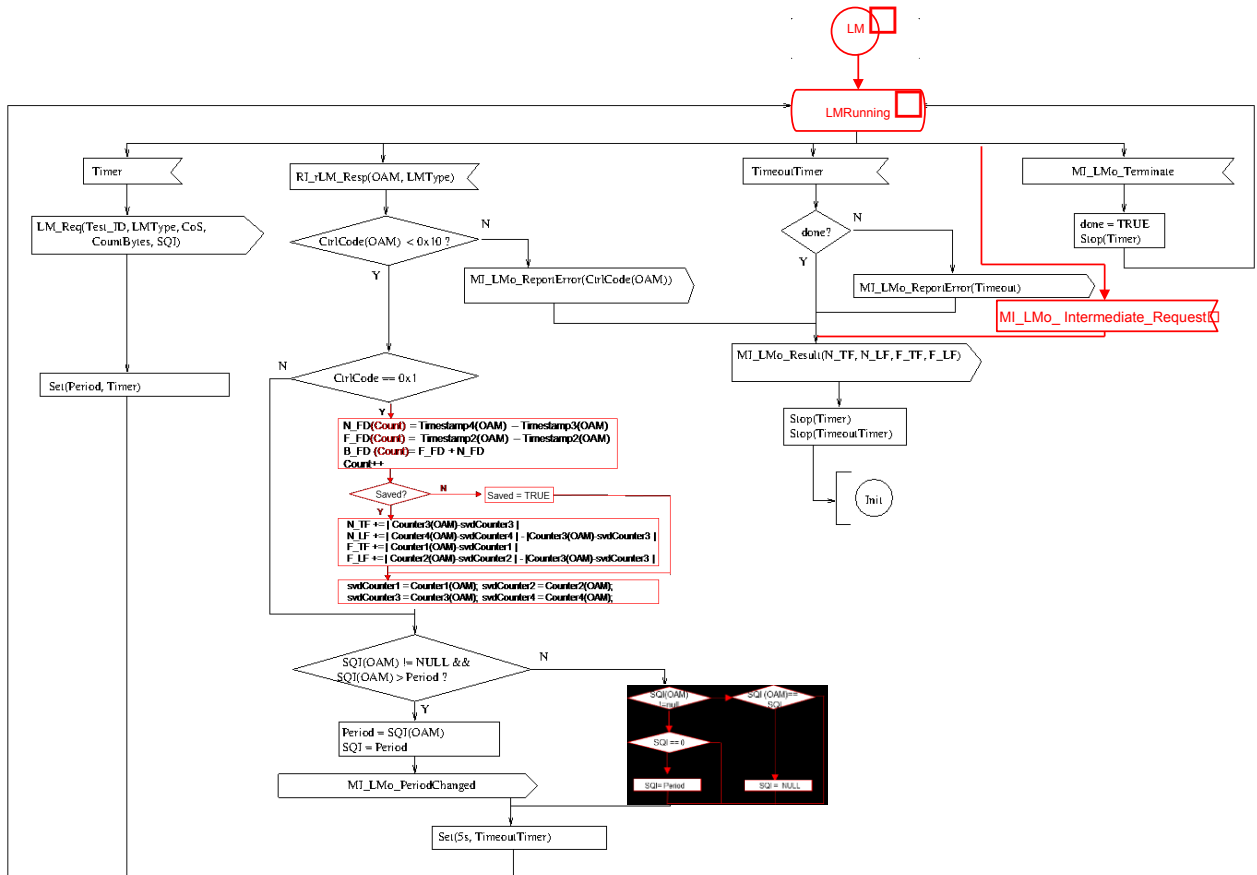
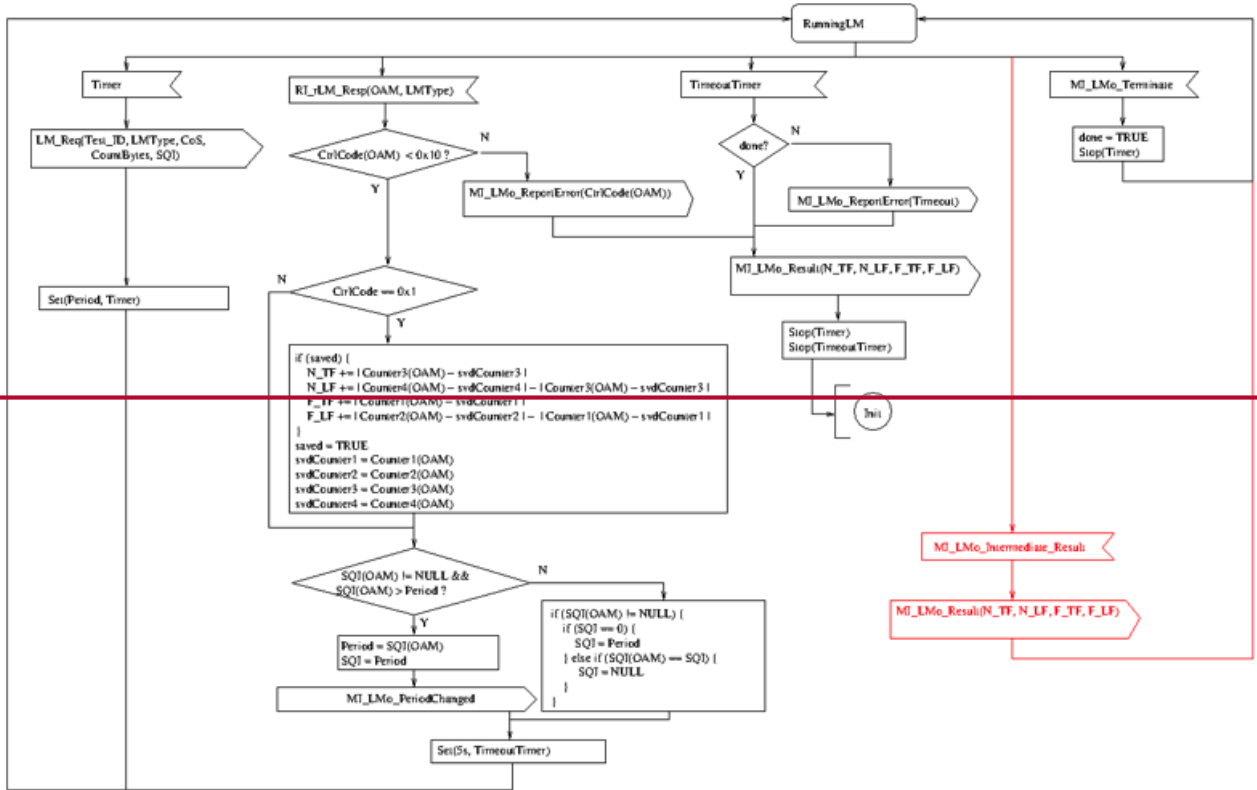


Figure 8-20d/G.8121.2/Y.1381.2 On-demand PM control process

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 for 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 for 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 for 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 in this Recommendation.

#### **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 in this Recommendation.

#### **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 in this Recommendation.

### **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.



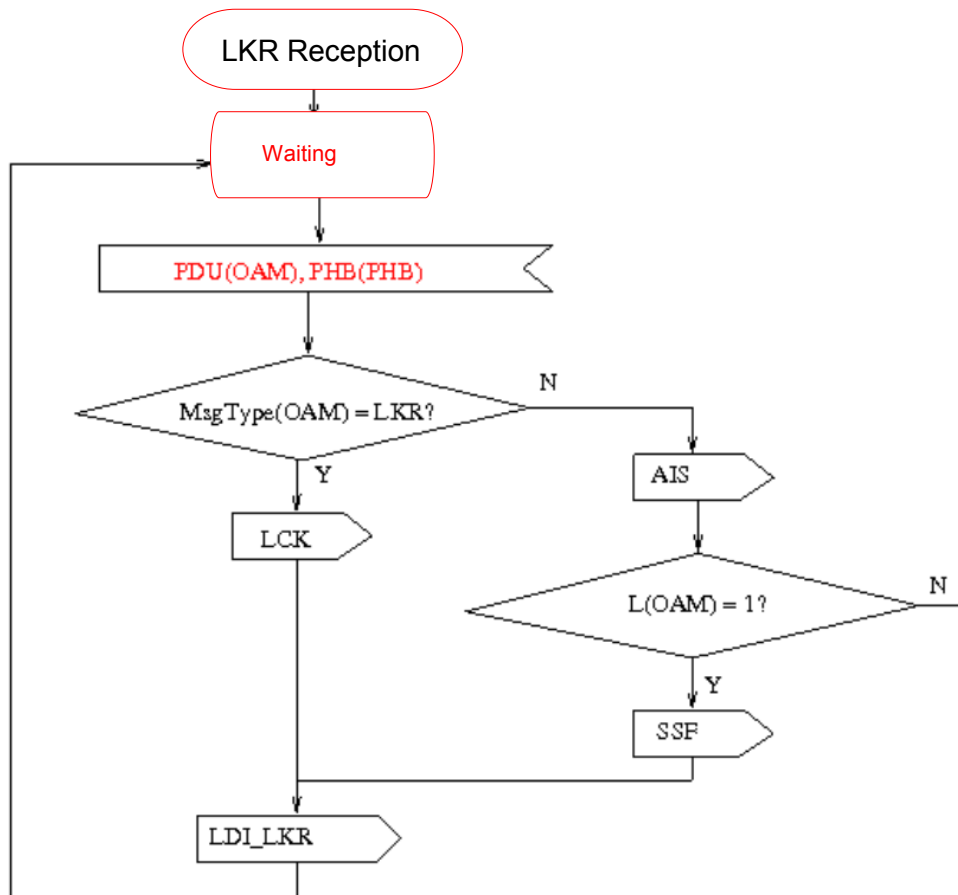
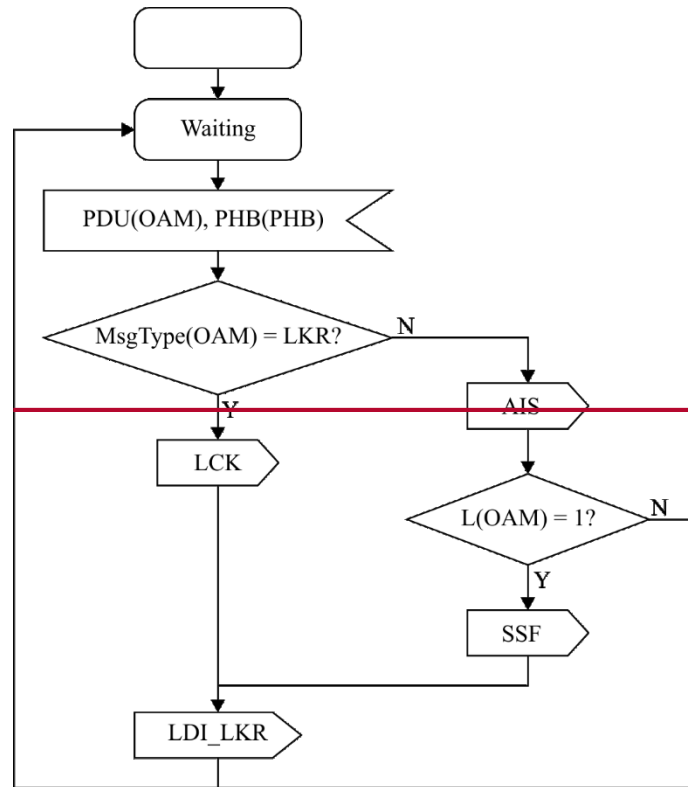
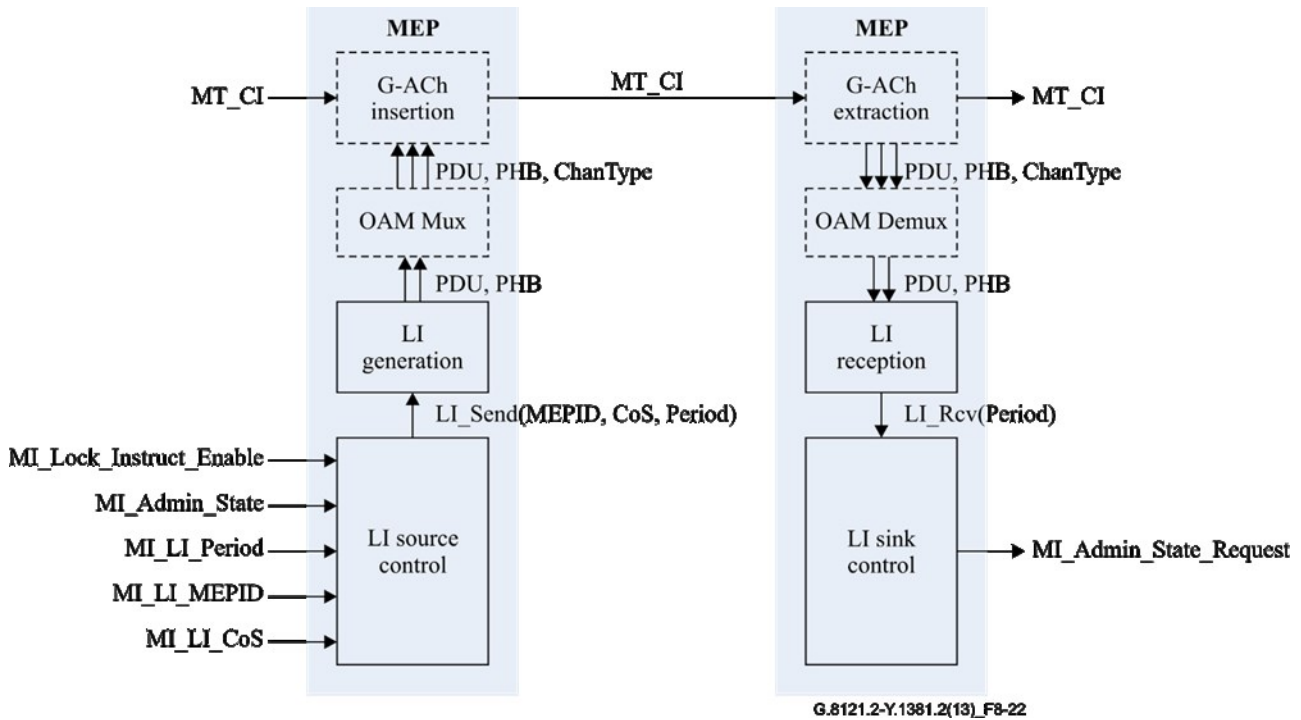


Figure 8-21/G.8121.2/Y.1381.2 LCR/AIS Reception behaviour

### 8.8.11. Lock Instruct processes

An overview of the processes relating to the Lock Instruct mechanism is shown in Figure 8-22. Note that G.8121 and rfc6371 use the abbreviation LKI for the same mechanism.



**Figure 8-22/G.8121.2/Y.1381.2 - 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 RFC6435].

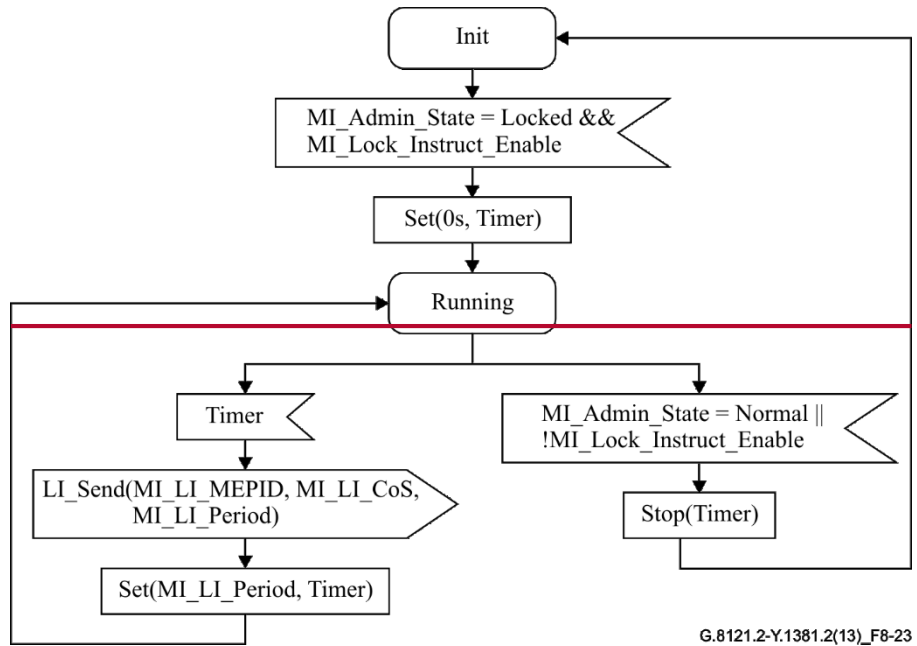
The LI Generation process formats LI messages and passes them to the OAM Mux process and hence to the G-ACh Insertion process.

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.



G.8121.2-Y.1381.2(13)\_F8-23

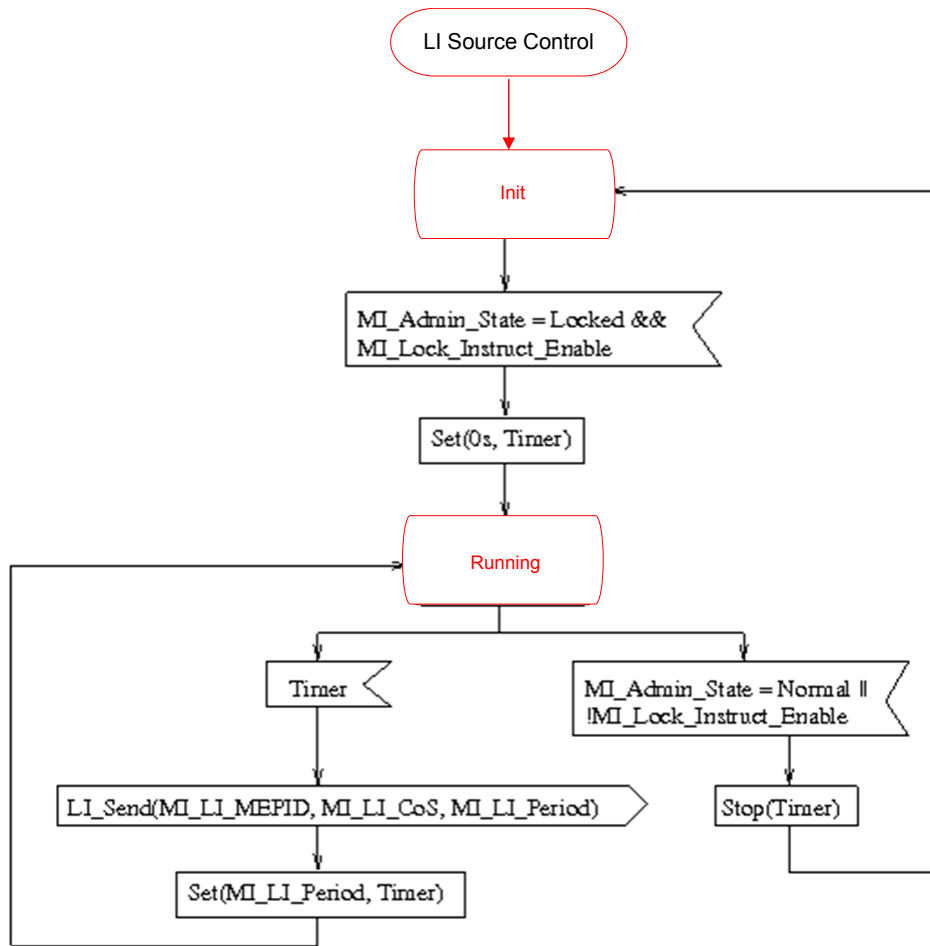
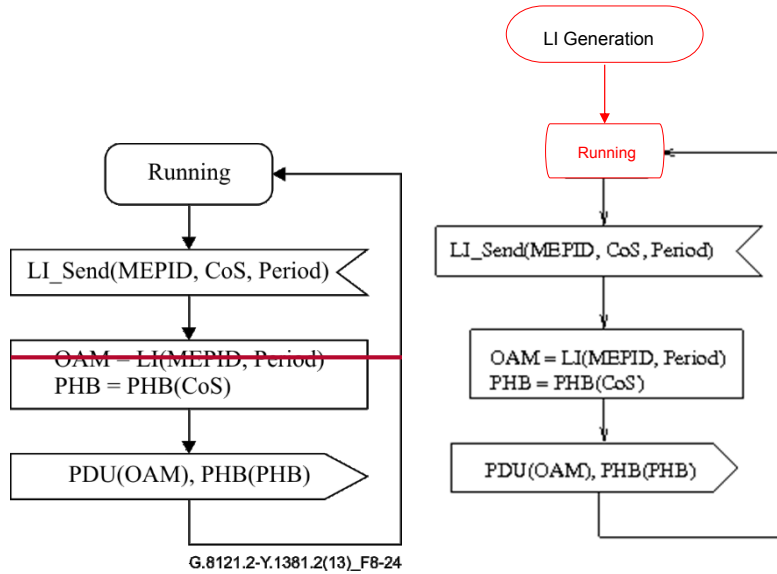


Figure 8-23/G.8121.2/Y.1381.2 - LI Source Control Process

### 8.8.11.2 LI Generation Process

The LI Generation Process is described in Figure 8-24.



**Figure 8-24/G.8121.2/Y.1381.2 - LI Generation Process**

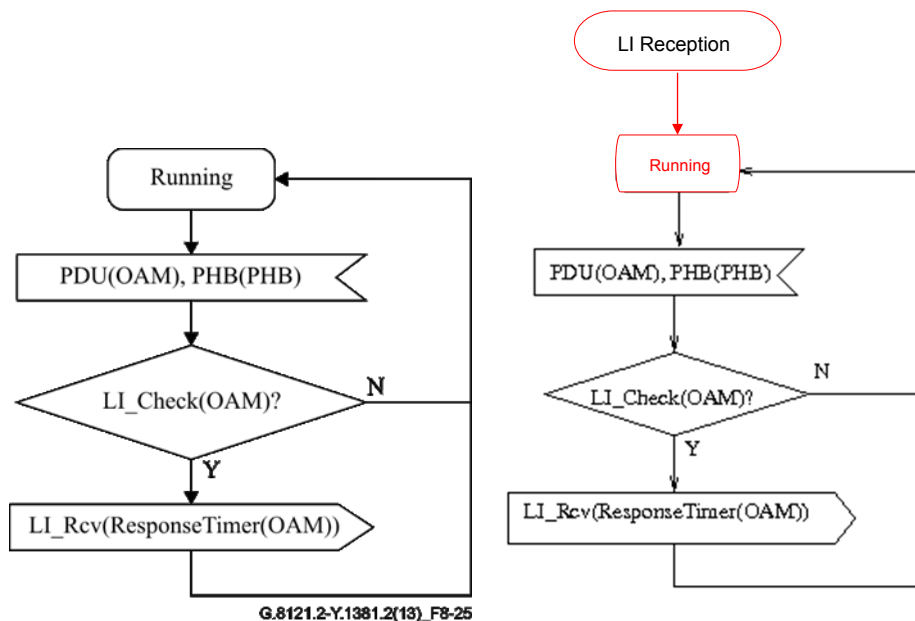
The LI(MEPID, Period) function formats an LI PDU according to [IETF RFC6435], as follows:

- 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.

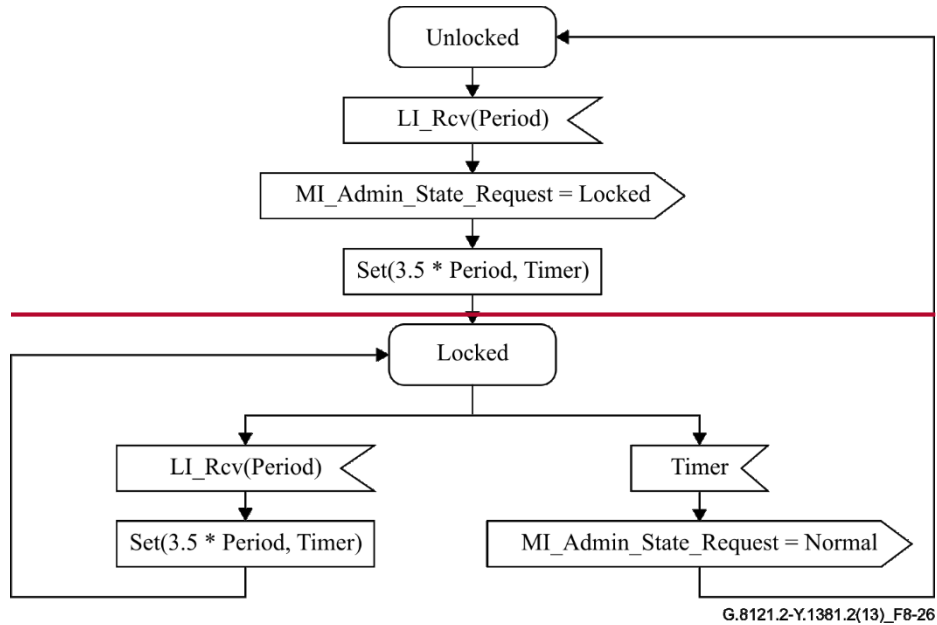


### Figure 8-25/G.8121.2/Y.1381.2 - LI Reception Process

The LI\_Check(OAM) function performs implementation-specific checks, including those described in [IETF RFC6435], 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.



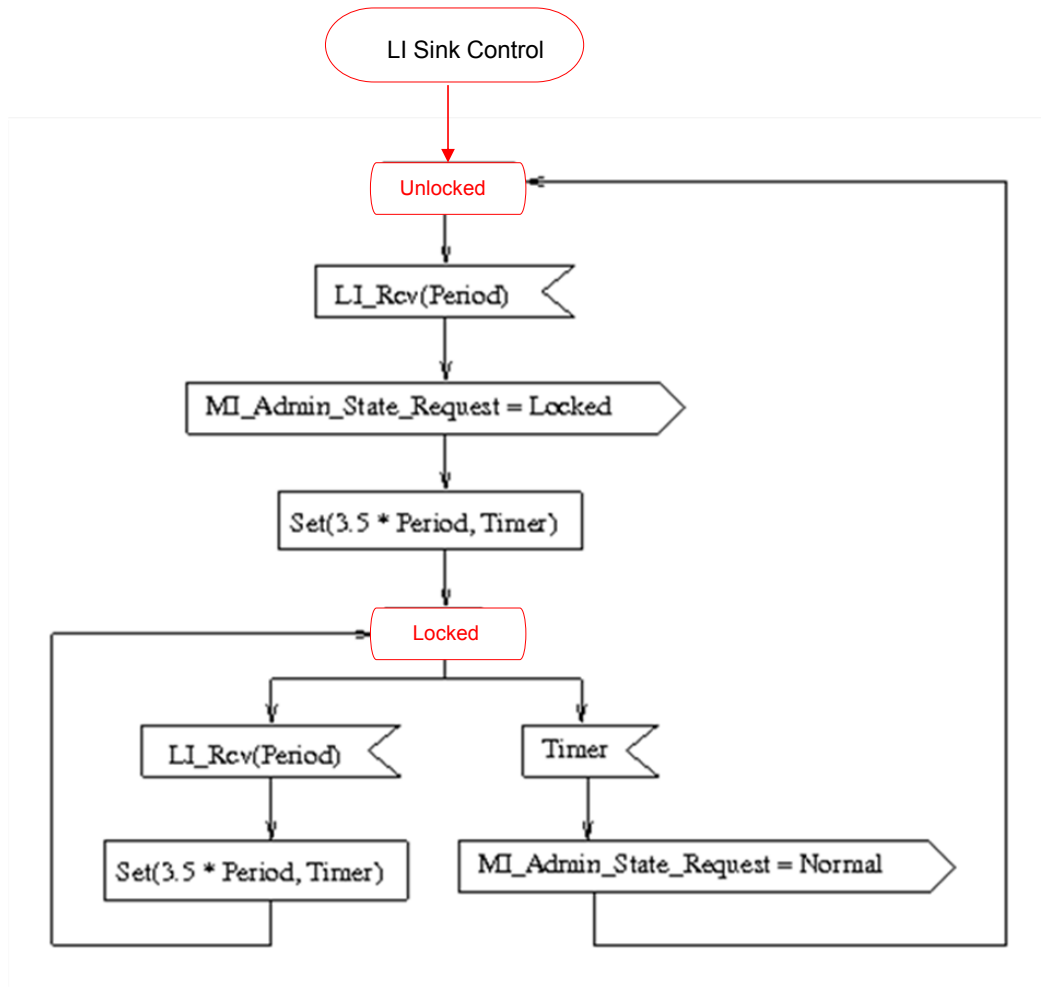


Figure 8-26/G.8121.2/Y.1381.2 - LI Sink Control Process

## 8.9 Dataplane Loopback Processes

See clause 8.9 in [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.

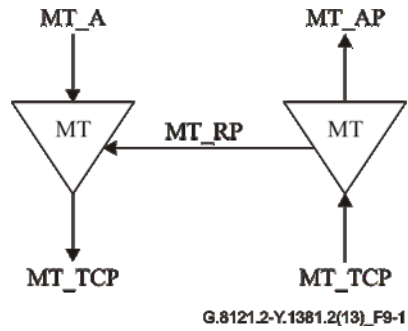


Figure 9-1/G.8121.2/Y.1381.2 MT\_TT

### 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 in Figure 9-2.

• **Symbol:**

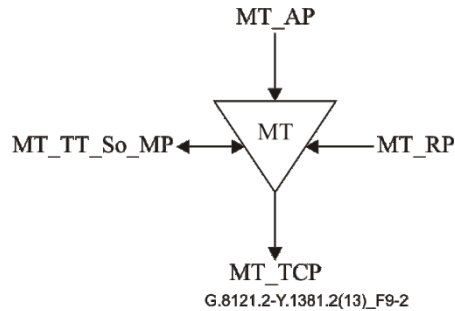


Figure 9-2G.8121.2/Y.1381.2 MT\_TT\_So symbol

• Interfaces:

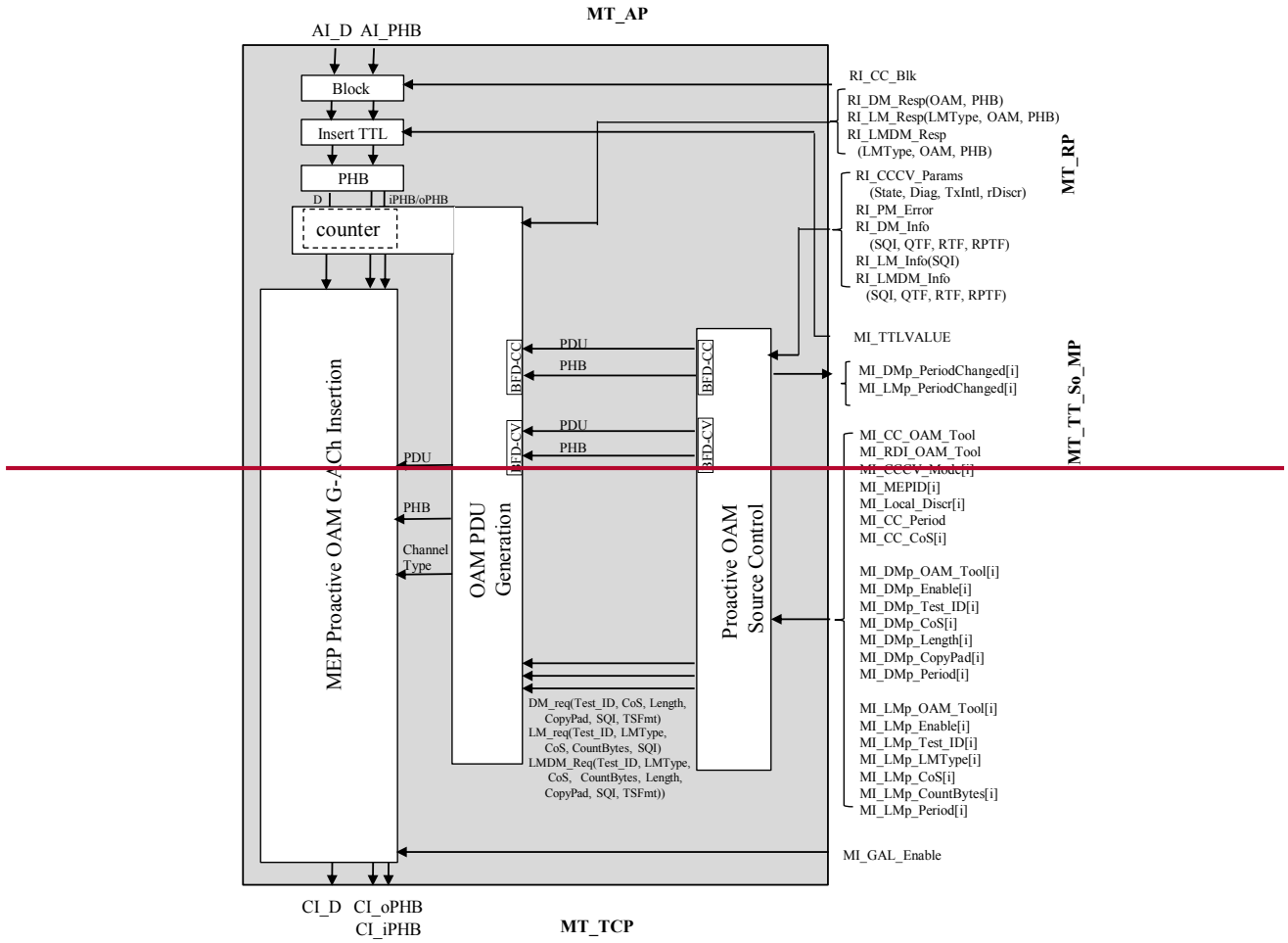
**Table 9-1/G.8121.2/Y.1381.2 – MT\_TT\_So inputs and outputs**

Input(s)	Output(s)
<p><b>MT_AP:</b> MT_AI_D MT_AI_PHB</p> <p><b>MT_RP:</b> MT_RI_CCCV_Params(State, Diag, TxIntl, rDiscr) MT_RI_CC_Blkc</p> <p>MT_RI_DM_Resp(OAM, PHB) MT_RI_LM_Resp(LMType, OAM, PHB) MT_RI_LMDM_Resp (LMType, OAM, PHB)</p> <p>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)</p> <p><b>MT_TT_So_MP:</b> MT_TT_So_MI_GAL_Enable MT_TT_So_MI_TTLValue</p> <p><u>MT_TT_So_MI_CC_OAM_Tool[...]</u> <u>MT_TT_So_MI_RDI_OAM_Tool[...]</u> MT_TT_So_MI_CCCV_Mode[1...M<sub>cccvcv</sub>] MT_TT_So_MI_MEPID[1...M<sub>cccvcv</sub>] MT_TT_So_MI_Local_Discr[1...M<sub>cccvcv</sub>] MT_TT_So_MI_CC_Period MT_TT_So_MI_CC_CoS[1...M<sub>cccvcv</sub>]</p> <p><u>MT_TT_So_MI_DMp_OAM_Tool[...]</u> MT_TT_So_MI_DMp_Enable[1...M<sub>DMp</sub>] MT_TT_So_MI_DMp_Test_ID[1...M<sub>DMp</sub>] MT_TT_So_MI_DMp_CoS[1...M<sub>DMp</sub>] MT_TT_So_MI_DMp_Length[1...M<sub>DMp</sub>] MT_TT_So_MI_DMp_CopyPad[1...M<sub>DMp</sub>] MT_TT_So_MI_DMp_Period[1...M<sub>DMp</sub>]</p> <p><u>MT_TT_So_MI_LMp_OAM_Tool[...]</u> MT_TT_So_MI_LMp_Enable[1...M<sub>LMp</sub>] MT_TT_So_MI_LMp_Test_ID[1...M<sub>LMp</sub>] MT_TT_So_MI_LMp_LMType[1...M<sub>LMp</sub>] MT_TT_So_MI_LMp_CoS[1...M<sub>LMp</sub>] MT_TT_So_MI_LMp_CountBytes[1...M<sub>LMp</sub>] MT_TT_So_MI_LMp_Period[1...M<sub>LMp</sub>]</p>	<p><b>MT_CP:</b> MT_CI_D MT_CI_oPHB MT_CI_iPHB</p> <p><b>MT_TT_So_MP:</b> MT_TT_So_MI_DMp_PeriodChanged[1...M<sub>DMp</sub>] MT_TT_So_MI_LMp_PeriodChanged[1...M<sub>LMp</sub>]</p>



- **Processes:**

The processes associated with the MT\_TT\_So function are as depicted in Figure 9-3. The sub-processes within each process, described in clause 8.8, are not shown separately



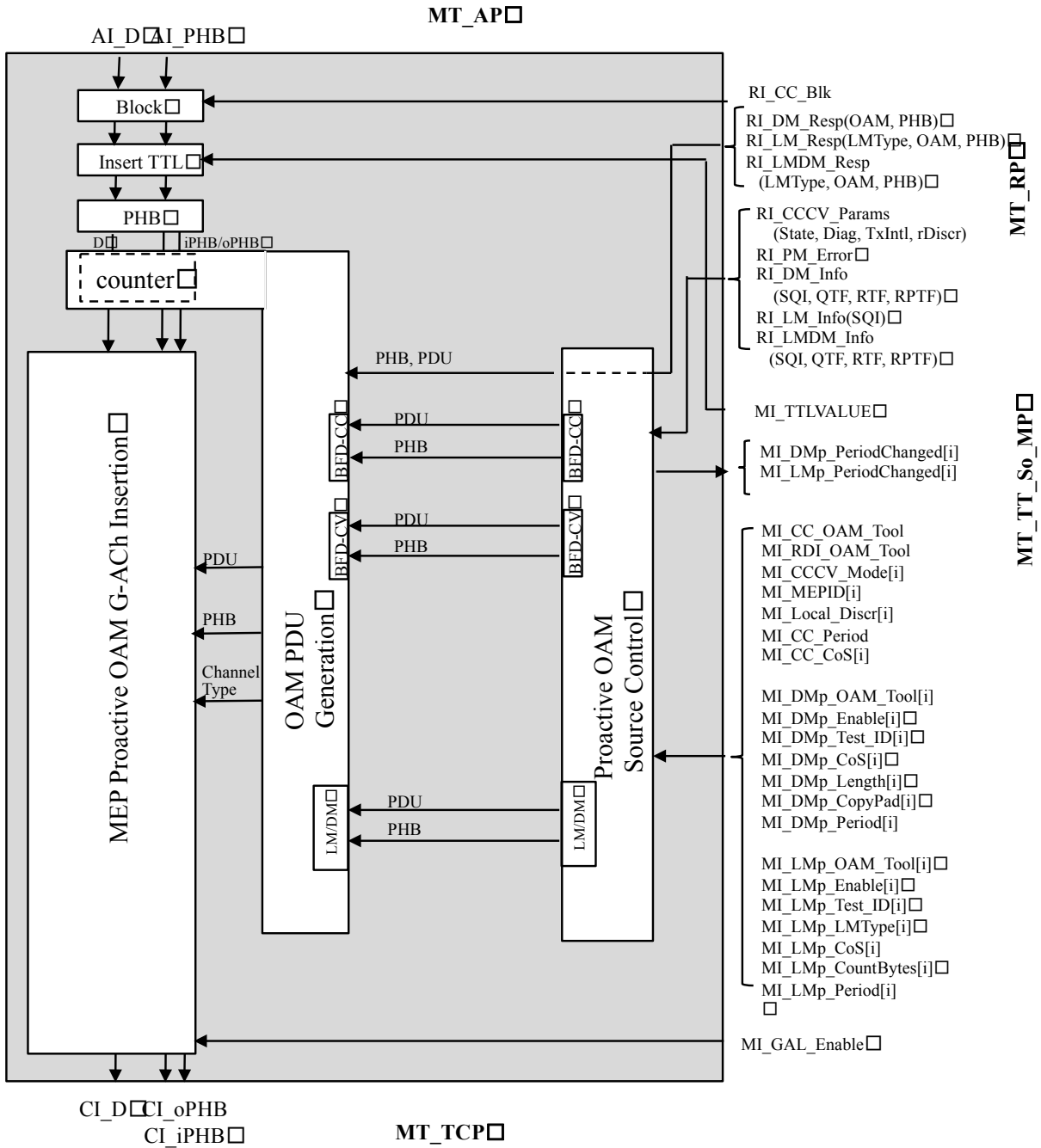


Figure 9-3/G.8121.2/Y.1381.2 MT\_TT\_So process

**PHB:** See 9.2.1.1 in [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 9.2.1.1 in [ITU-T G.8121].

**Block:** See 9.2.1.1 in [ITU-T G.8121].

**MEP Proactive OAM G-ACh Insertion:** See 8.1.2 in [ITU-T G.8121].

**OAM PDU Generation:** This contains following sub-processes as described in clause 8.8: PM Generation; OAM Mux; PM Mux.

**Proactive OAM Source Control:** This contains following sub-processes 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 RFC6374].

• **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 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:**

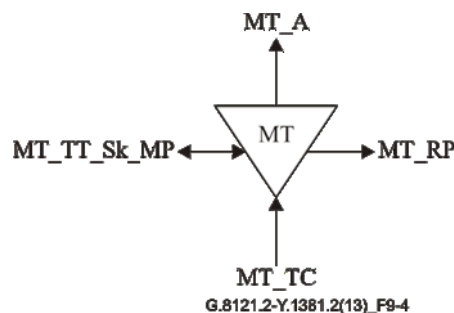


Figure 9-4/G.8121.2/Y.1381.2 MT\_TT\_Sk function symbol

• Interfaces:

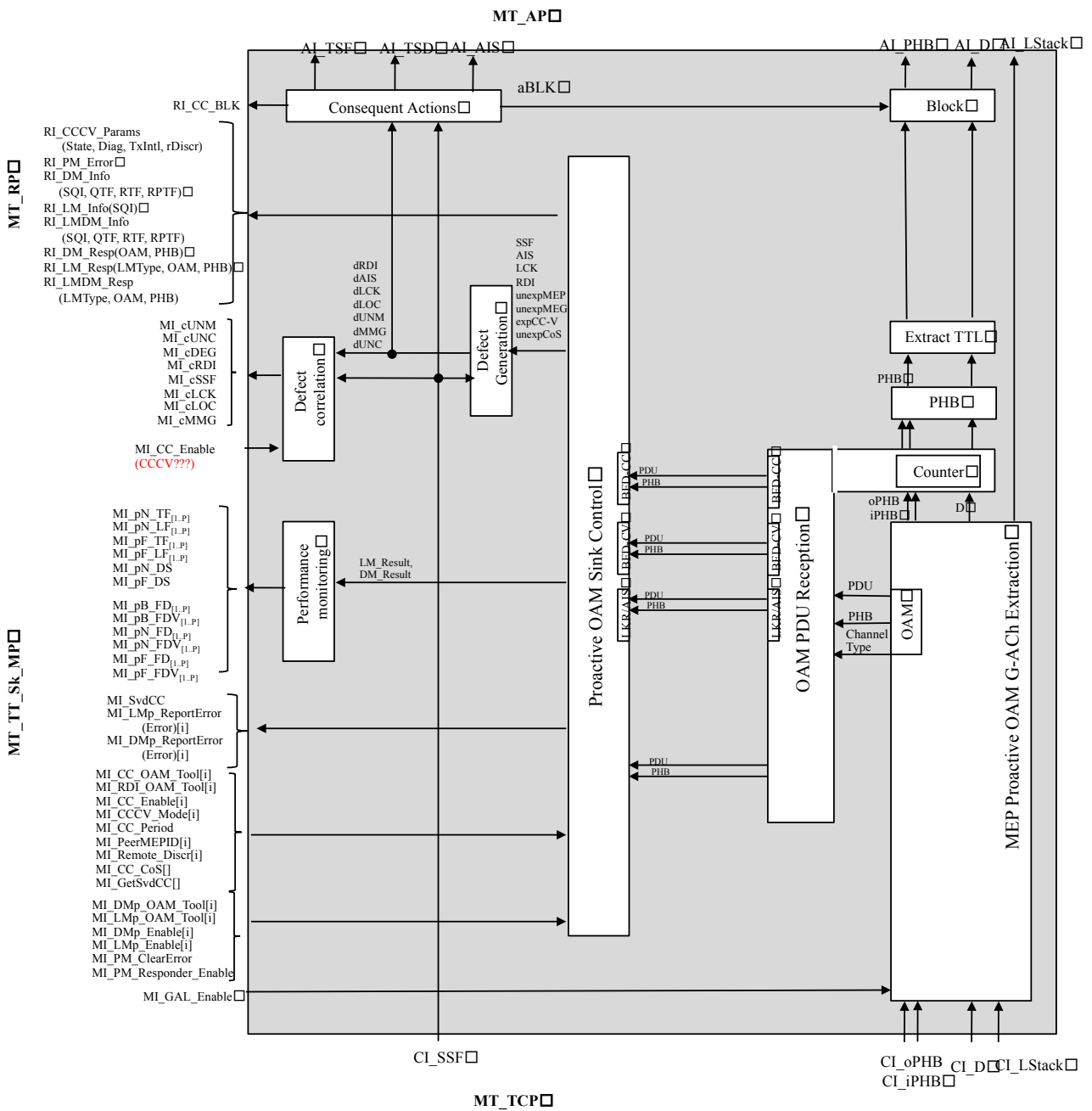
**Table 9-2/G.8121.2/Y.1381.2 – MT\_TT\_Sk inputs and outputs**

Input(s)	Output(s)
<b>MT_TCP:</b> MT_CI_D MT_CI_iPHB MT_CI_oPHB MT_CI_SSF MT_CI_LStack	<b>MT_AP:</b> MT_AI_D MT_AI_PHB MT_AI_TSF MT_AI_TSD MT_AI_AIS
<b>MT_TT_Sk_MP:</b> MT_TT_Sk_MI_GAL_Enable <u>MT_TT_Sk_MI_CC_OAM_Tool[...]</u> <u>MT_TT_Sk_MI_RDI_OAM_Tool[...]</u> MT_TT_Sk_MI_CC_Enable[1...Mcccv] MT_TT_Sk_MI_CCCV_Mode[1...Mcccv] MT_TT_Sk_MI_CC_Period MT_TT_Sk_MI_PeerMEPID[1...Mcccv] MT_TT_Sk_MI_Remote_Discr[1...Mcccv] MT_TT_Sk_MI_CC_CoS[...] MT_TT_Sk_MI_GetSvdCC[1...Mcccv]	MT_AI_LStack  <b>MT_RP:</b> MT_RI_CCCV_Params(State, Diag, TxIntl, rDiscr) MT_RI_CC_Blks  MT_RI_DM_Resp(OAM, PHB) MT_RI_LM_Resp(LMType, OAM, PHB) MT_RI_LMDM_Resp(LMType, OAM, PHB)
<u>MT_TT_Sk_MI_DMp_OAM_Tool[...]</u> <u>MT_TT_Sk_MI_LMp_OAM_Tool[...]</u> MT_TT_Sk_MI_DMp_Enable[1...M <sub>DMp</sub> ] MT_TT_Sk_MI_LMp_Enable[1...M <sub>LMp</sub> ]	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_MI_PM_ClearError MT_TT_Sk_MI_PM_Responder_Enable	<b>MT_TT_Sk_MP:</b> 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 <sub>DMp</sub> ]  MT_TT_Sk_MI_LMp_ReportError(Error)[1...M <sub>LMp</sub> ] 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]

Input(s)	Output(s)
	MT_TT_Sk_MI_pB_FD[1...P] MT_TT_Sk_MI_pN_FD[1...P] MT_TT_Sk_MI_pN_FD[1...P] MT_TT_Sk_MI_pF_FD[1...P] MT_TT_Sk_MI_pF_FD[1...P]

• **Processes:**

The processes associated with the MT\_TT\_Sk function are as depicted in Figure 9-5.



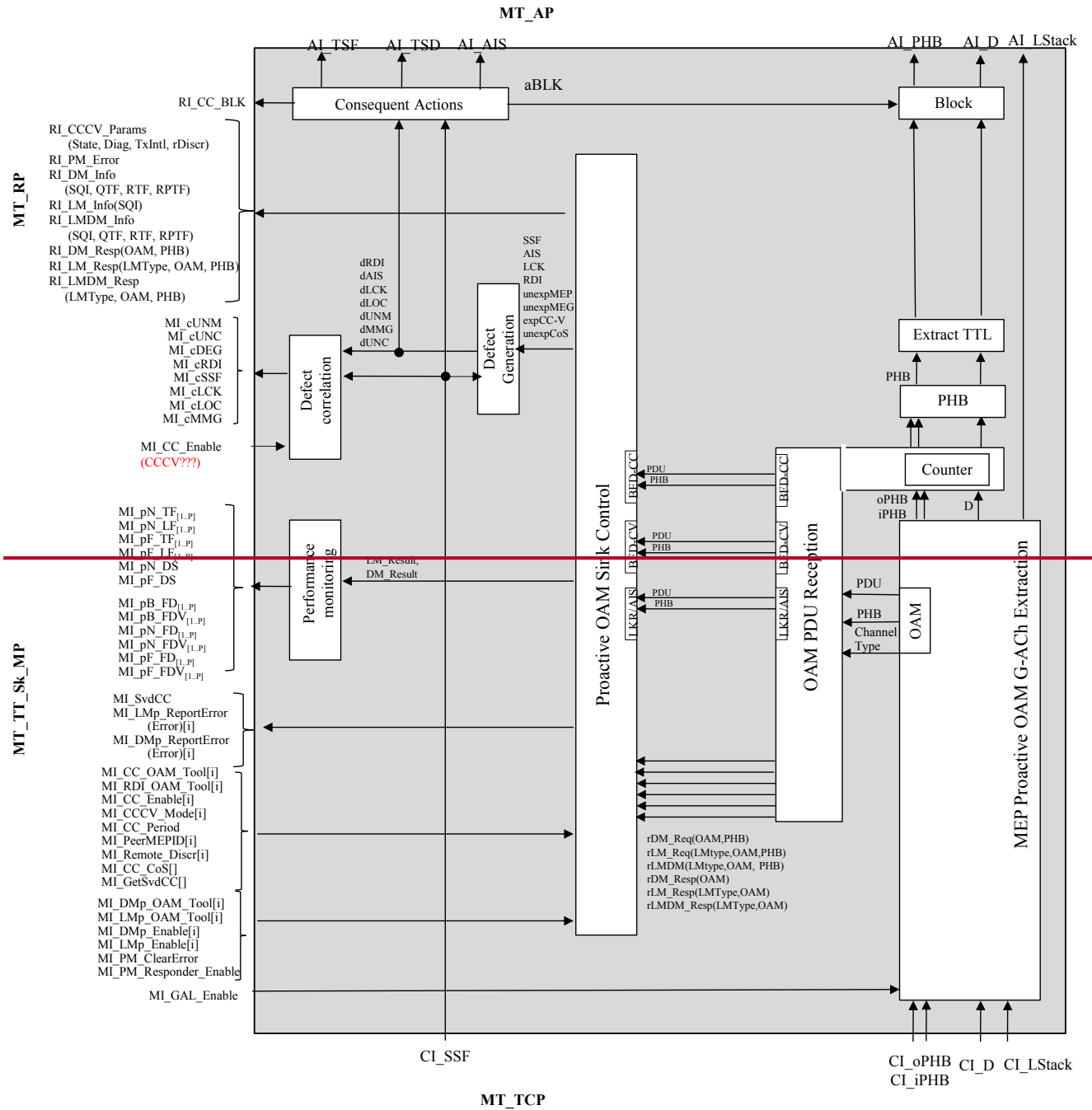


Figure 9-5/G.8121.2/G.8121.2/Y.1381.2 – MT\_TT\_Sk Process

**PHB:** See 9.2.1.2 in [ITU-T G.8121].

**Extract TTL:** See 9.2.1.2 in [ITU-T G.8121].

**Block:** See 9.2.1.2 in [ITU-T G.8121].

**MEP Proactive OAM G-ACh Extraction:** see 8.1.3 in [ITU-T G.8121].

**OAM PDU Reception:** This contains following sub-processes as described in clause 8.8: PM Reception; OAM Demux; Session Demux; PM Demux

**Proactive OAM Sink Control:** This contains following sub-processes as described in clause 8.8: CCCV Reception; LCK/AIS Reception; Proactive PM Sink Control; PM Responder.

**Performance Counter:** See 9.2.1.2 in [ITU-T G.8121].

**Defect Generation:** See 9.2.1.2 in [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 in [ITU-T G.8121]. They use the OAM protocol specific AIS insert process and LCK generation process as defined in clause 8.6.2 and 8.6.3. For the MT/MT\_A\_Sk function, in addition to the MI shown in Table 9.5 in [ITU-T G.8121] and Figure 9.11 in [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:

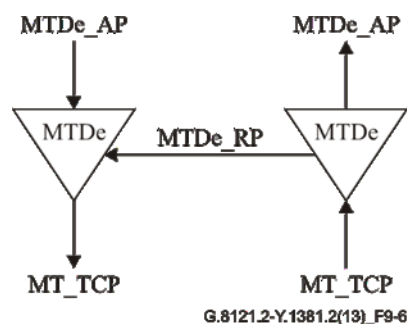


Figure 9-6/G.8121.2/Y.1381.2 - MTDe\_TT



### 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

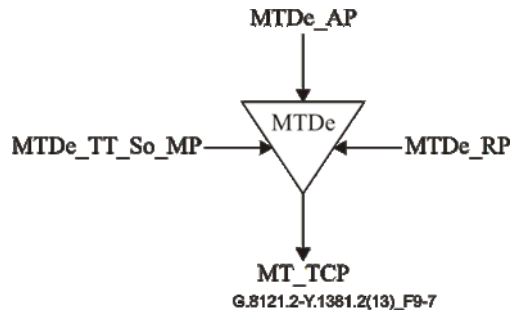


Figure 9-7/G.8121.2/Y.1381.2 - MTDe\_TT\_So\_Symbol

#### Interfaces

Table 9-3/G.8121.2/Y.1381.2 – MTDe\_TT\_So interfaces

Input(s)	Output(s)
<p><b>MT_AP:</b> MTDe_AI_D MTDe_AI_iPHB MTDe_AI_oPHB</p> <p><b>MTDe_RP:</b> MTDe_RI_LSPPing_Rsp(OAM, PHB, RC, SubRC, Err_TLV, include_ifstack, LStack Include_dsmmap, 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)</p> <p>MTDe_RI_rLSPPing_Rsp(OAM, PHB, LStack) MTDe_RI_rDM_Resp(OAM) MTDe_RI_rLM_Resp(LMtype, OAM) MTDe_RI_rLMDM_Resp(LMtype, OAM)</p> <p>MTDe_RI_CI(D, iPHB, oPHB)</p> <p><b>MTDe_TT_So_MP:</b> MTDe_TT_So_MI_GAL_Enable</p> <p><u>MTDe_TT_So_MI_CV_OAM_Tool</u></p>	<p><b>MT_CP:</b> MT_CI_D MT_CI_oPHB MT_CI_iPHB</p> <p><b>MTDe_RP:</b></p> <p><b>MTDe_TT_So_MP:</b> MTDe_TT_So_MI_DMo_ReportError(Error) [1...M<sub>DMo</sub>] MTDe_TT_So_MI_DMo_PeriodChanged [1...M<sub>DMo</sub>] MTDe_TT_So_MI_LMo_ReportError(Error) [1...M<sub>LMo</sub>] MTDe_TT_So_MI_LMo_PeriodChanged [1...M<sub>LMo</sub>] MTDe_TT_So_MI_DMo_Result(count, B_FD[], F_FD[], N_FD[])[1...M<sub>DMo</sub>] MTDe_TT_So_MI_LMo_Result(N_TF, N_LF, F_TF, F_LF)[1...M<sub>LMo</sub>]</p> <p>MTDe_TT_So_MI_CV_Series_Result(Session_ID, Rcv, OOO, FWErr, BWErr) MTDe_TT_So_MI_⊕CV_Trace_Result (Session_ID, Result) MTDe_TT_So_MI_⊕CV_FWErr(Session_ID, Seq, RC, SubRC, ErrTLV) MTDe_TT_So_MI_⊕CV_BWErr(Session_ID, Seq, RC, SubRC, ErrTLV)</p>

MTDe_TT_So_MI_CV_Series (Session_ID, Count, Period, CoS, Size, ValidateFEC, ValidateReverse, TargetFECStack) MTDe_TT_So_MI_⊖CV_Trace (Session_ID, CoS, ValidateFEC, ValidateReverse, TargetFECStack) MTDe_TT_So_MI_FEC_Checking MTDe_TT_So_MI_Target_FEC <del>MTDe_TT_So_MI_Iffnum</del> <del>MTDe_TT_So_MI_MTU [C.1074]</del>  <u>MTDe_TT_So_MI_LMo_OAM_Tool[...]</u> <u>MTDe_TT_So_MI_DMo_OAM_Tool[...]</u> MTDe_TT_So_MI_DMo_Start(CoS, Test_ID, Length, Period, CopyPad)[1...M <sub>DMo</sub> ] MTDe_TT_So_MI_LMo_Start(CoS, Test_ID, Period, LMType, CountBytes)[1...M <sub>LMo</sub> ] MTDe_TT_So_MI_LMDMo_Start(CoS, Test_ID, Length, Period, LMType, CountBytes, CopyPad)[1...M <sub>LMDMo</sub> ] <u>MTDe_TT_So_MI_DMo_Intermediate_Request [1...M<sub>DMo</sub>]</u> <u>MTDe_TT_So_MI_LMo_Intermediate_Request [1...M<sub>LMo</sub>]</u> MTDe_TT_So_MI_DMo_Terminate [1...M <sub>DMo</sub> ] MTDe_TT_So_MI_LMo_Terminate[1...M <sub>LMo</sub> ] MTDe_TT_So_MI_LMDMo_Terminate [1...M <sub>LMDMo</sub> ]  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

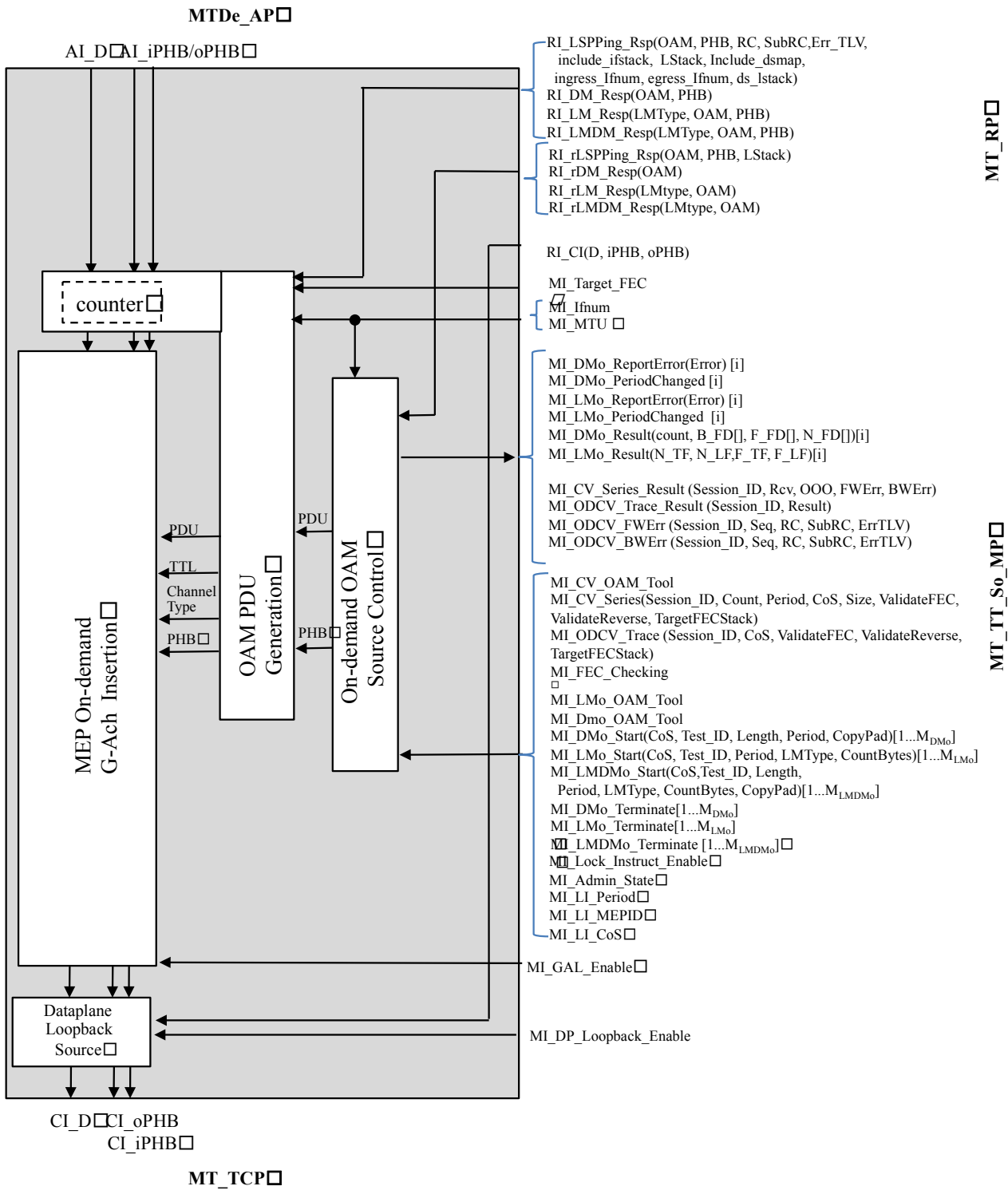


Figure 9-8/G.8121.2/Y.1381.2 – MTDe\_TT\_So Process

[Add MI\_Intermediate\_request in align with Table]  
[Remove MI\_Iifnum and MTU and OD from ODCV]

**MEP On-demand OAM G-ACh Insertion:** See 8.1.2 in [ITU-T G.8121]

**OAM PDU Generation:** This contains following sub-processes 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 sub-processes 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].

**Dataplane Loopback Source process:** see clause 8.9.1 in [ITU-T G.8121]

• **Defects:**

*None.*

• **Consequent actions:**

*None.*

• **Defect correlations:**

*None.*

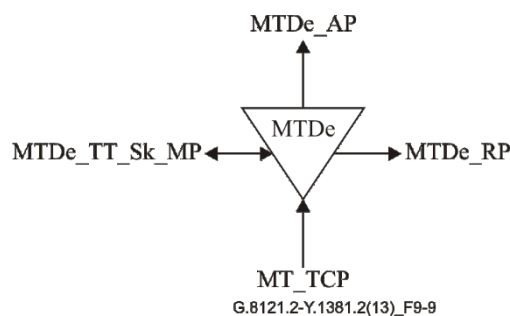
• **Performance monitoring:**

*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



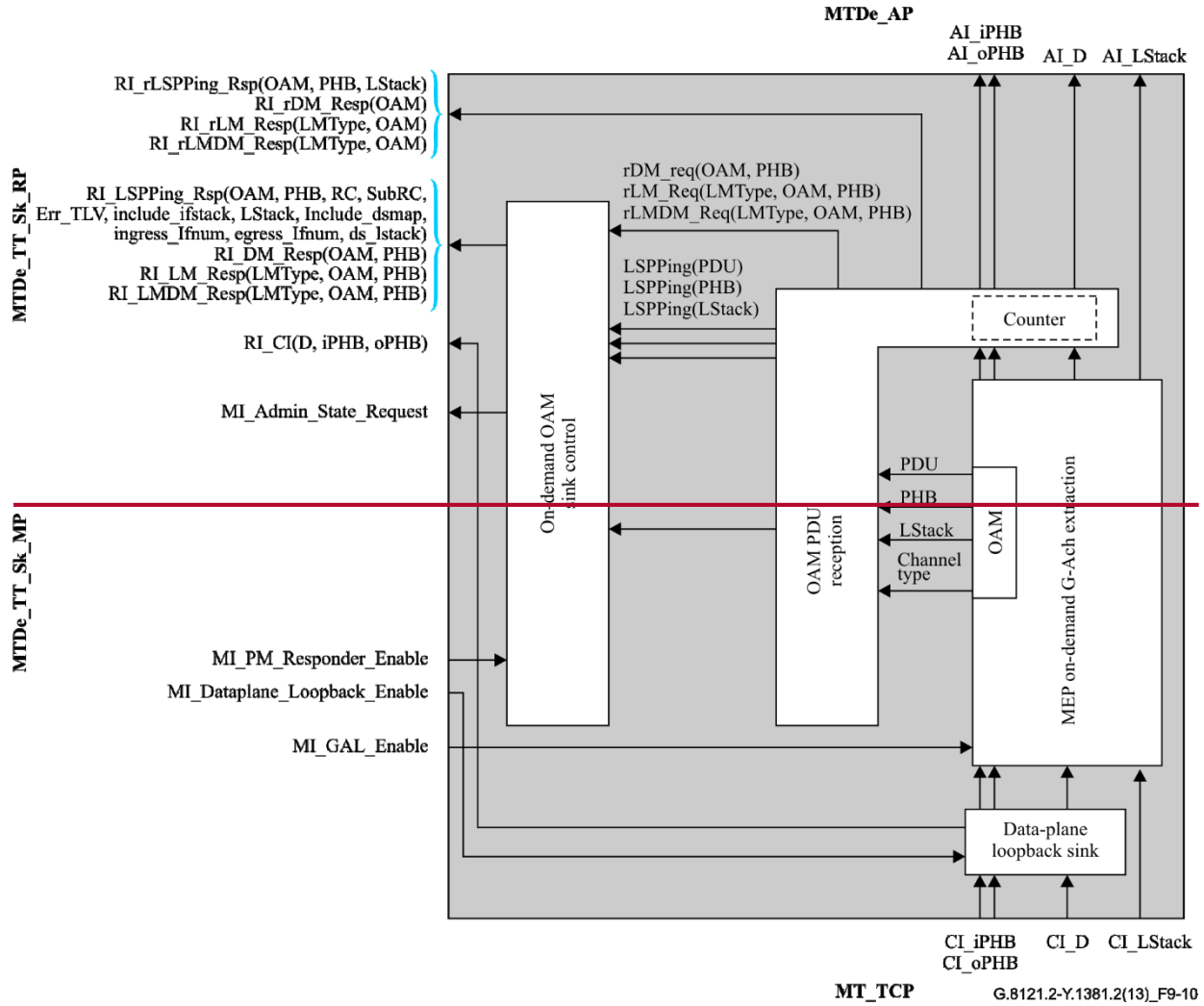
**Figure 9-9/G.8121.2/Y.1381.2 - MTDe\_TT\_Sk\_Symbol**

#### Interfaces

**Table 9-4/G.8121.2/Y.1381.2 – MTDe\_TT\_Sk interfaces**

Input(s)	Output(s)
<p><b>MT_CP:</b> MT_CI_D MT_CI_iPHB MT_CI_oPHB MT_CI_LStack</p> <p><b>MT_RP:</b></p> <p><b>MTDe_TT_Sk_MP:</b> MTDe_TT_Sk_MI_GAL_Enable MTDe_TT_Sk_MI_FEC_Checking <u>MTDe_TT_Sk_MI_CV_OAM_Tool</u> <u>MTDe_TT_Sk_MI_LMo_OAM_Tool</u> <u>MTDe_TT_Sk_MI_DMo_OAM_Tool</u></p> <p>MTDe_TT_Sk_MI_PM_Responder_Enable MTDe_TT_Sk_MI_DP_Loopback_Enable</p>	<p><b>MTDe_AP:</b> MTDe_AI_D MTDe_AI_iPHB MTDe_AI_oPHB MTDe_AI_LStack</p> <p><b>MTDe_RP:</b> MTDe_RI_rLSPPing_Rsp (OAM, PHB, LStack)</p> <p>MTDe_RI_LSPPing_Rsp(OAM, PHB, RC, SubRC,Err_TLV, include_ifstack, LStack, Include_dsmapi, ingress_ifnum, egress_ifnum, ds_lstack)</p> <p>MTDe_RI_rDM_Resp(OAM) MTDe_RI_rLM_Resp (LMtype,OAM) 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)</p> <p>MTDe_RI_CI(D, iPHB, oPHB)</p> <p><b>MTDe_FT_Sk_MP:</b> MTDe_TT_Sk_MI_Admin_State_Request</p>

**Processes**



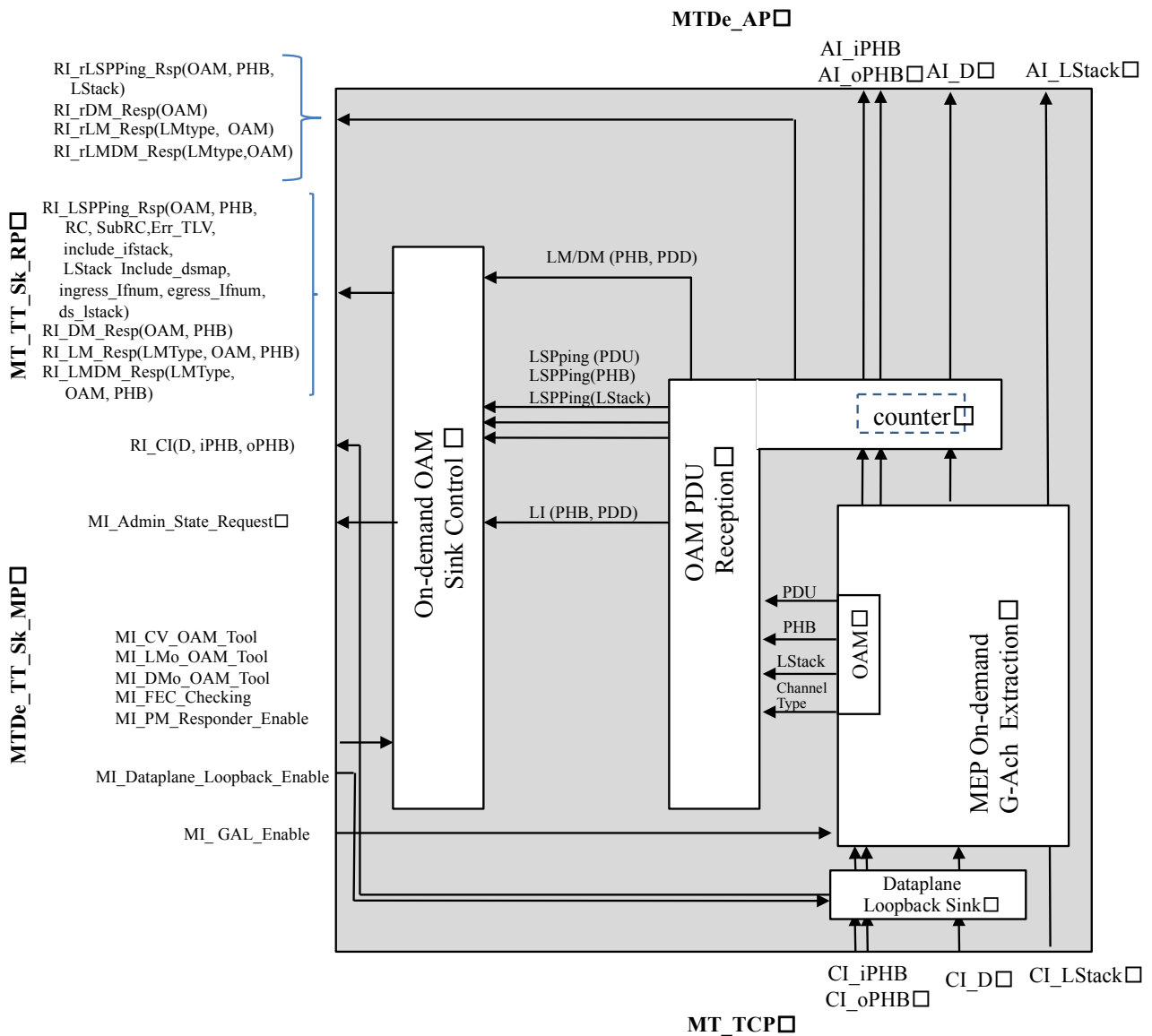


Figure 9-10/G.8121.2/Y.1381.2 – MTDe\_TT\_Sk Process

**MEP On-demand G-Ach Extraction:** see 8.1.3 in [ITU-T G.8121].

**OAM PDU Reception:** This contains following sub-processes 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 following sub-processes 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 RFC6374].

**Dataplane Loopback Sink:** see clause 8.9.1 in [ITU-T G.8121]

• **Defects:**

None.

• **Consequent actions:**

*None.*

• **Defect correlations:**

*None.*

• **Performance monitoring:**

*None.*

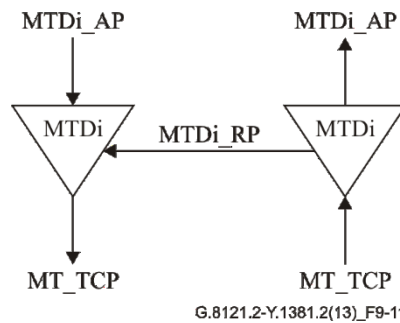
**9.4.1.2 MTDe to MT Adaptation Functions (MTDe/MT\_A)**

The MPLS-TP MEP Diagnostic Adaptation Function (MTDe/MT\_A) is described in Clause 9.4.1.2 in [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.



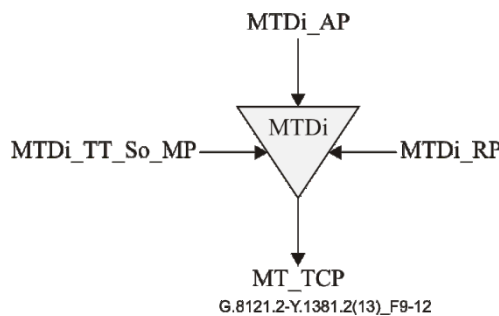
**Figure 9-11/G.8121.2/Y.1381.2 - MTDi\_TT**

**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 Figure 9-12.

• **Symbol:**



**Figure 9-12/G.8121.2/Y.1381.2 - MTDi\_TT\_So symbol**

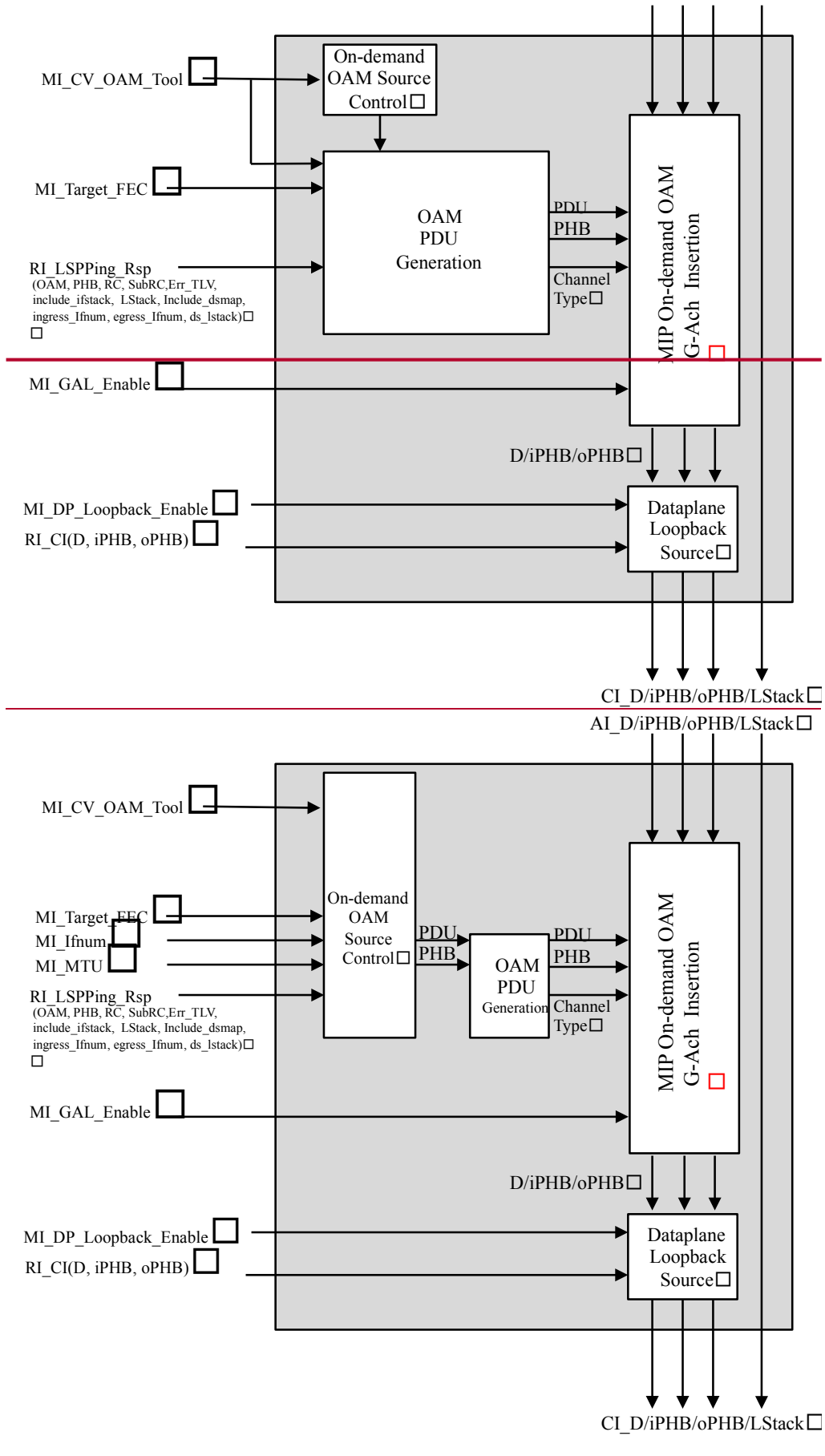


• Interfaces:

**Table 9-5/G.8121.2/Y.1381.2 – MTDi\_TT\_So inputs and outputs**

Input(s)	Output(s)
<p><b>MTDi_AP:</b>            MTDi_AI_D            MTDi_AI_iPHB            MTDi_AI_oPHB  <u>MT_AI_Lstack</u></p> <p><b>MTDi_RP:</b>            MTDi_RI_LSPPing_Rsp            MTDi_RI_CI</p> <p><b>MTDi_TT_So_MP:</b>            MTDi_TT_So_MI_Target_FEC  <del>MTDi_TT_So_MI&gt;Ifnum</del>  <del>MTDi_TT_So_MI&gt;MTU</del> <b>[C1074]</b>            MTDi_TT_So<del>k</del> MI_GAL_Enable  <del>MTDi_TT_So_MI&gt;CV&gt;OAM Tool</del>            MTDi_TT_So_MI_DP_Loopback_Enable</p>	<p><b>MT_CP:</b>            MT_CI_D            MT_CI_oPHB            MT_CI_iPHB  <u>MT_CI_Lstack</u></p>

• **Processes:**



**Figure 9-13/G.8121.2/Y.1381.2 - MTDi\_TT\_So Process**

The processes associated with the MTDi\_TT\_So function are depicted in the Figure below.

**G-ACh Insertion:** See 8.1.2 in [ITU-T G.8121].

**OAM PDU Generation:** This contains the following sub-processes as described in clause 8.8: On-demand CV Response Generation; OAM Mux

**On-demand OAM Source Control:** This process performs no operations.

**Dataplane Loopback Source:** see clause 8.9.1 in [ITU-T G.8121]

• **Defects:**

*None.*

• **Consequent actions:**

*None.*

• **Defect correlations:**

*None.*

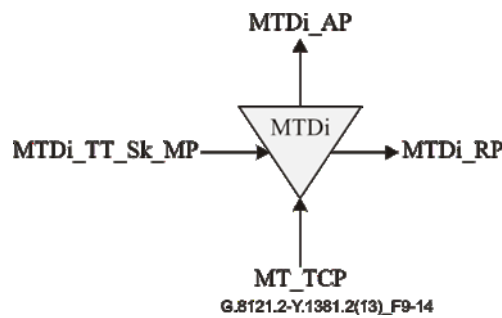
• **Performance monitoring:**

*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 14

• **Symbol:**



**Figure 9-14/G.8121.2/Y.1381.2 - MTDi\_TT\_Sk symbol**

• **Interfaces:**

**Table 9-6/G.8121.2/Y.1381.2 – MTDi\_TT\_Sk inputs and outputs**

Input(s)	Output(s)
<b>MT_TCP:</b> MT_CI_D MT_CI_iPHB MT_CI_oPHB	<b>MTDi_AP:</b> MTDi_AI_D MTDi_AI_iPHB MTDi_AI_oPHB

Input(s)	Output(s)
MT_CI_LStack  <b>MTDi_TT_Sk_MP:</b> MTDi_TT_Sk_MI_FEC_Checking MTDi_TT_Sk_MI_GAL_Enable <u>MTDi_TT_Sk_MI_CV_OAM_Tool</u> MTDi_TT_Sk_MI_DP_Loopback_Enable	MTDi_AI_LStack  <b>MTDi_RP:</b> MTDi_RI_LSPPing_Rsp MTDi_RI_CI

• **Processes:**

The processes associated with the MTDi\_TT\_Sk function are as depicted in Figure 9-15.

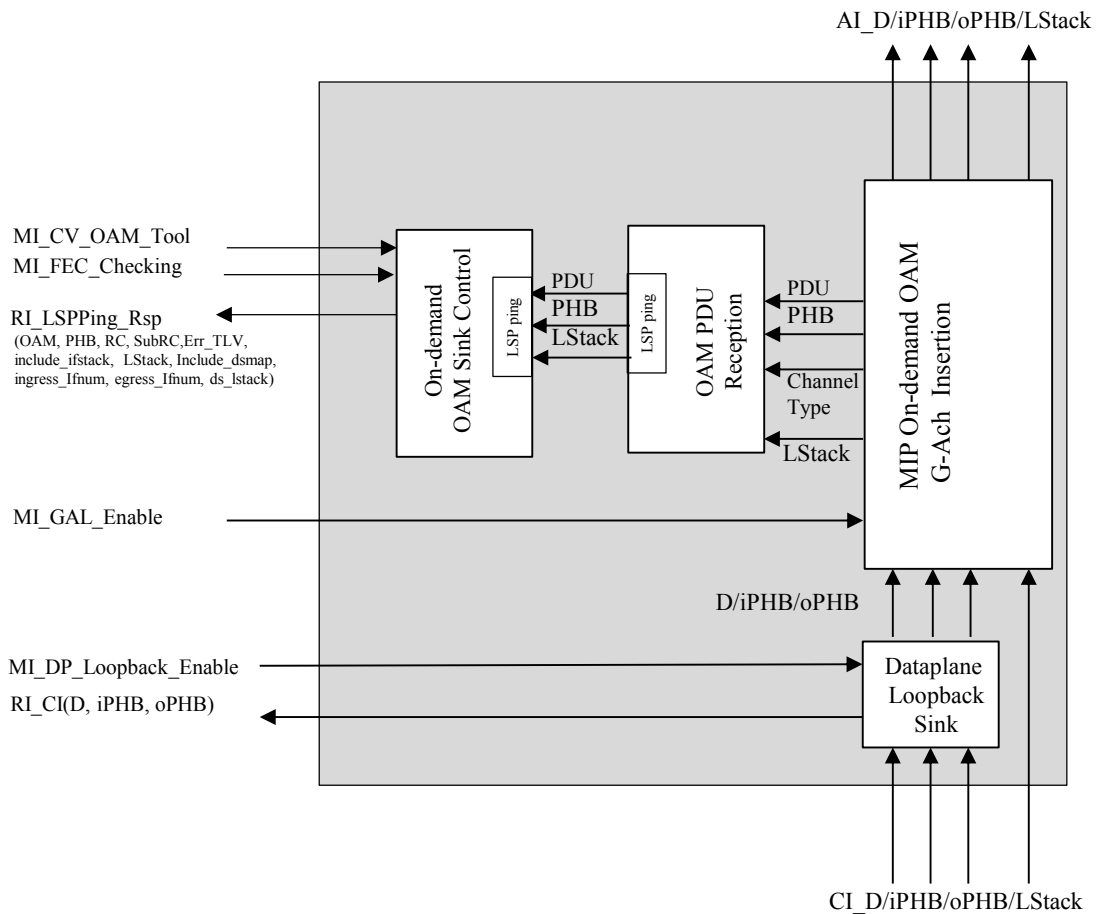


Figure 9-15/G.8121.2/Y.1381.2 - MTDi\_TT\_Sk process

**G-ACh Extraction:** see 8.1.3 in [ITU-T G.8121].

**OAM PDU Reception:** This contains the following sub-processes as described in clause 8.8: On-demand CV Reception; OAM Demux

**On-demand OAM Sink Control:** This contains the following sub-processes as described in clause 8.8: MIP On-demand CV Responder

**Dataplane Loopback Sink:** see clause 8.9.2 in [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 in [ITU-T G.8121].

#### **11 Non-MPLS-TP Server to MPLS-TP adaptation functions**

These atomic functions are defined in clause 11 in [ITU-T G.8121]. They use the OAM protocol specific AIS insert process and LCK generation process as defined in clause 8.6.2 and 8.6.3. For these adaptation functions, in addition to the MI shown in Table 9.5 in [ITU-T G.8121] and Figure 9.11 in [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].

|

## 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 sub-processes and where these (sub-)processes are implemented to the termination functions (MT\_TT, MTDe\_TT, and MTDi\_TT)

**Table I-1/G.8121.2/Y.1381.2 OAM Process and subprocesses**

<b>Process</b>	<b>Sub-processes</b>	<b>MT_TT</b>	<b>MTDe_TT</b>	<b>MTDi_TT</b>
<del>Proactive OAM Source Control</del>	<del>CCCV Generation</del>	<del>Yes</del>		
	<del>Proactive PM Source Control</del>	<del>Yes</del>		
<del>Proactive OAM Sink Control</del>	<del>CCCV Reception</del>	<del>Yes</del>		
	<del>LCK/AIS Reception</del>	<del>Yes</del>		
	<del>Proactive PM Sink Control</del>	<del>Yes</del>		
	<del>PM Responder</del>	<del>Yes</del>		
<del>On-demand OAM Source Control</del>	<del>On-demand CV Control</del>		<del>Yes</del>	
	<del>LI Source Control</del>		<del>Yes</del>	
	<del>On-demand PM Control</del>		<del>Yes</del>	
<del>On-demand OAM Sink Control</del>	<del>MIP On-demand CV Responder</del>			<del>Yes</del>
	<del>MEP On-demand CV Responder</del>		<del>Yes</del>	
	<del>LI Sink Control</del>		<del>Yes</del>	
	<del>PM Responder</del>		<del>Yes</del>	
<del>OAM PDU Generation</del>	<del>On-demand CV Request Generation</del>		<del>Yes</del>	
	<del>On-demand CV Response Generation</del>		<del>Yes</del>	<del>Yes</del>
	<del>OAM Mux</del>	<del>Yes</del>	<del>Yes</del>	<del>Yes</del>
	<del>LI Generation</del>		<del>Yes</del>	
	<del>PM Mux</del>	<del>Yes</del>	<del>Yes</del>	
	<del>PM Generation</del>	<del>Yes</del>	<del>Yes</del>	
<del>OAM PDU Reception</del>	<del>Session Demux</del>	<del>Yes</del>		
	<del>On-demand CV Reception</del>		<del>Yes</del>	<del>Yes</del>
	<del>OAM Demux</del>	<del>Yes</del>	<del>Yes</del>	<del>Yes</del>
	<del>LI Reception</del>		<del>Yes</del>	
	<del>PM Demux</del>	<del>Yes</del>	<del>Yes</del>	
	<del>PM Reception</del>	<del>Yes</del>	<del>Yes</del>	



<u>Process</u>	<u>Sub-processes</u>	<u>MT TT</u>	<u>MTDe TT</u>	<u>MTDi TT</u>
<u>Proactive OAM Source Control</u>	<u>CCCV Generation</u>	<u>Yes</u>		
	<u>Proactive PM Source Control</u>	<u>Yes</u>		
	<u>PM Generation</u>	<u>Yes</u>		
<u>Proactive OAM Sink Control</u>	<u>CCCV Reception</u>	<u>Yes</u>		
	<u>LCK/AIS Reception</u>	<u>Yes</u>		
	<u>Proactive PM Sink Control</u>	<u>Yes</u>		
	<u>PM Responder</u>	<u>Yes</u>		
	<u>PM Reception</u>	<u>Yes</u>		
<u>On-demand OAM Source Control</u>	<u>On-demand CV Control</u>		<u>Yes</u>	
	<u>On-demand CV Request Generation</u>		<u>Yes</u>	
	<u>On-demand CV Response Generation</u>		<u>Yes</u>	<u>Yes</u>
	<u>LI Source Control</u>		<u>Yes</u>	
	<u>On-demand PM Control</u>		<u>Yes</u>	
	<u>PM Generation</u>		<u>Yes</u>	
<u>On-demand OAM Sink Control</u>	<u>MIP On-demand CV Responder</u>			<u>Yes</u>
	<u>MEP On-demand CV Responder</u>		<u>Yes</u>	
	<u>LI Sink Control</u>		<u>Yes</u>	
	<u>PM Responder</u>		<u>Yes</u>	
	<u>PM Reception</u>		<u>Yes</u>	
<u>OAM PDU Generation</u>	<u>OAM Mux</u>		<u>Yes</u>	
	<u>PM Mux</u>	<u>Yes</u>	<u>Yes</u>	
<u>OAM PDU Reception</u>	<u>Session Demux</u>	<u>Yes</u>		
	<u>On-demand CV Reception</u>		<u>Yes</u>	<u>Yes</u>
	<u>OAM Demux</u>	<u>Yes</u>	<u>Yes</u>	<u>Yes</u>
	<u>PM Demux</u>	<u>Yes</u>	<u>Yes</u>	