Question(s): 9/15

Source: Editor G.8121/Y.1381
Title: Draft revised Recommendation ITU-T G.8121/Y.1381 (for Consent)

Abstract
This document contains the latest draft of G.8121 for consent, after a number of drafting sessions in this meeting.
ITU-T Recommendation G.8121/Y.1381

Characteristics of MPLS-TP equipment functional blocks

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

Source

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

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

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

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

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

NOTE

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

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

INTELLECTUAL PROPERTY RIGHTS

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

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

© ITU 2006

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

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Scope</td>
<td>.................................................................</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>References</td>
<td>..................................................................................</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>Definitions</td>
<td>..................................................................................</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>Abbreviations</td>
<td>..................................................................................</td>
<td>11</td>
</tr>
<tr>
<td>5</td>
<td>Conventions</td>
<td>..................................................................................</td>
<td>12</td>
</tr>
<tr>
<td>6</td>
<td>Supervision</td>
<td>..................................................................................</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>6.1</td>
<td>Defects</td>
<td>..................................................................................</td>
</tr>
<tr>
<td></td>
<td>6.2</td>
<td>Consequent actions</td>
<td>........................................................................</td>
</tr>
<tr>
<td></td>
<td>6.3</td>
<td>Defect correlations</td>
<td>........................................................................</td>
</tr>
<tr>
<td></td>
<td>6.4</td>
<td>Performance filters</td>
<td>........................................................................</td>
</tr>
<tr>
<td>7</td>
<td>Information flow across reference points</td>
<td>.................................................................</td>
<td>19</td>
</tr>
<tr>
<td>8</td>
<td>MPLS-TP processes</td>
<td>..................................................................................</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>8.1</td>
<td>G-ACh Process</td>
<td>........................................................................</td>
</tr>
<tr>
<td></td>
<td>8.1.1</td>
<td>Overview</td>
<td>........................................................................</td>
</tr>
<tr>
<td></td>
<td>8.1.2</td>
<td>G-ACh Insertion Process</td>
<td>..................................................................</td>
</tr>
<tr>
<td></td>
<td>8.1.3</td>
<td>G-ACh Reception Process</td>
<td>..................................................................</td>
</tr>
<tr>
<td></td>
<td>8.2</td>
<td>TC/Label processes</td>
<td>........................................................................</td>
</tr>
<tr>
<td></td>
<td>8.2.1</td>
<td>TC/Label source processes</td>
<td>..................................................................</td>
</tr>
<tr>
<td></td>
<td>8.2.2</td>
<td>TC/Label Sink Processes</td>
<td>...................................................................</td>
</tr>
<tr>
<td></td>
<td>8.2.3</td>
<td>Label Stack Copy Process</td>
<td>......................................................................</td>
</tr>
<tr>
<td></td>
<td>8.3</td>
<td>Queuing process</td>
<td>........................................................................</td>
</tr>
<tr>
<td></td>
<td>8.4</td>
<td>MPLS-TP-specific GFP-F processes</td>
<td>...................................................................</td>
</tr>
<tr>
<td></td>
<td>8.4.1</td>
<td>MPLS-TP-specific GFP-F source processes</td>
<td>...............................................................</td>
</tr>
<tr>
<td></td>
<td>8.4.2</td>
<td>MPLS-TP-specific GFP-F sink processes</td>
<td>..................................................................</td>
</tr>
<tr>
<td></td>
<td>8.5</td>
<td>Control Word (CW) processes</td>
<td>......................................................................</td>
</tr>
<tr>
<td></td>
<td>8.5.1</td>
<td>CW source process</td>
<td>........................................................................</td>
</tr>
<tr>
<td></td>
<td>8.5.2</td>
<td>CII Sink Process</td>
<td>........................................................................</td>
</tr>
<tr>
<td></td>
<td>8.6</td>
<td>OAM related Processes used by Server adaptation functions</td>
<td>..........................................................</td>
</tr>
<tr>
<td></td>
<td>8.6.1</td>
<td>Selector Process</td>
<td>........................................................................</td>
</tr>
<tr>
<td></td>
<td>8.6.2</td>
<td>AIS Insert Process</td>
<td>........................................................................</td>
</tr>
<tr>
<td></td>
<td>8.6.3</td>
<td>LCK Generate process</td>
<td>........................................................................</td>
</tr>
<tr>
<td></td>
<td>8.7</td>
<td>OAM related Processes used by adaptation functions</td>
<td>..........................................................</td>
</tr>
<tr>
<td></td>
<td>8.7.1</td>
<td>MCC and SCC Mapping and DeMapping</td>
<td>.....................................................................</td>
</tr>
<tr>
<td></td>
<td>8.7.2</td>
<td>APS Insert and Extract Processes</td>
<td>......................................................................</td>
</tr>
<tr>
<td></td>
<td>8.7.3</td>
<td>CSF Insert and Extract Processes</td>
<td>.......................................................................</td>
</tr>
</tbody>
</table>
8.8 Pro-active and on-demand OAM related Processes..........................36
8.8.1 Pro-active Continuity Check and Connectivity Verification (CC/CV)......38
8.8.2 Remote Defect Indication (RDI)........................................39
8.8.3 On-demand Connectivity Verification (CV)................................39
8.8.4 Pro-active Packet Loss Measurement (LMP)..........................40
8.8.5 On-demand Packet Loss Measurement (LMO)..........................41
8.8.6 Pro-active Packet Delay Measurement (DMP)..........................42
8.8.7 On-demand Packet Delay Measurement (DMO)..........................43
8.8.8 Diagnostic Test (DT).....................................................44
8.8.9 Route Tracing (RT).....................................................44
9  MPLS-TP layer functions ..................................................44
9.1 Connection Functions (MT_C).............................................45
9.1.1 Sub-network connection protection process..........................47
9.2 Termination functions ....................................................48
9.2.1 MPLS-TP Trail Termination function (MT_TT)..........................48
9.3 Adaptation functions ....................................................55
9.3.1 MPLS-TP to MPLS-TP adaptation function (MT/MT_A)..............55
9.4 MT Diagnostic Function..................................................60
9.4.1 MT Diagnostic Trail Termination Functions for MEPs (MTDe).........60
9.4.2 MT Diagnostic Flow Termination Functions for MIPs (MTDi_TT)......65
10 MPLS-TP to Non-MPLS-TP client adaptation functions......................72
10.1 MPLS-TP to ETH adaptation function (MT/ETH_A)......................72
10.2 MPLS-TP to SCC and MCC Adaptation functions-------------------79
10.2.1 MT/SCC_A Adaptation Function......................................79
10.2.2 MT/MCC_A Adaptation Function......................................83
11 Non-MPLS-TP Server to MPLS-TP adaptation functions.......................87
11.1 SDH to MPLS-TP adaptation function (S/MT_A)..........................88
11.1.1 VC-n to MPLS-TP adaptation functions (Sn/MT_A; n=3, 3-X, 4, 4-X) .88
11.1.2 LCAS-capable VC-n to MPLS-TP adaptation functions (Sn-X/L/MT_A; n=3, 4).......................................................93
11.1.3 VC-m to MPLS-TP adaptation functions (Sm/MT_A; m=11, 11-X, 12, 12-X).................................................................98
11.1.4 LCAS-capable VC-m to MPLS-TP adaptation functions (Sm-X/L/MT_A; m=11, 12)..........................................................105
11.2 OTH to MPLS-TP Adaptation function (O/MT_A)........................109
11.2.1 ODUk to MPLS-TP Adaptation functions.............................109
11.2.2 LCAS-capable ODUk to MPLS-TP Adaptation functions (ODUkP-X-L/MT_A; k=1,2,3).......................................................116
11.3 PDH to MPLS-TP adaptation function (P/MT_A)..........................123
11.3.1 Pq to MPLS-TP Adaptation functions (Pq/MT_A; q = 11s, 12s, 31s, 32e) ... 123
11.3.2 LCAS-capable Pq to MPLS-TP Adaptation functions (Pq-X-L/MT_A; q = 11s, 12s, 31s, 32e) ............................................................................ 130
11.4 Ethernet to MPLS-TP adaptation function ...................................................................... 136
11.4.1 ETY to MPLS-TP adaptation function (ETY/MT_A) ................................................Error! Bookmark not defined.
11.4.1.1 ETY to MPLS-TP adaptation function (ETY/MT_A) ........................................Error! Bookmark not defined.
11.4.1.2 ETY to MPLS-TP adaptation function (ETY/MT_A) ........................................Error! Bookmark not defined.
ITU-T Recommendation G.8121/Y.1381

Characteristics of MPLS-TP equipment functional blocks

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 a description of the MPLS-TP functional components using the same methodology that has been used for other transport technologies (e.g. SDH, OTN and Ethernet). This Recommendation is compliant with the transport profile of MPLS as defined by the IETF. In the event of a discrepancy between the MPLS-TP architecture or protocols described in this ITU-T Recommendation and the base definitions provided by the IETF RFCs that are normatively referenced from this ITU-T Recommendation, the IETF RFCs will take precedence. This Recommendation forms part of a suite of Recommendations covering the full functionality of network equipment. These Recommendations are [ITU-T G.806], [ITU-T G.798], [ITU-T G.783], [ITU-T G.705] and [ITU-T G.8021/Y.1341]. This Recommendation also follows the principles defined in [ITU-T G.805].

These Recommendations specify 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.

Figure 1 presents the set of atomic functions associated with the traffic signal transport. The functions are based on the functional architecture as described in [ITU-T G.8110.1]. It is noted that this recommendation only defines Ethernet for the client of MPLS-TP as MT/ETH adaptation function.

The representation of MPLS-TP provided in this ITU-T Recommendation is not intended to modify MPLS as defined by the IETF RFCs normatively referenced by this Recommendation.
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.


Figure 1/G.8121/Y.1381 – MPLS-TP atomic functions
3 Definitions

3.1 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 subnet: [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]
3.1.24 MPLS label stack: [ITU-T G.8101]
3.1.25 label switched path: [ITU-T G.8101]
3.1.26 Bottom of Stack: [ITU-T G.8101]
3.1.27 Time To Live: [ITU-T G.8101]
3.1.28 Label value: [ITU-T G.8101]
3.1.29 Per-Hop Behaviour: [ITU-T G.8101]
3.1.30 Associated Channel Header: [ITU-T G.8101]
3.1.31 Generic Associated Channel: [ITU-T G.8101]
3.1.32 G-ACh Label: [ITU-T G.8101]
3.1.33 traffic class: [ITU-T G.8101]
3.1.34 Explicitly TC-encoded-PSC LSP: [ITU-T G.8101]
3.1.35 label inferred PHB scheduling class LSP: [ITU-T G.8101]

3.2 Terms defined elsewhere:

None
Abbreviations

This Recommendation uses the following abbreviations:

- **ACH**: Associated Channel Header
- **AI**: Adapted Information
- **AIS**: Alarm indication signal
- **AP**: Access Point
- **APS**: Automatic protection switching
- **CC**: Continuity Check
- **CI**: Characteristic Information
- **CII**: Common Interworking Indicator
- **CoS**: Class of Service
- **CP**: Connection Point
- **CV**: Connectivity Verification
- **CW**: Control Word
- **DM**: Delay Measurement
- **DP**: Drop Precedence
- **DT**: Diagnostic Test
- **ETH**: Ethernet MAC layer network
- **ETY**: Ethernet PHY layer network
- **E-LSP**: Explicitly TC-encoded-PSC LSP
- **FP**: Flow Point
- **FTP**: Flow termination point
- **G-ACh**: Generic Associated Channel
- **GAL**: G-ACh Label
- **GFP**: Generic Framing Procedure
- **L-LSP**: Label-Only-Inferred PSC LSP
- **LCAS**: Link Capacity Adjustment Scheme
- **LCK**: Locked
- **LM**: Loss Measurement
- **LSP**: Label Switched Path
- **MCC**: Maintenance Communication Channel
- **MPLS**: Multi-Protocol Label Switching
- **MPLS-TP**: MPLS Transport Profile
- **OAM**: Operation, Administration and Maintenance
- **on-demand**: Per Hop Behaviour
- **p**: Per Hop Behaviour
- **PSC**: PHB Scheduling Class
- **RDI**: Remote Detect Indication
- **RT**: Route Trace
- **S**: Bottom of Stack
- **SCC**: Signalling Communication Channel
- **TCP**: Termination Connection Point
- **TFP**: Termination Flow Point
- **TH**: Throughput
- **TTL**: Time-To-Live
- **TTSI**: Trail Termination Source Identifier
- **ODU**: Optical Channel Data Unit
- **ODUk**: Optical Channel Data Unit – order k
- **ODUk-Xv**: Virtual concatenated Optical Channel Data Unit – order k
OPU  Optical Payload Unit
OPUk  Optical Payload Unit of level k
OPUk-Xv  Virtually concatenated Optical Payload Unit of level k
OTH  Optical Transport Hierarchy
P11s  1544 kbit/s PDH path layer with synchronous 125 µs frame structure according to ITU-T G.704
P12s  2048 kbit/s PDH path layer with synchronous 125 µs frame structure according to ITU-T G.704
P31s  34368 kbit/s PDH path layer with synchronous 125 µs frame structure according to ITU-T G.832
P32e  44 736 kbit/s PDH path layer with frame structure according to ITU-T G.704
PM  Performance Monitoring
PSI  Payload Structure Indication
PT  Payload Type
RES  Reserved overhead
TC  Traffic Class
TLV  Type Length Value
vcPT  virtual concatenation Payload Type
VcPLM  Virtual concatenation Payload Mismatch

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

6  Supervision
The generic supervision functions are defined in clause 6 of [ITU-T G.806]. Specific supervision functions for the MPLS-TP network are defined in this clause.

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.
In the following:
Valid means a received value is equal to the value configured via the MI input interface(s).
Invalid means a received value is not equal to the value configured via the MI input interface(s).
The events defined for this Recommendation are summarized in Table 6-1 as a quick overview. Events, other than the protection switching events, are generated by processes in the MT_TT_Sk function as defined in clause 9.2. These processes define the exact conditions for these events; Table 6-1 only provides a quick overview.

Further details of the specific events relating to each protocol can be found in G.8121.1 and G.8121.2.

<table>
<thead>
<tr>
<th>Event</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event</td>
<td>Meaning</td>
</tr>
</tbody>
</table>

Table 6-1/ G.8121/Y.1381 – Overview of Events
unexpMEG  | Reception of a CV packet with an invalid MEG value.
unexpMEP  | Reception of a CV packet with an invalid MEP value, but with a valid MEG value.
unexpCCPeriod | Reception of a CC packet with an invalid Periodicity value.
unexpCVPeriod | Reception of a CV packet with an invalid Periodicity value, but with valid MEG and MEP values.
unexpCoS-CC | Reception of a CC packet with an invalid TC value.
unexpCoS-CV | Reception of a CV packet with an invalid TC value, but with valid MEG and MEP values.
expCC  | Reception of a CC packet.
expCV  | Reception of a CV packet with valid MEG and MEP values.
RDI=x  | Reception of a CC packet for the peer MEP with the RDI information indicate to x; where x=0 (remote defect clear) and x=1 (remote defect set).
LCK  | Reception of a LCK packet. (Note 1)
AIS  | Reception of an AIS packet.
BS  | Bad Second, a second in which the Lost Frame Ratio exceeds the Bad Second Threshold (BS_THR).
CSF-LOS  | Reception of a CSF packet that indicates Client Loss of Signal.
CSF-FDI  | Reception of a CSF packet that indicates Client Forward Defect Indication.
CSF-RDI  | Reception of a CSF packet that indicates Client Reverse Defect Indication.

Note 1: IETF uses this term LCK as LKR and LKI in [IETF RFC 6371]

The occurrence or absence of these events may detect or clear a defect. An overview of the conditions is given in Table 6-2. The notation “#event=x (K*period)” is used to indicate the occurrence of x events within the period as specified between the brackets.

Table 6-2 gives a quick overview of the types of defects for MPLS-TP layer and the raising and clearing conditions for these defects as described in [IETF RFC 6371].

<table>
<thead>
<tr>
<th>Defect</th>
<th>RFDIe Condition</th>
<th>Clearing Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>dLOC</td>
<td>#expCC==0 (K*CC_Period)</td>
<td>expCC</td>
</tr>
<tr>
<td>dUNC-CC</td>
<td>unexpCoS-CC</td>
<td>#unexpCoS-CC==0 (K*CC_Period)</td>
</tr>
<tr>
<td>dUNC-CV</td>
<td>unexpCoS-CV</td>
<td>#unexpCoS-CV==0 (K*CV_Period)</td>
</tr>
</tbody>
</table>

Table 6-2/ G.8121/Y.1381 – Overview of Detection and Clearing Conditions
<table>
<thead>
<tr>
<th>dMMG</th>
<th>unexpMEG</th>
<th>#unexpMEG==0 (K* CV_Period)</th>
</tr>
</thead>
<tbody>
<tr>
<td>dUNM</td>
<td>unexpMEP</td>
<td>#unexpMEP==0 (K*CV_Period)</td>
</tr>
<tr>
<td>dUNP-CC</td>
<td>unexpCCPeriod</td>
<td>#unexpCCPeriod==0 (K*CC_Period)</td>
</tr>
<tr>
<td>dUNP-CV</td>
<td>unexpCVPeriod</td>
<td>#unexpCVPeriod==0 (K*CV_Period)</td>
</tr>
<tr>
<td>dRDI</td>
<td>RDI==1</td>
<td>RDI==0</td>
</tr>
<tr>
<td>dAIS</td>
<td>AIS</td>
<td>#AIS==0 (K*AIS_Period)</td>
</tr>
<tr>
<td>dLCK</td>
<td>LCK</td>
<td>#LCK==0 (K*LCK_Period)</td>
</tr>
<tr>
<td>dCSF-LOS</td>
<td>CSF-LOS</td>
<td>#CSF-LOS == 0 (K*CSF_Period or CSF-DCI)</td>
</tr>
<tr>
<td>dCSF-FDI</td>
<td>CSF-FDI</td>
<td>#CSF-FDI == 0 (K*CSF_Period or CSF-DCI)</td>
</tr>
<tr>
<td>dCSF-RDI</td>
<td>CSF-RDI</td>
<td>#CSF-RDI == 0 (K*CSF_Period or CSF-DCI)</td>
</tr>
<tr>
<td>dDEG</td>
<td>#BS==DEGM (DEGM*1second)</td>
<td>#BS==0 (M*1second)</td>
</tr>
</tbody>
</table>
6.1.2 Continuity Supervision

Figure 6-1/G.8121/Y.1381 – dLOC detection and clearance process

6.1.2.1 Loss Of Continuity defect (dLOC)

The Loss of Connectivity Verification defect is calculated at the MT layer. It monitors the presence of continuity in MT trails.

Its detection and clearance are defined in Figure 6-1. The ‘period’ in Figure 6-1 is set to K*MI_CC_Period, where MI_CC_Period corresponds to the configured CC Period and K is such that 3.25≤K≤3.5.

6.1.3 Connectivity Supervision
Figure 6-2/G.8021/Y.1341 – Defect detection and clearance process for dMMG, dUNM, dUNP, dUNPr, dAIS, dLCK and dCSF

Figure 6-2 shows a generic state diagram that is used to detect and clear the dMMG, dUNM, dUNP, dUNPr, dAIS, dLCK and dCSF defects. In this diagram <Defect> needs to be replaced with the specific defect and <Event> with the specific event related to this defect. Furthermore in Figure 6-2 3.25 ≤ K ≤ 3.5.

Figure 6-2 shows that the Timer is set based on the last received period value, unless an earlier OAM packet triggering <Event> (and therefore the detection of <Defect>) carried a longer period. As a consequence clearing certain defects may take more time than necessary.

6.1.3.1 Mismerge defect (dMMG)

The Mismerge defect detect is calculated at the MT layer. It monitors the connectivity in a Maintenance Entity Group.
Its detection and clearance are defined in Figure 6-2. The <Defect> in Figure 6-2 is dMMG. The <Event> in Figure 6-2 is the unexpectedMEG event and the Period is the Period carried in the CV packet that triggered the event, unless an earlier CV packet triggering an unexpectedMEG event carried a greater period.

6.1.3.2 Unexpected MEP defect (dUNM)

The Unexpected MEP defect is calculated at the MT layer. It monitors the connectivity in a Maintenance Entity Group. Its detection and clearance are defined in Figure 6-2. The <Defect> in Figure 6-2 is dUNM. The <Event> in Figure 6-2 is the unexpectedMEP event and the Period is the Period carried in the CV packet that triggered the event, unless an earlier CV packet triggering an unexpectedMEP event carried a greater period.

6.1.3.3 Degraded Signal defect (dDEG)

Figure 6-3/G.8121 – dDEG detection and clearance process
The Degraded Signal defect is calculated at the MT layer. It monitors the connectivity of a MT Trail.
Its detection and clearance are defined Figure 6-3.
Every second the statemachine receives the 1 second counters for near end received and transmitted frames and determines whether the second was a Bad Second. The defect is detected if there are MI_LM_DEGM consecutive Bad Seconds and cleared if there are MI_LM_M consecutive Good Seconds.
In order to declare a Bad Second the number of transmitted frames must exceed a threshold (TF_MIN). If this is true then a Bad Second is declared if either the Frame Loss is negative (i.e. there are more frames received than transmitted) or the Frame Loss Ratio (lost frames/transmitted frames) is greater than MI_LM_DEGTHR.

6.1.4 Protocol Supervision

6.1.4.1 Unexpected Periodicity defect (dUNP-CC/dUNP-CV)
The Unexpected Periodicity defect is calculated at the MT layer. It detects the configuration of different periodicities at different MEPs belonging to the same MEG.
Its detection and clearance are defined Figure 6-2. The <Defect> in Figure 6-2 is dUNP-CC or dUNP-CV. The <Event> in Figure 6-2 is the unexpectedCCPeriod event or the unexpectedCVPeriod event and the Period is the Period carried in the CC or CV packet that triggered the event, unless an earlier CC or CV packet triggering an unexpectedCCPeriod event or the unexpectedCVPeriod event carried a greater period.

6.1.4.2 Unexpected CoS defect (dUNC-CC/dUNC-CV)
The Unexpected CoS defect is detected at the MT layer. It detects the configuration error of different CoS for CC or CV at different MEPs belonging to the same MEG.
Its detection and clearance are defined Figure 6-2.
The <Defect> in Figure 6-2 is dUNC-CC or dUNC-CV. The <Event> in Figure 6-2 is the unexpectedCos-CC event or the unexpectedCos-CV event and the Period is the Period associated with the CC or CV packet that triggered the event, unless an earlier CC or CV packet triggering an unexpectedCos-CC event or an unexpectedCos-CV event associated with a greater period.

6.1.4.3 Protection protocol supervision
For further study

6.1.5 Maintenance Signal Supervision

6.1.5.1 Remote Defect Indicator defect (dRDI)
The Remote Defect Indicator defect is detected at the MT layer. It monitors the presence of the RDI maintenance signal.
dRDI is detected on receipt of the RDI=1 event and cleared on receipt of the RDI=0 event.
6.1.5.2 Alarm Indicate Signal defect (dAIS)
The Alarm Indicate Signal defect is detected at the MT layer. It monitors the presence of the AIS maintenance signal.
Its detection and clearance are defined Figure 6-2. The <Defect> in Figure 6-2 is dAIS. The <Event> in Figure 6-2 is the AIS event and the Period is the Period associated with the AIS packet unless an earlier AIS packet associated with a greater period.

6.1.5.3 Locked Defect (dLCK)
The Locked defect is detected at the MT layer. It monitors the presence of the Locked maintenance signal.
Its detection and clearance are defined Figure 6-2. The <Defect> in Figure 6-2 is dLCK. The <Event> in Figure 6-2 is the LCK event and the Period is the Period associated with the LCK packet unless an earlier LCK packet associated with a greater period.

6.1.5.4 Client Signal Fail defect (dCSF)
The CSF (CSF-LOS, CSF-FDI, and CSF-RDI) defect is detected at the MT layer. It monitors the presence of the CSF maintenance signal.
Its detection and clearance conditions are defined in Figure 6-2. The <Defect> in Figure 6-2 is dCSF-LOS, dCSF-FDI, or dCSF-RDI. The <Event> in Figure 6-2 is the CSF event (as generated by the CSF reception process in clause 8.7.6) and the Period is the Period associated with the CSF packet unless an earlier CSF packet associated with a greater period
The <Clear_event> in Figure 6-2 is the CSF event which indicates Detect Clearance Indication (DCI).

6.2 Consequent actions
For generic consequent actions, see [ITU-T G.806]. For the specific consequent actions applicable to MPLS-TP, refer the specific atomic functions.

6.3 Defect correlations
For the defect correlations, see the specific atomic functions.

6.4 Performance filters
F/s.

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
This clause defines the specific processes for the MPLS-TP network. Generic processes are defined in clause 8 [ITU-TG.806].
8.1 G-ACh Process

8.1.1 Overview

In order to ensure proper operational control, MPLS-TP network elements exchange OAM packets that strictly follow the same path as user traffic packets; that is, OAM packets are subject to the exact same forwarding schemes (e.g. fate sharing) as the user traffic packets. These OAM packets can be distinguished from the user traffic packets by using the G-ACh and GAL constructs.

The G-ACh is a generic associated control channel mechanism for Sections, LSPs and PWs, over which OAM and other control messages can be exchanged. The GAL is a label based exception mechanism to alert LERs/LSRs of the presence of an Associated Channel Header (ACH) after the bottom of the stack.

The format of GAL and ACH is described in [IETF RFC 5586].

8.1.2 G-ACh Insertion Process

Figure 8-1 describes G-ACh Insertion process.

The G-ACh Insertion process encapsulates OAM packets and multiplexes them with the data packets. The data packets are passed through unchanged, while the OAM packets are encapsulated as follows:

A G-ACh header is prepended to the OAM PDU, with the Channel Type. If MI_GAL_Enable is true, the process then further prepends a G-ACh Label (GAL) as described in [IETF RFC 5586]. If the TTL signal is not specified, the TTL field in the GAL is set to 255; otherwise it is set to the value in the TTL signal. Note: certain OAM packets can be addressed to a MIP and thus need to be inserted with a specific TTL to ensure that the TTL expires at the target MIP. OAM packets addressed to a MEP have the TTL set to 255.
Note: MI_GAL_Enable must be set to true on LSPs and to false on PWs. Setting it to true for PWs is FFS.

8.1.3 G-Ach Reception Process

Figure 8-2 describes G-ACh Reception process.

The G-ACh Traffic Unit will be extracted if it includes GAL and ACH in incoming Data when MI_GAL_Enable is set.
8.2 TC/Label processes

8.2.1 TC/Label source processes

Figure 8-3/G.8121/Y.1381 – TC/Label source processes

Figure 3 shows the TC/Label source processes. These processes are performed on a frame-per-frame basis.

Client Specific Processes: The function supports M (M ≤ 2^N – 16, with N = 20 for MPLS label) client specific processes (CSP#1 to CSP#M), each connected to a single MPLS-TP connection point. CSP#m (1 ≤ m ≤ M) is active when Label[m] has a value in the range 16 to 2^N – 1.

**TC Insertion process**: Insert the TC field encoding the PHB information according to the following rules:

- If LSPType[m] = L-LSP, the DP information is encoded into the TC field according to [RFC 3270] and CoS[m].
- If LSPType[m] = E-LSP, the PHB information is encoded into the TC field according to the 1:1 mapping configured in the PHB2TCMapping[m].

Note - E-LSP and L-LSP are referred to [ITU-T G.8110.1]

The PHB information to map into the TC field is selected according to the following rules:

- If QoSEncodingMode[m] = A, the iPHB information is mapped into the TC field.
- If QoSEncodingMode[m] = B, the oPHB information is mapped into the TC field.

**Label Insertion process**: Insert the 20-bit MPLS Label field with the value provided via Label[m].

**Interleave process**: Interleave the MPLS-TP traffic units from the client specific processes into a single stream.
8.2.2 TC/Label Sink Processes

Figure 8-4/G.8121/Y.1381 – TC/Label sink processes

Figure 8-4 shows the TC/Label sink processes. These processes are performed on a frame-per-frame basis.

**De-Interleave process:** De-interleave the MPLS-TP traffic units and forwards each of its Client Specific Process #m based on the value in the Label field of the traffic unit. Relation between CSP and MPLS label value is provided by Label[1..M].

Traffic units received with a label value identifying a non-active CSP are dropped.

**Client Specific Processes:** The function supports M (M ≤ 2^N – 16, with N = 20 for MPLS label) client specific processes (CSP#1 to CSP#M), each connected to a single MPLS-TP connection point. CSP#m (1 ≤ m ≤ M) is active when Label[m] has a value in the range 16 to 2^N – 1.

**Label and TC Extraction process:** Extract the MPLS label and the TC fields from the traffic unit.

**TTL Decrement Process:** Decrement the TTL. If the MPLS-TP CP is not a TCP and the decremented TTL is less than or equal to zero, the traffic unit is dropped silently.

**NOTE** – MIPs and MEPs compound functions are connected to the Server/MT_A (or MT/MT_A) functions via an MPLS-TP TCP.

**PHB Generation process:** Processes the TC field.

The iPHB signal is generated according to the following rules:

- If LSPTypem = L-LSP, the CoS information is equal to the CoS[m] while the DP information is decoded from the TC field according to RFC 3270 and the CoS[m].
• If LSPType[m] = E-LSP, the PHB information is decoded from the TC field according to the 1:1 mapping configured in the TC2PHBMapping[m].

Note - E-LSP and L-LSP are refered to [ITU-T G.8110.1]

The CI_oPHB is generated according to the following rule:
• If QoSDecodingMode = A, the oPHB is equal to the generated iPHB.
• If QoSDecodingMode = B, the oPHB is equal to the received PHB.

8.2.3 Label Stack Copy Process

![Label Stack Copy Process](image)

Figure 8-5/G.8121/Y.1381 – Label Stack Copy Process

Figure 8-5 shows Label Stack Copy Process. It passes through the CI_D unchanged and copies from the CI_D traffic unit the complete label stack.

8.3 Queuing process

The queuing process buffers received MPLS packets for output according to the CI_oPHB. Figure 8-6 shows Queuing process. The details of the queuing process implementation are out of the scope of this Recommendation.

The queuing process is also responsible for dropping frames if their rate at the MT_CI is higher than the <Srv>_AI_D can accommodate. Performance monitor counters are for further study.

![Queuing process](image)

Figure 8-6/G.8121/Y.1381 – Queuing process
8.4 MPLS-TP-specific GFP-F processes

8.4.1 MPLS-TP-specific GFP-F source processes

Figure 8-7/G.8121/Y.1381 – MPLS-TP-specific GFP-F source process

Figure 8-7 shows the MPLS-TP-specific GFP-F source processes. These processes are performed on a frame-per-frame basis.

**Mapping of MPLS-TP data**: The MPLS-TP packet is inserted into the client payload information field of the GFP frame as defined in 7.6 of [ITU-T G.7041]. One MPLS-TP packet results in one GFP frame.

**Mapping of SCC data**: The SCC frame is inserted into the client payload information field of the GFP frame as defined in clause 7/G.7041/Y.1303. One SCC packet results in one GFP frame.

**Frame Count**: It counts the number of frames (n_FramesOK) and of octets (n_OctetsOK) that passes through.

**pFCS generation**: See 8.5.4.1.1 in [ITU-T G.806]. GFP FCS is always enabled (FCSEnable=true).

**Generate PTI and UPI, Interleave**: The PTI field of the GFP type header is set fixed to "000". The UPI field of the GFP type header is set to:
- the MPLS UPI (as defined in Table 6-3 of [ITU-T G.7041]), for frames coming from the Map MPLS-TP data process;
- the SCC UPI according to SCC Type for frames coming from the Map SCC data process.

The frames are then interleaved to form a single stream.

**NOTE 2** – GFP Client Management frames are not defined for MPLS-TP over GFP-F mapping.
8.4.2 MPLS-TP-specific GFP-F sink processes

Figure 8-8/G.8121/Y.1381 – MPLS-TP-specific GFP-F sink process

Figure 8-8 shows the MPLS-TP-specific GFP-F sink processes. These processes are performed on a frame-per-frame basis.

**Check PTI and UPI, Deinterleave:** GFP frames with an accepted PTI (AcPTI, see 8.5.1.1/G.806) of "000" are client data frames. All GFP frames with an accepted PTI (AcPTI, see 8.5.1.1/G.806) value other than "000" shall be discarded.

The UPI of client data frames is checked to generate dUPM as follows:

- a "valid-UPI frame" is a frame with a UPI that equals either the MPLS UPI (as defined in Table 6-3 of [ITU-T G.7041]) or the SCC UPI according to SCCType. All other frames are "invalid-UPI frames".
- dUPM is raised as soon as one "invalid-UPI frame" is received.
- dUPM is cleared if no "invalid-UPI frames" have been received for the last Tclear seconds.

Tclear is ffs. If dUPM is active, the latest received invalid UPI is available at LastInvalidUPI. If dUPM is not active, LastInvalidUPI is "n/a".

The UPI of client data frames is further used to deinterleave the frames:

- "valid-UPI frames" with UPI equalling the MPLS UPI (as defined in Table 6-3 of [ITU-T G.7041]) are sent towards the “Demap MPLS-TP data” process.
- "valid-UPI frames" with UPI equalling the SCC UPI according to SCCType (as defined in Table 6-3 of [ITU-T G.7041]) are sent towards the “Demap SCC data” process.
- "invalid-UPI frames" are discarded.

**GFP-F frame length:** It checks whether the length of the GFP-F frame is allowed. Frames longer than GFP_Length bytes are dropped and counted (n_FramesTooLong).
NOTE 1 – GFP.Length is for further study.

pFCS Supervision: See 8.5.4.1.2/G.806. The discarding of errored frames is always enabled (FCSdiscard=true). If the accepted PFI is 0, the frame is dropped and counted (n_FDis_PFI).

Frame Count: It counts the number of frames (n_FramesOK) and of octets (n_OctetsOK) that passes through.

Demapping of SCC data: The SCC packet is extracted from the client payload information field of the GFP frame as defined in clause 7 of [ITU-T G.7041]. One GFP frame results in one SCC frame.

Demapping of unicast MPLS-TP data: The MPLS-TP upacket is extracted from the client payload information field of the GFP frame as defined in 7.6 of [ITU-T G.7041]. One GFP frame results in one MPLS-TP packet.

8.5 Control Word (CW) processes

This function performs the Control Word (CW) processing as described in [IETF RFC 4448]. The CW is known as the common interworking indicators (CII) in [ITU-T Y.1415].

8.5.1 CW source process

```
CWEEnable
SF
```

Figure 8-9/G.8121/Y.1381 – CW source process

Figure 8-9 shows CW source process. This function should generate and insert the CWs as described in [IETF RFC 4448], in case the indication CIIEnable is true. Otherwise no insertion should be performed. If the indication SQUse is false, the sequence number field should be set at all zeroes.

8.5.2 CW Sink Process

```
CWEEnable
SF
```

Figure 8-10/G.8121/Y.1381 – CW sink process

Figure 8-9 shows CW sink process. This function should process the Common Interworking Indicator as described in [ITU-T Y.1415], in case the indication CIIEnable is true. In this case, if the indication SQUse is true, the sequence number field should be processed and out-of-sequence packets dropped (no reordering is performed by this process).

In addition, the SF indication is passed through unaltered to the next process.
8.6 OAM related Processes used by Server adaptation functions

8.6.1 Selector Process

Figure 8-11/G.8121/Y.1381 – Selector process

Figure 8-11 shows the Selector Process Symbol. The Selector process selects the valid signal from the input of the normal MT_CI signal or the MT_CI LCK signal (as generated by the LCK Generation process in 8.6.3). The normal signal is blocked if MI_Admin_State is LOCKED. The behaviour is defined in Figure 8-12.

Figure 8-12/G.8121/Y.1381 – Selector Behaviour
8.6.2 AIS Insert Process

Figure 8-13/G.8121/Y.1381 – AIS Insert process
Figure 8-13 shows the AIS Insert Process Symbol. The generated AIS traffic units are inserted in the incoming stream, i.e., the output stream contains the incoming traffic units and the generated AIS traffic units that contains G-ACh traffic unit as described in Clause 8.1 which includes GAL or not depending on MI_GAL_Enable. The value of the MT_CI_CoS signal associated with the generated AIS traffic units is defined by the MI_AIS_CoS input parameter. As described in [IETF RFC 6371], AIS packets are transmitted with the "minimum loss probability PHB".

8.6.3 LCK Generate process

Figure 8-14/G.8121/Y.1381 – LCK Generation process
Figure 8-14 shows the LCK\(^1\) Insert Process Symbol. The LCK Generation Process generates MT_CI traffic units where the MT_CI_D signal contains the LCK signal. Figure 8-15 defines the behaviour of the LCK Generation Process.

\(^1\) IETF uses the term LKR for this function
The LCK Generation Process continuously generates LCK Traffic Units that contains G-ACh traffic unit as described in Clause 8.1 which includes GAL or not depending on MI_GAL_Enable. The period between consecutive LCK traffic units is determined by the MI_LCK_Period parameter. The value of the MT_CI_PHB signal associated with the generated LCK traffic units is defined by the MI_LCK_CoS input parameter.

8.7 OAM related Processes used by adaptation functions

8.7.1 MCC and SCC Mapping and DeMapping

As defined in [ITU-T G.7712], an embedded communication channel (ECC) provides a logical operations channel between NEs that can be utilized by various applications. An MCC is an ECC dedicated for management plane communications. An SCC is an ECC dedicated for control plane communications.

The MCC mapping and de-mapping processes are provided to support the MT to MCC adaptation function for accessing to the MCC. The SCC mapping and de-mapping processes are provided to support the MT to SCC adaptation function for accessing to the SCC. The mapping and de-mapping processes for MCC is very similar to that of SCC. In the following description of this sub-clause and sub-clause 8.7.2, the term ECC will be used, which applies to both MCC and SCC.
8.7.1.1 ECC Mapping

The ECC mapping process is associated with the MT/MCC_A_So and MT/SCC_A_So functions, which are described in Clauses 10.2.2.1 and 10.2.1.1 respectively.

This process shall map the incoming ECC packet into G-ACh encapsulated ECC traffic unit (i.e., an MT_AI_D traffic units carrying an ECC packet).

The ECC traffic units generated by this process are encapsulated into the G_ACh, as defined in [IETF RFC5718], using or not the GAL depending on the MI_GAL_Enable configuration parameters. The value of the MT_AI_PHB associated with the generated ECC traffic units is defined by the MI_ECC_CoS input parameter.

8.7.1.2 ECC DeMapping

The ECC DeMapping process is associated with the MT/MCC_A_Sk and MT/MCC_A_Sk functions, which are described in Clauses 10.2.2.2 and 10.2.1.2 respectively.

This process shall extract the ECC packet from the G-ACh encapsulated ECC traffic unit (i.e., MT_AI_D traffic units carrying ECC packets).

The criteria for selecting ECC traffic units are based on the values of the fields within the MT_AI_D signal:

- GAL included to the MT_AI_D if GAL usage is enabled via MI_GAL_Enable
- The Channel type of G-ACh indicates an MCC packet (in MT/MCC_A_Sk) or an SCC packet (in MT/SCC_A_Sk), as defined in [IETF RFC 5718]

8.7.2 APS Insert and Extract Processes

Figure 8-16 shows a protocol-neutral abstract model of the different processes inside MEPs and MIPs that are involved in APS function. [to]

![Figure 8-16/ – Overview of the processes involved with APS function](image)

---

2 IETF uses the term PCS for this function in [IETF RFC 6378]
APS Insert and Extract processes are located in MT/MT_Adaptation function. CI_APS signal carries APS specific information which is for further study. APS traffic units are inserted into and extracted from the stream of MT_CI_D traffic units.

8.7.2.1 APS Insert Process

Figure 8-17/G.8121/Y.1381 – APS Insert process
Figure 8-17 shows the APS Insert process and Figure 8-18 defines the behaviour. The resulting APS traffic unit is inserted into the stream of incoming traffic units, i.e., the outgoing stream consists of the incoming traffic units and the inserted APS traffic units. The APS traffic unit contain G-ACh traffic unit as described in Clause 8.1 which includes GAL or not depending on MI_GAL_Enable.

Figure 8-18/G.8121/Y.1381 – APS Insert Behaviour
8.7.2.2 APS Extract Process

The APS Extract process extracts MT_CI_APS signals from the incoming stream of MT_CI traffic units.

The MT_CI_APS is the APS Specific Information contained in the received Traffic Unit. All other traffic units will be transparently forwarded.

The criteria for filtering are based on the values of the fields within the MT_CI_D signal:

- GAL included to the MT_CI_D if GAL usage is enabled via MI_GAL_Enable
- OAM type that is defined in Channel type of G-ACh indicates APS

This is defined in Figure 8-20. The function APS(D) extracts the APS specific information from the received Traffic Unit.

Figure 8-19/G.8121/Y.1381 – APS Extract process

Figure 8-20/G.8121/Y.1381 – APS Extract Behaviour
8.7.3 CSF Insert and Extract Processes

Figure 8-21 shows the different processes inside MEPs and MIPs that are involved in the CSF Protocol.

Figure 8-21/G.8121/Y.1381 – Overview of Processes involved with CSF Protocol

The MPLS-TP Client Signal Fail function (MT-CSF) is used by a MEP to propagate to a peer MEP the detection of a failure or defect event in an MPLS-TP client signal when the client itself does not support appropriate fault or defect detection or propagation mechanisms, such as MT-CC or MT-AIS. The MT-CSF messages propagate in the direction from MPLS-TP MEP function detecting the failure or defect event to the MPLS-TP sink-adaptation function associated with the peer MEP.

MT-CSF generation is located at MT/Client_A_So to insert CSF traffic unit and ProActive OAM Insertion is located at MT_TT.

8.7.3.1 CSF Insert Process

The CSF Insert Process is located at MT/Client_A_So as a part of CSF generation. Figure 8-22 shows the CSF Insert Process Symbol and Figure 8-23 defines the behaviour. If the aCSF signal is true, the CSF Insert process periodically generates MT_CI traffic units where the MT_CI_D signal contains the CSF signal until the aCSF signal is false. The generated CSF traffic units are inserted in the incoming stream, i.e., the output stream contains the incoming traffic units and the generated CSF traffic unit that contains G-ACh traffic unit as described in Clause 8.1 which includes GAL.

IETF uses the term “Client Failure Indication” for this function in [IETF RFC 6371]
not depending on MI_GAL_Enable. The period between consecutive CSF traffic units is determined by the MI_CSF_Period parameter.

The specific CSF traffic unit is for further study.

**Figure 8-23/G.8121/Y.1381 – CSF Insert behaviour**

Note: generation of CSF(0) and CSF(1) events as well as determination of CSF type is FFS.

### 8.7.3.2 CSF Extract Process

**Figure 8-24/G.8121/Y.1381 – CSF Extract process**

The CSF Extract process is located at MT/Client_A_sk and extracts MT-CSF from MI_AI_D. Figure 8-24 shows the CSF Extract Process Symbol.
The encoding of the MT_CI_D signal for CSF frames is for further study.
The criteria for filtering are based on the values of the fields within the MT_CI_D signal:

- GAL included to the MT_CI_D if GAL usage is enabled via MI_GAL_Enable
- OAM type that is defined in Channel type of G-ACh indicates CSF

This behaviour is defined in Figure 8-25. The function CSF(D) extracts the CSF specific information from the received Traffic Unit.

Note: G-ACh process is done at G-ACh process as defined in clause 8.1. The CSF traffic unit in MT_CI_D is forwarded to the CSF extract process.

---

**Figure 8-25/G.8121/Y.1381 – CSF Extract Behaviour**

---

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

As described in [IETF RFC 6371], OAM functions are categorized as pro-active and on-demand and these OAM functionalities provide the different interfaces.

OAM functions can be also categorised as single-ended and dual-ended. Single-ended functions are those in which an initiating MEP sends OAM PDUs to a target MEP, which processes it and sends a response OAM PDU back to the initiating MEP. The results of the function are available only on the initiating MEP. Dual-ended functions are those in which an initiating MEP sends OAM PDUs to a target MEP, which processes it and does not send a response. The results of the function are available only on the target MEP. Dual-ended functions are typically deployed in pairs, one in each direction.
Figure 8-26 shows an OAM protocol-neutral abstract model of the different processes inside MEPs and MIPs that are involved in performing single-ended pro-active or on-demand OAM functions. In the case of dual-ended functions, the model is equivalent to the top half of the diagram only, and the results are reported by the OAM sink control process on the Target MEP.

**NOTE** – The MT_CI signals at the input of the G-ACh/GAL Insertion process and at output of the G-ACh/GAL Extraction process are not input/output signals of the Initiation/Target MEPs but signals which are internal to these MEPs.

The proactive OAM Source and Sink Control processes perform all the OAM control procedures (e.g., they maintain the necessary state machine) that are required for a specific OAM protocol within the MT_TT_So and MT_TT_Sk atomic functions respectively.

The OAM Source Control process within the initiating MEP requests the OAM PDU Generation process to generate OAM Request PDUs toward the target MEP on the basis of the local state machine and the relevant Management Information (MI). This supports both single-ended and dual-ended pro-active or on-demand OAM transactions.

In the case of a dual-ended OAM transaction, the OAM Sink Control process within the target MEP reports the dual-ended OAM results on the basis of the OAM Request PDUs received by the OAM PDU Reception process.

In the case of single-ended OAM transactions, the following actions are taken:
- The OAM Sink Control process within the target MEP provides the local OAM Source Control process the relevant Remote Information (RI_) to generate a reply to the OAM Request PDU received by the local PDU Reception process.
- The OAM Source Control process within the target MEP requests the OAM PDU Generation process to generate OAM Reply PDUs toward the initiating MEP based on the information it receives from the local OAM Sink Control process via the relevant Remote Information (RI_).
- The OAM Sink Control process within the initiating MEP reports the unidirectional or bidirectional OAM results based on the OAM Reply PDUs received by the local OAM PDU Reception process.

The OAM PDU Generation process builds, when instructed by its control process, the required OAM PDU and passes it to the G-ACh/GAL process, defined in clause 8.1, for insertion within the MPLS-TP CI traffic flow. It also passes the following information elements that are required by the G-ACh/GAL process: the CoS associated to the OAM packet (on the basis of the instruction received by the control process); the ACH Channel Type that identifies the OAM PDU and the TTL value which it is either the TTL distance to a MIP (for OAM PDUs targeted to a MIP and properly requested by the control process) or the default value as configured via MI_TTLValue.

Similarly, the OAM PDU generation process consists of a number of OAM protocol-specific PDU generation sub-processes and a sub-process that multiplexes all the PDUs generated by these OAM protocol-specific PDU generation sub-processes into a single stream of OAM PDUs, which is sent to the G-ACh/GAL Insertion process along with the appropriate ACH Channel Type.

The OAM PDU Reception process receives an OAM PDU, together with the ACH Channel Type value identifying the PDU type and the associated CoS, from the G-ACh/GAL process and passes the relevant information to its control process.

Similarly, the OAM PDU Reception process consists of a number of OAM protocol-specific PDU reception sub-processes and a sub-process that demultiplexes OAM PDUs received from the G-ACh/GAL Extraction process towards these OAM protocol-specific PDU reception sub-processes, based on the ACH Channel Type.

The relevant Management Information (MI_) and Remote Information (RI_) used by these processes depend on the OAM function to be performed and it is defined in the next sub-clauses.

The detailed specification, including further process decomposition and the interface between them, of these pro-active and on-demand OAM control processes and of the OAM PDU Generation and Reception processes are OAM protocol-specific and therefore outside the scope of this Recommendation.

The Proactive OAM Source Control process, Proactive OAM Sink Control, On-demand OAM Source Control and On-Demand OAM Sink Control processes each consist of a number of protocol-specific control sub-processes, relating to different types of OAM PDU.

8.8.1 Pro-active Continuity Check and Connectivity Verification (CC/CV)

[CD07: asks for single-ended mechanism (BFD CC and CV packets in each direction are not independent) while The current content in section 8.8.1 describes only the dual-ended
mechanism and is therefore incompatible with G.8121.2. As described in [IETF RFC 6371], both CC and CV OAM functions are based on the (proactive) generation of OAM packets by the source MEP that are processed by the peer sink MEP(s).

The source MEP generates CC/CV OAM packets if it is enabled via Management Information. As described in [IETF RFC 6371], the CC/CV OAM packets are generated at a regular rate which is configured by the operator via the MI_CC_Period. These packets are also transmitted using PHB which is configured via MI_CC_CoS and MI_CC_DP (and that is typically the “minimum loss probability PHB”).

In order to perform Connectivity Verification, the generated CC/CV packets also includes a globally unique Source MEP identifier: the transmitted value is configured via protocol-specific management information on the source MEP while the expected value is configured via different protocol-specific management information on the sink MEP.

The sink MEP always processes received CC/CV OAM packets and detects the following CC/CV defects, as defined in clause 6.1:

- dLOC
- dUNC-CC
- dUNC-CV
- dMMG
- dUNM
- dUNP-CC
- dUNP-CV

CC/CV OAM packets pass transparently through MIPs as described in [IETF RFC 6371]. The EMF can retrieve from the sink MEP the latest CC/CV OAM packet which caused a defect condition via the MI_GetSvdCC command: the CC/CV OAM packet is returned to the EMF via the MI_SvdCC.

8.8.2 Remote Defect Indication (RDI)

As in [IETF RFC6371], in case of co-routed and associated bidirectional transport paths, RDI is associated with proactive CC/CV, and the RDI indicator can be piggy-backed onto the CC/CV packet.

RDI information is carried in the CC/CV packets based upon the R1_CC/CV_RDI input. It is extracted in the CC/CV Reception Process.

In case of unidirectional transport paths, the RDI related OAM process is for further study.

8.8.3 On-demand Connectivity Verification (CV)

As described in [IETF RFC 6371], on-demand CV OAM functions are based on the (on-demand) generation of OAM packets by the source MEP, that are processed and responded to by the peer sink MIP(s) or MEP(s).
The source MEP generates on-demand CV OAM packets when requested via protocol-specific MI signals. The results of the on-demand CV operation are returned by the source MEP using additional protocol-specific management information.

8.8.4 Pro-active Packet Loss Measurement (LMP)

As described in [IETF RFC 6371], pro-active LM is performed by periodically sending LM OAM packets from the initiating MEP to the target MEP and by receiving LM OAM packets from the target MEP on a co-routed bidirectional connection. Each MEP performs measurements of its transmitted and received user data packets (TxFCI and RxFCI). These measurements are then correlated in real time with the target MEP in the ME to derive the impact of packet loss on a number of performance metrics for the ME in the MEG.

For single-ended:

- The initiating MEP generates pro-active LM OAM Request packets if MI_LMp_Enable is true. These packets are generated at the rate configured via the MI_LMp_Period and, as described in [IETF RFC 6371], with the PHB configured via MI_LMp_CoS that yields the lowest drop precedence within the measured PHB Scheduling Class, in order to maximize reliability of measurement within the traffic class. The local value of transmitted user data packets (TxFCI) is inserted within the LM OAM packet by the OAM PDU Generation process.

- The target MEP replies to the LM OAM packets if it is enabled. The local value of the received user data packets (RxFCI) at the time the pro-active LM OAM Request packet has been received is passed by OAM PDU Reception process to the pro-active OAM sink control process, then to the pro-active OAM source control process (via RI_OAM_Info) and inserted by the OAM PDU Generation process within the transmitted pro-active LM OAM Reply: the actual behaviour on how this information is passed is OAM protocol specific and therefore outside the scope of this Recommendation. The OAM PDU Generation process also inserts the local value of the transmitted user data packets (TxFCI) in the reverse direction within the transmitted pro-active LM OAM Reply.

- The initiating MEP processes the received pro-active LM OAM Reply packet, together with the local value of the received used data packets (RxFCI) in the reverse direction at the time this OAM packet is received, as passed by the OAM PDU Reception process, and generates LM results.

- Depending on the LMP OAM tool that it is used, the LM results can be either calculated by the pro-active OAM sink control process or by the pro-active OAM source control process. In the latter case, the the pro-active OAM sink control process passes the required information in the received LM OAM Reply to the the pro-active OAM source control process via the RI_LMP and receives the LM results back via the RI_LMP_Result. In both cases, the pro-active OAM sink control process passes the LM Results to the relevant performance monitoring processes within the MT_TT_SK atomic function for reporting to the EMF.

For dual-ended:

- the initiating MEP generates pro-active LM OAM Request packets if it is enabled to do so via management information. These packets are generated at the rate configured via the MI_LMp_Period and, as described in [IETF RFC 6371], with the PHB configured via MI_LMp_CoS that yields the lowest drop precedence within the measured PHB Scheduling Class, in order to maximize reliability of measurement within the traffic class. The local value of transmitted user data packets (TxFCI) is inserted within the LM OAM packet by the OAM PDU Generation process.
The target MEP receives the LM OAM packets if it is enabled to do so via management information. The local value of the received user data packets (RxFCI) at the time the pro-active LM OAM Request packet has been received is passed by OAM PDU Reception process to the pro-active OAM sink control process and generates LM results. Pro-active LM OAM packets pass transparently through MIPs as described in [IETF RFC6371].

8.8.5 On-demand Packet Loss Measurement (LMo)

As described in [IETF RFC 6371], on-demand LM is performed by the command that sends LM OAM packets from the initiating MEP to the target MEP and by receiving LM OAM packets from the target MEP on a co-routed bidirectional connection. The initiating MEP performs measurements of its transmitted and received user data packets (TxFCI and RxFCI). These measurements are then correlated in real time with values received from the target MEP in the ME to derive the impact of packet loss for the ME in the MEG.

For single-ended measurement,

- The initiating MEP generates on-demand LM OAM Request packets when enabled via management information. These packets are generated with the PHB configured via management information that yields the lowest drop precedence within the measured PHB Scheduling Class, in order to maximize reliability of measurement within the traffic class. The local value of transmitted user data packets (TxFCl) is inserted within the LM OAM packet by the OAM PDU Generation process.

- The target MEP replies to the LM OAM packets if enabled via management information. The local value of the received user data packets (RxFCI) at the time the on-demand LM OAM Request packet has been received is passed by OAM PDU Reception process to the on-demand OAM sink control process, then to the on-demand OAM source control process (via RI_OAM_Info) and inserted by the OAM PDU Generation process within the transmitted on-demand LM OAM Reply; the actual behaviour on how this information is passed is OAM protocol specific and therefore outside the scope of this Recommendation. The OAM PDU Generation process also inserts the local value of the transmitted user data packets (TxFCl) in the reverse direction within the transmitted on-demand LM OAM Reply.

- The initiating MEP processes the received on-demand LM OAM Reply packet, together with the local value of the received used data packets (RxFCI) in the reverse direction at the time this OAM packet is received, as passed by the OAM PDU Reception process, and generates LM results.

- Depending on the LMo OAM tool that it is used, the LM results can be either calculated by the on-demand OAM sink control process or by the on-demand OAM source control process. In both cases, the LM results are reported to EMF by the MTDe_TT_So by the protocol-specific management information.

For dual-ended measurement,

- The initiating MEP generates on-demand LM OAM Request packets when enabled via management information. These packets are generated at the rate configured via the MI_LMo_Period and, as described in [IETF RFC6371], with the PHB configured via MI_LMo_CoS that yields the lowest drop precedence within the measured PHB Scheduling Class, in order to maximize reliability of measurement within the traffic class. The local value
of transmitted user data packets (TxFCI) is inserted within the LM OAM packet by the OAM PDU Generation process.

- The target MEP receives the LM OAM packets when enabled via management information. The local value of the received user data packets (RxFCI) at the time the on-demand LM OAM Request packet has been received is passed by OAM PDU Reception process to the on-demand OAM sink control process and generates LM results.

On demand LM OAM packets pass transparently through MIPs as described in [IETF RFC6371].

8.8.6 Pro-active Packet Delay Measurement (DMp)

As described in [IETF RFC6371], pro-active DM is performed by periodically sending DM OAM packets from the initiating MEP to the target MEP and by receiving DM OAM packets from the target MEP on a co-routed bidirectional connection. Each MEP records its transmitted and received Timestamps. The timestamps from the initiating and target MEPs are then correlated to derive a number of performance metrics relating to delay for the ME in the MEG.

For single-ended:

- The initiating MEP generates pro-active DM OAM Request packets if MI_DMp_Enable is true. These packets are generated at the rate configured via the MI_DMp_Period and, as described in [IETF RFC6371], with the PHB configured via MI_DMp_CoS that yields the lowest drop precedence within the measured PHB Scheduling Class, in order to maximize reliability of measurement within the traffic class. The local value of transmitted timestamp (TxTimeStampf) is inserted within the DM OAM packet by the OAM PDU Generation process.

- The target MEP replies to the DM OAM packets if enabled by management information. The local value of the received timestamp (RxTimeStampl) at the time the pro-active DM OAM Request packet has been received is passed by OAM PDU Reception process to the pro-active OAM sink control process, then to the pro-active OAM source control process (via RI_OAM_Info) and inserted by the OAM PDU Generation process within the transmitted pro-active DM OAM Reply: the actual behaviour on how this information is passed is OAM protocol specific and therefore outside the scope of this Recommendation. The OAM PDU Generation process also inserts the local value of the transmitted timestamp (TxTimeStampf) in the reverse direction within the transmitted pro-active DM OAM Reply.

- The initiating MEP processes the received pro-active DM OAM Reply packet, together with the local value of the received timestamp (RxTimeStampl) in the reverse direction at the time this OAM packet is received, as passed by the OAM PDU Reception process, and generates DM results.

- Depending on the DMp OAM tool that it is used, the DM results can be either calculated by the pro-active OAM sink control process or by the pro-active OAM source control process. In the latter case, the the pro-active OAM sink control process passes the required information in the received DM OAM Reply to the the pro-active OAM source control process via the RI_DMRp and receives the DM results back via the RI_DMp_Result. In both cases, the pro-active OAM sink control process passes the DM Results to the relevant performance monitoring processes within the MT_TT_Sk atomic function for reporting to the EMF.

For dual-ended:

- the initiating MEP generates pro-active DM OAM Request packets if enabled by management information. These packets are generated at the rate configured via the
MI_DMp_Period and, as described in [IETF RFC6371], with the PHB configured via MI_DMp_CoS that yields the lowest drop precedence within the measured PHB Scheduling Class, in order to maximize reliability of measurement within the traffic class. The local value of transmitted Timestamp (TxTimeStampl) is inserted within the DM OAM packet by the OAM PDU Generation process.

- The target MEP receives the DM OAM packets if enabled by management information. The local value of the received Timestamp (RxTimeStampl) at the time the pro-active DM OAM Request packet has been received is passed by OAM PDU Reception process to the pro-active OAM sink control process which generates DM results.

Pro-active DM OAM packets pass transparently through MIPs as described in [IETF RFC6371].

8.8.7 On-demand Packet Delay Measurement (DMo)

As described in [IETF RFC6371], on-demand DM is performed by the command that sends DM OAM packets from the initiating MEP to the target MEP and by receiving DM OAM packets from the target MEP on a co-routed bidirectional connection. Each MEP records its transmitted and received Timestamps. The timestamps from the initiating and target MEPs are then correlated to derive a number of performance metrics relating to delay for the ME in the MEG.

For single-ended:

- The initiating MEP generates on-demand PM OAM Request packets if MI_DMo_Enable is true. These packets are generated with the PHB configured via MI_DMo_CoS that yields the lowest drop precedence within the measured PHB Scheduling Class, in order to maximize reliability of measurement within the traffic class. The local value of transmitted timestamp (TxTimeStampl) is inserted within the DM OAM packet by the OAM PDU Generation process. The target MEP replies to the PM OAM packets if the MI_PMo_Enable is true. The local value of the received timestamp (RxTimeStampl) at the time the on-demand PM OAM Request packet has been received is passed by OAM PDU Reception process to the on-demand OAM sink control process, then to the on-demand OAM source control process (via RI_OAM_Info) and inserted by the OAM PDU Generation process within the transmitted on-demand DM OAM Reply: the actual behaviour on how this information is passed is OAM protocol specific and therefore outside the scope of this Recommendation.

- The initiating MEP processes the received on-demand DM OAM Reply packet, together with the local value of the received timestamp (RxTimeStampl) in the reverse direction at the time this OAM packet is received, as passed by the OAM PDU Reception process, and generates DM results.

- Depending on the DMo OAM tool that it is used, the DM results can be either calculated by the on-demand OAM sink control process or by the on-demand OAM source control process. In both cases, the DM results are reported to EMF by the MTDe_TT_So by the protocol-specific management information.

dual-ended:

- The initiating MEP generates on-demand DM OAM Request packets if MI_DMo_Enable is true. These packets are generated at the rate configured via the MI_DMo_Period and, as described in [IETF RFC6371], with the PHB configured via MI_DMo_CoS that yields the lowest drop precedence within the measured PHB Scheduling Class, in order to maximize reliability of measurement within the traffic class. The local value of transmitted Timestamp (TxTimeStampl) is inserted within the DM OAM packet by the OAM PDU Generation process.
The target MEP receives DM OAM packets if the MI_DMo_Enable is true. The local value of the received Timestamp (RxTimestamp) at the time the on-demand DM OAM Request packet has been received is passed by OAM PDU Reception process to the on-demand OAM sink control process and generates DM results.

DM OAM packets pass transparently through MIPs as described in [IETF RFC6371].

### 8.8.8 Throughput Test (TH)

For dual-ended Throughput Test:

- As described in [IETF RFC 6371], out of service on demand throughput estimation can be performed by sending OAM test packets at increasing rate (up to the theoretical maximum), computing the percentage of OAM test packets received and reporting the rate at which OAM test packets begin to drop. In general, this rate is dependent on the OAM test packet size.

- The source MEP starts generating test packets when requested via protocol-specific management information and continues generating these packets at the configured period until requested to stop; at this time the number of sent packets is reported via protocol-specific management information.

- The sink MEP, when enabled via protocol-specific management information, starts processing the received OAM test packets until the test is terminated; at this time, the calculated test results are reported.

For single-ended Throughput Test:

- For further study

### 8.8.9 Route Tracing (RT)

For further study

### 9 MPLS-TP layer functions

Figure 9-0 illustrates the MPLS-TP layer network and server and client layer adaptation functions. The information crossing the MPLS-TP connection point (MT_CP) is referred to as the MPLS-TP characteristic information (MT_CI). The information crossing the MPLS-TP access point (MT_AP) is referred to as the MPLS-TP adapted information (MT_AI).

The MPLS-TP layer network provides embedded hierarchy via the label stacking mechanism. This is represented in the model by MPLS-TP Tunnel sublayers, which contain MT_TT and MT/MT_A functions. The figure shows a generic example for the connection of the MPLS-TP Tunnel functions. It is not required to connect them via a MT_C function; they can be directly inserted without a connection function. It is noted that this recommendation only defines Ethernet for the client of MPLS-TP as MT/ETH adaptation function.

This mechanism (MPLS-TP tunnel sublayers) is also used when sublayer (tandem connection) monitoring is required.
9.1 Connection Functions (MT_C)

MT_C is the function that assigns MPLS packets at its input ports to MPLS-TP packets at its output ports.

The MT_C connection process is a unidirectional function as illustrated in Figure 9-1. The signal formats at the input and output ports of the function are similar, differing only in the logical sequence of the MPLS-TP packets. As the process does not affect the nature of the characteristic information of the signal, the reference point on either side of the MT_C function is the same, as illustrated in Figure 9-1.

Incoming MPLS-TP packets at the MT_CP are assigned to available outgoing MPLS-TP capacity at the MT_CP.
• Symbol:

![Diagram of MT_C symbol]

Figure 9-1/G.8121/Y.1381 – MT_C symbol

• Interfaces:

<table>
<thead>
<tr>
<th>Input(s)</th>
<th>Output(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Per MT_CP, n × for the function:</td>
<td>per MT_CP, m × per function:</td>
</tr>
<tr>
<td>MT_CI_D</td>
<td>MT_CI_D</td>
</tr>
<tr>
<td>MT_CI_iPHB</td>
<td>MT_CI_iPHB</td>
</tr>
<tr>
<td>MT_CI_oPHB</td>
<td>MT_CI_oPHB</td>
</tr>
<tr>
<td>MT_CI_SSF</td>
<td>MT_CI_SSF</td>
</tr>
<tr>
<td>MT_AI_TSF</td>
<td></td>
</tr>
<tr>
<td>per input and output connection point:</td>
<td></td>
</tr>
<tr>
<td>for further study</td>
<td></td>
</tr>
<tr>
<td>per matrix connection:</td>
<td></td>
</tr>
<tr>
<td>MT_C_MI_ConnectionType</td>
<td></td>
</tr>
<tr>
<td>MT_C_MI_Return_CP_ID</td>
<td></td>
</tr>
<tr>
<td>MT_C_MI_ConnectionPortIds</td>
<td></td>
</tr>
<tr>
<td>per SNC protection group:</td>
<td></td>
</tr>
<tr>
<td>for further study</td>
<td></td>
</tr>
</tbody>
</table>

• Processes:

In the MT_C function MPLS-TP Characteristic Information is routed between input (termination) connection points ((T)CPs) and output (T)CPs by means of matrix connections. (T)CPs may be allocated within a protection group.

NOTE – Neither the number of input/output signals to the connection function, nor the connectivity is specified in this Recommendation. That is a property of individual network elements.
Routing process:
This process passes all the traffic units received from a specific input to the corresponding output according to the matrix connection between the specified input and output.

Each (matrix) connection in the MT_C function shall be characterized by the:

| Type of connection (MI_ConnectionType): | unprotected, protected |
| Traffic direction (MI_Return_CP_ID): | Unidirectional if NULL, otherwise it identifies the CP of the return connection (Note) |
| Input and output connection points (MI_ConnectionPortIDs): | set of connection point identifiers |

NOTE – Bidirectional LSPs are supported by associating two unidirectional LSPs in the data plane, as per ITU-T Rec. G.8110.1/Y.1370.1.

Protection Switching process:
For further study.

• Performance Monitoring:
None.

• Defects:
None.

• Consequent actions:
If an output of this function is not connected to one of its inputs, the connection function shall send no traffic units and SSF = false to the output.

• Defect correlations:
None.

9.1.1 Sub-network connection protection process
For further study.
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 9.3.

![Figure 9-3/G.8121/Y.1381 – MT_TT](image)

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 for pro-active monitoring to the MT_AI signal at its MT_AP.

The information flow and processing of the MT_TT_So function is defined with reference to Figure 9.4.

- **Symbol:**

![Figure 9-4/G.8121/Y.1381 – MT_TT_So function](image)

- **Interfaces:**

Table 9-2/G.8121/Y.1381 – MT_TT_So inputs and outputs
<table>
<thead>
<tr>
<th>Input(s)</th>
<th>Output(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MT_AP:</strong></td>
<td><strong>MT_CP:</strong></td>
</tr>
<tr>
<td>MT_AI_D</td>
<td>MT_CI_D</td>
</tr>
<tr>
<td>MT_AI_PHB</td>
<td>MT_CI_oPHB</td>
</tr>
<tr>
<td>MT_AI_LStack</td>
<td>MT_CI_iPHB</td>
</tr>
<tr>
<td><strong>MT_RP:</strong></td>
<td><strong>MT_RP:</strong></td>
</tr>
<tr>
<td>MT_RI_CC_RDI</td>
<td></td>
</tr>
<tr>
<td>MT_RI_CC_Blk</td>
<td></td>
</tr>
<tr>
<td>MT_RI_OAM_Info(D,CoS,DP)</td>
<td></td>
</tr>
<tr>
<td><strong>MT_TT_So_MP:</strong></td>
<td></td>
</tr>
<tr>
<td>MT TT So MI GAL_Enable</td>
<td></td>
</tr>
<tr>
<td>MT TT So MI TTLVALUE</td>
<td></td>
</tr>
<tr>
<td>MT TT So MI MEG ID</td>
<td></td>
</tr>
<tr>
<td>MT TT So MI MEP_ID</td>
<td></td>
</tr>
<tr>
<td>MT TT So MI CC OAM_Tool</td>
<td></td>
</tr>
<tr>
<td>MT TT So MI RD</td>
<td>OAM Tool</td>
</tr>
<tr>
<td>MT TT So MI CC_Enable</td>
<td></td>
</tr>
<tr>
<td>MT TT So MI CC CoS</td>
<td></td>
</tr>
<tr>
<td>MT TT So MI CC_Period</td>
<td></td>
</tr>
<tr>
<td>MT TT So MI CV_Period</td>
<td></td>
</tr>
<tr>
<td>MT TT So MI LMp OAM_Tool</td>
<td></td>
</tr>
<tr>
<td>MT TT So MI LMp Enable[1...MLMp]</td>
<td></td>
</tr>
<tr>
<td>MT TT So MI LMp Period[1...MLMp]</td>
<td></td>
</tr>
<tr>
<td>MT TT So MI LMp CoS[1...MLMp]</td>
<td></td>
</tr>
<tr>
<td>MT TT So MI DMp OAM_Tool</td>
<td></td>
</tr>
<tr>
<td>MT TT So MI DMp Enable[1...MDMp]</td>
<td></td>
</tr>
<tr>
<td>MT TT So MI DMp Period[1...MDMp]</td>
<td></td>
</tr>
<tr>
<td>MT TT So MI DMp Test ID[1...MDMp]</td>
<td></td>
</tr>
<tr>
<td>MT TT So MI DMp CoS[1...MDMp]</td>
<td></td>
</tr>
<tr>
<td>MT TT So MI DMp Length[1...MDMp]</td>
<td></td>
</tr>
<tr>
<td>MT TT So MI 1DMp OAM_Tool</td>
<td></td>
</tr>
<tr>
<td>MT TT So MI 1DMp Enable[1...MDMp]</td>
<td></td>
</tr>
<tr>
<td>MT TT So MI 1DMp Period[1...MDMp]</td>
<td></td>
</tr>
<tr>
<td>MT TT So MI 1DMp Test ID[1...MDMp]</td>
<td></td>
</tr>
<tr>
<td>MT TT So MI 1DMp CoS[1...MDMp]</td>
<td></td>
</tr>
<tr>
<td>MT TT So MI 1DMp Length[1...MDMp]</td>
<td></td>
</tr>
<tr>
<td>MT TT So MI SLp OAM Tool</td>
<td></td>
</tr>
<tr>
<td>MT TT So MI SLp Enable[1...MLSp]</td>
<td></td>
</tr>
<tr>
<td>MT TT So MI SLp Period[1...MLSp]</td>
<td></td>
</tr>
<tr>
<td>MT TT So MI SLp Test ID[1...MLSp]</td>
<td></td>
</tr>
<tr>
<td>MT TT So MI SLp Length[1...MLSp]</td>
<td></td>
</tr>
<tr>
<td>MT TT So MI SLp CoS[1...MLSp]</td>
<td></td>
</tr>
</tbody>
</table>
• Processes:
The processes associated with the MT_TT_So function are as depicted in Figure 9-5.
Notes:

1. The interface between Pro-active OAM Control and OAM PDU Generation is protocol specific.
2. Note that the parameters & values in the MT_TT_So_MI_XX_OAM_Tool are outside the scope of this recommendation.

**PHB**: The AI_PHB signal is assigned to both the CI_iPHB and CI_oPHB signals at the MT_TCP reference point.

**Insert TTL**: The Time To Live value is inserted in the outer shim header's TTL field within the MT_AI traffic unit.

**Block process**: When RI_CC_Blk is raised, the Block process will discard all AI_D traffic units it receives. If RI_CC_Blk is cleared, the received AI_D traffic units will be passed to the output port.

**Counter process**: For Further Study.


**Pro-active OAM Source Control Process**: See 8.8/G.8121

**OAM PDU Generation Process**: See 8.8/G.8121

- **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 for pro-active monitoring - 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 9-6.
• Symbol:

```
MT_AP
MT_TCP
MT_RP

MT_TT_Sk_MP
```

Figure 9-6/G.8121/Y.1381 – MT_TT_Sk function

• Interfaces:

Table 9-3/G.8121/Y.1381 – MT_TT_Sk inputs and outputs

<table>
<thead>
<tr>
<th>Input(s)</th>
<th>Output(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MT_TCP:</strong></td>
<td><strong>MT_AP:</strong></td>
</tr>
<tr>
<td>MT_CI_D</td>
<td>MT_AI_D</td>
</tr>
<tr>
<td>MT_CI_iPHB</td>
<td>MT_AI_PHB</td>
</tr>
<tr>
<td>MT_CI_oPHB</td>
<td>MT_AI_TSF</td>
</tr>
<tr>
<td>MT_CI_SSF</td>
<td>MT_AL_TSD</td>
</tr>
<tr>
<td>MT_CI_Lstack</td>
<td>MT_AL_AIS</td>
</tr>
</tbody>
</table>

| **MT_RP:**                                     | **MT_AP:**                                     |
| **MT_TT_Sk_MP:**                              | **MT_RP:**                                     |
| MT_TT_Sk_MI_MEG_ID                            | MT_RI_CC_RDI                                   |
| MT_TT_Sk_MI_PeerMep_ID                        | MT_RI_CC_Blk                                   |
| MT_TT_Sk_MI_CC_OAM_Tool                      | MT_RI_OAM_Info(D,P,DP)                         |
| MT_TT_Sk_MI_RDIOAM_Tool                      | **MT_TT_Sk_MP:**                              |
| MT_TT_Sk_MI_CC_Enable                        | MT_TT_Sk_MI_SvdcC                              |
| MT_TT_Sk_MI_CC_Period                        | MT_TT_Sk_MI_CSSF                               |
| MT_TT_Sk_MI_CC_CoS                          | MT_TT_Sk_MI_CC_Enable                         |
| MT_TT_Sk_MI_CC_Period                        | MT_TT_Sk_MI_CoS                                |
| MT_TT_Sk_MI_CC_Get_SvdcC                    | MT_TT_Sk_MI_CC_Period                         |
| **MT_TT_Sk_MI_LMp_OAM_Tool**                 | **MT_TT_Sk_MI_LMp_OAM_Tool**                  |
| MT_TT_Sk_MI_LMp_Enable[1...M_Lmp]            | MT_TT_Sk_MI_LMp_Enable[1...M_Lmp]             |
| MT_TT_Sk_MI_LMp_CoS[1...M_Lmp]               | MT_TT_Sk_MI_LMp_CoS[1...M_Lmp]                |
| MT_TT_Sk_MI_LM_DEG                          | MT_TT_Sk_MI_LM_DEG                            |
| MT_TT_Sk_MI_LM_M                            | MT_TT_Sk_MI_LM_M                              |
| MT_TT_Sk_MI_LM_DEGTHR                        | MT_TT_Sk_MI_LM_DEGTHR                         |
| MT_TT_Sk_MI_LM_TFMN                        | MT_TT_Sk_MI_LM_TFMN                           |

| **MT_TT_Sk_MI_DMp_OAM_Tool**                  | **MT_TT_Sk_MI_DMp_OAM_Tool**                  |
### Input(s)
- MT_TT_Sk_MI_DMp_Enable[1...M_{DMp}]
- MT_TT_Sk_MI_DMp_CoS[1...M_{DMp}]
- MT_TT_Sk_MI_1DMp_OAM_Tool
- MT_TT_Sk_MI_1DMp_Enable[1...M_{1DMp}]
- MT_TT_Sk_MI_1DMp_Test_ID[1...M_{1DMp}]
- MT_TT_Sk_MI_SLp_OAM_Tool
- MT_TT_Sk_MI_SLp_Enable[1...M_{SLp}]
- MT_TT_Sk_MI_SLp_CoS[1...M_{SLp}]
- MT_TT_Sk_MI_AIS_OAM_Tool
- MT_TT_Sk_MI_LCK_OAM_Tool
- MT_TT_Sk_MI_1second
- MT_TT_Sk_MI_SSF_Reported
- MT_TT_Sk_MI_RDI_Reported
- MT_TT_Sk_MI_pF_LF[1...P]
- MT_TT_Sk_MI_pF_TF[1...P]
- MT_TT_Sk_MI_pF_DS
- MT_TT_Sk_MI_pN_DS
- MT_TT_Sk_MI_pB_FD[1...P]
- MT_TT_Sk_MI_pB_FDV[1...P]
- MT_TT_Sk_MI_pN_FD[1...P]
- MT_TT_Sk_MI_pN_FDV[1...P]
- MT_TT_Sk_MI_pF_FD[1...P]
- MT_TT_Sk_MI_pF_FDV[1...P]

### Output(s)

**Processes:**
The processes associated with the MT_TT_Sk function are as depicted in Figure 9-7.
Figure 9-7/G.8121/Y.1381 – MT_TT_Sk process diagram

Note - The parameters & values in the MT_TT_Sk_MI_XX_OAM_Tool are outside the scope of this recommendation.

**PHB:** The CI_oPHB signal is assigned to the AI_PHB signal at the reference point MT_AP.

Note that the CI_ipHB signal is not used by any of the processes in the function.

**Extract TTL:** The Time To Live value is extracted from the outer shim header's TTL field within the MT_CI traffic unit.

**Block:** When the aBlock consequent action is asserted, this process drops all traffic units arriving at its input.

**Counter process:** For further study.

**G-Ach/GAL Extraction Process:** See 8.1.

**Pro-active OAM Sink Control Process:** See 8.8

**OAM PDU Reception Process:** See 8.8

**Defect Generation:** This process raises and clears the defects as defined in clause 6.1.
• Defects:
For further study

• Consequent actions:
For further study

• Defect correlations:
For further study

• Performance monitoring:
For further study

9.3 Adaptation functions

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

9.3.1.1 MPLS-TP to MPLS-TP adaptation source function (MT/MT_A_So)
This function maps client MT_CI traffic units into server MT_AI traffic units.

Figure 9-8/G.8121/Y.1381 – MT/MT_A_So function
• Interfaces:

<table>
<thead>
<tr>
<th>Each MT_CP:</th>
<th>MT_AP:</th>
</tr>
</thead>
<tbody>
<tr>
<td>MT_CI_Data</td>
<td>MT_AI_Data</td>
</tr>
<tr>
<td>MT_CI_iPHB</td>
<td>MT_AI_PHB</td>
</tr>
<tr>
<td>MT_CI_oPHB</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MT/MT_A_So_MI:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MT/MT_A_So_MI_Admin_State</td>
<td></td>
</tr>
<tr>
<td>MT/MT_A_So_MI_Label[1…M]</td>
<td></td>
</tr>
<tr>
<td>MT/MT_A_So_MI_LSPType[1…M]</td>
<td></td>
</tr>
<tr>
<td>MT/MT_A_So_MI_CoS[1…M]</td>
<td></td>
</tr>
<tr>
<td>MT/MT_A_So_MI_PHB2TCMapping[1…M]</td>
<td></td>
</tr>
<tr>
<td>MT/MT_A_So_MI_QoSEncodingMode[1…M]</td>
<td></td>
</tr>
<tr>
<td>MT/MT_A_So_MI_LCK_Period[1…M]</td>
<td></td>
</tr>
<tr>
<td>MT/MT_A_So_MI_LCK_CoS[1…M]</td>
<td></td>
</tr>
<tr>
<td>MT/MT_A_So_MI_GAL_Enable[1…M]</td>
<td></td>
</tr>
</tbody>
</table>

• Processes:

A process diagram of this function is shown in Figure 9-9.

![Figure 9-9/G.8121/Y.1381 – MT/MT_A_So process diagram](image-url)

Table 9-4/G.8121/Y.1381 – MT/MT_A_So interfaces
– **LCK Generate process:**
See 8.6.3. Each CP has its LCK Generate process.

– **Selector process:**
See 8.6.1. The normal CI is blocked if Admin_State = LOCKED.

– **TC/Label processes:**
See 8.2.1.

– **S Field Insertion:**
A 1-bit S Field set to 0 (not bottom of label stack) is inserted to indicate the client is MPLS.

  • **Defects:**
  None.

  • **Consequent actions:**
  None.

  • **Defect correlations:**
  None.

  • **Performance monitoring:**
  None.

9.3.1.2 **MPLS-TP to MPLS-TP adaptation sink function (MT/MT_A_Sk)**
This function retrieves client MT_CI traffic units from server MT AI traffic units.

![Figure 9-10/G.8121/Y.1381 – MT/MT_A_Sk function](image)

**Interfaces:**

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>MT_AP:</td>
<td>Each MT_CP:</td>
</tr>
<tr>
<td>MT_AI.Data</td>
<td>MT_CI.Data</td>
</tr>
<tr>
<td>MT_AI_PHB</td>
<td>MT_CI_iPHB</td>
</tr>
<tr>
<td>MT_AI_TSF</td>
<td>MT_CI_oPHB</td>
</tr>
<tr>
<td>-----------------</td>
<td>----------------</td>
</tr>
<tr>
<td>MT_AI_AIS</td>
<td>MT_CI_SSF</td>
</tr>
<tr>
<td>MT_AI_LStack</td>
<td>MT_AI_LStack</td>
</tr>
</tbody>
</table>

**MT/MT_A_Sk_MP:**

- MT/MT_A_Sk_MI_AdminState
- MT/MT_A_Sk_MI_Label[1...M]
- MT/MT_A_Sk_MI_LSPType[1...M]
- MT/MT_A_Sk_MI_CoS[1...M]
- MT/MT_A_Sk_MI_TC2PHBMapping[1...M]
- MT/MT_A_Sk_MI_QoSDecodingMode[1...M]
- MT/MT_A_Sk_MI_AIS_Period[1...M]
- MT/MT_A_Sk_MI_AIS_CoS[1...M]
- MT/MT_A_Sk_MI_LCK_Period[1...M]
- MT/MT_A_Sk_MI_LCK_CoS[1...M]
- MT/MT_A_Sk_MI_GAL_Enable [1...M]

**Processes:**

A process diagram of this function is shown in Figure 9-11.
- Selector process:
See 8.6.1. The normal CI is blocked if Admin_State = LOCKED.

- LCK Generate process:
See 8.6.3.

- AIS process:
See 8.6.2.

- TC/Label processes:
See 8.2.2.

- Label Stack Copy process:
See 8.2.3.

- S field extraction:
Extract and process the 1-bit S Field: the retrieved S Field should have the value 0 (not bottom of label stack) to indicate the client is MPLS; for such case the traffic unit is accepted and forwarded.
(together with the PHB information) after extraction of the S-bit field to the next process. For the case the S-bit has the value 1, the traffic unit is silently discarded.

- **Defects:**
  
  *None.*

- **Consequent actions:**

  The function shall perform the following consequent actions:

  aSSF $\leftarrow$ AI_TSF
  
aAIS $\leftarrow$ AI_AIS • Defect correlations:

  *None.*

- **Performance monitoring:**

  *None.*

9.4 **MT Diagnostic Function**

9.4.1 **MT Diagnostic Trail Termination Functions for MEPs (MTDe)**

The bidirectional MTDe Flow Termination (MTDe_TT) function is performed by a co-located pair of MTDe flow termination source (MTDe_TT_So) and sink (MTDe_TT_Sk) functions as shown in Figure 9-12.

![Figure 9-12/G.8121/Y.1381 – MTDe_TT](image)

9.4.1.1 **MT Diagnostic Flow Termination Source Function for MEPs (MTDe_FT_So)**

The MTDe_FT_So Process diagram is shown in Figure 9-13.

**Symbol**
Figure 9-13/G.8121/Y.1381– MTDe_TT_So symbol

Interfaces

Table 9-6/G.8121/Y.1381 – MTDe_TT_So interfaces

<table>
<thead>
<tr>
<th>Input(s)</th>
<th>Output(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTDe_AP:</td>
<td>MT_CP:</td>
</tr>
<tr>
<td>MTDe_AI_D</td>
<td>MT_CI_D</td>
</tr>
<tr>
<td>MTDe_AI_oPHB</td>
<td>MT_CI_oPHB</td>
</tr>
<tr>
<td>MTDe_AI_iPHB</td>
<td>MT_CI_iPHB</td>
</tr>
<tr>
<td>MTDe_AI_LStack</td>
<td>MT_CI_LStack</td>
</tr>
</tbody>
</table>

MTDe_TT_So_MP:
- MTDe_TT_So_MI_GAL_Enable
- MTDe_TT_So_MI_TTLVALUE
- MTDe_TT_So_MI.CV_OAM_Tool
- MTDe_TT_So_MI.CV_Series
  (Target_MEP/MIP_ID,TTL,CoS,N,Length,Period)
- MTDe_TT_So_MI.IETH_OAM_Tool
- MTDe_TT_So_MI.IETH_Start
  (CoS,Length,Period)
- MTDe_TT_So_MI.IETH_Terminate
- MTDe_TT_So_MI.LMo.OAM_Tool
- MTDe_TT_So_MI.LMo.Start(CoS,Period) [1,...MLoMo]
- MTDe_FT_So_MI.LMo_TERMINATE[1,...MLoMo]
- MTDe_TT_So_MI.DMo.OAM_Tool
- MTDe_TT_So_MI.DMo.Start
  (CoS,Test_ID,Length,Period)[1,...MDMo]
- MTDe_TT_So_MI.DMo_TERMINATE[1,...MDMo]
- MTDe_TT_So_MI.1DMo.OAM_Tool
- MTDe_TT_So_MI.1DMo.Start

MTDe_TT_So_MP:
- MTDe_TT_So_MI.CV_Series_Result(REC,ERR,OO)
- MTDe_TT_So_MI.IETH_Result(Sent)
- MTDe_TT_So_MI.LMo_Result(N_TF,N_LF,F_TF,F_LF)[1,...MLoMo]
- MTDe_TT_So_MI.DMo_Result(count,B_FD[],F_FD[],N_FD[],N_DM[])[1,...MDMo]
- MTDe_TT_So_MI.SLo_Result(N_TF,N_LF,F_TF,F_LF)[1,...MSoLo]
Processes

MTDe_AP

G-Ach Insertion

On-demand OAM PDU Process

On-demand OAM Source Control Process

MT_TCP

Figure 9-14/G.8121/Y.1381 – MTDe_FT_So Process

On-demand OAM Source Control Process: See 8.8
OAM PDU Generation Process: See 8.8

Defects
None

Consequent actions
None

Defect correlations
None

Performance monitoring
None

9.4.1.2 MT Diagnostic Trail Termination Sink Function for MEPs (MTDe_TT_Sk)

Symbol

Figure 9-15/G.8121/Y.1381 – MTDe_TT_Sk symbol

Interfaces

Table 9-7/G.8121/Y.1381 – MTDe_TT_Sk interfaces

<table>
<thead>
<tr>
<th>Input(s)</th>
<th>Output(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MT_TCP:</strong></td>
<td><strong>MTDe_AP:</strong></td>
</tr>
<tr>
<td>MT_CI_D</td>
<td>MTDe_AI_D</td>
</tr>
<tr>
<td>MT_CI_iPHB</td>
<td>MTDe_AI_oPHB</td>
</tr>
<tr>
<td>MT_CI_oPHB</td>
<td>MTDe_AI_iPHB</td>
</tr>
<tr>
<td>MT_CI_LStack</td>
<td>MTDe_AI_LStack</td>
</tr>
<tr>
<td><strong>MT_TT_Sk_MP:</strong></td>
<td><strong>MTDe_RP:</strong></td>
</tr>
<tr>
<td>MTDe_TT_Sk_M1_GAL_Enable</td>
<td>MTDe_RI_OAM_Info(D,CoS,DP)</td>
</tr>
<tr>
<td>MTDe_TT_Sk_M1_MEG_ID</td>
<td></td>
</tr>
<tr>
<td>MTDe_TT_Sk_M1_PeerMEP_ID</td>
<td></td>
</tr>
<tr>
<td>MTDe_TT_Sk_M1.CV_OAM_Tool</td>
<td></td>
</tr>
<tr>
<td>MTDe_TT_Sk_M1.1TH_OAM_Tool</td>
<td></td>
</tr>
<tr>
<td>MTDe_TT_Sk_M1.1TH_Start</td>
<td></td>
</tr>
<tr>
<td>MTDe_TT_Sk_M1.1TH_Terminate</td>
<td></td>
</tr>
<tr>
<td>MTDe_TT_Sk_M1.LMo_OAM_Tool</td>
<td></td>
</tr>
<tr>
<td><strong>MT_RP:</strong></td>
<td><strong>MTDe_FT_Sk_MP:</strong></td>
</tr>
<tr>
<td>MTDe_FT_Sk_MP:</td>
<td>MTDe_TT_Sk_M1.1TH_Result(REC,CRC,BER,OO)</td>
</tr>
<tr>
<td>MTDe_FT_Sk_MP:</td>
<td>MTDe_TT_Sk_M1.DMo_Result(count,N_FD[])[1 ...M000]]</td>
</tr>
</tbody>
</table>
**Input(s)**
- MTDe_TT_Sk_MI_DMo_OAM_Tool
- MTDe_TT_Sk_MI_1DMo_OAM_Tool
- MTDe_TT_Sk_MI_1DMo_Start(Test_ID)[1...M]
- MTDe_TT_Sk_MI_1DMo_Terminate[1...M]
- MTDe_TT_Sk_MI_SLo_OAM_Tool

**Output(s)**
- MTDe_TP
- MTDe_TT_Sk_TI_TimeStamp

**Processes**

---

**Figure 9-16/G.8121/Y.1381 – MTDe_TT_Sk Process**
On-demand OAM Sink Control Process: See 8.8/G.811
OAM PDU Reception Process: See 8.8

Defects
None

Consequent actions
None

Defect correlations
None

Performance monitoring
None

9.4.2 MT Diagnostic Flow Termination Functions for MIPs (MTDi_TT)
9.4.2.1 MT Diagnostic Trail Termination Functions for MIPs (MTDi_TT_So)
9.4.2.1.1 MT Diagnostic Trail Termination Source Function for MIPs (MTDi_TT_So)

Symbol

Figure 9-17/G.8121/Y.1381 – MTDi_TT_So symbol

Interfaces
Table 9-8/G.8121/Y.1381 – MTDi_TT_So interfaces

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MTDi_AP</strong></td>
<td><strong>MTDi_TCP</strong></td>
</tr>
<tr>
<td>MT_AI_D</td>
<td>MT_CI_D</td>
</tr>
<tr>
<td>MT_AI_iPHB</td>
<td>MT_CI_iPHB</td>
</tr>
<tr>
<td>MT_AI_oPHB</td>
<td>MT_CI_oPHB</td>
</tr>
<tr>
<td>MT_AI_Lstack</td>
<td>MT_CI_LStack</td>
</tr>
<tr>
<td><strong>MTDi_RP</strong></td>
<td><strong>MTDi_TT_So_MP</strong></td>
</tr>
<tr>
<td>MT_RI_OAM_Info (D, CoS, DP)</td>
<td>MTdi_TT_So_MI_GAL_Enable</td>
</tr>
<tr>
<td><strong>MTDi_TT_So_MP</strong></td>
<td>MTdi_TT_So_MI_TTLVALUE</td>
</tr>
<tr>
<td>MTdi_TT_So_MI_MIP_ID</td>
<td>MTdi_TT_So_MI_CI_Lstack</td>
</tr>
<tr>
<td>MTdi_TT_So_MI_CV_OAM_Tool</td>
<td></td>
</tr>
</tbody>
</table>

**Processes**

![Diagram](image)

**MTDi_TCP**

*Figure 9-18/G.8121/Y.1381 – MTDi_TT_So Process (remove RT)*

**MIP OAM insertion:**

The MIP OAM Insertion process inserts OAM Traffic Units that are generated in the MTDi_TT_So process into the stream of Traffic Units.
The GAL is used or not according to the MI_GAL_Enable parameter.

**On-demand OAM PDU Generation Process**: See clause 8.8.
**On-demand OAM Source Control Process**: See clause 8.8.

**Defects** None.

**Consequent actions** None.

**Defect correlations** None.

**Performance monitoring** None.

9.4.2.1.2 MT Diagnostic Trail Termination Sink Function for MIPs (MTDi_TT_Sk)

**Symbol**

![Diagram](figure)

**Interfaces**

**Table**

Table 9-9/G.8121/Y.1381 – MTDi_TT_Sk interfaces
Inputs

<table>
<thead>
<tr>
<th>MTDi_TCP</th>
<th>MTDi_AP</th>
<th>MTDi_TT_Sk_MP</th>
<th>MTDi_AP</th>
</tr>
</thead>
<tbody>
<tr>
<td>MT_CI_D</td>
<td>MT_AI_D</td>
<td>MTDi_TT_Sk_MP</td>
<td>MT_AI_D</td>
</tr>
<tr>
<td>MT_CI_iPHB</td>
<td>MT_AI_iPHB</td>
<td>MTDi_TT_Sk_MI_GAL_Enable</td>
<td>MT_AI_iPHB</td>
</tr>
<tr>
<td>MT_CI_oPHB</td>
<td>MT_AI_oPHB</td>
<td>MTDi_TT_Sk_MI_MIP_ID</td>
<td>MT_AI_oPHB</td>
</tr>
<tr>
<td>MT_CI_LStack</td>
<td>MT_AI_LStack</td>
<td>MTDi_TT_Sk_MI_CV_OAM_Tool</td>
<td>MTDi_AP</td>
</tr>
</tbody>
</table>

Outputs

<table>
<thead>
<tr>
<th>MTDi_AP</th>
<th>MT_AI_D</th>
<th>MTDi_TT_Sk_MP</th>
<th>MTDi_AP</th>
</tr>
</thead>
<tbody>
<tr>
<td>MT_CI_D</td>
<td>MT_AI_D</td>
<td>MTDi_TT_Sk_MP</td>
<td>MT_AI_D</td>
</tr>
<tr>
<td>MT_CI_iPHB</td>
<td>MT_AI_iPHB</td>
<td>MTDi_TT_Sk_MI_GAL_Enable</td>
<td>MT_AI_iPHB</td>
</tr>
<tr>
<td>MT_CI_oPHB</td>
<td>MT_AI_oPHB</td>
<td>MTDi_TT_Sk_MI_MIP_ID</td>
<td>MT_AI_oPHB</td>
</tr>
<tr>
<td>MT_CI_LStack</td>
<td>MT_AI_LStack</td>
<td>MTDi_TT_Sk_MI_CV_OAM_Tool</td>
<td>MT_AI_LStack</td>
</tr>
</tbody>
</table>

MTDi_AP

MT_TTCP

Figure 9-20/G.8121/Y.1381 – MTDi_TT_Sk Process (remove RT)
**MIP OAM extraction**:  
The MIP OAM Extraction process classifies the OAM traffic units targeted to the MIP to which this MTDi_TT belongs, as configured by MI_MIP_ID, and delivers them to the On-demand OAM PDU Reception Process. All the other traffic units are delivered to MTDi_AP.

**On-demand OAM PDU Reception Process**: See clause 8.8.

**On-demand OAM Sink Control Process**: See clause 8.8.

<table>
<thead>
<tr>
<th><strong>Defects</strong></th>
<th>None.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Consequent actions</strong></td>
<td>None.</td>
</tr>
<tr>
<td><strong>Defect correlations</strong></td>
<td>None.</td>
</tr>
<tr>
<td><strong>Performance monitoring</strong></td>
<td>None.</td>
</tr>
</tbody>
</table>
9.4.2.2 MTDi to MT Adaptation functions (MTDi/MT_A)

The MTDi/MT adaptation function is an empty function; it is included to satisfy the modelling rules.

The bidirectional MTD/MT adaptation function is performed by a co-located pair of MTDi/MT adaptation source (MTDi/MT_A_So) and sink (MTDi/MT_A_Sk) functions.

9.4.2.2.1 MTDi to MT adaptation source functions (MTDi/MT_A_So)

The MTDi/MT_A_So function symbol is shown in Figure 9-xx and the process in Figure 9-21.

![Figure 9-21/G.8121/Y.1381 – MTDi/MT_A_So symbol](image)

**Interfaces**

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTDi/MT_CP</td>
<td>MTDi/MT_AP</td>
</tr>
<tr>
<td>MTDi/MT</td>
<td></td>
</tr>
<tr>
<td>MTDi/MT_AP</td>
<td></td>
</tr>
</tbody>
</table>

**Table 9-10 – MTDi/MT_A_So interfaces**

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTDi/MT_CP</td>
<td>MTDi/MT_AP</td>
</tr>
<tr>
<td>MTDi/MT</td>
<td></td>
</tr>
<tr>
<td>MTDi/MT_AP</td>
<td></td>
</tr>
<tr>
<td>MTDi/MT_CP</td>
<td></td>
</tr>
<tr>
<td>MTDi/MT</td>
<td></td>
</tr>
<tr>
<td>MTDi/MT_AP</td>
<td></td>
</tr>
</tbody>
</table>

**Processes**

![Figure 9-22/G.8121/Y.1381 – MTDi/MT_A_So Process](image)
Defects
None.

Consequent Actions
None.

Defect correlations
None.

Performance Monitoring
None.

9.4.2.2.2 MTDi to MT adaptation sink function (MTDi/MT_A_Sk)
The MTDi/MT_A_So function symbol is shown in Figure 9-xx and the process in Figure 9-xx.

![Diagram of MTDi/MT_A_Sk function symbol]

Figure 9-23/G.8121/Y.1381 – MTDi/MT_A_Sk symbol

Interfaces

Table 9-11/G.8121/Y.1381 – MTDi/MT_A_Sk interfaces

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>MT_AP:</td>
<td>MT_C_P:</td>
</tr>
<tr>
<td>MT_AI_D</td>
<td>MT_AI_D</td>
</tr>
<tr>
<td>MT_AI_iPHB</td>
<td>MT_AI_iPHB</td>
</tr>
<tr>
<td>MT_AI_oPHB</td>
<td>MT_AI_oPHB</td>
</tr>
<tr>
<td>MT_AI_LStack</td>
<td>MT_AI_LStack</td>
</tr>
<tr>
<td>MT_MP:</td>
<td></td>
</tr>
<tr>
<td>MI_DS_MP_Type</td>
<td></td>
</tr>
</tbody>
</table>

Processes
The TTL check process drops all MPLS-TP packets with TTL = 0 by default (MI_DS_MP_Type set to none).

When MI_DS_MP_Type is set to MIP, TTL check process drops only user data MPLS-TP packets with TTL = 0 while OAM packets with TTL = 0 are not dropped in this process and forwarded.

When the MI_DS_MP_Type is set to MEP, TTL check process does not block any MPLS-TP packet with TTL = 0: all MPLS-TP packets with TTL = 0 are forwarded.

NOTE – The MI_DS_MP_Type parameter should be properly configured by the EMF on the basis of the MPLS-TP connection configuration within the node and not exposed to the operator as a configuration parameter of the Equipment Management Interface. Examples of MI_DS_MP_Type configuration are described in Appendix I.

Defects None.
Consequent Actions None.
Defect correlations None.
Performance Monitoring None.

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

10.1 MPLS-TP to ETH adaptation function (MT/ETH_A)

10.1.2 MPLS-TP to ETH adaptation source function (MT/ETH_A_So)

This function maps the ETH_CI information for transport in an MT_AI signal.

The information flow and processing of the MT/ETH_A_So function is defined with reference to Figure 10-1.

• Symbol:
Figure 10-1/G.8121/Y.1381 – MT/ETH_A_So function

- Interfaces:

Table 10-1/G.8121/Y.1381 – MT/ETH_A_So Inputs and Outputs

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>ETH_FP: ETH_CI Data ETH_CI P ETH_CI_DE</td>
<td>MT_AP: MT_AI_Data MT_AI_PHB</td>
</tr>
<tr>
<td>MT/ETH_A_So_MP: ETH_CI_MI_AdminState</td>
<td></td>
</tr>
<tr>
<td>MT/ETH_A_So_MI_FCSEnable</td>
<td></td>
</tr>
<tr>
<td>MT/ETH_A_So_MI CWEnable</td>
<td></td>
</tr>
<tr>
<td>MT/ETH_A_So_MI_SQUse</td>
<td></td>
</tr>
<tr>
<td>MT/ETH_A_So_MI_PRI2PSCMapping</td>
<td></td>
</tr>
<tr>
<td>MT/ETH_A_So_MI MEP_MAC*</td>
<td></td>
</tr>
<tr>
<td>MT/ETH_A_So_MI_Common_MEL*</td>
<td></td>
</tr>
<tr>
<td>MT/ETH_A_So_MI_LCK_Period*</td>
<td></td>
</tr>
<tr>
<td>MT/ETH_A_So_MI_LCK_Pri*</td>
<td></td>
</tr>
<tr>
<td>MT/ETH_A_So_MI_MEL*</td>
<td></td>
</tr>
<tr>
<td>* ETH OAM related</td>
<td></td>
</tr>
</tbody>
</table>
• Processes:
The processes associated with the MT/ETH_A_So function are as depicted in Figure 10-2.

Figure 10-2/G.8121/Y.1381 – MT/ETH_A_So process diagram

- **LCK Generate process:**
  See 8.1.2 of [ITU-T G.8021].
- **Selector process:**
  See 8.1.3 of [ITU-T G.8021]. The normal CI is blocked if Admin_State = LOCKED.
- **OAM MEL Filter process:**
  See 8.1.1 of [ITU-T G.8021].
802.3 MAC FCS generation:
See 8.8.1 of [ITU-T G.8021]. MAC FCS generation is optional (see [IETF RFC 4720] and [ITU-T Y.1415]): MAC FCS is generated if MI_FCSEnabled is True.

CW Insertion process:
See 8.5.1.

QoS mapping process:
This process maps the Ethernet-based QoS signals into MPLS-based QoS signals.
The CoS part of the AI_PHB is generated by the received CI_P according to the 1:1 mapping configured by the MI_PRI2PSCMapping.
The DP part of the AI_PHB is generated by the received CI_DE according to the following rule:

\[
\begin{align*}
\text{If } \text{CI}_{\text{DE}} &= \text{True} \\
\text{Then } DP(\text{AI}_{\text{PHB}}) &= \text{Yellow} \\
\text{Else } DP(\text{AI}_{\text{PHB}}) &= \text{Green}
\end{align*}
\]

S Field insertion:
A 1-bit S Field set to 1 (bottom of label stack) is inserted to indicate the client is not MPLS.

- **Defects:**
  None.

- **Consequent actions:**
  None.

- **Defect correlations:**
  None.

- **Performance monitoring:**
  None.

10.1.2 MPLS-TP to ETH adaptation sink function (MT/ETH_A_Sk)
This function extracts the ETH_CI information from an MT_AI signal.
The information flow and processing of the MT/ETH_A_Sk function is defined with reference to Figure 10-3.

- **Symbol:**
- Interfaces:

**Table 10-2/G.8121/Y.1381 – MT/ETH_A_Sk Inputs and Outputs**

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Each MT_AP:</td>
<td>ETH_FP:</td>
</tr>
<tr>
<td>MT_AI_Data</td>
<td>ETH_CI_Data</td>
</tr>
<tr>
<td>MT_AI_PHB</td>
<td>ETH_CI_P</td>
</tr>
<tr>
<td>MT_AI_TSF</td>
<td>ETH_CI_DE</td>
</tr>
<tr>
<td><strong>MT/ETH_A_Sk_MP:</strong></td>
<td>ETH_CI_SSF</td>
</tr>
<tr>
<td>MT/ETH_A_Sk_MI_FCSEnable</td>
<td></td>
</tr>
<tr>
<td>MT/ETH_A_Sk_MI_CIIEnable</td>
<td></td>
</tr>
<tr>
<td>MT/ETH_A_So_MI_SQUse</td>
<td></td>
</tr>
<tr>
<td>MT/ETH_A_Sk_MI_CoS2PRIMapping</td>
<td></td>
</tr>
<tr>
<td>MT/ETH_A_Sk_MI_Admin_State</td>
<td></td>
</tr>
<tr>
<td>MT/ETH_A_Sk_MI_LCK_Period *</td>
<td></td>
</tr>
<tr>
<td>MT/ETH_A_Sk_MI_LCK_Pri *</td>
<td></td>
</tr>
<tr>
<td>MT/ETH_A_Sk_MI_Client_MEL *</td>
<td></td>
</tr>
<tr>
<td>MT/ETH_A_Sk_MI_MEP_MAC *</td>
<td></td>
</tr>
<tr>
<td>MT/ETH_A_Sk_MI_AIS_Pri *</td>
<td></td>
</tr>
<tr>
<td>MT/ETH_A_Sk_MI_AIS_Period *</td>
<td></td>
</tr>
<tr>
<td>* ETH OAM related</td>
<td></td>
</tr>
</tbody>
</table>

* ETH OAM related
• Processes:

![Figure 10-4/G.8121/Y.1381 – MT/ETH_A_Sk process diagram](image)

- **Selector process:**
  See 8.1.3 of [ITU-T G.8021]. The normal CI is blocked if Admin_State = LOCKED.
– **LCK Generate process:**

See 8.1.2 of [ITU-T G.8021].

– **AIS Insert process:**

See 8.1.4 of [ITU-T G.8021].

– **OAM MEL Filter process:**

See 8.1.1 of [ITU-T G.8021].

– "802.3 MAC Frame Check" process:

See 8.9.2/G.8021/Y.1341. MAC Frame Check is optional (see [IETF RFC 4720] and [ITU-T Y.1415]): MAC FCS is checked if MI_FCSEnabled is True.

– **CW Extraction process:**

See 8.5.2.

– **QoS mapping process:**

This process maps the MPLS-based QoS signals into Ethernet-based QoS signals.

The CI_P is generated by the received PSC part of the AI_PHB according to the 1:1 mapping configured by the MI_CoS2PRIMapping.

The CI_DE is generated by the received DP part of the AI_PHB according to the following rule:

\[
\text{If } \text{DP}(\text{AI_PHB}) = \text{Green} \\
\quad \text{CI_DE} = \text{False} \\
\text{Else} \\
\quad \text{CI_DE} = \text{True}
\]

– **S field extraction:**

Extract and process the 1-bit S Field: the retrieved S Field should have the value 1 (bottom of label stack) to indicate the client is not MPLS: for such case the traffic unit is accepted and forwarded (together with the PHB information) after extraction of the S-bit field to the next process. For the case the S-bit has the value 0, the traffic unit is silently discarded.

- **Defects:**

  None.

- **Consequent actions:**

The function shall perform the following consequent actions:

aSSF ← AI_TSF and (not MI_Admin_State == LOCKED)

aAIS ← AI_AIS

- **Defect correlations:**

  None.

- **Performance monitoring:**

  Ffs.
10.2  MPLS-TP to SCC and MCC Adaptation functions

This clause provides the descriptions of the MPLS-TP adaptation functions for the MPLS-TP MCC and SCC.

Figure 10-5 shows the MPLS-TP adaptation functions providing access to the MCC and SCC. These MT/MCC and MT/SCC adaptation functions are defined in more detail below.

In case that the client is MPLS-TP, MT/Client will be MT/MT,

![Diagram showing adaptation functions](image)

**Figure 10-5/G.8121/Y.1382 – MT/SCC_A function, MT/MCC_A function, and MT/client_A function**

### 10.2.1  MT/SCC_A Adaptation Function

The MT to SCC adaptation function provides access to the SCC for signalling communication. It is used for the scenarios where the SCN utilizes the SCC as defined in [IETF RFC5718].

#### 10.2.1.1  MT to SCC adaptation source function (MT/SCC_A_So function)

The MT/SCC_A_So function maps the SCN data into the G-ACh SCC packets as defined in [IETF RFC5718]. The diamonds in Figure 10-6 represent traffic shaping and conditioning functions that may be needed to prevent the SCC forwarding points from exceeding their committed bandwidth in congestion situations. These traffic shaping and conditioning functions as well as the related bandwidth management and bandwidth assignment functions are outside the scope of this recommendation.

The information flow and processing of the MT/SCC_A_So functions is defined with reference to Figures 10-6 and 10-7.
Symbol

Figure 10-6/G.8121/Y.1382 – MT/SCC_A_So function

Interfaces

Table 10-3/G.8121/Y.1382 – MT/SCC_A_So inputs and outputs

<table>
<thead>
<tr>
<th>Input(s)</th>
<th>Output(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCC_FP:</td>
<td>MT_AP:</td>
</tr>
<tr>
<td>SCC_CI_D</td>
<td>MT_AI_D</td>
</tr>
<tr>
<td>MT/SCC_A_So_MP:</td>
<td>MT_AI_PHB</td>
</tr>
<tr>
<td>MT/SCC_A_So_MI_Active</td>
<td></td>
</tr>
<tr>
<td>MT/SCC_A_So_MI_ECC_CoS</td>
<td></td>
</tr>
<tr>
<td>MT/SCC_A_So_MI_GAL_Enable</td>
<td></td>
</tr>
</tbody>
</table>

Processes

Activation

The MT/SCC_A_So function shall access the access point when it is activated (MI_Active is true). Otherwise, it shall not access the access point.

The process associated with the MT/SCC_A_So function is as depicted in Figure 10-7.
ECC Mapping process: See clause 8.7.1.1

Defects: None.
Consequent actions: None.
Defect correlations: None.
Performance Monitoring: None.

10.2.1.2 MT to SCC adaptation sink function (MT/SCC_A_Sk function)
The MT/SCC_A_Sk function extracts the SCN from the G-ACh SCC packets as defined in [IETF RFC5718].

The information flow and processing of the MT/SCC_A_Sk functions is defined with reference to Figures 10-8 and 10-9.
Symbol

![Diagram](image)

Figure 10-8/G.8121/Y.1382 – MT/SCC_A_Sk function

Interfaces

Table 10-4/G.8121/Y.1382 – MT/SCC_A_Sk inputs and outputs

<table>
<thead>
<tr>
<th>Input(s)</th>
<th>Output(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MT_AP:</strong></td>
<td><strong>SCC_FP:</strong></td>
</tr>
<tr>
<td>MT_AI_D</td>
<td>SCC_CI_D</td>
</tr>
<tr>
<td>MT_AI_PHB</td>
<td>SCC_CI_SSF</td>
</tr>
<tr>
<td>MT_AI_TSF</td>
<td></td>
</tr>
<tr>
<td><strong>MT/SCC_A_Sk_MP:</strong></td>
<td></td>
</tr>
<tr>
<td>MT/SCC_A_Sk_MI_Active</td>
<td></td>
</tr>
<tr>
<td>MT/SCC_A_Sk_MI_GAL_Enable</td>
<td></td>
</tr>
</tbody>
</table>

Processes

Activation

The MT/SCC_A_Sk function shall access the access point and perform the common and specific processes operation specified below when it is activated (MI_Active is true). Otherwise, it shall activate the SSF signals at its output (CI_SSF).

The processes associated with the MT/SCC_A_Sk function are as depicted in Figure 10-9.
ECC Demapping process: See clause 8.7.1.2

Defects: None.

Consequent actions
The function shall perform the following consequent actions:
\[ \text{aSSF} \leftarrow \text{AI\_TSF or (not MI\_Active)} \]

Defect correlations: None.
Performance monitoring: None.

10.2.2 MT/MCC_A Adaptation Function
The MT to MCC adaptation function provides access to the MCC for signalling communication. It is used for the scenarios where the MCN utilizes the MCC as defined in [IETF RFC5718].

10.2.2.1 MT to MCC adaptation source function (MT/MCC_A_So function)
The MT/MCC_A_So function maps the MCN data into the G-ACh MCC packets as defined in [IETF RFC5718]. The diamonds in Figure 10-A.6 represent traffic shaping and conditioning functions that may be needed to prevent the MCC forwarding points from exceeding their committed bandwidth in congestion situations. These traffic shaping and conditioning functions as well as the related bandwidth management and bandwidth assignment functions are outside the scope of this recommendation.

The information flow and processing of the MT/MCC_A_So functions is defined with reference to Figures 10-10 and 10-11.
Symbol

MT/MCC
MCC_FPs
IP v4
FwP
IP v6
FwP
OSI
FwP
MT/MCC_A_So_MP
MT_AP

Figure 10-10/G.8121/Y.1382 – MT/MCC_A_So function

Interfaces

Table 10-5/G.8121/Y.1382 – MT/MCC_A_So inputs and outputs

<table>
<thead>
<tr>
<th>Input(s)</th>
<th>Output(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCC_FPs:</td>
<td>MT_AP:</td>
</tr>
<tr>
<td>MCC_CI_D</td>
<td>MT_AI_D</td>
</tr>
<tr>
<td>MT/MCC_A_So_MP:</td>
<td>MT_AI_PHB</td>
</tr>
<tr>
<td>MT/MCC_A_So_MI_Active</td>
<td></td>
</tr>
<tr>
<td>MT/MCC_A_So_MI_ECC_CoS</td>
<td></td>
</tr>
<tr>
<td>MT/MCC_A_So_MI_GAL_enable</td>
<td></td>
</tr>
</tbody>
</table>

Processes

Activation

The MT/MCC_A_So function shall access the access point when it is activated (MI_Active is true). Otherwise, it shall not access the access point.

The process associated with the MT/MCC_A_So function is as depicted in Figure 10-11.
10.2.2 MT to MCC adaptation source function (MT/SCC_A_Sk function)

The MT/MCC_A_Sk function extracts the MCN data from the G-ACh MCC packets as defined in [IETF RFC5718].

The information flow and processing of the MT/MCC_A_Sk functions is defined with reference to Figures 10-12 and 10-13.
**Symbol**

![Symbol Diagram]

**Figure 10-12/G.8121/Y.1382 – MT/MCC_A_Sk function**

**Interfaces**

**Table 10-6/G.8121/Y.1382 – MT/MCC_A_Sk inputs and outputs**

<table>
<thead>
<tr>
<th>Input(s)</th>
<th>Output(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MT_AP:</strong></td>
<td><strong>MCC_FP:</strong></td>
</tr>
<tr>
<td>MT_AI_D</td>
<td>MCC_CI_D</td>
</tr>
<tr>
<td>MT_AI_PHB</td>
<td>MCC_CI_SSF</td>
</tr>
<tr>
<td>MT_AI_TSF</td>
<td></td>
</tr>
<tr>
<td><strong>MT/MCC_A_Sk_MP:</strong></td>
<td></td>
</tr>
<tr>
<td>MT/MCC_A_Sk_MI_Active</td>
<td></td>
</tr>
<tr>
<td>MT/SCC_A_Sk_MI_GAL_Enable</td>
<td></td>
</tr>
</tbody>
</table>

**Processes**

*Activation*

The MT/MCC_A_Sk function shall access the access point and perform the common and specific processes operation specified below when it is activated (MI_Active is true). Otherwise, it shall activate the SSF signals at its output (CI_SSF).

The processes associated with the MT/MCC_A_Sk function are as depicted in Figure 10-13.
**ECC Demapping process**: See clause 8.7.1.2

**Defects**: None.

**Consequent actions**

The function shall perform the following consequent actions:

\[ aSSF \leftarrow AI_{TSF} \text{ or } (\text{not MI}_{Active}) \]

**Defect correlations**: None.

**Performance monitoring**: None.

11 **Non-MPLS-TP Server to MPLS-TP adaptation functions**
11.1 SDH to MPLS-TP adaptation function (S/Mt_A)

11.1.1 VC-n to MPLS-TP adaptation functions (Sn/Mt_A; n=3, 3-X, 4, 4-X)

11.1.1.1 VC-n to MPLS-TP adaptation source function (Sn/Mt_A_So)

This function maps Mt_CI information onto an Sn_AI signal (n=3, 3-X, 4, 4-X).

Data at the Sn_AP is a VC-n (n = 3, 3-X, 4, 4-X), having a payload as described in [ITU-T G.707], but with indeterminate POH bytes: J1, B3, G1.

- Symbol:

![Symbol Diagram](image)

Figure 11-1/G.8121/Y.1381 – Sn/Mt_A_So symbol

- Interfaces:

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Each Mt_CP:</td>
<td>Sn_AP:</td>
</tr>
<tr>
<td>Mt_CI Data</td>
<td>Sn_AI_Data</td>
</tr>
<tr>
<td>Mt_CI_iPHB</td>
<td>Sn_AI_Clock</td>
</tr>
<tr>
<td>Mt_CI_oPHB</td>
<td>Sn_AI_FrameStart</td>
</tr>
<tr>
<td>SCC_CP:</td>
<td></td>
</tr>
<tr>
<td>SCC_CI_Data</td>
<td></td>
</tr>
<tr>
<td>Sn_TP:</td>
<td></td>
</tr>
<tr>
<td>Sn_TI_Clock</td>
<td></td>
</tr>
<tr>
<td>Sn_TI_FrameStart</td>
<td></td>
</tr>
<tr>
<td>Sn/Mt_A_So_MP:</td>
<td></td>
</tr>
<tr>
<td>Sn/Mt_A_So_MI_SCCType</td>
<td></td>
</tr>
<tr>
<td>Sn/Mt_A_So_MI_Label[1…M]</td>
<td></td>
</tr>
<tr>
<td>Sn/Mt_A_So_MI_LSPType[1…M]</td>
<td></td>
</tr>
<tr>
<td>Sn/Mt_A_So_MI_CoS[1…M]</td>
<td></td>
</tr>
<tr>
<td>Sn/Mt_A_So_PHB2TCMapping[1…M]</td>
<td></td>
</tr>
<tr>
<td>Sn/Mt_A_So_QoSEncodingMode[1…M]</td>
<td></td>
</tr>
</tbody>
</table>
• Processes:

A process diagram of this function is shown in Figure 11-2.

![Process Diagram](image)

**Figure 11-2/G.8121/Y.1381 – Sn/MT_A_So process diagram**

- **TC/Label processes:**
  See 8.2.1.

- **Queueing process:**
  See 8.3.

- **MPLS-TP-specific GFP-F source process:**
  See 8.4.1.

- **Common GFP source process:**
  See 8.5.3.1/G.806. GFP channel multiplexing is not supported (CMuxActive=false).
– **VC-n specific GFP source process:**

See 8.5.2.1/G.806. The GFP frames are mapped into the VC-n payload area according to 10.6/G.707/Y.1322.

– **VC-n specific source process:**

C2: Signal label information is derived directly from the Adaptation function type. The value for "GFP mapping" in Table 9-11/G.707/Y.1322 is placed in the C2 byte position.

H4: For Sn/MT_A_So with n=3, 4, the H4 byte is sourced as all-zeros.

NOTE 1 – For Sn/MT_A_So with n=3-X, 4-X, the H4 byte is undefined at the Sn-X_AP output of this function (as per clause 12/G.783).

NOTE 2 – For Sn/MT_A_So with n=3, 4, 3-X, 4-X, the K3, F2, F3 bytes are undefined at the Sn-X_AP output of this function (as per clause 12/G.783).

• **Defects:**

  None.

• **Consequent actions:**

  None.

• **Defect correlations:**

  None.

• **Performance monitoring:**

  Ffs.

11.1.1.2 **VC-n to MPLS-TP adaptation sink function (Sn/MT_A_Sk)**

This function extracts MT_CI information from the Sn_AI signal (n=3, 3-X, 4, 4-X), delivering MT_CI.

Data at the Sn_AP is a VC-n (n=3, 3-X, 4, 4-X) but with indeterminate POH bytes J1, B3, G1, as per [ITU-T G.707].

• **Symbol:**

```
Sn/MT_A_Sk

Sn_CI

SCC_CI

MT_CI

...```

```
Sn/MT_A_Sk_MI
```

**Figure 11-3/G.8121/Y.1381 – Sn/MT_A_Sk symbol**
**Interfaces:**

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sn_AP:</td>
<td>Each MT_CP:</td>
</tr>
<tr>
<td>Sn_AI_Data</td>
<td>MT_CI_Data</td>
</tr>
<tr>
<td>Sn_AI_ClocK</td>
<td>MT_CI_iPHB</td>
</tr>
<tr>
<td>Sn_AI_FrameStart</td>
<td>MT_CI_oPHB</td>
</tr>
<tr>
<td>Sn_AI_TSF</td>
<td>MT_CI_SSF</td>
</tr>
<tr>
<td>Sn/MT_A_Sk_MP:</td>
<td>MT_CI_LStack</td>
</tr>
<tr>
<td>Sn/MT_A_Sk_MI_SCCType</td>
<td>SCC_CP:</td>
</tr>
<tr>
<td>Sn/MT_A_Sk_MI_Label[1…M]</td>
<td>SCC_CI_Data</td>
</tr>
<tr>
<td>Sn/MT_A_Sk_MI_LSPType[1…M]</td>
<td>SCC_CI_SSF</td>
</tr>
<tr>
<td>Sn/MT_A_Sk_MI_CoS[1…M]</td>
<td>Sn/MT_A_Sk_MP:</td>
</tr>
<tr>
<td>Sn/MT_A_Sk_MI_TC2PHBMapping[1…M]</td>
<td>Sn/MT_A_Sk_MI_AcSL</td>
</tr>
<tr>
<td>Sn/MT_A_Sk_MI_QoSDecodingMode[1…M]</td>
<td>Sn/MT_A_Sk_MI_AcEXI</td>
</tr>
<tr>
<td>Sn/MT_A_Sk_MI_LCK_Period[1…M]</td>
<td>Sn/MT_A_Sk_MI_LastValidUPI</td>
</tr>
<tr>
<td>Sn/MT_A_Sk_MI_LCK_CoS[1…M]</td>
<td>Sn/MT_A_Sk_MI_cPLM</td>
</tr>
<tr>
<td>Sn/MT_A_Sk_MI_Admin_State</td>
<td>Sn/MT_A_Sk_MI_cLFD</td>
</tr>
<tr>
<td>Sn/MT_A_Sk_MI_AIS_Period[1…M]</td>
<td>Sn/MT_A_Sk_MI_cEXM</td>
</tr>
<tr>
<td>Sn/MT_A_Sk_MI_AIS_CoS[1…M]</td>
<td>Sn/MT_A_Sk_MI_cUPM</td>
</tr>
<tr>
<td>Sn/MT_A_Sk_MI_GAL_enable[1…M]</td>
<td></td>
</tr>
</tbody>
</table>

**Processes:**

A process diagram of this function is shown in Figure 11-4.
Figure 11-4/G.8121/Y.1381 – Sn/MT_A_Sk process diagram

- **Selector generation process:**
  See 8.6.1 The normal CI is blocked if Admin_State = LOCKED.

- **AIS Insert process:**
  See 8.6.2. There is a single AIS Insert process for each MT.

- **LCK generation process:**
  See 8.6.3. There is a single LCK Insert process for each MT.

- **TC/Label processes:**
  See 8.2.2.

- **Label Stack Copy process:**
  See 8.2.3.

- **MPLS-TP-specific GFP-F sink process:**
  See 8.4.2.
– **Common GFP sink process:**
See 8.5.3.2 in [ITU-T G.806]. GFP channel multiplexing is not supported (CMuxActive=false).

– **VC-n specific GFP sink process:**
See 8.5.2.2 in [ITU-T G.806]. The GFP frames are demapped from the VC-n payload area according to 10.6 in [ITU-T G.707].

– **VC-n-specific sink process:**
C2: The signal label is recovered from the C2 byte as per 6.2.4.2 in [ITU-T G.806]. The signal label for "GFP mapping" in Table 9-11 of [ITU-T G.707] shall be expected. The accepted value of the signal label is also available at the Sn/MT_A_Sk_MP.

**Defects:**
dPLM – See 6.2.4.2 in [ITU-T G.806].
dLFD – See 6.2.5.2 in [ITU-T G.806].
dEXM – See 6.2.4.4 in [ITU-T G.806].
dUPM – See 8.4.2.

**Consequent actions:**
The function shall perform the following consequent actions:

- aSSF ← AI_TSF or dPLM or dLFD or dUPM or dEXM
- aAIS ← AI_TSF or dPLM or dLFD or dUPM or dEXM

**Defect correlations:**
The function shall perform the following defect correlations to determine the most probable fault cause (see 6.4/G.806). This fault cause shall be reported to the EMF.

- cPLM ← dPLM and (not AI_TSF)
- cLFD ← dLFD and (not dPLM) and (not AI_TSF)
- cEXM ← dEXM and (not dPLM) and (not dLFD) and (not AI_TSF)
- cUPM ← dUPM and (not dEXM) and (not dPLM) and (not dLFD) and (not AI_TSF)

**Performance monitoring:**
Ffs.

11.1.2 **LCAS-capable VC-n to MPLS-TP adaptation functions (Sn-X-L/MT_A; n=3, 4)**
11.1.2.1 **LCAS-capable VC-n to MPLS-TP adaptation source function (Sn-X-L/MT_A_So)**
This function maps MT_CI information onto an Sn-X-L_AI signal (n=3, 4).
Data at the Sn-X-L_AP is a VC-n-X (n = 3, 4), having a payload as described in [ITU-T G.707], but with indeterminate POH bytes: J1, B3, G1.
• Symbol:

```
  SCC_CI  MT_CI
   ↓     ↓
 Sn-X-L/MT_A_So  Sn-X-L/MT_A_So_MI
  Sn-X-L_AL_XAT  Sn-X-L_AL_D
   ↓     ↓
 Sn-X-L_AL_D  Sn-X-L_AL_D
```

Figure 11-5/G.8121/Y.1381 – Sn-X-L/MT_A_So symbol

• Interfaces:

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Each MT_CP:</td>
<td>Sn-X-L_AP:</td>
</tr>
<tr>
<td>MT_CI_Data</td>
<td>Sn-X-L_AL_Data</td>
</tr>
<tr>
<td>MT_CI_iPHB</td>
<td>Sn-X-L_AL_Clock</td>
</tr>
<tr>
<td>MT_CI_oPHB</td>
<td>Sn-X-L_AL_FrameStart</td>
</tr>
<tr>
<td>SCC_CP:</td>
<td></td>
</tr>
<tr>
<td>SCC_CI_Data</td>
<td></td>
</tr>
<tr>
<td>Sn-X-L_AL_XAT</td>
<td></td>
</tr>
<tr>
<td>Sn-X-L TP:</td>
<td></td>
</tr>
<tr>
<td>Sn-X-L_TI_Clock</td>
<td></td>
</tr>
<tr>
<td>Sn-X-L_TI_FrameStart</td>
<td></td>
</tr>
<tr>
<td>Sn-X-L/MT_A_So_MP:</td>
<td></td>
</tr>
<tr>
<td>Sn-X-L/MT_A_So_MI_SCCType</td>
<td></td>
</tr>
<tr>
<td>Sn-X-L/MT_A_So_MI_Label[1…M]</td>
<td></td>
</tr>
<tr>
<td>Sn-X-L/MT_A_So_MI_LSPType[1…M]</td>
<td></td>
</tr>
<tr>
<td>Sn-X-L/MT_A_So_MI_CoS[1…M]</td>
<td></td>
</tr>
<tr>
<td>Sn-X-L/MT_A_So_PHB2TCMapping[1…M]</td>
<td></td>
</tr>
<tr>
<td>Sn-X-L/MT_A_So_MI_QoSEncodingMode[1…M]</td>
<td></td>
</tr>
</tbody>
</table>

• Processes:

A process diagram of this function is shown in Figure 11-6.
Figure 11-6/G.8121/Y.1381 – Sn-X-L/MT_A_So process diagram

The processes have the same definition as in 11.1.1.1.

• Defects:
None.

• Consequent actions:
None.

• Defect correlations:
None.

• Performance monitoring:
Ffs.
11.1.2.2 LCAS-capable VC-n to MPLS-TP adaptation sink function (Sn-X-L/MT_A_Sk)

This function extracts MT_CI information from the Sn-X-L_AI signal (n=3, 4), delivering MT_CI. Data at the Sn-X-L_AP is a VC-n-Xv (n=3, 4) but with indeterminate POH bytes J1, B3, G1, as per [ITU-T G.707].

- **Symbol:**

```
+-----------------+         +-----------------+
| MT_CI           |         | SCC_CI           |
|                 |         |                  |
+-----------------+         +-----------------+
| Sn-X-L/MT_A_Sk  |         | Sn-X-L/MT_A_Sk_MI|
| Sn-X-L_AI_XAT   |         | Sn-X-L_AI_D      |
| Sn-X-L_AI_D     |         |                  |
+-----------------+         +-----------------+
```

**Figure 11-7/G.8121/Y.1381 – Sn-X-L/MT_A_Sk symbol**

- **Interfaces:**

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sn-X-L_AP:</td>
<td>Each MT_CP:</td>
</tr>
<tr>
<td>Sn-X-L_AI_Data</td>
<td>MT_CI_Data</td>
</tr>
<tr>
<td>Sn-X-L_AI_ClocK</td>
<td>MT_CI_iPHB</td>
</tr>
<tr>
<td>Sn-X-L_AI_FrameStart</td>
<td>MT_CI_oPHB</td>
</tr>
<tr>
<td>Sn-X-L_AI_TSF</td>
<td>MT_CI_SSF</td>
</tr>
<tr>
<td>Sn-X-L_AI_XAR</td>
<td>MI_CI_LStack</td>
</tr>
<tr>
<td>Sn-X-L/MT_A_Sk_MP:</td>
<td>SCC_CP:</td>
</tr>
<tr>
<td>Sn-X-L/MT_A_Sk_MI_SCCType</td>
<td>SCC_CI_Data</td>
</tr>
<tr>
<td>Sn-X-L/MT_A_Sk_MI_Label[1…M]</td>
<td>SCC_CI_SSF</td>
</tr>
<tr>
<td>Sn-X-L/MT_A_Sk_MI_LSPTtype[1…M]</td>
<td></td>
</tr>
<tr>
<td>Sn-X-L/MT_A_Sk_MI_CoS[1…M]</td>
<td>Sn-X-L/MT_A_Sk_MP:</td>
</tr>
<tr>
<td>Sn-X-L/MT_A_Sk_CI_TC2PHBMapping[1…M]</td>
<td>Sn-X-L/MT_A_Sk_MI_AcSL</td>
</tr>
<tr>
<td>Sn-X-L/MT_A_Sk_CI_QoSDecodingMode[1…M]</td>
<td>Sn-X-L/MT_A_Sk_MI_AcEI</td>
</tr>
<tr>
<td>Sn-X-L/MT_A_Sk_MI_LCK_Period[1…M]</td>
<td>Sn-X-L/MT_A_Sk_MI_LastValidUPI</td>
</tr>
<tr>
<td>Sn-X-L/MT_A_Sk_MI_LCK_CoS[1…M]</td>
<td>Sn-X-L/MT_A_Sk_MI_cPLM</td>
</tr>
<tr>
<td>Sn-X-L/MT_A_Sk_MI_Admin_State</td>
<td>Sn-X-L/MT_A_Sk_MI_cLFD</td>
</tr>
<tr>
<td>Sn-X-L/MT_A_Sk_MI_AIS_Period[1…M]</td>
<td>Sn-X-L/MT_A_Sk_MI_cEXM</td>
</tr>
<tr>
<td>Sn-X-L/MT_A_Sk_MI_AIS_CoS[1…M]</td>
<td>Sn-X-L/MT_A_Sk_MI_cUPM</td>
</tr>
<tr>
<td>Sn-X-L/MT_A_Sk_MI_GAL_Enable[1…M]</td>
<td></td>
</tr>
</tbody>
</table>

- **Processes:**

A process diagram of this function is shown in Figure 11-8.
Figure 11-8/G.8212/Y.1381 – Sn-X-L/MT_A_Sk process diagram

See process diagram and process description in 11.1.2. The additional Sn-X-L_AI_XAR interface is not connected to any of the internal processes.

- **Defects:**
  - dPLM – See 6.2.4.2 in [ITU-T G.806].
  - dLFD – See 6.2.5.2 in [ITU-T G.806].
  - dUPM – See 8.4.2.
  - dEXM – See 6.2.4.4 in [ITU-T G.806].
• **Consequent actions:**
The function shall perform the following consequent actions:

aSSF  \(\leftarrow\)  AI_TSF or dPLM or dLFD or dUPM or dEXM

aAIS  \(\leftarrow\)  AI_TSF or dPLM or dLFD or dUPM or dEXM

• **Defect correlations:**
The function shall perform the following defect correlations to determine the most probable fault cause (see 6.4 in [ITU-T G.806]). This fault cause shall be reported to the EMF.

cPLM  \(\leftarrow\)  dPLM and (not AI_TSF)

cLFD  \(\leftarrow\)  dLFD and (not dPLM) and (not AI_TSF)

cEXM  \(\leftarrow\)  dEXM and (not dPLM) and (not dLFD) and (not AI_TSF)

cUPM  \(\leftarrow\)  dUPM and (not dEXM) and (not dPLM) and (not dLFD) and (not AI_TSF)

• **Performance monitoring:**

**Ffs.**

11.1.3 VC-m to MPLS-TP adaptation functions (Sm/MT_A; \(m=11, 11-X, 12, 12-X\))

11.1.3.1 VC-m to MPLS-TP adaptation source function (Sm/MT_A_So)

This function maps MT_CI information onto an Sm_AI signal (m=11, 11-X, 12, 12-X).

Data at the Sm_AP is a VC-m (m = 11, 11-X, 12, 12-X), having a payload as described in [ITU-T G.707], but with indeterminate POH bytes: J2, V5[1-4], V5[8].

• **Symbol:**

![Symbol Diagram]

**Figure 11-9/G.8121/Y.1381 – Sm/MT_A_So symbol**
• Interfaces:

Table 11-5/G.8121/Y.1381 – Sm/MT_A_So interfaces

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Each MT_CP:</td>
<td>Sm_AP:</td>
</tr>
<tr>
<td>MT_CI_Data</td>
<td>Sm_AI_Data</td>
</tr>
<tr>
<td>MT_CI_iPHB</td>
<td>Sm_AI.Clock</td>
</tr>
<tr>
<td>MT_CI_oPHB</td>
<td>Sm_AI.FrameStart</td>
</tr>
<tr>
<td>SCC_CP:</td>
<td></td>
</tr>
<tr>
<td>SCC_CI_Data</td>
<td></td>
</tr>
<tr>
<td>Sm_TP:</td>
<td></td>
</tr>
<tr>
<td>Sm_TI_Clock</td>
<td></td>
</tr>
<tr>
<td>Sm_TI_FrameStart</td>
<td></td>
</tr>
<tr>
<td>Sm/MT_A_So_MP:</td>
<td></td>
</tr>
<tr>
<td>Sm/MT_A_So_MI_SCCType</td>
<td></td>
</tr>
<tr>
<td>Sm/MT_A_So_MI_Label[1…M]</td>
<td></td>
</tr>
<tr>
<td>Sm/MT_A_So_MI_LSPType[1…M]</td>
<td></td>
</tr>
<tr>
<td>Sm/MT_A_So_MI_CoS[1…M]</td>
<td></td>
</tr>
<tr>
<td>Sm/MT_A_So_PHB2TCMapping[1…M]</td>
<td></td>
</tr>
<tr>
<td>Sm/MT_A_So_MI_QoSEncodingMode[1…M]</td>
<td></td>
</tr>
</tbody>
</table>

• Processes:

A process diagram of this function is shown in Figure 11-10.
Figure 11-10/G.8121/Y.1381 – Sm/MT_A_So process diagram

- **TC/Label processes:**
  See 8.2.1.

- **Queuing process:**
  See 8.3.

- **MPLS-TP-specific GFP-F source process:**
  See 8.4.1.

- **Common GFP source process:**
  See 8.5.3.1/G.806. GFP channel multiplexing is not supported (CMuxActive=false).
– VC-m-specific GFP source process:
See 8.5.2.1/G.806. The GFP frames are mapped into the VC-m payload area according to 10.6/G.707/Y.1322.

– VC-m-specific source process:

V5[5-7] and K4[1]: Signal label information is derived directly from the Adaptation function type. The value for "GFP mapping" in Table 9-13/G.707/Y.1322 is placed in the K4[1] Extended Signal Label field as described in 8.2.3.2/G.783.

K4[2]: For Sm/MT_A_So with m = 11, 12, the K4[2] bit is sourced as all-zeros.
NOTE 1 – For Sm/MT_A_So with m = 11-X, 12-X, the K4[2] bit is undefined at the Sm-X_AP output of this function (as per clause 13/G.783).
NOTE 2 – For Sm/MT_A_So with m = 11, 11-X, 12, 12-X, 2, the K4[3-8], V5[1-4] and V5[8] bits are undefined at the Sm-X_AP output of this function (as per clause 13/G.783).

- Defects:
None.
- Consequent actions:
None.
- Defect correlations:
None.
- Performance monitoring:
Ffs.

11.1.3.2 VC-m to MPLS-TP adaptation sink function (Sm/MT_A_Sk)
This function extracts MT_CI information from the Sm_AI signal (m=11, 11-X, 12, 12-X), delivering MT_CI.
Data at the Sm_AP is a VC-m (m=11, 11-X, 12, 12-X) but with indeterminate POH bytes J2, V5[1-4], V5[8], as per [ITU-T G.707].

- Symbol:

\[ \text{Figure 11-11/G.8121/Y.1381 – Sm/MT_A_Sk symbol} \]
• **Interfaces:**

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sm_AP:</strong></td>
<td></td>
</tr>
<tr>
<td>Sm_AP Data</td>
<td></td>
</tr>
<tr>
<td>Sm_AP ClocK</td>
<td></td>
</tr>
<tr>
<td>Sm_AP FrameStart</td>
<td></td>
</tr>
<tr>
<td>Sm_AP TSF</td>
<td></td>
</tr>
<tr>
<td><strong>Sm/MT_A_Sk_MP:</strong></td>
<td></td>
</tr>
<tr>
<td>Sm/MT_A_Sk_MI_SCCType</td>
<td></td>
</tr>
<tr>
<td>Sm/MT_A_Sk_MI_Label[1…M]</td>
<td></td>
</tr>
<tr>
<td>Sm/MT_A_Sk_MI_LSPType[1…M]</td>
<td></td>
</tr>
<tr>
<td>Sm/MT_A_Sk_MI_CoS[1…M]</td>
<td></td>
</tr>
<tr>
<td>Sm/MT_A_Sk_MI_TC2PHBMapping[1…M]</td>
<td></td>
</tr>
<tr>
<td>Sm/MT_A_Sk_MI_QoSDecodingMode[1…M]</td>
<td></td>
</tr>
<tr>
<td>Sm/MT_A_Sk_MI_LCK_Period[1…M]</td>
<td></td>
</tr>
<tr>
<td>Sm/MT_A_Sk_MI_LCK_CoS[1…M]</td>
<td></td>
</tr>
<tr>
<td>Sm/MT_A_Sk_MI_Admin_State</td>
<td></td>
</tr>
<tr>
<td>Sm/MT_A_Sk_MI_AIS_Period[1…M]</td>
<td></td>
</tr>
<tr>
<td>Sm/MT_A_Sk_MI_AIS_CoS[1…M]</td>
<td></td>
</tr>
<tr>
<td>Sm/MT_A_Sk_MI_GAL_Enable[1…M]</td>
<td></td>
</tr>
</tbody>
</table>

**Sm/MT_A_Sk_MP:**

- Sm/MT_A_Sk_MI_AcSL
- Sm/MT_A_Sk_MI_AcEXI
- Sm/MT_A_Sk_MI_LastValidUPI
- Sm/MT_A_Sk_MI_cPLM
- Sm/MT_A_Sk_MI_cLFD
- Sm/MT_A_Sk_MI_cEXM
- Sm/MT_A_Sk_MI_cUPM

**Each MT_CP:**

- MT_CI_Data
- MT_CI_iPHB
- MT_CI_oPHB
- MT_CI_SSF
- MI_CI_LStack

**SCC_CP:**

- SCC_CI_Data
- SCC_CI_SSF

**Sm/MT_A_Sk_MP:**

- Sm/MT_A_Sk_MI_AcSL
- Sm/MT_A_Sk_MI_AcEXI
- Sm/MT_A_Sk_MI_LastValidUPI
- Sm/MT_A_Sk_MI_cPLM
- Sm/MT_A_Sk_MI_cLFD
- Sm/MT_A_Sk_MI_cEXM
- Sm/MT_A_Sk_MI_cUPM

• **Processes:**

A process diagram of this function is shown in Figure 11-12
Figure 11-12/G.8121/Y.1381 – Sm/MT_A_Sk process diagram

– Selector generation process:
See 8.6.1 The normal CI is blocked if Admin_State = LOCKED.

– AIS Insert process:
See 8.6.2. There is a single AIS Insert process for each MT.

– LCK generation process:
See 8.6.3. There is a single LCK Insert process for each MT.

– TC/Label processes:
See 8.2.2.

– Label Stack Copy process:
See 8.2.3.
– **MPLS-TP specific GFP-F sink process:**

  See 8.4.2.

– **Common GFP sink process:**

  See 8.5.3.2 in [ITU-T G.806]. GFP channel multiplexing is not supported (CMuxActive=false).

– **VC-m-specific GFP sink process:**

  See 8.5.2.2 in [ITU-T G.806]. The GFP frames are demapped from the VC-m payload area according to 10.6 in [ITU-T G.707].

– **VC-m-specific sink process:**

  **V5[5-7] and K4[1]**: The signal label is recovered from the extended signal label position as described in 8.2.3.2 in [ITU-T G.783] and 6.2.4.2 in [ITU-T G.806]. The signal label for "GFP mapping" in Table 9-13 in [ITU-T G.707] shall be expected. The accepted value of the signal label is also available at the Sm/MT_A_Sk_MP.

• **Defects:**

  dPLM – See 6.2.4.2 in [ITU-T G.806].
  dLFD – See 6.2.5.2 in [ITU-T G.806].
  dUPM – See 8.4.2.
  dEXM – See 6.2.4.4 in [ITU-T G.806].

• **Consequent actions:**

  The function shall perform the following consequent actions:
  aSSF ← AI_TSF or dPLM or dLFD or dUPM or dEXM
  aAIS ← AI_TSF or dPLM or dLFD or dUPM or dEXM

• **Defect correlations:**

  The function shall perform the following defect correlations to determine the most probable fault cause (see 6.4/G.806). This fault cause shall be reported to the EMF.
  cPLM ← dPLM and (not AI_TSF)
  cLFD ← dLFD and (not dPLM) and (not AI_TSF)
  cEXM ← dEXM and (not dPLM) and (not dLFD) and (not AI_TSF)
  cUPM ← dUPM and (not dEXM) and (not dPLM) and (not dLFD) and (not AI_TSF)

• **Performance monitoring:**

  Ffs.
11.1.4 LCAS-capable VC-m to MPLS-TP adaptation functions (Sm-X-L/MT_A; m=11, 12)

11.1.4.1 LCAS-capable VC-m to MPLS-TP Adaptation Source function (Sm-X-L/MT_A_So)

This function maps MT_CI information onto an Sm-X-L_AI signal (m=11, 12).

Data at the Sm-X-L_AP is a VC-m-X (m = 11, 12), having a payload as described in [ITU-T G.707], but with indeterminate POH bytes: J2, V5[1-4], V5[8].

- **Symbol:**

  ![Figure 11-13/G.8121/Y.1381 – Sm-X-L/MT_A_So symbol](image)

- **Interfaces:**

  **Table 11-7/G.8121/Y.1381 – Sm-X-L/MT_A_So interfaces**

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Each MT_CP:</td>
<td>Sm-X-L_AP:</td>
</tr>
<tr>
<td>MT_CI Data</td>
<td>Sm-X-L_AI_Data</td>
</tr>
<tr>
<td>MT_CI iPHB</td>
<td>Sm-X-L_AI_Clock</td>
</tr>
<tr>
<td>MT_CI oPHB</td>
<td>Sm-X-L_AI_FrameStart</td>
</tr>
<tr>
<td>SCC_CP:</td>
<td>Sm-X-L_AP:</td>
</tr>
<tr>
<td>SCC_CI_Data</td>
<td>Sm-X-L_AI_Data</td>
</tr>
<tr>
<td>Sm-X-L_AP:</td>
<td>Sm-X-L_AI_Clock</td>
</tr>
<tr>
<td>Sm-X-L_AI_XAT</td>
<td>Sm-X-L_AI_FrameStart</td>
</tr>
<tr>
<td>Sm-X-L_TP:</td>
<td>Sm-X-L_AP:</td>
</tr>
<tr>
<td>Sm-X-L_TI_Clock</td>
<td>Sm-X-L_AP:</td>
</tr>
<tr>
<td>Sm-X-L_TI_FrameStart</td>
<td>Sm-X-L_AP:</td>
</tr>
<tr>
<td>Sm-X-L/MT_A_So_MP:</td>
<td>Sm-X-L_AP:</td>
</tr>
<tr>
<td>Sm-X-L/MT_A_So_MI_SCCType</td>
<td>Sm-X-L_AP:</td>
</tr>
<tr>
<td>Sm-X-L/MT_A_So_MI_Label[1…M]</td>
<td>Sm-X-L_AP:</td>
</tr>
<tr>
<td>Sm-X-L/MT_A_So_MI_LSPType[1…M]</td>
<td>Sm-X-L_AP:</td>
</tr>
<tr>
<td>Sm-X-L/MT_A_So_MI_CoS[1…M]</td>
<td>Sm-X-L_AP:</td>
</tr>
<tr>
<td>Sm-X-L/MT_A_So_PHB2TCMapping[1…M]</td>
<td>Sm-X-L_AP:</td>
</tr>
<tr>
<td>Sm-X-L/MT_A_So_MI_QoSEncodingMode[1…M]</td>
<td>Sm-X-L_AP:</td>
</tr>
</tbody>
</table>

- **Processes:**

  A process diagram of this function is shown in Figure 11-14.
Figure 11-14/G.8121/Y.1381 – Sm-X-L/MT_A_So process diagram

The processes have the same definition as in 11.1.1.1.

- **Defects:**
  
  None.

- **Consequent actions:**
  
  None.

- **Defect correlations:**
  
  None.

- **Performance monitoring:**
  
  Ffs.
11.1.4.2 LCAS-capable VC-m to MPLS-TP adaptation sink function (Sm-X-L/MT_A_Sk)

This function extracts MT_CI information from the Sm-X-L_AI signal (m=11, 12), delivering MT_CI.

Data at the Sm-X-L_AP is a VC-m-Xv (m=11, 12) but with indeterminate POH bytes J2, V5[1-4], V5[8], as per [ITU-T G.707].

- **Symbol:**

  ![Figure 11-15/G.8121/Y.1381 – Sm-X-L/MT_A_Sk symbol](image)

- **Interfaces:**

  **Table 11-8/G.8121/Y.1381 – Sm-X-L/MT_A_Sk interfaces**

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sm-X-L_AP:</strong></td>
<td><strong>Each MT_CP:</strong></td>
</tr>
<tr>
<td>Sm-X-L_AI_Data</td>
<td>MT_CI_Data</td>
</tr>
<tr>
<td>Sm-X-L_AI_ClockK</td>
<td>MT_CI_iPHB</td>
</tr>
<tr>
<td>Sm-X-L_AI_FrameStart</td>
<td>MT_CI_oPHB</td>
</tr>
<tr>
<td>Sm-X-L_AI_TSF</td>
<td>MT_CI_SSF</td>
</tr>
<tr>
<td>Sm-X-L_AL_AR</td>
<td>MI_CI_LStack</td>
</tr>
<tr>
<td><strong>Sm-X-L/MT_A_Sk_MP:</strong></td>
<td><strong>SCC_CP:</strong></td>
</tr>
<tr>
<td>Sm-X-L/MT_A_Sk_MI_SCCType</td>
<td>SCC_CI_Data</td>
</tr>
<tr>
<td>Sm-X-L/MT_A_Sk_MI_Label[1…M]</td>
<td>SCC_CI_SSF</td>
</tr>
<tr>
<td>Sm-X-L/MT_A_Sk_MI_LSPType[1…M]</td>
<td></td>
</tr>
<tr>
<td>Sm-X-L/MT_A_Sk_MI_CoS[1…M]</td>
<td></td>
</tr>
<tr>
<td>Sm-X-L/MT_A_Sk_MI_TC2PHBMapping[1…M]</td>
<td></td>
</tr>
<tr>
<td>Sm-X-L/MT_A_Sk_MI_QoSDecodingMode[1…M]</td>
<td></td>
</tr>
<tr>
<td>Sm-X-L/MT_A_Sk_MI_LCK_Periode[1…M]</td>
<td></td>
</tr>
<tr>
<td>Sm-X-L/MT_A_Sk_MI_LCK_Cos[1…M]</td>
<td></td>
</tr>
<tr>
<td>Sm-X-L/MT_A_Sk_MI_Admin_State</td>
<td></td>
</tr>
<tr>
<td>Sm-X-L/MT_A_Sk_MI_AIS_Periode[1…M]</td>
<td></td>
</tr>
<tr>
<td>Sm-X-L/MT_A_Sk_MI_AIS_Cos[1…M]</td>
<td></td>
</tr>
<tr>
<td>Sm-X-L/MT_A_Sk_MI_GAL_Enable[1…M]</td>
<td></td>
</tr>
</tbody>
</table>

- **Sm-X-L/MT_A_Sk_MP:**

  Sm-X-L/MT_A_Sk_MI_AcSL
  Sm-X-L/MT_A_Sk_MI_AcEXI
  Sm-X-L/MT_A_Sk_MI_LastValidUPI
  Sm-X-L/MT_A_Sk_MI_cPLM
  Sm-X-L/MT_A_Sk_MI_cLFD
  Sm-X-L/MT_A_Sk_MI_cEXM
  Sm-X-L/MT_A_Sk_MI_cUPM
• Processes:

**Figure 11-16/G.8121/Y.1381 – Sm-X-L/MT_A_Sk process diagram**

See process diagram and process description in 11.1.1.2. The additional Sm-X-L_AI_XAR interface is not connected to any of the internal processes.

• Defects:
  
dPLM – See 6.2.4.2 in [ITU-T G.806].
  
dlFD – See 6.2.5.2 in [ITU-T G.806].
  
dUPM – See 8.4.2.
  
dEXM – See 6.2.4.4 in [ITU-T G.806].

• Consequent actions:
  
The function shall perform the following consequent actions:
aSSF $\leftarrow$ AI_TSF or dPLM or dLFD or dUPM or dEXM

aAIS $\leftarrow$ AI or dPLM or dLFD or dUPM or dEXM

**Defect correlations:**
The function shall perform the following defect correlations to determine the most probable fault cause (see 6.4 in [ITU-T G.806]). This fault cause shall be reported to the EMF.

cPLM $\leftarrow$ dPLM and (not AI_TSF)
cLFD $\leftarrow$ dLFD and (not dPLM) and (not AI_TSF)
cEXM $\leftarrow$ dEXM and (not dPLM) and (not dLFD) and (not AI_TSF)
cUPM $\leftarrow$ dUPM and (not dEXM) and (not dPLM) and (not dLFD) and (not AI_TSF)

**Performance monitoring:**

F/s.

11.2 OTH to MPLS-TP Adaptation function (O/MT_A)

11.2.1 ODUk to MPLS-TP Adaptation functions

11.2.1.1 ODUk to MPLS-TP adaptation source function (ODUkP/MT_A_So)
The ODUkP/MT_A_So function creates the ODUk signal from a free running clock. It maps the MT_CI information into the payload of the OPUk, adds OPUk Overhead (RES, PT) and default ODUk Overhead.

**Symbol:**

![Figure 11-17/G.8121/Y.1381 – ODUkP/MT_A_So symbol](image)

**Interfaces:**

| Table 11-9/G.8121/Y.1381 – ODUkP/MT_A_So interfaces |
|-----------------------------------------------|---|
| **Inputs**                                      | **Outputs**                        |
| Each MT_CP:                                     | ODUkP_AP:                           |
| MT_CI Data                                      | ODUkP_AP_Data                       |
| MT_CI iPHB                                      | ODUkP_AP_Clock                      |
| MT_CI oPHB                                      | ODUkP_AP_FrameStart                 |
| SCC_CP:                                        | ODUkP_AP_MultiFrameStart            |
Processes:
A process diagram of this function is shown in Figure 11-18.

![Figure 11-18/G.8121/Y.1381 – ODUkP/MT_A_So process diagram](image-url)

- **TC/Label processes:**
  See 8.2.1.
– Queuing process:
See 8.3.
– MPLS-TP-specific GFP-F source process:
See 8.4.1.
– Common GFP source process:
See 8.5.3.1 in [ITU-T G.806]. GFP channel multiplexing is not supported (CMuxActive=false).
– ODUk specific GFP source process:
See 8.5.2.1/G.806. The GFP frames are mapped into the ODUk payload area according to 17.3 in [ITU-T G.709].
– ODUk specific source process:

![Diagram](Image)

**Figure 11-19/G.8121/Y.1381 – ODUkP specific source processes**

**Clock and (Multi)Frame Start signal generation:** The function shall generate a local ODUk clock (ODUkP_AL_CK) of "239/(239 – k) * 4^{k-1} * 2 488 320 kHz ± 20 ppm" from a free running oscillator. The jitter and wander requirements as defined in Annex A in [ITU-T G.8251] (ODCa clock) apply.

The function shall generate the (multi)frame start reference signals Al_FS and Al_MFS for the ODUk signal. The Al_FS signal shall be active once per 122368 clock cycles. Al_MFS shall be active once every 256 frames.
PT: The payload type information is derived directly from the Adaptation function type. The value for “GFP mapping” shall be inserted into the PT byte position of the PSI overhead as defined in 15.9.2.1.1 in [ITU-TG.709].

RES: The function shall insert all-0's into the RES bytes.

All other bits of the ODUk overhead should be sourced as "0"s, except the ODUk-PM STAT field which should be set to the value "normal path signal" (001).

Defects:
None.

Consequent actions:
None.

Defect correlations:
None.

Performance monitoring:
Ffs.

11.2.1.2 ODUk to MPLS-TP adaptation sink function (ODUkP/MT_A_Sk)

The ODUkP/MT_A_Sk extracts MT_CI information from the ODUkP payload area. It extracts the OPUk Overhead (PT and RES) and monitors the reception of the correct payload type.

Symbol:

![Symbol diagram](image)

Figure 11-20/G.8121/Y.1381 – ODUkP/MT_A_Sk symbol
Interfaces:

Table 11-10/G.8121/Y.1381 – ODUkP/MT_A_Sk interfaces

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>ODUkP_AP:</td>
<td>Each MT_CP:</td>
</tr>
<tr>
<td>ODUkP_AP_CI_Data</td>
<td>MT_CI_Data</td>
</tr>
<tr>
<td>ODUkP_AP_CI_ScK</td>
<td>MT_CI_ScK</td>
</tr>
<tr>
<td>ODUkP_AP_CI_FrameStart</td>
<td>MT_CI_oPHB</td>
</tr>
<tr>
<td>ODUkP_AP_CI_MultiFrameStart</td>
<td>MT_CI_SSF</td>
</tr>
<tr>
<td>ODUkP_AP_CI_TSF</td>
<td>MT_CI_LStack</td>
</tr>
<tr>
<td>ODUkP/MT_A_Sk_MP:</td>
<td>SCC_CP:</td>
</tr>
<tr>
<td>ODUkP/MT_A_Sk_MI_Active</td>
<td>SCC_CI_Data</td>
</tr>
<tr>
<td>ODUkP/MT_A_Sk_MI_SCCType</td>
<td>SCC_CI_ScK</td>
</tr>
<tr>
<td>ODUkP/MT_A_Sk_MI_Label[1…M]</td>
<td>ODUkP/MT_A_Sk_MP:</td>
</tr>
<tr>
<td>ODUkP/MT_A_Sk_MI_LSPType[1…M]</td>
<td>ODUkP/MT_A_Sk_MI_AcPT</td>
</tr>
<tr>
<td>ODUkP/MT_A_Sk_MI_CoS[1…M]</td>
<td>ODUkP/MT_A_Sk_MI_AcEXI</td>
</tr>
<tr>
<td>ODUkP/MT_A_Sk_MI_TC2PHBMapping[1…M]</td>
<td>ODUkP/MT_A_Sk_MI_LastValidUPI</td>
</tr>
<tr>
<td>ODUkP/MT_A_Sk_MI_QoSDecodingMode[1…M]</td>
<td>ODUkP/MT_A_Sk_MI_cPLM</td>
</tr>
<tr>
<td>ODUkP/MT_A_Sk_MI_LCK_Peri0d[1…M]</td>
<td>ODUkP/MT_A_Sk_MI_cLFD</td>
</tr>
<tr>
<td>ODUkP/MT_A_Sk_MI_LCK_CoS[1…M]</td>
<td>ODUkP/MT_A_Sk_MI_cEXM</td>
</tr>
<tr>
<td>ODUkP/MT_A_Sk_MI_Admin_State</td>
<td>ODUkP/MT_A_Sk_MI_cUPM</td>
</tr>
<tr>
<td>ODUkP/MT_A_Sk_MI_AIS_Period[1…M]</td>
<td>ODUkP/MT_A_Sk_MI_GAL_Enable[1…M]</td>
</tr>
</tbody>
</table>
Processes:

A process diagram of this function is shown in Figure 11-21.

Figure 11-21/G.8121/Y.1381 – ODUkP/MT_A_Sk process diagram

- **Selector generation process:**
  See 8.6.1. The normal CI is blocked if Admin State = LOCKED.

- **AIS Insert process:**
  See 8.6.2. There is a single AIS Insert process for each MT.

- **LCK generation process:**
  See 8.6.3. There is a single LCK Insert process for each MT.

- **TC/Label processes:**
  See 8.2.2.
– **Label Stack Copy process:**
See 8.2.3.

– **MPLS-TP-specific GFP-F sink process:**
See 8.4.2.

– **Common GFP sink process:**
See 8.5.3.2 in [ITU-T G.806]. GFP channel multiplexing is not supported (CMuxActive=false).

– **ODUk specific GFP sink process:**
See 8.5.2.2 in [ITU-T G.806]. The GFP frames are demapped from the ODUk payload area according to 17.3 in [ITU-T G.709].

– **ODUk-specific sink process:**

![Diagram](image_url)

**Figure 11-22/G.8121/Y.1381 – ODUkP specific sink processes**

**PT:** The function shall extract the PT byte from the PSI overhead as defined in 8.7.1 in [ITU-T G.798]. The payload type value for "GFP mapping" in 15.9.2.1.1 in [ITU-T G.709] shall be expected. The accepted PT value is available at the MP (MI_AcPT) and is used for PLM defect detection.

**RES:** The value in the RES bytes shall be ignored.

**Defects:**
- dPLM – See 6.2.4.1 in [ITU-T G.798].
- dLFD – See 6.2.5.2 in [ITU-T G.806].
dEXM – See 6.2.4.4 in [ITU-T G.806].
dUPM – See 8.4.2.

**Consequent actions:**
The function shall perform the following consequent actions:

\[ aSSF \leftarrow \text{AI}_\text{TSF} \text{ or dPLM or dLFD or dUPM or dEXM} \]
\[ aAIS \leftarrow \text{AI}_\text{TSF} \text{ or dPLM or dLFD or dUPM or dEXM} \]

**Defect correlations:**
The function shall perform the following defect correlations to determine the most probable fault cause (see 6.4/G.806). This fault cause shall be reported to the EMF.

\[ c\text{PLM} \leftarrow \text{dPLM} \text{ and (not AI}_\text{TSF)} \]
\[ c\text{LFD} \leftarrow \text{dLFD} \text{ and (not dPLM) and (not AI}_\text{TSF)} \]
\[ c\text{EXM} \leftarrow \text{dEXM} \text{ and (not dPLM) and (not dLFD) and (not AI}_\text{TSF)} \]
\[ c\text{UPM} \leftarrow \text{dUPM} \text{ and (not dEXM) and (not dPLM) and (not dLFD) and (not AI}_\text{TSF)} \]

**Performance monitoring:**
\[ Ffs. \]

11.2.2 LCAS-capable ODU\(k\) to MPLS-TP Adaptation functions (ODU\(k\)P-X-L/MT_A; \(k=1,2,3\))

11.2.2.1 LCAS-capable ODU\(k\) to MPLS-TP adaptation source function (ODU\(k\)P-X-L/MT_A_So)

The ODU\(k\)P-X-L/MT_A_So function creates the ODU\(k\)-X-L signal from a free running clock. It maps the MT_CI information into the payload of the OPU\(k\)-Xv (\(k = 1, 2, 3\)), adds OPU\(k\)-Xv Overhead (RES, vcPT).
Symbol:

![Diagram of ODUkP-X-L/MT_A_So interfaces]

Figure 11-23/G.8121/Y.1381 – ODUkP-X-L/MT_A_So symbol

Interfaces:

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Each MT_CP:</strong></td>
<td><strong>ODUkP-X-L_AP:</strong></td>
</tr>
<tr>
<td>MT_CI_Data</td>
<td>ODUkP-X-L_AP_Data</td>
</tr>
<tr>
<td>MT_CI_iPHB</td>
<td>ODUkP-X-L_AP_iPHB</td>
</tr>
<tr>
<td>MT_CI_oPHB</td>
<td>ODUkP-X-L_AP_oPHB</td>
</tr>
<tr>
<td><strong>SCC_CP:</strong></td>
<td><strong>ODUkP-X-L_HHMP:</strong></td>
</tr>
<tr>
<td>SCC_CI_Data</td>
<td>ODUkP-X-L_HHMP_Data</td>
</tr>
<tr>
<td><strong>ODUkP-X-L_HHMP:</strong></td>
<td><strong>ODUkP-X-L_PHB:</strong></td>
</tr>
<tr>
<td>ODUkP-X-L_MT_A_So_MI_Active</td>
<td>ODUkP-X-L_PHB_Active</td>
</tr>
<tr>
<td>ODUkP-X-L_MT_A_So_MI_SCCType</td>
<td>ODUkP-X-L_PHB_SCCType</td>
</tr>
<tr>
<td>ODUkP-X-L_MT_A_So_MI_Label[1…M]</td>
<td>ODUkP-X-L_PHB_Label[1…M]</td>
</tr>
<tr>
<td>ODUkP-X-L_MT_A_So_MI_LSPType[1…M]</td>
<td>ODUkP-X-L_PHB_LSPType[1…M]</td>
</tr>
<tr>
<td>ODUkP-X-L_MT_A_So_PHB2TCMapping[1…M]</td>
<td>ODUkP-X-L_PHB_PHB2TCMapping[1…M]</td>
</tr>
<tr>
<td>ODUkP-X-L_MT_A_So_MI_QoSEncodingMode[1…M]</td>
<td>ODUkP-X-L_PHB_QoSEncodingMode[1…M]</td>
</tr>
</tbody>
</table>

Table 11-11/G.8121/Y.1381 – ODUkP-X-L/MT_A_So interfaces
Processes:
A process diagram of this function is shown in Figure 48.

Figure 11-24/G.8121/Y.1381 – ODUkP-X-L/MT_A_So process diagram

The processes have the same definition as in 11.2.1.1.

ODUkP-X-L specific source process:
Clock and (Multi)Frame Start signal generation: The function shall generate a local ODUk clock (ODUkP AI_CK) of "X AT * 239/(239 – k) * 4^(k-1) * 2 488 320 kHz ± 20 ppm" from a free running oscillator. The jitter and wander requirements as defined in Annex A in [ITU-T G.8251] (ODCa clock) apply.

The function shall generate the (multi)frame start reference signals AI_FS and AI_MFS for the ODUk signal. The AI_FS signal shall be active once per 122368 clock cycles. AI_MFS shall be active once every 256 frames.

vcPT: The payload type information is derived directly from the Adaptation function type. The value for “GFP mapping” shall be inserted into the vcPT byte position of the PSI overhead as defined in 18.1.2.2 in [ITU-T G.709].

RES: The function shall insert all-0's into the RES bytes.

Defects:
None.

Consequent actions:
None.

Defect correlations:
None.

Performance monitoring:
11.2.2.2 LCAS-capable ODUk to MPLS-TP adaptation sink function (ODUkP-X-L/MT_A_Sk)

The ODUkP-X-L/MT_A_Sk extracts MT_CI information from the ODUkP-Xv payload area. It extracts the OPUk-Xv Overhead (vcPT and RES) and monitors the reception of the correct payload type.

**Symbol:**

![Figure 11-26/G.8121/Y.1381 – ODUkP-X-L/MT_A_Sk symbol](image)

**Interfaces:**

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ODUkP-X-L_AP:</strong></td>
<td><strong>Each MT_CP:</strong></td>
</tr>
<tr>
<td>ODUkP-X-L_AI_Data</td>
<td>MT_CI Data</td>
</tr>
<tr>
<td>ODUkP-X-L_AI_ClocK</td>
<td>MT_CI iPHB</td>
</tr>
<tr>
<td>ODUkP-X-L_AI_FrameStart</td>
<td>MT_CI oPHB</td>
</tr>
<tr>
<td>ODUkP-X-L_AI_MultiFrameStart</td>
<td>MT_CI SSF</td>
</tr>
<tr>
<td>ODUkP-X-L_AI_TSF</td>
<td>MT_CI LStack</td>
</tr>
<tr>
<td>ODUkP-X-L_AI_Xaf</td>
<td><strong>SCC_CP:</strong></td>
</tr>
<tr>
<td>ODUkP-X-L_MT_A_Sk_MP:</td>
<td>SCC_CI Data</td>
</tr>
<tr>
<td>ODUkP-X-L_MT_A_Sk MI Active</td>
<td>SCC_CI SSF</td>
</tr>
<tr>
<td>ODUkP-X-L_MT_A_Sk MI SCCType</td>
<td></td>
</tr>
<tr>
<td>ODUkP-X-L_MT_A_Sk MI Label[1...M]</td>
<td></td>
</tr>
<tr>
<td>ODUkP-X-L_MT_A_Sk MI LSPType[1...M]</td>
<td></td>
</tr>
<tr>
<td>ODUkP-X-L_MT_A_Sk MI CoS[1...M]</td>
<td></td>
</tr>
<tr>
<td>ODUkP-X-L_MT_A_Sk MI TC2PHBMapping[1...M]</td>
<td></td>
</tr>
<tr>
<td>ODUkP-X-L_MT_A_Sk MI QoSDecodingMode[1...M]</td>
<td></td>
</tr>
<tr>
<td>ODUkP-X-L_MT_A_Sk MI _LCK_Period[1...M]</td>
<td></td>
</tr>
<tr>
<td>ODUkP-X-L_MT_A_Sk MI _LCK_CoS[1...M]</td>
<td></td>
</tr>
<tr>
<td>ODUkP-X-L_MT_A_Sk MI _Admin State</td>
<td></td>
</tr>
<tr>
<td>ODUkP-X-L_MT_A_Sk MI _AIS_Period[1...M]</td>
<td></td>
</tr>
<tr>
<td>ODUkP-X-L_MT_A_Sk MI _AIS_CoS[1...M]</td>
<td></td>
</tr>
<tr>
<td>ODUkP-X-L_MT_A_Sk MI _GAL_Enable[1...M]</td>
<td></td>
</tr>
</tbody>
</table>
Processes:

Figure 11-27/G.8121/Y.1381 – ODUkP-X-L/MT_A_Sk process diagram

See process diagram and process description in 11.2.1.2. The additional ODUkP-X-L.AI.XAR interface is not connected to any of the internal processes.

ODUkP-X-L specific sink process:
**Figure 11-28/G.8121/Y.1341 – ODUkP-X-L specific sink processes**

**PT:** The function shall extract the vcPT byte from the PSI overhead as defined in 8.7.3 in [ITU-T G.798]. The payload type value for “GFP mapping” in 18.1.2.2 in [ITU-T G.709] shall be expected. The accepted PT value is available at the MP (MI_AcPT) and is used for PLM defect detection.

**RES:** The value in the RES bytes shall be ignored.

**Defects:**
dVcPLM – See 6.2.4.2 in [ITU-T G.798].
dLFD – See 6.2.5.2 in [ITU-T G.806].
dUPM – See 8.4.2.
dEXM – See 6.2.4.4 in [ITU-T G.806].

**Consequent actions:**
The function shall perform the following consequent actions:
aSSF  $\Leftarrow$  AI_TSF or dVcPLM or dLFD or dUPM or dEXM
aAIS  $\Leftarrow$  AI_TSF or dVcPLM or dLFD or dUPM or dEXM

**Defect correlations:**
The function shall perform the following defect correlations to determine the most probable fault cause (see 6.4 in [ITU-T G.806]). This fault cause shall be reported to the EMF.
cVcPLM  $\Leftarrow$  dVcPLM and (not AI_TSF)
cLFD  $\Leftarrow$  dLFD and (not dVcPLM) and (not AI_TSF)
cEXM ↔ dEXM and (not dVcPLM) and (not dLFD) and (not AI_TSF)
cUPM ↔ dUPM and (not dEXM) and (not dVcPLM) and (not dLFD) and (not AI_TSF)

Performance monitoring:

11.3 PDH to MPLS-TP adaptation function (P/MT_A)

11.3.1 Pq to MPLS-TP Adaptation functions (Pq/MT_A; q = 11s, 12s, 31s, 32e)

11.3.1.1 Pq to MPLS-TP Adaptation Source function (Pq/MT_A_So)

This function maps MT_CI information onto a Pq_AI signal (q = 11s, 12s, 31s, 32e).

Data at the Pq_AP is a Pq (q = 11s, 12s, 31s, 32e), having a payload as described in [ITU-T G.7043] with a value of N=1. The VLI byte is reserved and not used for payload data.

Symbol

![Diagram](image-url)

Figure 11-29/G.8121/Y.1381 – Pq/MT_A_So symbol
## Interfaces

Table 11-13/G.8121/Y.1381: Pq/MT_A_So interfaces

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Each MT CP:</td>
<td>Pq_AP:</td>
</tr>
<tr>
<td>MT_CI_Data</td>
<td>Pq_AI_Data</td>
</tr>
<tr>
<td>MT_CI_iPHB</td>
<td>Pq_AI_Clock</td>
</tr>
<tr>
<td>MT_CI_oPHB</td>
<td>Pq_AI_FrameStart</td>
</tr>
<tr>
<td>SCC_CP:</td>
<td></td>
</tr>
<tr>
<td>SCC_CI_Data</td>
<td></td>
</tr>
<tr>
<td>Pq TP:</td>
<td></td>
</tr>
<tr>
<td>Pq_TI_Clock</td>
<td></td>
</tr>
<tr>
<td>Pq_TI_FrameStart</td>
<td></td>
</tr>
<tr>
<td>Pq/MT_A_So MP:</td>
<td></td>
</tr>
<tr>
<td>Pq/MT_A_So_MI_SCCType</td>
<td></td>
</tr>
<tr>
<td>Pq/MT_A_So_MI_Label[1…M]</td>
<td></td>
</tr>
<tr>
<td>Pq/MT_A_So_MI_LSPType[1…M]</td>
<td></td>
</tr>
<tr>
<td>Pq/MT_A_So_MI_CoS[1…M]</td>
<td></td>
</tr>
<tr>
<td>Pq/MT_A_So_PHB2TCMapping[1…M]</td>
<td></td>
</tr>
<tr>
<td>Pq/MT_A_So_MI_QoSEncodingMode[1…M]</td>
<td></td>
</tr>
</tbody>
</table>
Processes

A process diagram of this function is shown in Figure 11-30.

Figure 11-30/G.8121/Y.1381 – Pq/MT_A_So process diagram

**TC/Label processes:**
See 8.2.1.

**Queuing process:**
See 8.3.

**MPLS-TP-specific GFP-F source process:**
See 8.4.1.

**Common GFP source process:**
See 8.5.3.1 in [ITU-T G.806]. GFP channel multiplexing is not supported (CMuxActive=false).
Pq specific GFP source process:
See 8.5.2.1 in [ITU-T G.806]. The GFP frames are mapped into the Pq payload area according to [ITU-T G.8040].

Pq specific source process:
Note: the VLI byte is fixed stuff equal to 0x00 at the Pq_AP output of this function.

P31s specific:

MA: Signal label information is derived directly from the Adaptation function type. The value for “GFP mapping” in clause 2.1/G.832 is placed in the Payload Type field of the MA byte.

Defects
None.

Consequent actions
None.

Defect correlations
None.

Performance monitoring
Ffs.

11.3.1.2 Pq to MPLS-TP Adaptation Sink function (Pq/MT_A_Sk)
This function extracts MT_CI information from the Pq_AI signal (q = 11s, 12s, 31s, 32e), delivering MT_CI.

Data at the Pq_AP is a Pq (q = 11s, 12s, 31s, 32e), having a payload as described in [ITU-T G.7043] with a value of N=1. The VLI byte is reserved and not used for payload data.

Symbol

![Symbol diagram]

Figure 11-31/G.8121/Y.1381 – Pq/MT_A_Sk symbol
Interfaces

Table 11-14/G.8121/Y.1381: Pq/MT_A_Sk interfaces

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pq_AP:</td>
<td>Each MT_CP:</td>
</tr>
<tr>
<td>Pq_AI_Data</td>
<td>MT_CI_Data</td>
</tr>
<tr>
<td>Pq_AI_Clock</td>
<td>MT_CI_iPHB</td>
</tr>
<tr>
<td>Pq_AI_FrameStart</td>
<td>MT_CI_oPHB</td>
</tr>
<tr>
<td>Pq_AI_TSF</td>
<td>MT_CI_SSF</td>
</tr>
<tr>
<td></td>
<td>MT_CI_LStack</td>
</tr>
<tr>
<td>Pq/MT_A_Sk_MP:</td>
<td>SCC_CP:</td>
</tr>
<tr>
<td>Pq/MT_A_Sk_MI_SCCType</td>
<td>SCC_CI_Data</td>
</tr>
<tr>
<td>Pq/MT_A_Sk_MI_Label[1…M]</td>
<td>SCC_CI_SSF</td>
</tr>
<tr>
<td>Pq/MT_A_Sk_MI_LSPType[1…M]</td>
<td>Pq/MT_A_Sk_MP:</td>
</tr>
<tr>
<td>Pq/MT_A_Sk_MI_CoS[1…M]</td>
<td>Pq/MT_A_Sk_MI_AcSL</td>
</tr>
<tr>
<td>Pq/MT_A_Sk_MI_TC2PHBMapping[1…M]</td>
<td>Pq/MT_A_Sk_MI_AcEXI</td>
</tr>
<tr>
<td>Pq/MT_A_Sk_MI_QoSDecodingMode[1…M]</td>
<td>Pq/MT_A_Sk_MI_LastValidUPI</td>
</tr>
<tr>
<td>Pq/MT_A_Sk_MI_LCK_Period[1…M]</td>
<td>Pq/MT_A_Sk_MI_cPLM</td>
</tr>
<tr>
<td>Pq/MT_A_Sk_MI_LCK_CoS[1…M]</td>
<td>Pq/MT_A_Sk_MI_cLFD</td>
</tr>
<tr>
<td>Pq/MT_A_Sk_MI_Admin_State</td>
<td>Pq/MT_A_Sk_MI_cEXM</td>
</tr>
<tr>
<td>Pq/MT_A_Sk_MI_AIS_Period[1…M]</td>
<td>Pq/MT_A_Sk_MI_cUPM</td>
</tr>
<tr>
<td>Pq/MT_A_Sk_MI_AIS_CoS[1…M]</td>
<td></td>
</tr>
<tr>
<td>Pq/MT_A_Sk_MI_GAL_Enable [1…M]</td>
<td></td>
</tr>
</tbody>
</table>

Processes

A process diagram of this function is shown in Figure 11-32.
**MPLS-TP Specific GFP-F Processes**

- **Selector generation process:**
  See 8.6.1 The normal CI is blocked if Admin State = LOCKED.

- **AIS Insert process:**
  See 8.6.2. There is a single AIS Insert process for each MT.

- **LCK generation process:**
  See 8.6.3. There is a single LCK Insert process for each MT.

**TC/Label processes:**
See 8.2.2.

- **Label Stack Copy process:**
  See 8.2.3.

**MPLS-TP specific GFP-F sink process:**
See 8.4.2/G.8121/Y.1381.
**Common GFP sink process:**
See 8.5.3.1/G.806. GFP channel multiplexing is not supported (CMuxActive=false).

**Pq specific GFP sink process:**
See 8.5.2.1/G.806. The GFP frames are demapped from the Pq payload area according to G.8040/Y.1340.

**Pq specific sink process:**
Note: the VLI byte at the Pq_AP input of this function is ignored.

**P31s specific:**
MA: The signal label is recovered from the Payload Type field in the MA byte as per 6.2.4.2 in [ITU-T G.806]. The signal label for “GFP mapping” in clause 2.1 in [ITU-T G.832] shall be expected. The accepted value of the signal label is also available at the P31s/ETH_A_Sk_MP.

**Defects**
dPLM – See 6.2.4.2 in [ITU-T G.806].
dLFD – See 6.2.5.2 in [ITU-T G.806]
dUPM - See 8.4.2/G.8121/Y.1381
dEXM – See 6.2.4.4 in [ITU-T G.806].
Note: dPLM is only defined for q = 31s. dPLM is assumed to be false for q = 11s, 12s, 32e.

**Consequent actions**
The function shall perform the following consequent actions:
aSSF ← AI_TSF or dPLM or dLFD or dUPM or dEXM
aAIS ← AI_TSF or dPLM or dLFD or dUPM or dEXM

**Defect correlations**
The function shall perform the following defect correlations to determine the most probable fault cause (see 6.4/G.806). This fault cause shall be reported to the EMF.
cPLM ← dPLM and (not AI_TSF)
cLFD ← dLFD and (not dPLM) and (not AI_TSF)
cEXM ← dEXM and (not dPLM) and (not dLFD) and (not AI_TSF)
cUPM ← dUPM and (not dEXM) and (not dPLM) and (not dLFD) and (not AI_TSF)

**Performance monitoring**
Ffs.

11.3.2 LCAS-capable Pq to MPLS-TP Adaptation functions (Pq-X-L/MT_A; q=11s, 12s, 31s, 32e)

11.3.2.1 LCAS-capable Pq to MPLS-TP Adaptation Source function (Pq-X-L/MT_A_So)
This function maps MT_CI information onto an Pq-X-L_AI signal (q=11s, 12s, 31s, 32e).
Data at the Pq-X-L AP is a Pq-X (q = 11s, 12s, 31s, 32e), having a payload as described in [ITU-T G.7043].
Symbol

![Symbol Diagram]

**Figure 11-33/G.8121/Y.1381 – Pq-X-L/MT_A_So symbol**

**Interfaces**

<table>
<thead>
<tr>
<th>Table 11-15/G.8121/Y.1381: Pq-X-L/MT_A_So interfaces</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Inputs</strong></td>
</tr>
<tr>
<td>Each MT_CP:</td>
</tr>
<tr>
<td>MT_CI Data</td>
</tr>
<tr>
<td>MT_CI iPHB</td>
</tr>
<tr>
<td>MT_CI oPHB</td>
</tr>
<tr>
<td>SCC_CI Data</td>
</tr>
<tr>
<td>Pq-X-L AP:</td>
</tr>
<tr>
<td>Pq-X-L_AL_XAT</td>
</tr>
<tr>
<td>Pq-X-L_AL_TP:</td>
</tr>
<tr>
<td>Pq-X-L_AL_XAT</td>
</tr>
<tr>
<td>Pq-X-L/MT_A_So MP:</td>
</tr>
<tr>
<td>Pq-X-L/MT_A_So_MI_SCCType</td>
</tr>
<tr>
<td>Pq-X-L/MT_A_So_MI_Label[1…M]</td>
</tr>
<tr>
<td>Pq-X-L/MT_A_So_MI_LSPType[1…M]</td>
</tr>
<tr>
<td>Pq-X-L/MT_A_So_MI_CoS[1…M]</td>
</tr>
<tr>
<td>Pq-X-L/MT_A_So_MI_PHB2TCMapping[1…M]</td>
</tr>
<tr>
<td>Pq-X-L/MT_A_So_MI_QoSEncodingMode[1…M]</td>
</tr>
</tbody>
</table>

**Processes**

A process diagram of this function is shown in Figure 11-34.
The processes have the same definition as in 11.1.1.1.

**Defects**
*None.*

**Consequent actions**
*None.*

**Defect correlations**
*None.*

**Performance monitoring**
Ffs.

11.3.2.2 LCAS-capable Pq to MPLS-TP Adaptation Sink function (Pq-X-L/MT_A_Sk)
This function extracts MT_CI information from the Pq-X-L_AI signal (q = 11s, 12s, 31s, 32e), delivering MT_CI.

Data at the Pq-X-L_AP is a Pq-Xv (q = 11s, 12s, 31s, 32e), having a payload as described in [ITU-T G.7043].

**Symbol**

![Symbol Diagram](image)

**Interfaces**

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Pq-X-L_AP:</strong></td>
<td>Each MT_CP:</td>
</tr>
<tr>
<td>Pq-X-L_AI_Data</td>
<td>MT_CI_Data</td>
</tr>
<tr>
<td>Pq-X-L_AI_ClocK</td>
<td>MT_CI_iPHB</td>
</tr>
<tr>
<td>Pq-X-L_AI_FrameStart</td>
<td>MT_CI_oPHB</td>
</tr>
<tr>
<td>Pq-X-L_AI_TSF</td>
<td>MT_CI_SSF</td>
</tr>
<tr>
<td>Pq-X-L_AI_Xat</td>
<td>MT_CI_LStack</td>
</tr>
<tr>
<td><strong>Pq-X-L/MT_A_Sk_MP:</strong></td>
<td>SCC CP:</td>
</tr>
<tr>
<td>Pq-X-L/MT_A_Sk_MI_SCCType</td>
<td>SCC_CI_Data</td>
</tr>
<tr>
<td>Pq-X-L/MT_A_Sk_MI_Label[1…M]</td>
<td>SCC_CI_SSF</td>
</tr>
<tr>
<td>Pq-X-L/MT_A_Sk_MI_LSPType[1…M]</td>
<td></td>
</tr>
<tr>
<td>Pq-X-L/MT_A_Sk_MI_CoS[1…M]</td>
<td></td>
</tr>
<tr>
<td>Pq-X-L/MT_A_Sk_MI_TC2PHBMapping[1…M]</td>
<td>Pq-X-L/MT_A_Sk_MP:</td>
</tr>
<tr>
<td>Pq-X-L/MT_A_Sk_MI_QoSDecodingMode[1…M]</td>
<td>Pq-X-L/MT_A_Sk_MI_AcSL</td>
</tr>
<tr>
<td>Pq-X-L/MT_A_Sk_MI_LCK_Period[1…M]</td>
<td>Pq-X-L/MT_A_Sk_MI_AcEXI</td>
</tr>
<tr>
<td>Pq-X-L/MT_A_Sk_MI_LCK_P[1…M]</td>
<td>Pq-X-L/MT_A_Sk_MI_LastValidUPI</td>
</tr>
<tr>
<td>Pq-X-L/MT_A_Sk_MI_Admin_State</td>
<td>Pq-X-L/MT_A_Sk_MI_cPLM</td>
</tr>
<tr>
<td>Pq-X-L/MT_A_Sk_MI_AIS_Period[1…M]</td>
<td>Pq-X-L/MT_A_Sk_MI_cLFD</td>
</tr>
<tr>
<td>Pq-X-L/MT_A_Sk_MI_AIS_P[1…M]</td>
<td>Pq-X-L/MT_A_Sk_MI_cEXM</td>
</tr>
<tr>
<td>Pq-X-L/MT_A_Sk_MI_GAL_Enable[1…M]</td>
<td>Pq-X-L/MT_A_Sk_MI_cUPM</td>
</tr>
</tbody>
</table>
Processes

Figure 11-36/G.8121/Y.1381 – Pq-X-L/MT_A_Sk process diagram

See process diagram and process description in 11.1.1.2. The additional Pq-X-L_AI_XAR interface is not connected to any of the internal processes.

Defects

dPLM – See 6.2.4.2 in [ITU-T G.806].
dLFD – See 6.2.5.2 in [ITU-T G.806].
dUPM - See 8.4.2

dEXM – See 6.2.4.4 in [ITU-T G.806].

Note: dPLM is only defined for q = 31s. dPLM is assumed to be false for q = 11s, 12s, 32e.

Consequent actions

The function shall perform the following consequent actions:

aSSF $\leftarrow$ AI_TSF or dPLM or dLFD or dUPM or dEXM
aAIS ← AI_TSF or dPLM or dLFD or dUPM or dEXM

**Defect correlations**

The function shall perform the following defect correlations to determine the most probable fault cause (see 6.4/G.806). This fault cause shall be reported to the EMF.

cPLM ← dPLM and (not AI_TSF)
cLFD ← dLFD and (not dPLM) and (not AI_TSF)
cEXM ← dEXM and (not dPLM) and (not dLFD) and (not AI_TSF)
cUPM ← dUPM and (not dEXM) and (not dPLM) and (not dLFD) and (not AI_TSF)

**Performance monitoring**

Ffs.

**11.4 Ethernet to MPLS-TP adaptation function**

FFS
Appendix I

Examples of processing of packets with expired TTL

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

MPLS-TP packets received with an expired TTL shall not be forwarded. However, MPLS-TP OAM packets received with an expired TTL can be processed and their processing can be happen at different locations (i.e., from different atomic functions) within an MPLS-TP Equipment.

The proper behavior depends on the MPLS-TP connection configuration within the node. The following examples are considered and described:
- Intermediate node with no MIPs
- Intermediate node – interface MIPs
- Intermediate node – node MIP
- Terminating Node - Down MEP or node MEP
- Terminating Node - Up MEP (with interface MIP)

NOTE – As indicated in clause 9.4.2.2.2, the MI_DS_MP_Type parameter should be properly configured by the EMF and not exposed to the operator as a configuration parameter of the NE Management. The examples described in this appendix provides guidelines on how the EMF can properly configure the MI_DS_MP_Type.

Figure I.1 describes the behavior of an intermediate node with no MIPs using the atomic functions defined in this Recommendation:

![Diagram](image)

Figure I.1 – Intermediate node with no MIPs

The Server/MT_A_Sk is connected to the MT_C via an MT_CP. Therefore the TTL Decrement Process, as defined in clause 8.2.2, will discard all the MPLS-TP packets (user data or OAM) that are received with an expired TTL.

Figure I.2 describes the behavior of an intermediate node supporting per-interface MIPs using the atomic functions defined in this Recommendation:
The Server/MT_A_Sk is connected to ingress MIP via an MT_TCP. Therefore the TTL Decrement Process, as defined in clause 8.2.2, will forward all the MPLS-TP packets (user data or OAM) that are received with an expired TTL to the ingress MIP.

The MTDi_TT_Sk atomic function within the ingress MIP will process all the MPLS-TP OAM packets received with an expired TTL and targeted to the ingress MIP.

The TTL check process in the MTDi/MT_A_Sk within the ingress MIP, as defined in clause 9.4.2.2.2, is properly configured by the EMF (MI_DS_MP_Type=MIP) to drop all the MPLS-TP user data packets received with an expired TTL and to forward all the MPLS-TP OAM packets received with an expired TTL together (i.e., with fate share) with all the MPLS-TP packets received with an non-expired TTL.

These packets are forwarded up to the egress MIP where the MTDi_TT_Sk atomic function will process all the MPLS-TP OAM packets received with an expired TTL and targeted to the egress MIP.

The TTL check process in the MTDi/MT_A_Sk within the egress MIP, as defined in clause 9.4.2.2.2, is properly configured by the EMF (MI_DS_MP_Type=none) to drop all the MPLS-TP packets received with an expired TTL. Although MPLS-TP user data packets with an expired TTL will never arrive at this point, this check will ensure also that any MPLS-TP OAM packet with an expired TTL is not forwarded.

Figure I.3 describes the behavior of an intermediate node with a per-node MIP using the atomic functions defined in this Recommendation. The per-node MIP is modeled as being composed by two half-MIPs on each side of the MT_C.
Bibliography

[IETF b-RFC 6378] IETF RFC 6378 *MPLS Transport Profile (MPLS-TP) Linear Protection.*