Question(s):	**9**, 10, 12, 14	**Meeting, date:**	Beijing, September 5 – 9, 2011
Study Group:	15 **Working Party:** 3	**Intended type of document**(R-C-TD) :	WD44r6
Source:	Editors G.8121		
Title:	Draft new G.8121		
Contact:	Yuji Tochio Fujitsu Japan		Tel: +81-44-754-8829 Fax: +81-44-754-2741 Email: tochio@jp.fujitsu.com
Contact:	Huub van Helvoort Huawei Technologies Co., Ltd. P.R.China		Tel: +31 649248936 Fax: Email: hhelvoort@huawei.com

Please don't change the structure of this table, just insert the necessary information.

Abstract

This document contains the consolidation of all WDs to new G.8121 as proposed in WD04.

Update from R3 is:

- Add new subclauses 8.2.3 for LStack (WD20)
- Add AI_LStack and CI_LStack to some functions
- Update input/output parameters and figures like P→CoS
  - Figures not updated are shown by yellow with notes
- Change the titles

Formatted: Bulleted + Level: 1 + Aligne at: 0 cm + Indent at: 0.63 cm

Formatted: Bulleted + Level: 2 + Aligne at: 0.74 cm + Indent at: 1.48 cm

Formatted: Highlight

Formatted: Bulleted + Level: 1 + Aligne at: 0 cm + Indent at: 0.63 cm

Formatted: Space Before: 6 pt, Snap t grid, Tab stops: 1.19 cm, Left + Not at cm

Drafting status

Clause	Title	WDs, note
1	**Scope**	
2	**References**	WD12
3	**Definitions**	
4	**Abbreviations**	
5	**Conventions**	
6	**Supervision**	
6.1	**Defects**	WD19 for Table 6-1, 2 and Figure 6-x WD37 proposed full update
6.2	**Consequent actions**	
6.3	**Defect correlations**	
6.4	**Performance filters**	
7	**Information flow across reference points**	
8	**MPLS-TP processes**	WD20 (General comments to clause 8)
8.1	**G-ACh Process**	Updated per WD04, i.e. only ex-8.1.1 is left. Note that some proposed updates in other WDs requires to update this clause.
8.2	**TC/Label processes**	WD20
8.3	**Queuing process**	
8.4	**MPLS-TP-specific GFP-F processes**	
8.5	**Control Word (CW) processes**	
8.6	**OAM related Processes used by Server adaptation functions**	WD13
8.7	**OAM related Processes used by adaptation functions**	
8.7.1	MCC/SCC Insert Process	WD09
8.7.2	MCC/SCC Extract Process	
8.7.3	APS Insert Process	WD14 (with merging them into single 8.7.3)
8.7.4	APS Extract Process	
8.7.5	CSF Insert Process	WD15 (with merging them into single 8.7.5)
8.7.6	CSF Extract Process	
8.8	**Pro-active and on-demand OAM related Processes**	WD16
9	**MPLS-TP layer functions**	WD21 (General comments to clause 9)
9.1	**Connection Functions (MT_C)**	
9.2	**Termination functions**	
9.2.1	MPLS-TP Trail Termination function (MT_TT)	WD07, *WD17*
9.3	**Adaptation functions**	
9.3.1	MPLS-TP to MPLS-TP adaptation function (MT/MT_A)	
9.4	**MT Diagnostic Function**	
9.4.1	MT Diagnostic Trail Termination Functions for MEPs	WD08~~7~~, *WD17*
9.4.2	MT Diagnostic Flow Termination Functions for MIPs	WD17
10	**MPLS-TP to Non-MPLS-TP client adaptation functions**	
10.1	MPLS-TP to ETH adaptation function (MT/ETH_A)	*WD21,* WD22
10.2	MPLS-TP to SCC and MCC Adaptation functions	WD10 (New from WD04)
11	**Non-MPLS-TP Server to MPLS-TP adaptation functions**	WD18, WD23
11.1	SDH to MPLS-TP adaptation function (S/MT_A)	
11.2	OTH to MPLS-TP Adaptation function (O/MT_A)	
11.3	PDH to MPLS-TP adaptation function (P/MT_A)	
11.4	ETY to MPLS-TP adaptation function (ETY/MT_A)	
Appendix 1	Examples of processing of packets with expired TTL	WD17 (New from WD04)

International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

G.8121/Y.1381

(03/2006)
Corrigendum (12/2006)
Amendment (10/2007)

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

Ethernet over Transport aspects – MPLS over Transport aspects

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

Internet protocol aspects – Transport

Characteristics of MPLS-TP equipment functional blocks

Editors version

Formatted

ITU-T Recommendation G.8121/Y.1381

International Telecommunication Union

ITU-T

ITU-T G-SERIES RECOMMENDATIONS

TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS

INTERNATIONAL TELEPHONE CONNECTIONS AND CIRCUITS	G.100–G.199
GENERAL CHARACTERISTICS COMMON TO ALL ANALOGUE CARRIER-TRANSMISSION SYSTEMS	G.200–G.299
INDIVIDUAL CHARACTERISTICS OF INTERNATIONAL CARRIER TELEPHONE SYSTEMS ON METALLIC LINES	G.300–G.399
GENERAL CHARACTERISTICS OF INTERNATIONAL CARRIER TELEPHONE SYSTEMS ON RADIO-RELAY OR SATELLITE LINKS AND INTERCONNECTION WITH METALLIC LINES	G.400–G.449
COORDINATION OF RADIOTELEPHONY AND LINE TELEPHONY	G.450–G.499
TRANSMISSION MEDIA CHARACTERISTICS	G.600–G.699
DIGITAL TERMINAL EQUIPMENTS	G.700–G.799
DIGITAL NETWORKS	G.800–G.899
DIGITAL SECTIONS AND DIGITAL LINE SYSTEM	G.900–G.999
QUALITY OF SERVICE AND PERFORMANCE – GENERIC AND USER-RELATED ASPECTS	G.1000–G.1999
TRANSMISSION MEDIA CHARACTERISTICS	G.6000–G.6999
DATA OVER TRANSPORT – GENERIC ASPECTS	G.7000–G.7999
ETHERNET OVER TRANSPORT ASPECTS	G.8000–G.8999
General aspects	G.8000–G.8099
MPLS over Transport aspects	**G.8100–G.8199**
Quality and availability targets	G.8200–G.8299
ACCESS NETWORKS	G.9000–G.9999

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

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

ITU-T Recommendation G.8121/Y.1381 was approved on 29 March 2006 by ITU-T Study Group 15 (2005-2008) under the ITU-T Recommendation A.8 procedure.

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.

CONTENTS

ITU-T Recommendation G.8121/Y.1381

Characteristics of MPLS-TP equipment functional blocks

1 Scope

[Editor's note: Need to be updated in terms of protocol neutral per WD04. Contributions invited]

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

This Recommendation is compliant with the transport profile of MPLS as defined by the IETF. In the event of a misalignment in MPLS-TP related architecture, framework, and protocols between this ITU-T Recommendation and the referenced IETF RFCs, the 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 Recs G.806 (Conventions and Generic Equipment Functions), G.798 (OTN functions), G.783 (SDH functions), G.705 (PDH functions), G.781 (Synchronization functions), I.732 (ATM functions), G.8021/Y.1341 (ETH functions), G.7710/Y.1701, G.784 and G.874 (Management functions). This Recommendation also follows the principles defined in ITU-T Rec. G.805.

[Note: Recommendations marked grey are not covered at this of time. Proposed to remove]

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. It is noted that this recommendation only defines Ethernet for the client of MPLS-TP as MT/ETH adaptation function.

[Ed note:No contributions received to address BFD based processes. Contribution are invited]

Formatted: Highlight

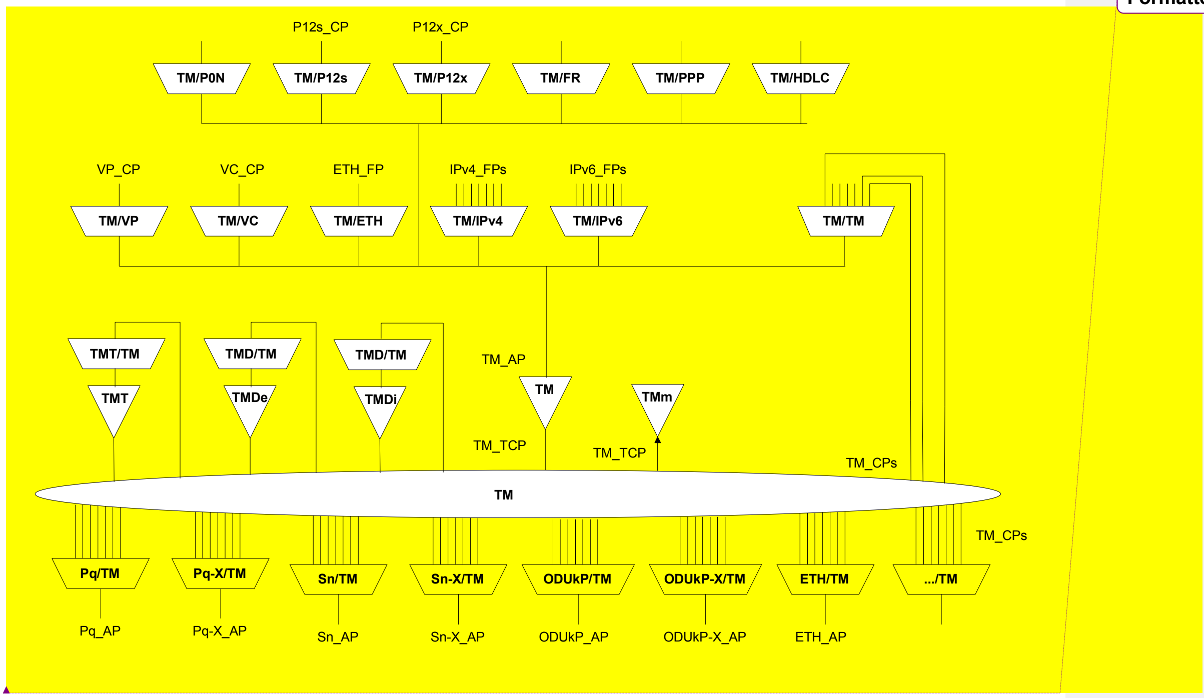
Formatted: Highlight

Formatted: Highlight

Formatted: Highlight

Formatted: Highlight

Formatted: Highlight



Formatted: Highlight

Figure 1/G.8121/Y.1381 – MPLS-TP atomic functions
[Replace TM by MT. Needs to clarify what is currently defined]

Formatted: Highlight

2 References

[Note: It will be updated in September time after the relevant documents such as IETF RFCs/I-Ds are available/updated / Need to be updated per current template]

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

– ITU-T Recommendation G.705 (2000), *Characteristics of plesiochronous digital hierarchy (PDH) equipment functional blocks*.

– ITU-T Recommendation G.707/Y.1322 (2003), *Network node interface for the synchronous digital hierarchy (SDH).*

– ITU-T Recommendation G.709/Y.1331 (2009), *Interfaces for the Optical Transport Network (OTN).*

– ITU-T Recommendation G.780/Y.1351 (2004), *Terms and definitions for synchronous digital hierarchy (SDH) networks.*

– ITU-T Recommendation G.783 (2006), *Characteristics of synchronous digital hierarchy (SDH) equipment functional blocks.*

– ITU-T Recommendation G.798 (2010), *Characteristics of optical transport network hierarchy equipment functional blocks*

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

– ITU-T Recommendation G.806 (2009), *Characteristics of transport equipment – Description methodology and generic functionality.*

– ITU-T Recommendation G.808.1 (2010), *Generic protection switching – Linear trail and subnetwork protection.*

– ITU-T Recommendation G.870/Y.1352 (2004), *Terms and definitions for Optical Transport Networks (OTN).*

– ITU-T Recommendation G.8101/Y.1355 (2010), *Terms and definitions for MPLS transport profile* .

– ITU-T Recommendation G.8110/Y.1370 (2005), *MPLS layer network architecture*.

– ITU-T Recommendation G.8110.1/Y.1370.1 (2006), *Architecture of MPLS-TP (MPLS-TP) layer network*.

– ITU-T Recommendation G.7042/Y.1305 (2006), *Link capacity adjustment scheme (LCAS) for virtual concatenated signals.*

– ITU-T Recommendation G.8021/Y.1341 (2010), *Characteristics of Ethernet transport network equipment functional blocks.*

– ITU-T Recommendation Y.1415 (2005), *Ethernet-MPLS network interworking – User plane interworking*.

–

- ITU-T Recommendation G.7043/Y.1343 (2004), *Virtual Concatenation of Plesiochronous Digital Hierarchy (PDH) signals.*
- ITU-T Recommendation G.8040/Y.1340 (2005), *GFP frame mapping into Plesiochronous Digital Hierarchy (PDH)*
- ITU-T Recommendation G.7712/Y.1703 (2010), *Architecture and specification of data communication network*
- ITU-T Recommendation G.8113.1 (201x), ~~*Architecture and specification of data communication network*~~ [put the title of G.8113.1] *[Note: add when approved]*
- ITU-T Recommendation G.8131/Y.1382 (201x), *Linear protection switching for MPLS-TP networks*. *[Note: add when revised G.8131 is approved]*
- ITU-T Draft Recommendation G.8132/Y.1383 (201x), *Ring protection switching for MPLS-TP networks*. *[add when G.8132 is approved]*

[Note: needs to add RFC5586 for GAL, RFC 4448 and RFC 4720 for Ethernet as client]

[ed note: these RFCs, whether keep or remove, will be reviewed after the text completed

~~*WD 12 proposes to add:*~~

[IETF RFC xxxx] IETF RFC ietf-mpls-tp-oam-framework, *Operations, Administration and Maintenance Framework for MPLS-based Transport Networks*

[IETF RFC xxxx] IETF RFC ietf-mpls-tp-cc-cv-rdi, *Proactive Connectivity Verification, Continuity Check and Remote Defect Indication for MPLS-TP.*

[IETF RFC xxxx] IETF RFC ietf-mpls-tp-fault, *MPLS Fault Management OAM.*

[IETF RFC xxxx] IETF RFC ietf-mpls-tp-on-demand-cv, *MPLS On-demand Connectivity Verification and Route Tracing.*

[IETF RFC xxxx] IETF RFC ietf-mpls-tp-li-lb, *MPLS-TP Lock Instruct and Loopback Functions.*

[IETF RFC xxxx] IETF RFC ietf-mpls-loss-delay, *Packet Loss and Delay Measurement for the MPLS-TP.*

[IETF RFC xxxx] IETF RFC ietf-mpls-tp-loss-delay-profile, *A Packet Loss and Delay Measurement Profile for MPLS-based Transport Networks.*

]

[Note: RFCs are removed due to dependency on other Recommendations]

3 Definitions

[Need to be update per current template]

This Recommendation uses the following terms defined in ITU-T Rec. G.805:

3.1 access point

3.2 adapted information

Formatted: Highlight
Formatted: Highlight
Formatted: Highlight
Formatted: Indent: Before: 0 cm, First line: 0 cm
Formatted: Highlight
Formatted: Highlight
Formatted: Indent: Before: 0 cm, Hanging: 3.3 ch, First line: -3.3 ch
Formatted: Highlight

3.3 characteristic information

3.4 client/server relationship

3.5 connection

3.6 connection point

3.7 layer network

3.8 matrix

3.9 network

3.10 network connection

3.11 reference point

3.12 subnetwork

3.13 subnetwork connection

3.14 termination connection point

3.15 trail

3.16 trail termination

3.17 transport

3.18 transport entity

3.19 transport processing function

3.20 unidirectional connection

3.21 unidirectional trail

[Note: Following terms should be refer to G.8101 or other MPLS-TP recommendations. Update will be made by Feb 2010]

This Recommendation uses the following terms defined in RFC 3031:

3.22 label

3.23 label stack

3.24 label switched path

This Recommendation uses the following terms defined in RFC 3032:

3.25 Bottom of Stack

3.26 Time To Live

3.27 Label value

This Recommendation uses the following terms defined in RFC 3270:

3.28 Per-Hop Behaviour

This Recommendation uses the following terms defined in RFC 5586

3.29 Associated Channel Header

3.30 Generic Associated Channel

3.31 G-ACh Label

Formatted: Highlight

Formatted: Highlight

Formatted: Tab stops: 1.4 cm, Left

4 Abbreviations

This Recommendation uses the following abbreviations:

AI	Adapted Information
AP	Access Point
BDI	Backward Defect Indication [Note: Aberration based on Y.1711 so propose to remove]
BIP	Bit Interleaved Parity [Note: Aberration based on Y.1711 so propose to remove]
CI	Characteristic Information
CII	Common Interworking Indicator
CP	Connection Point
CV	Connectivity Verification
DL	Defect Location [Note: Aberration based on Y.1711 so propose to remove]
DT	Defect Type [Note: Aberration based on Y.1711 so propose to remove]
FDI	Forward Defect Indication [Note: Aberration based on Y.1711 so propose to remove]
FFD	Fast Failure Detection [Note: Aberration based on Y.1711 so propose to remove]
FP	Flow Point
FTP	Flow termination point
LSP	Label Switched Path
MPLS	Multi-Protocol Label Switching
OAM	Operation, Administration and Maintenance
PHB	Per Hop Behaviour
PSC	PHB Scheduling Class
S	Bottom of Stack
SCC	Signalling Communication Channel
TCP	Termination Connection Point
TFP	Termination Flow Point
MPLS-TP	MPLS Transport Profile
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

Formatted: French (France)

Formatted: Highlight

Formatted: English (U.S.)

Formatted: Highlight

Formatted: English (U.S.)

Formatted: Highlight

Formatted: Highlight

Formatted: Highlight

Formatted: Highlight

Formatted: Font color: Auto

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
PSI	Payload Structure Indication
PT	Payload Type
RES	Reserved overhead
vcPT	virtual concatenation Payload Type
VcPLM	Virtual concatenation Payload Mismatch
TC	Traffic Class
ACH	Associated Channel Header
G-Ach	Generic Associated Channel
GAL	G-ACh Label
TLV	Type Length Value
DP	Drop Precedence
CoS	Class of service

Formatted: Highlight

5 Conventions

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

6 Supervision

The generic supervision functions are defined in clause 6/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 6-1/ G.8121/Y.1381 – Overview of Events

Event	Meaning
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 set to x; where x=0 (remote defect clear) and x=1 (remote defect set).
LCK	Reception of a LCK packet.
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.

<<APS related events are TBD>>]

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 the these defects as described in [MPLS-TP OAM FWK];.

Table 6-2/ G.8121/Y.1381 – Overview of Detection and Clearing Conditions

Defect	RFDIe Condition	Clearing Condition
dLOC	#expCC==0 (K*CC_Period)	expCC
dUNC-CC	unexpCoS-CC	#unexpCoS-CC==0 (K*CC_Period)
dUNC-CV	unexpCoS-CV	#unexpCoS-CV==0 (K*CV_Period)
dMMG	unexpMEG	#unexpMEG==0 (K* CV _Period)
dUNM	unexpMEP	#unexpMEP==0 (K*CV_Period)
dUNP-CC	unexpCCPeriod	#unexpCCPeriod==0 (K*CC_Period)
dUNP-CV	unexpCVPeriod	#unexpCVPeriod==0 (K*CV_Period)
dRDI	RDI==1	RDI==0
dAIS	AIS	#AIS==0 (K*AIS_Period)
dLCK	LCK	#LCK==0 (K*LCK_Period)
dCSF-LOS	CSF-LOS	#CSF-LOS == 0 (K*CSF_Period or CSF-DCI)
dCSF-FDI	CSF-FDI	#CSF-FDI == 0 (K*CSF_Period or CSF-DCI)
dCSF-RDI	CSF-RDI	#CSF-RDI == 0 (K*CSF_Period or CSF-DCI)
dDEG	#BS==DEGM (DEGM*1second)	#BS==0 (M*1second)

6.1.2 Continuity Supervision

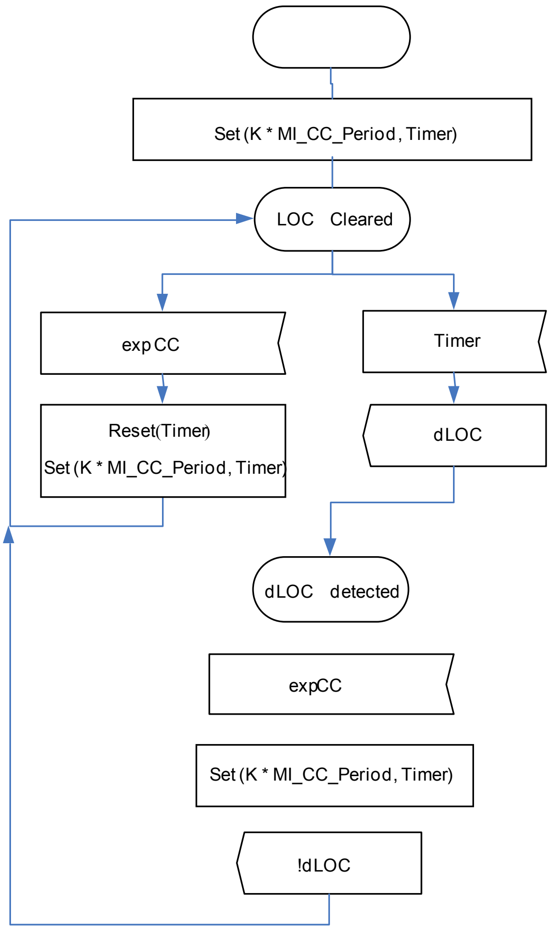


Figure 6-x/G.8121/Y.1381 – dLOC detection and clearance process

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 $3.25 \le K \le 3.5$.

6.1.3 Connectivity Supervision

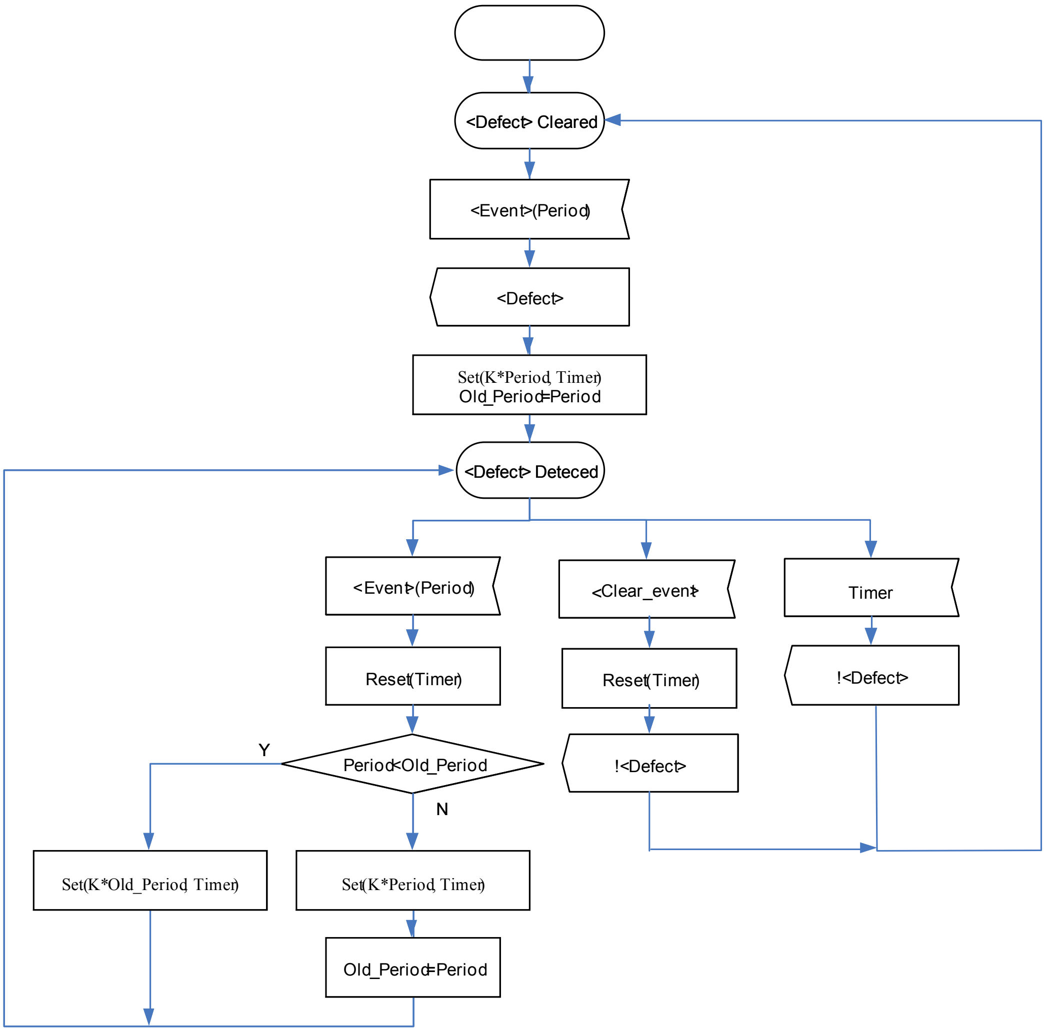


Figure 6-2/G.8021/Y.1341 – Defect detection and clearance process for dMMG, dUNM, dUNP, dUNPr, dAIS, dLCK and dCSF

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 $3.25 \leq K \leq 3.5$.

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.

6.1.3.4 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.

6.1.3.5 Degraded Signal defect (dDEG)

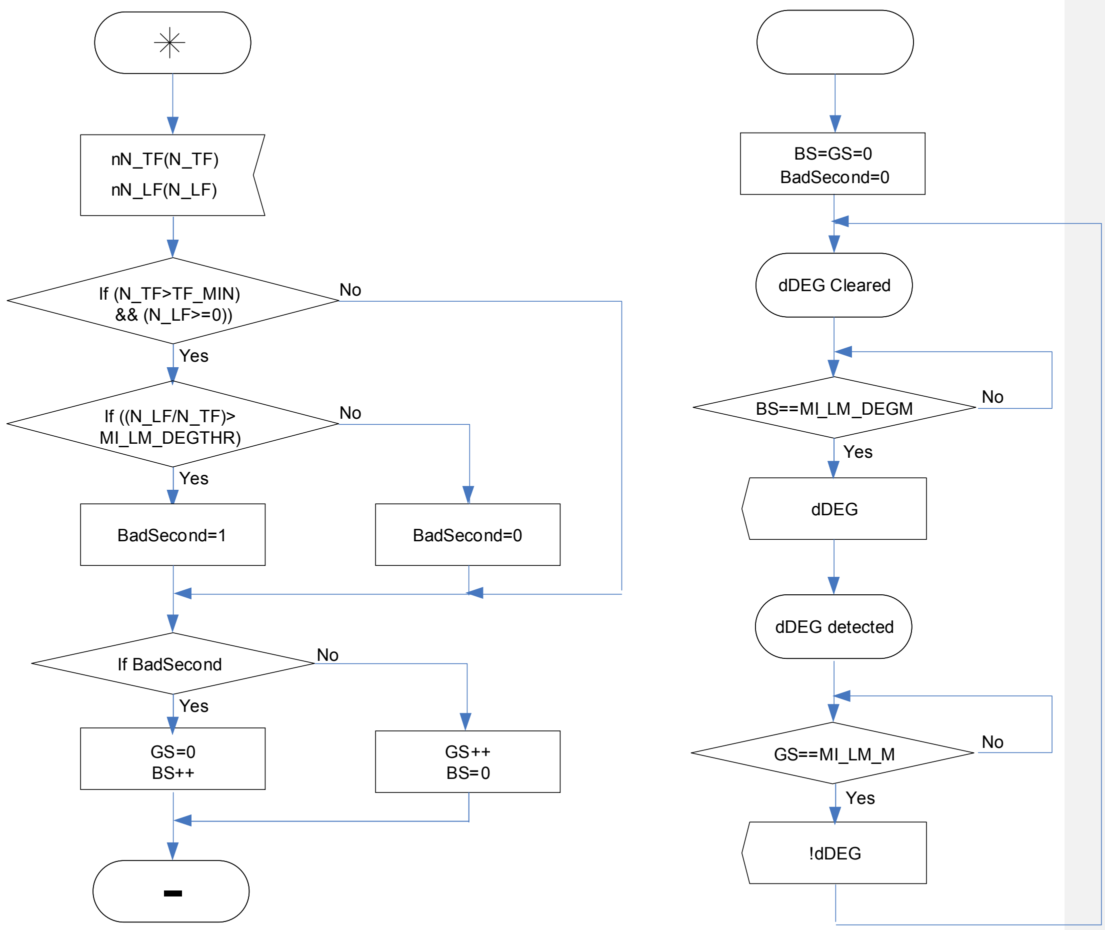


Figure 6-3/G.8121 – dDEG detection and clearance process

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 accociated 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 accociated with a greater period.

[Editor's note:The text on the 3rd paragraph should be cleaned up]

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.

Formatted: Highlight

Formatted: Highlight

Formatted: Superscript, Highlight

Formatted: Highlight

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 accociated with a greater period.

Formatted: Not Highlight

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 accociated with a greater period.

Formatted: Not Highlight

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 accociated 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 Rec. 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

Ffs.

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/G.806.

8 MPLS-TP processes

This clause defines the specific processes for the MPLS-TP network. Generic processes are defined in clause 8/G.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 [RFC5586].

8.1.2 G-ACh Insertion Process

Figure 8-1 describes G-ACh Insertion process.

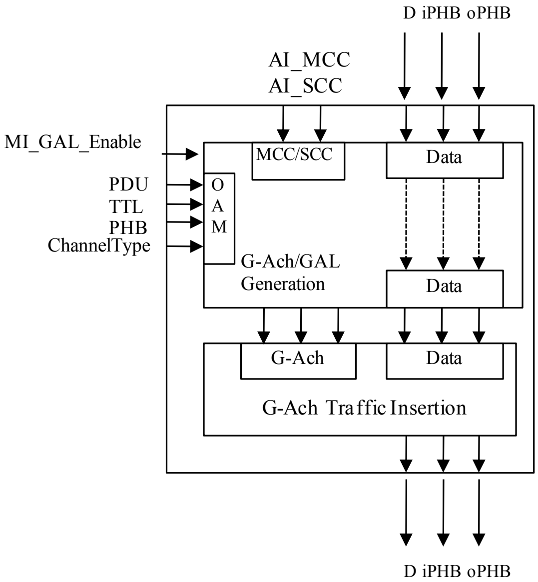


Figure 8-1 – G-ACh Insertion Process

Editor Note: G-Ach processing is performed per [RFC5586]. MCN and SCN frame processing is performed per [RFC5718]. The MCN and SCN frame should be assigned with the same priority as the MI_CC_P. Other OAM Fframes that require G-Ach processing is in Clause8.1.1.

[Editor Note: MI_GAL must be set to True on LSPs and to False on PWs (in the future the MI_GAL may be set to True also for PWs but this is still work-in-progress within IETF). Note this in Clause 8.1.1/G.8121 and also in G.8151.]

Formatted: Highlight

Formatted: Highlight

[Editor Note: Describe the following in 8.1.1/G.8121 (G-ACh Process): The TTL for OAM traffic is inserted by the G-ACh/GAL Insert process. The reason is that some OAM packets addressed to a MIP (e.g., LBM) needs to be inserted with a TTL value different than the TTL value used for user data traffic to ensure that the TTL expires at the target MIP. The Ach/GAL Insertion process is the same for proactive (MT_TT) and on-demand (MTDe_FT). For MT_TT, the TTL field for the GAL when used, can be set to 1 by the G-ACh/GAL process.]

8.1.3 G-Ach Reception Process

Figure 8-2 describes G-ACh Reception process.

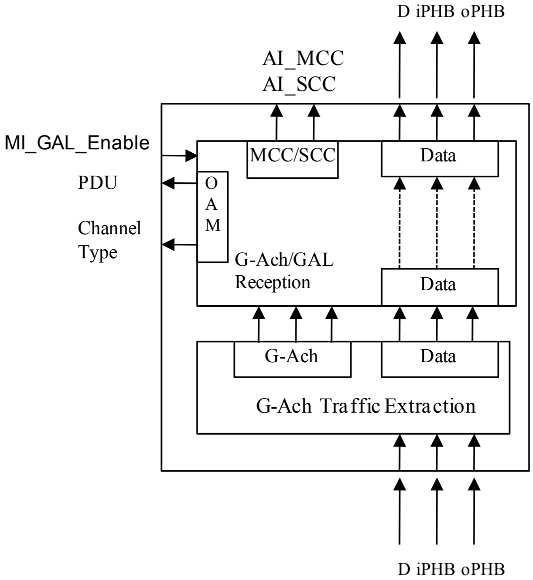


Figure 8-2 – G-ACh Reception Process

The G-ACh Traffic Unit will be extracted if it includes GAL and ACH in incoming Data when MI_GAL_Enable is set. Figure 8-3 shows G-ACh traffic Extract behaviour.

Formatted: Highlight

Formatted: Font: (Asian) MS Mincho

Formatted: None

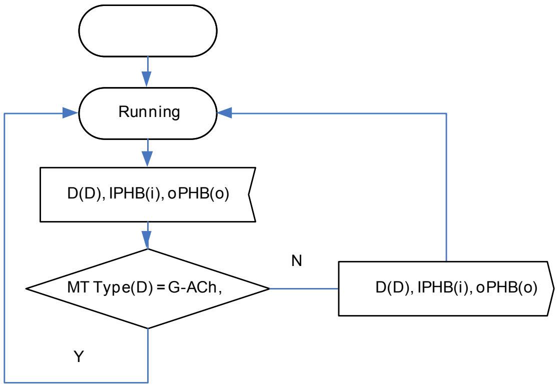


Figure 8-4 – G-ACh traffic Extract behaviour

[Note: It is required to describe malformed G-ACh such as only GAL without ACH]

Formatted: English (U.K.)

8.2 TC/Label processes

8.2.1 TC/Label source processes

Formatted: Highlight

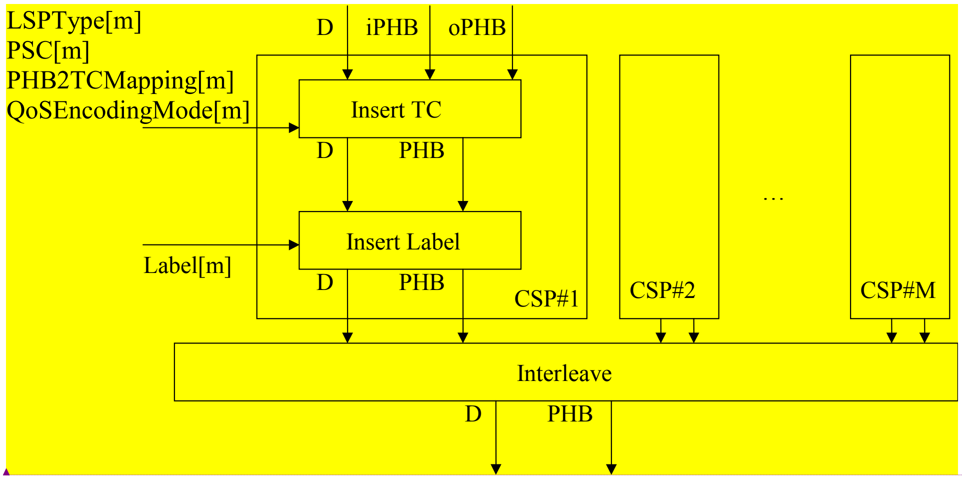


Figure 3/G.8121/Y.1381 – TC/Label source processes

[ed note: PSC should be CoS]

Formatted: Highlight

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 ($M \leq 2^N - 16$, 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 ($1 \leq m \leq M$) 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].

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

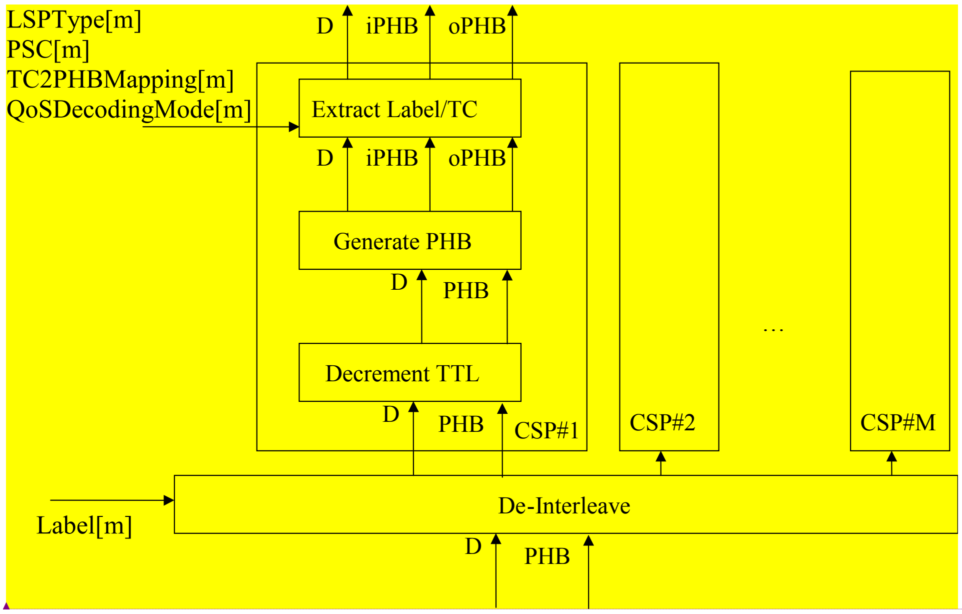


Figure 4/G.8121/Y.1381 – TC/Label sink processes
[ed note: PSC should be CoS]

Formatted: Highlight

Formatted: English (U.K.)

Formatted: Normal_after_title

Figure 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 ($M \leq 2^N - 16$, 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 ($1 \leq m \leq M$) 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: Decrements 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.

Formatted: Page Number

PHB Generation process: Processes the TC field.

The iPHB signal is generated according to the following rules:

- If LSPType[m] = 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].

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

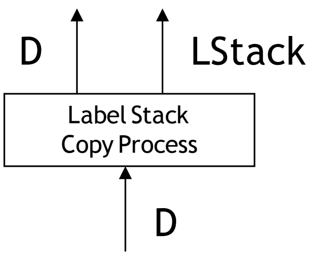


Figure x/G.8121/Y.1381 – Label Stack Copy Process

Figure X 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. 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.

Formatted: Font: Not Italic

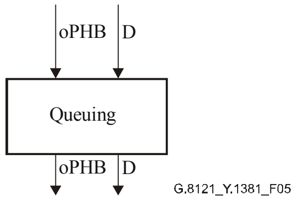


Figure 5/G.8121/Y.1381 – Queuing process

8.4 MPLS-TP-specific GFP-F processes

8.4.1 MPLS-TP-specific GFP-F source processes

Formatted: Highlight

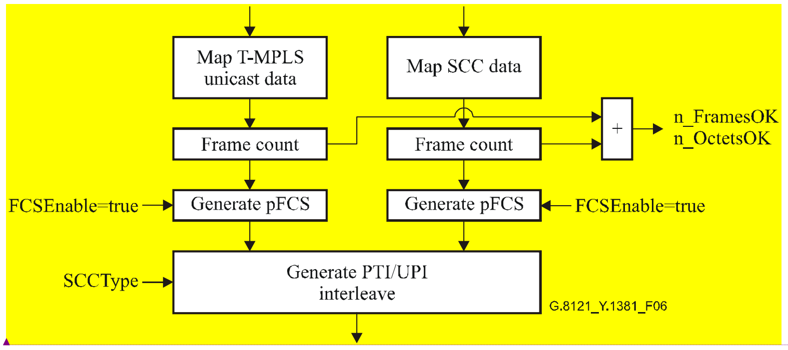


Figure 6/G.8121/Y.1381 – MPLS-TP-specific GFP-F source process
[Replace T-MPLS by MPLS-TP. Check with G.7712]

Figure 6 shows the MPLS-TP-specific GPF-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/G.7041/Y.1303. One MPLS-TP packetresults 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/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/G.7041/Y.1303), 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.

Formatted: Not Highlight

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

Formatted: Highlight

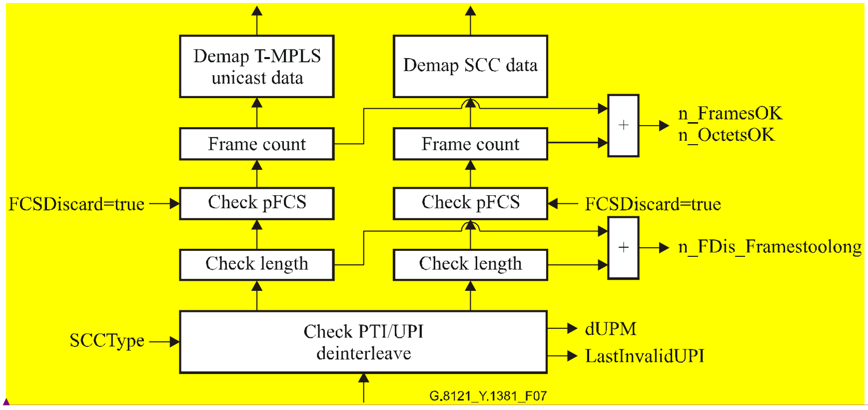


Figure 7/G.8121/Y.1381 – MPLS-TP-specific GFP-F sink process
[Replace T-MPLS by MPLS-TP Check with G.7712]

Figure 7 shows the MPLS-TP-specific GPF-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/G.7041/Y.1303) 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/G.7041/Y.1303) are sent towards the "Demap MPLS-TP data" process.

Formatted: Not Highlight

– "valid-UPI frames" with UPI equalling the SCC UPI according to SCCType (as defined in Table 6-3/G.7041/Y.1303) 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/G.7041/Y.1303. 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/G.7041/Y.1303. 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

Formatted: Highlight

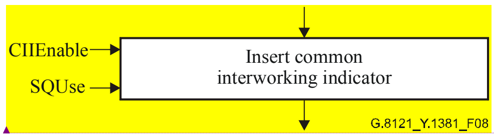


Figure 8/G.8121/Y.1381 – CW source process
[Replace "Insert Common Interworking Indicator" with "Insert Control Word". Replace "CIIenable" with "CWEnable"]

This function should generate and insert the CWas 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 CII Sink Process

Formatted: Highlight

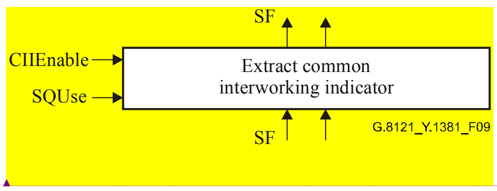


Figure 9/G.8121/Y.1381 – CII sink process
[Replace "Insert Common Interworking Indicator" with "Insert Control Word". Replace "CIIenable" with "CWEnable"]

Formatted: (Asian) Japanese

This function should process the Common Interworking Indicator as described in ITU-T Rec. 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

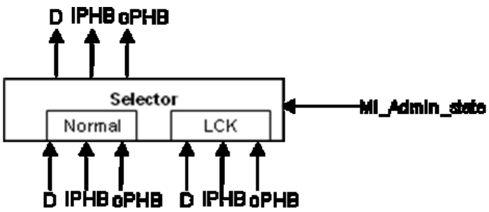


Figure 8-x1 – Selector process

Figure 8-x1 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-x2.

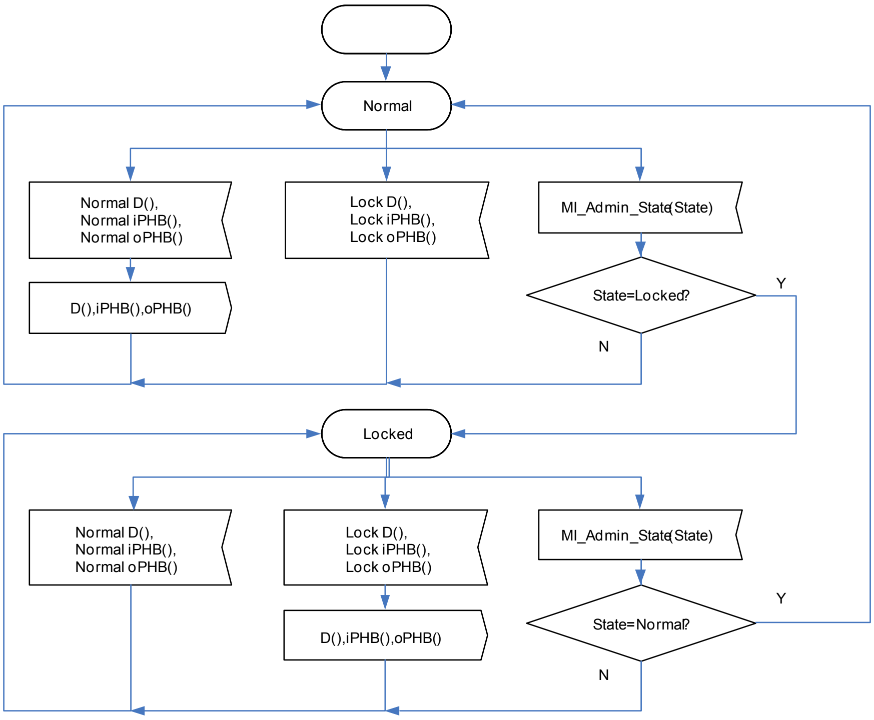


Figure 8-x2 – Selector Behaviour

8.6.2 AIS Insert Process

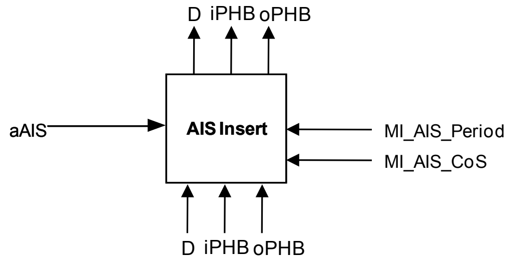


Figure 8-x3 – AIS Insert process

Figure 8-x3 shows the AIS Insert Process Symbol and Figure 8-x4 defines the behaviour. If the aAIS signal is true, the AIS Insert process continuously generates MT_CI traffic units where the MT_CI_D signal contains the AIS signal until the aAIS signal is false. 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. The value of the MT_CI_PHB signal associated with the generated AIS traffic units is defined by the MI_AIS_CoS input parameter. As described in [IETF tp-oam-framework], AIS packets are transmitted with the "minimum loss probability PHB".

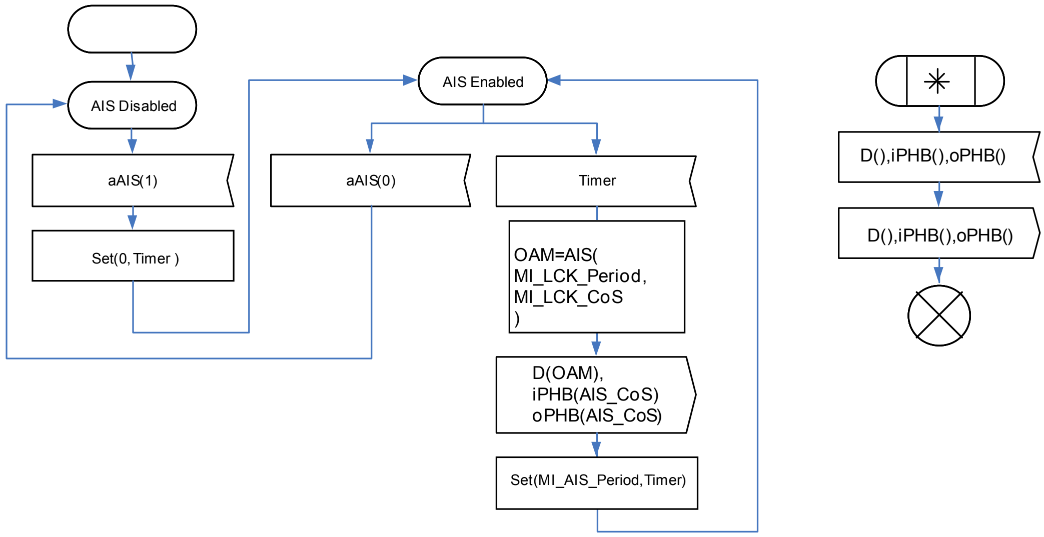


Figure 8-x4 – AIS Insert behaviour

Formatted: Justified

8.6.3 LCK Generate process

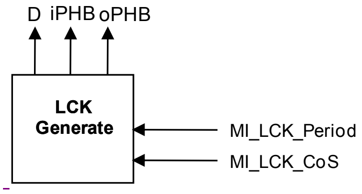


Figure 8-x5 – LCK Generation process

Formatted: Font: (Asian) MS Mincho, (Asian) Japanese

Figure 8-x5 shows the LCK Insert Process Symbol. The LCK Generation Process generates MT_CI traffic units where the MT_CI_D signal contains the LCK signal. Figure 8-x6 defines the behaviour of the LCK Generation Process.

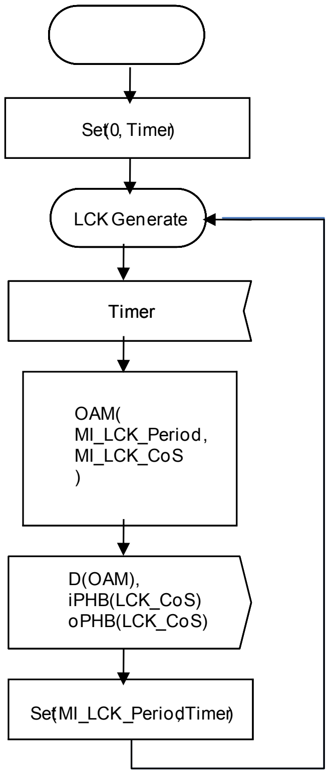


Figure 8-x6 – LCK Generation behaviour

The LCK Generation Process continuously generates LCK Traffic Units. 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 G.7712/Y.1702, 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 MCC and SCC 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 (ECC_CI_D) into the payload of a G-ACh ECC packet (MT_AI_ECC) as defined in [IETF RFC5718].

8.7.1.2 MCC and SCC 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 (ECC_CI_D) from the payload of a G-ACh ECC packet (MT_AI_ECC) as defined in [IETF RFC5718].

8.7.2 APS Insert and Extract Processes

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

[Add foot note to APS as "The IETF uses the term PSC for this function"]

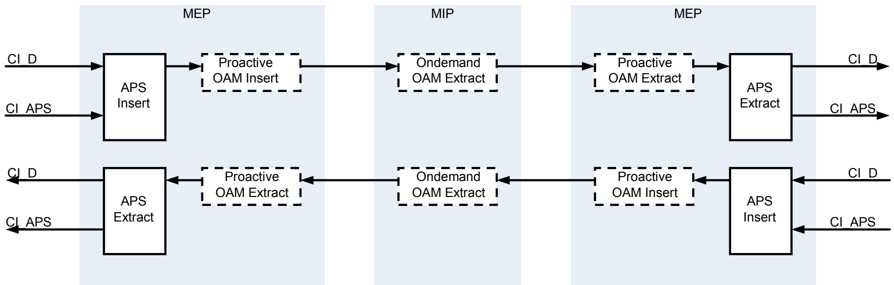


Figure 8-xx – Overview of the processes involved with APS function

APS Insert and Extract processes are located in MT/MT_Adaptation function. CI_APS signal which carries APS specific information defined in [ITU-T G.8131.1] and [ITU-T G.8131.2], is inserted into and extracted from the stream of MT_CI_D traffic units.

Note – Protection switching architecture is defined in [ITU-T G.8131.1] and [ITU-T G.8131.2] and relevant equipment model is to be described in clause 9.

[ed note: remove references G.8131.1 and G.8131.2 unless they are consented at the same meeting]

8.7.2.1 APS Insert Process

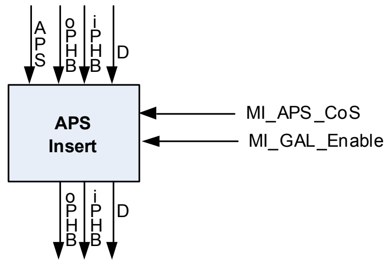


Figure 8-xx+1 – APS Insert process

Figure 8-xx+1 shows the APS Insert process and Figure 8-xx+2 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 Specific Information is defined in [ITU-T G.8131.1] and [ITU-T G.8131.2].

[Ed note: Text needs to be added to indicate that packet is G-ACh encapsulated with or without depending on MI_GAL_Enabled]

[ed note: remove references G.8131.1 and G.8131.2 unless they are consented at the same meeting]

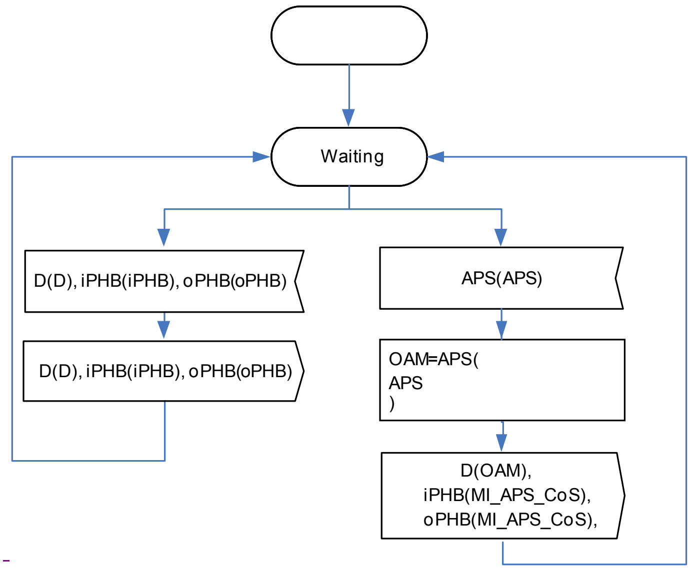


Figure 8-xx+2 – APS Insert Behaviour

Formatted: Highlight

Formatted: Highlight

Formatted: Don't snap to grid

Formatted: Highlight

Formatted: Font: (Asian) MS Mincho, (Asian) Japanese

8.7.2.2 APS Extract Process

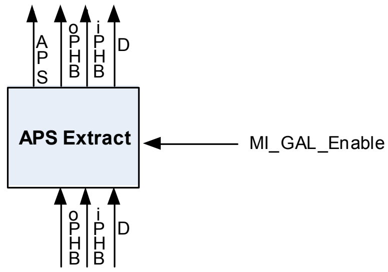


Figure 8-xx+3– APS Extract process

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 encoding of the MT_CI_D signal for APS frames is defined in [ITU-T G.8113.1] and [ITU-T G.8113.2].

[ed note: remove references G.8131.1 and G.8131.2 unless they are consented at the same meeting]

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, as defined in [ITU-T G.8113.1] and [ITU-T G.8113.2] *[Ed note: remove refs unless they are consented]*

This is defined in Figure 8-xx+4. The function APS(D) extracts the APS specific information from the received Traffic Unit.

Formatted: Font: (Asian) MS Mincho, (Asian) Japanese

Formatted: Highlight

Formatted: Highlight

Formatted: Font: Italic

Formatted: Font: Italic

Formatted: Font: (Asian) MS Mincho, (Asian) Japanese

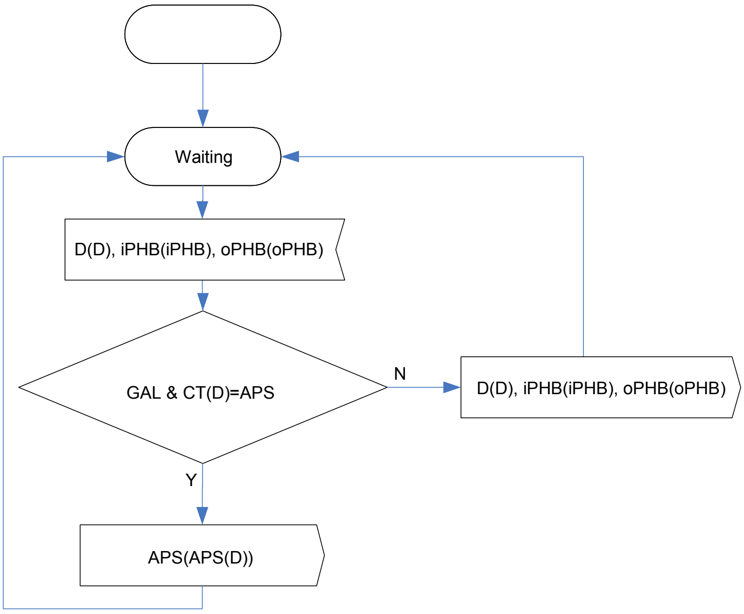


Figure 8-xx+4 – APS Extract Behaviour

8.7.3 CSF Insert and Extract Processes

Formatted: Not Highlight

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

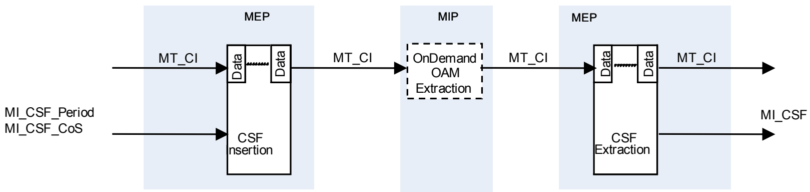


Figure 8-xx – Overview of Processes involved with CSF Protocol

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

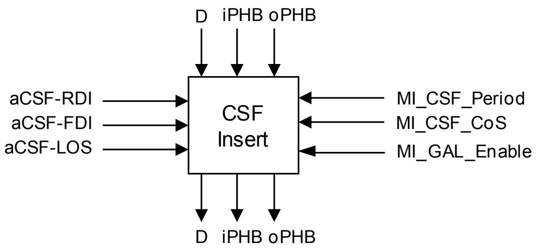


Figure 8-zz – CSF Insert process

The CSF Insert Process is located at MT/Client_A_So as a part of CSF generation. Figure 8-zz shows the CSF Insert Process Symbol and Figure 8-zz+1 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 units. The CSF traffic unit is defined in [ITU-T G.8113.1] and [ITU-T G.8113.2].

[Ed note: Text needs to be added to indicate that packet is G-ACh encapsulated with or without depending on MI_GAL_Enabled]

[Ed note: remove references unless they are approved/consented]

Formatted: Highlight

Formatted: Highlight

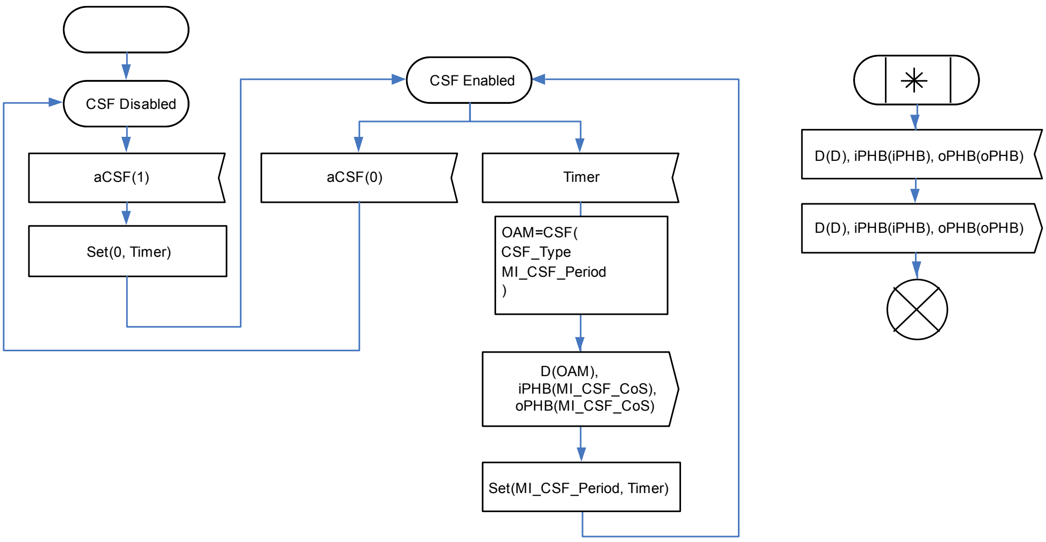


Figure 8-zz+1 – CSF Insert behaviour

The period between consecutive CSF traffic units is determined by the MI_CSF_Period parameter. Table 8-zz shows allowed values for CSF transmission period.

Table 8-zz – CSF period values

Period Value	Comments
1s	1 packet per second
1 min	1 packet per minute
Others	FFS

The CSF_Type shown on the Table 8-zz+1 means a type of client failure or defect event. The encoding of each type is defined in [ITU-T G.8113.1] and [ITU-T G.8113.2].

[Ed note: remove references unless they are approved/consented]

Table 8-zz+1 – CSF type values

Type	Comments
LOS	Client Loss of Signal
FDI/AIS	Client Forward Defect Indication
RDI	Client Reverse Defect Indication
DCI	Client Defect Clear Indication

8.7.3.2 CSF Extract Process

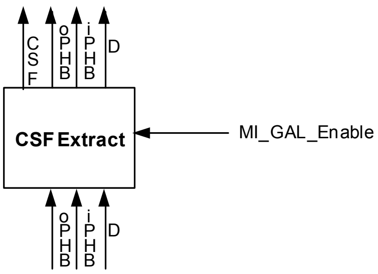


Figure 8-zz+3 – CSF Extract process

The CSF Extract process is located at MT/Client_A_sk and extracts MT-CSF from MI_AI_D. Figure 8-zz-3 shows the CSF Extract Process Symbol.

The encoding of the MT_CI_D signal for CSF frames is defined in [ITU-T G.8113.1] and [ITU-T G.8113.2].

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, as defined in [ITU-T G.8113.1] and [ITU-T G.8113.2]. (remove ref…)

This behaviour is defined in Figure 8-zz+4. 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.

Formatted: Highlight

Formatted: Highlight

Formatted

Formatted: Normal

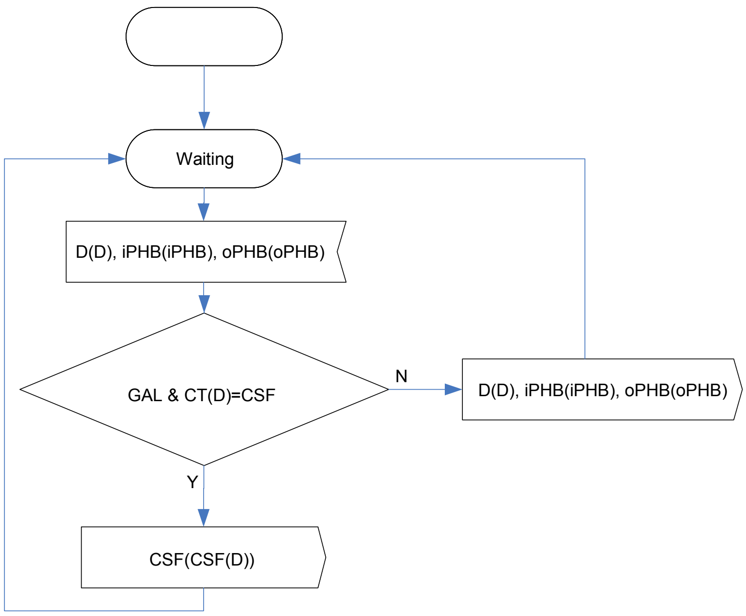
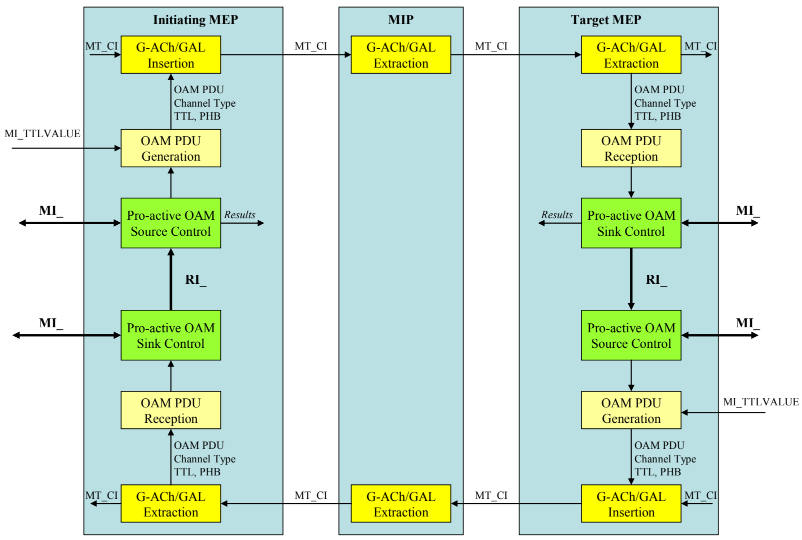


Figure 8- zz+4 – CSF Extract Behaviour

8.8 Pro-active and on-demand OAM related Processes

[Editor's note: Need to develop the whole text in clause 8 by correspondence]

Figure 8-xx shows an OAM protocol-neutral abstract model of the different processes inside MEPs and MIPs that are involved in performing pro-active OAM functions.



Formatted: Highlight

Formatted: Highlight

Formatted: Highlight

Formatted: Normal

Figure 8-xx – Overview of the processes involved with pro-active OAM functions

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 unidirectional and bidirectional pro-active OAM transactions.

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

In the case of bidirectional OAM transactions, the following actions are taken:

- The OAM Sink Control process within the target MEP provides the local pro-active 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 provides the local pro-active OAM Source Control process the Remote Information (RI_) required for reporting the bidirectional OAM measurements based on the OAM Reply PDUs received by the local OAM PDU Reception process.
- The OAM Source Control process within the initiating MEP reports the bidirectional OAM measurements based on the information it receives from the local OAM Sink Control process via the Remote Information (RI_).

Figure 8-xx+1 shows an OAM protocol-neutral abstract model of the different processes inside MEPs and MIPs that are involved in performing on-demand OAM functions.

<<Add Figure 8-xx+1>>

Figure 8-xx+1 – Overview of the processes involved with on-demand OAM functions

<<Add description of on-demand OAM Control processes>>

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 PHB 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

Formatted: Not Highlight

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.

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

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.

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

Figure 8-xx+2 shows an OAM protocol-neutral abstract model of the different processes inside MEPs and MIPs that are involved in performing pro-active Continuity Check and Connectivity Verification (CC/CV) OAM functions.

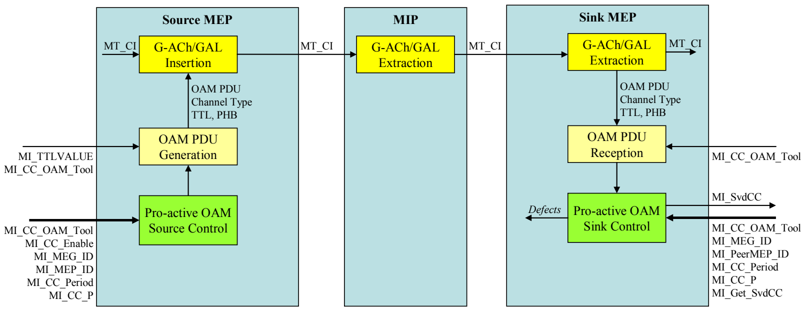


Figure 8-xx+2 – Overview of the processes involved with pro-active CC/CV
(_P→_CoS, add MI_CC_DP)

As described in [b-IETF tp-oam-fw], 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 MI_CC_Enable is true. As described in [b-IETF tp-oam-fw], 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 MI_MEG_ID and MI_MEP_ID on the source MEP while the expected value is configured via the MI_MEG_ID and MI_Peer_MEP_ID on the sink MEP.

Formatted: Normal

Formatted: Font: Not Bold, Highlight

Formatted: Highlight

Formatted: Font: Not Bold, Highlight

Formatted: Font: (Asian) MS Mincho, (Asian) Japanese

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 [b-IETF tp-oam-fw].

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)

8.8.3 On-demand Connectivity Verification (CV)

8.8.4 Pro-active Packet Loss Measurement (LMp)

Figure 8-xx+5 shows an OAM protocol-neutral abstract model of the different processes inside MEPs and MIPs that are involved in performing pro-active Loss Measurement (LMp) OAM functions.

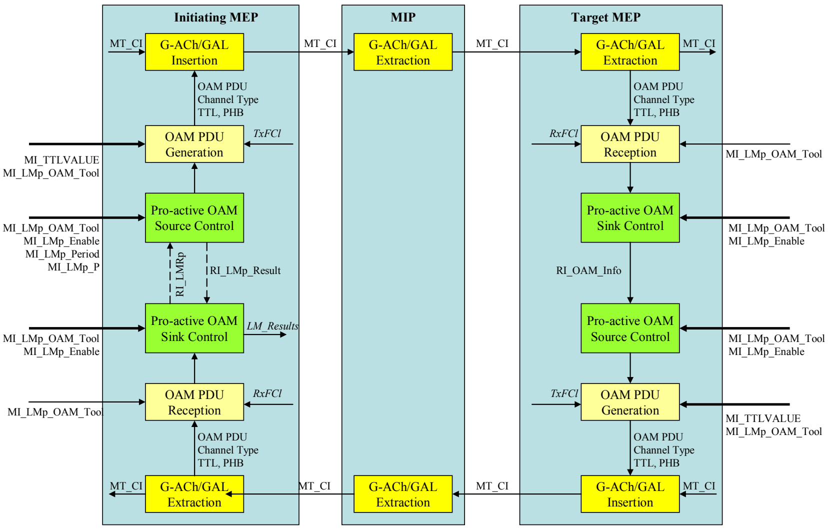


Figure 8-xx+5 – Overview of the processes involved with pro-active LMp

As described in [b-IETF tp-oam-fw], pro-active LM is performed by periodically sending LM OAM packets from the initiatingMEP 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 (TxFCl and RxFCl). 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.

[Ed note:One way should be added to this clause]

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 [b-IETF tp-oam-fw], 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 (TxFCl) is inserted within the LM OAM packet by the OAM PDU Generation process.

The target MEP replies to the LM OAM packets if the MI_LMp_Enable is true. The local value of the received user data packets (RxFCl) 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 (TxFCl) 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 (RxFCl) 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_LMRp 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.

Pro-active LM OAM packets pass transparently through MIPs as described in [b-IETF tp-oam-fw].

8.8.5 On-demand Packet Loss Measurement (LMo)

To be added

8.8.6 Pro-active Packet Delay Measurement (DMp)

To be added

8.8.7 On-demand Packet Delay Measurement (DMo)

To be added

8.8.8 Diagnostic Test (DT)

[Note: 2 way should be added to this clause]

Figure 8-xx+6 shows an OAM protocol-neutral abstract model of the different processes inside MEPs and MIPs that are involved in One Way Diagnostic Test (1DT) OAM functions.

Formatted: Highlight

Formatted: Normal

Formatted: Normal

Formatted: Normal

Formatted: Highlight

Formatted: Normal

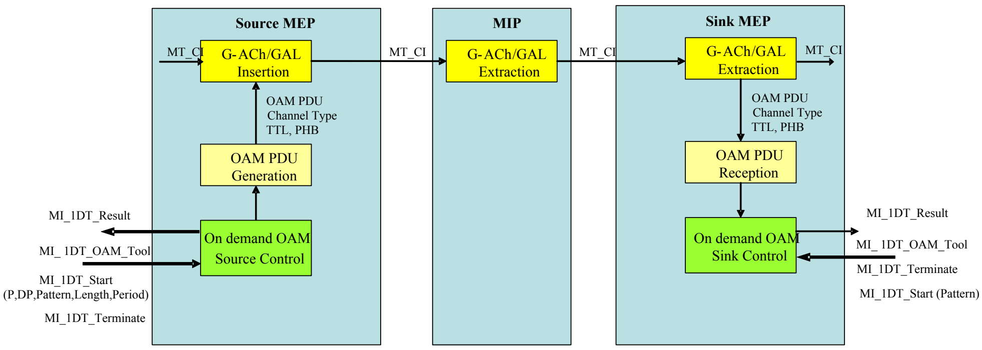


Figure 8-xx+6 – Overview of the processes involved with One Way Diagnostic Test (1DT)

As described in [b-IETF tp-oam-fw], 1DT can be used for out of service on demand throughput estimation, the test is performed by sending 1DT 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 1DT OAM test packet size.

The source MEP starts generating 1DT packets when requested via the MI_1DT_Start (P, DP, Pattern, Length, Period) and continues generating 1DT packets at the configured period until requested to stop via the MI_Terminate; at this time the number of sent packets are reported by the MI_1DT_Result.

The generated 1DT packets contain a Test pattern of the configured Length. The following pattern types are supported:

0: “Null signal without CRC-32”

1: “Null signal with CRC-32”

2: “PRBS 2^31-1 without CRC-32”

3: “PRBS 2^31-1 with CRC-32”

The sink MEP, when enabled via the MI_1DT_Start (Pattern), start processing the received 1DT packet until the test is terminated via the MI_Terminate; at this time, the calculated test results are reported by MI_1DT_Result.

8.8.9 Route Tracing (RT)

To be added

9 MPLS-TP layer functions

Formatted: Normal

Figure 10 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.

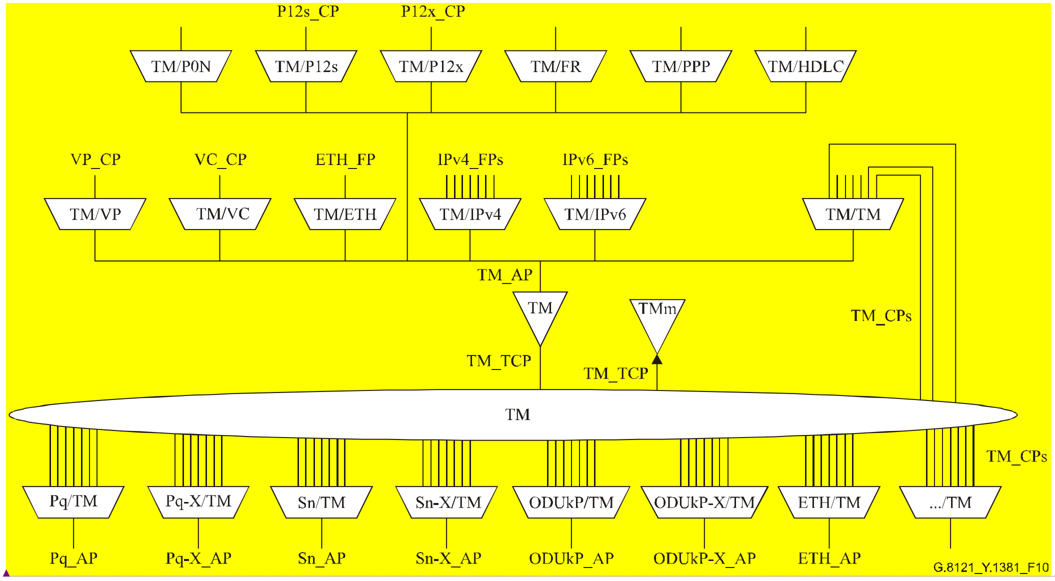


Figure 10/G.8121/Y.1381 – MPLS-TP atomic functions
[Replace TM by MT (same as fig1)]
[MTDe, MTDi related functions are needed]

Formatted: Highlight

Formatted: English (U.K.)

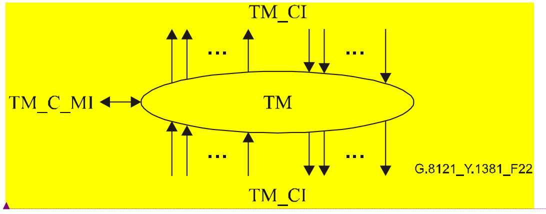
9.1 Connection Functions (MT_C)

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 22. 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 22.

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

• **Symbol:**



Formatted: Highlight

Figure 22/G.8121/Y.1381 – MT_C symbol
[Replace TM by MT]

• **Interfaces:**

Table 3/G.8121/Y.1381 – MT_C input and output signals

Input(s)	Output(s)
Per MT_CP, n × for the function: MT_CI_D MT_CI_iPHB MT_CI_oPHB MT_CI_SSF MT_AI_TSF per input and output connection point: *for further study* per matrix connection: MT_C_MI_ConnectionType MT_C_MI_Return_CP_ID MT_C_MI_ConnectionPortIds per SNC protection group: *for further study*	per MT_CP, m × per function: MT_CI_D MT_CI_iPHB MT_CI_oPHB MT_CI_SSF

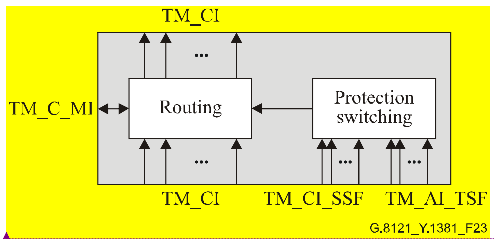
Formatted: Highlight

Formatted: Highlight

• **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.



Formatted: Highlight

Figure 23/G.8121/Y.1381 – MT_C process diagram
[Replace TM by MT]

– *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-a1.

Formatted: Not Highlight

Formatted: Not Highlight

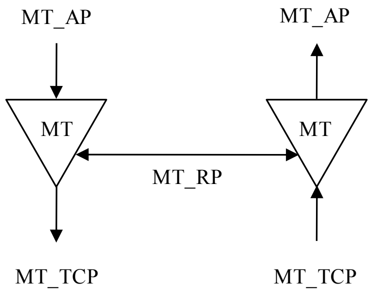


Figure 9-a1/G.8121/Y.1381 – MT_TT

9.2.1.1 MPLS-TP Trail Termination Source function (MT_TT_So)

The MT_TT_So function determines and inserts the TTL value in the shim header TTL field and adds MPLS-TP OAM 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 Figure9-a2.

• Symbol:

Formatted: Highlight

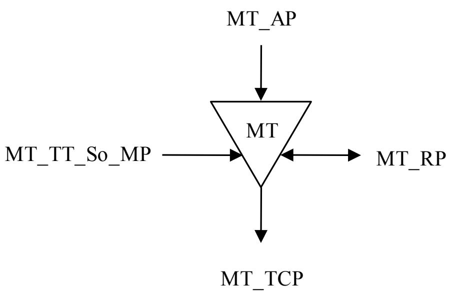


Figure9-a2/G.8121/Y.1381 –MT_TT_So function

• Interfaces:

Table 9-a1/G.8121/Y.1381 – MT_TT_So inputs and outputs

Input(s)	**Output(s)**
MT_AP: MT_AI_D MT_AI_PHB MT_AI_MCC MT_AI_SCC MT_AI_LStack **MT_RP:** MT_RI_CC_RDI MT_RI_CC_Blk	**MT_CP:** MT_CI_D MT_CI_oPHB MT_CI_iPHB MT_CI_LStack **MT_RP:** MT_TT_So_RI_LMp_ Result(CoS,N_TF,N_LF,F_TF,F_LF) MT_TT_So_RI_DMp_

Formatted: Not Highlight

MT_RI_OAM_Info(D,CoS,DP) MT_RI_LMRp(TxFCf,RxFCf,TxFCb,RxFCl,CoS) MT_RI_DMRp(TxTimeStampf,RxTimeStampf, TxTimeStampb,RxTimeb,CoS) MT_RI_SLRp(TxFCf,TxFCb,rTestID) **MT_TT_So_MP:** MT_TT_So_MI_GAL_Enable MT_TT_So_MI_TTLVALUE MT_TT_So_MI_ MEG_ID MT_TT_So_MI_ MEP_ID MT_TT_So_MI_CC_OAM_Tool MT_TT_So_MI_RDI_OAM_Tool MT_TT_So_MI_CC_Enable MT_TT_So_MI_CC_CoS MT_TT_So_MI_CC_DP MT_TT_So_MI_CC_Period MT_TT_So_MI_CV_Period MT_TT_So_MI_ LMp_OAM_Tool MT_TT_So_MI_LMp_Enable[1...M_{LMp}] MT_TT_So_MI_LMp_Period[1...M_{LMp}] MT_TT_So_MI_LMp_CoS[1...M_{LMp}] MT_TT_So_MI_ DMp_OAM_Tool MT_TT_So_MI_DMp_Enable[1...M_{DMp}] MT_TT_So_MI_DMp_Period[1...M_{DMp}] MT_TT_So_MI_DMp_Test_ID[1...M_{DMp}] MT_TT_So_MI_DMp_CoS[1...M_{DMp}] MT_TT_So_MI_DMp_Length[1...M_{DMp}] MT_TT_So_MI_ 1DMp_OAM_Tool MT_TT_So_MI_1DMp_Enable[1...M_{1DMp}] MT_TT_So_MI_1DMp_Period[1...M_{1DMp}] MT_TT_So_MI_1DMp_Test_ID[1...M_{1DMp}] MT_TT_So_MI_1DMp_Length[1...M_{1DMp}] MT_TT_So_MI_1DMp_CoS[1...M_{1DMp}] MT_TT_So_MI_ SLp_OAM_Tool MT_TT_So_MI_SLp_Enable[1...M_{SLp}] MT_TT_So_MI_SLp_Period[1...M_{SLp}] MT_TT_So_MI_SLp_Test_ID[1...M_{SLp}] MT_TT_So_MI_SLp_Length[1...M_{SLp}] MT_TT_So_MI_SLp_CoS[1...M_{SLp}] **MT_TP:** MT_TT_So_TI_ TimeStampl	Result(CoS,B_FD,F_FD,N_FD) MT_TT_So_RI_SLp_ Result(CoS,N_TF,N_LF,F_TF,F_LF)

• Processes:

The processes associated with the MT_TT_So function are as depicted in Figure 9-a3.

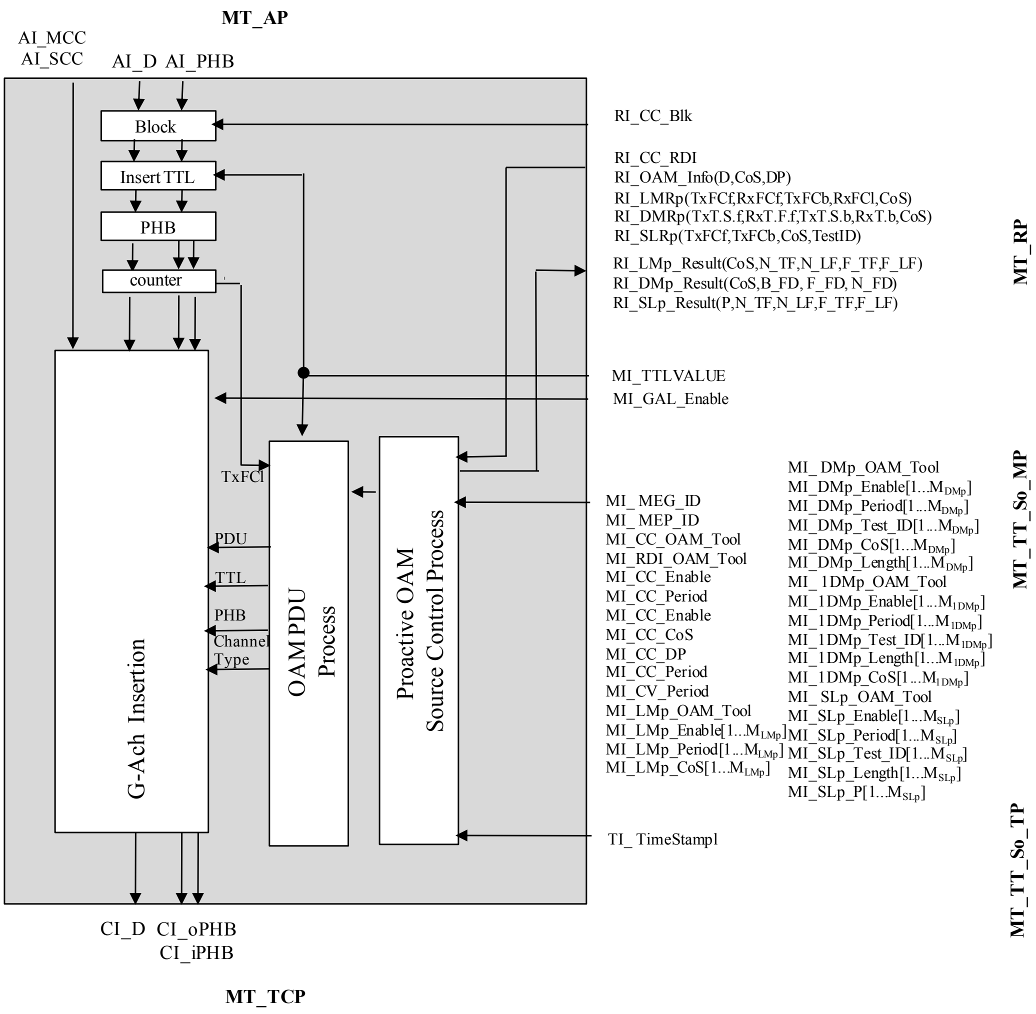


Figure xx/G.8121/Y.1381 –MT_TT_So process diagram

Notes:

Formatted: English (U.S.)

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.

[Editor's Note: In the block diagram, shows that no need to have SLp and LMp result simultaneously for a given priority.]

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: [Editor Note: Text or pointer will be provided.]

G-Ach/GAL Insertion Process: See 8.1/G.8121.

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-a4.

Formatted: Highlight

Formatted: (Asian) Japanese, (Other) English (U.K.), Highlight

Formatted: Normal

Formatted: Highlight

• **Symbol:**

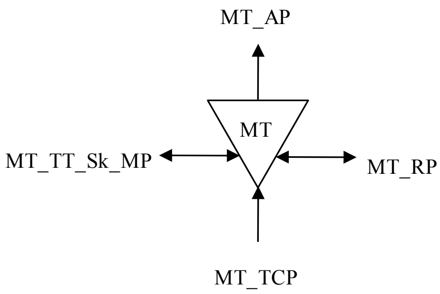


Figure 9-a4/G.8121/Y.1381 – MT_TT_Sk function

• **Interfaces:**

Table 9-2/G.8121/Y.1381 – MT_TT_Sk inputs and outputs

Input(s)	Output(s)
MT_TCP: MT_CI_D MT_CI_iPHB MT_CI_oPHB MT_CI_SSF MT_CI_Lstack	**MT_AP:** MT_AI_D MT_AI_PHB MT_AI_TSF MT_AI_TSD MT_AI_AIS MT_AI_MCC MT_AI_SCC MT_AI_LStack
MT_RP: MT_TT_Sk_RI_LMp_ Result(P,N_TF,N_LF,F_TF,F_LF) MT_TT_Sk_RI_DMp_ Result(P,B_FD,F_FD,N_FD) MT_TT_Sk_RI_SLp_ Result(P,N_TF,N_LF,F_TF,F_LF)	**MT_RP:** MT_RI_CC_RDI MT_RI_CC_Blk MT_RI_OAM_Info(D,P,DP) MT_RI_LMRp(TxFCf,RxFCf,TxFCb,RxFCl,CoS) MT_RI_DMRp(TxTimeStampf,RxTimeStampf,TxTimeStampb,RxTimeb,CoS) MT_RI_SLRp(TxFCf,TxFCb,rTestID)
MT_TT_Sk_MP: MT_TT_Sk_MI_ MEG_ID MT_TT_Sk_MI_ PeerMEP_ID MT_TT_Sk_MI_ CC_OAM_Tool MT_TT_Sk_MI_ RDI_OAM_Tool MT_TT_Sk_MI_CC_Enable MT_TT_Sk_MI_CC_Period MT_TT_Sk_MI_CC_CoS MT_TT_Sk_MI_CV_Period MT_TT_Sk_MI_Get_SvdCC MT_TT_Sk_MI_ LMp_OAM_Tool MT_TT_Sk_MI_LMp_Enable[1... M_{LMp}] MT_TT_Sk_MI_LMp_CoS[1... M_{LMp}]	**MT_TT_Sk_MP:** MT_TT_Sk_MI_SvdCC MT_TT_Sk_MI_cSSF MT_TT_Sk_MI_cLCK MT_TT_Sk_MI_cLOC MT_TT_Sk_MI_cMMG MT_TT_Sk_MI_cUNM MT_TT_Sk_MI_cUNP-CC

Formatted: Normal

Formatted: Highlight

Formatted: Font: (Asian) MS Mincho

Formatted: Highlight

Formatted: Highlight

Input(s)	Output(s)
MT_TT_Sk_MI_LM_DEGM MT_TT_Sk_MI_LM_M MT_TT_Sk_MI_LM_DEGTHR MT_TT_Sk_MI_LM_TFMIN MT_TT_Sk_MI_ DMp_OAM_Tool 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_TP:** MT_TT_Sk_TI_ TimeStampl	MT_TT_Sk_MI_cUNP-CV MT_TT_Sk_MI_cUNC-CC MT_TT_Sk_MI_cUNC-CV MT_TT_Sk_MI_cDEG MT_TT_Sk_MI_cRDI MT_TT_Sk_MI_pN_LF[1...P] MT_TT_Sk_MI_pN_TF[1...P] MT_TT_Sk_MI_pF_LF[1...P] MT_TT_Sk_MI_pF_TF[1...P] MT_TT_Sk_MI_pF_DS MT_TT_Sk_MI_pN_DS MT_TT_Sk_MI_pB_FD[1...P] MT_TT_Sk_MI_pB_FDV[1...P] MT_TT_Sk_MI_pN_FD[1...P] MT_TT_Sk_MI_pN_FDV[1...P] MT_TT_Sk_MI_pF_FD[1...P] MT_TT_Sk_MI_pF_FDV[1...P]

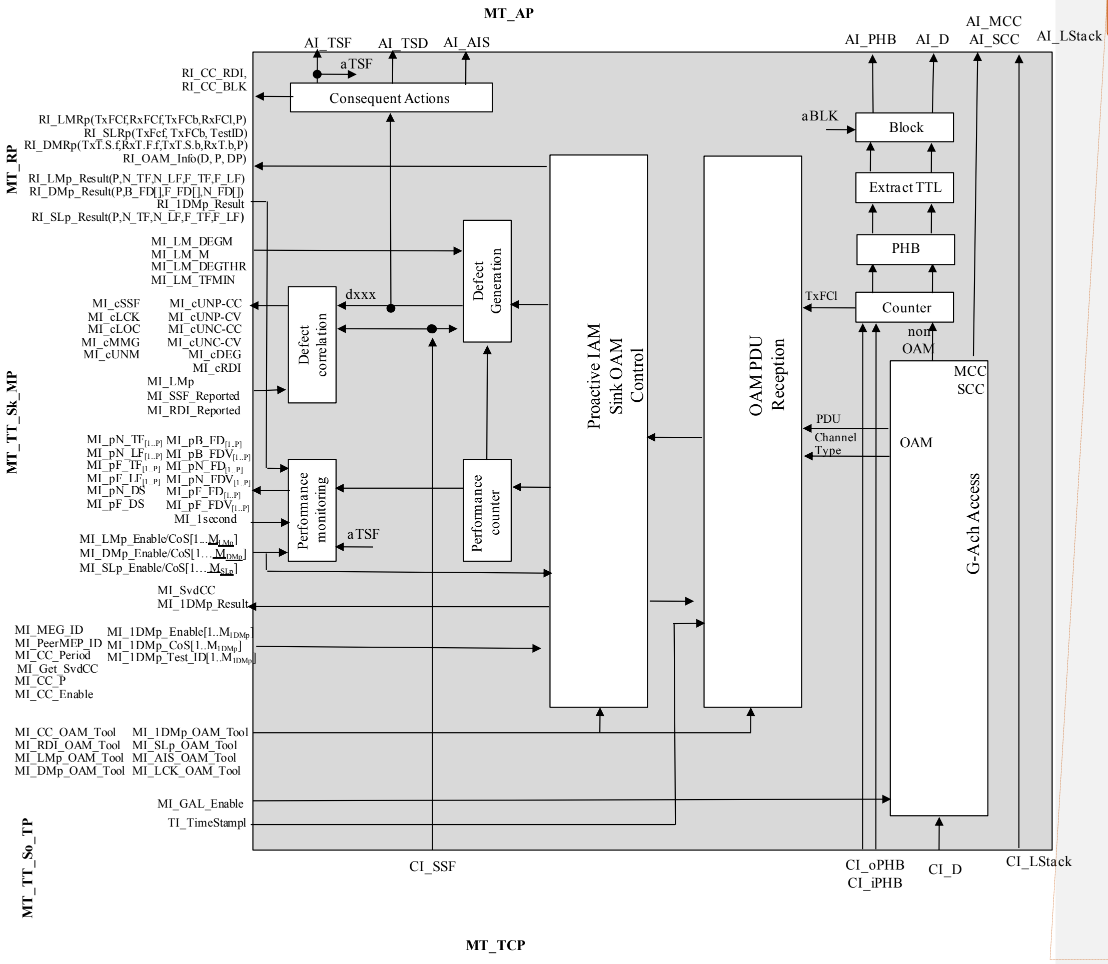
Formatted: Not Highlight

Formatted: Not Highlight

Formatted: Font: (Asian) MS Mincho, (Asian) Japanese, Highlight

• Processes:

The processes associated with the MT_TT_Sk function are as depicted in Figure xx.



Formatted: (Asian) Japanese, (Other) English (U.S.)

Figure xx/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: [Editor Note: to be provided]

G-Ach/GAL Extraction Process: See 8.1/G.8121.

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

OAM PDU Reception Process: See 8.8/G.8121

Defect Generation: This process raises and clears the defects as defined in clause 6.1/G.8121.•

• Defects:

ffs

• Consequent actions:

ffs

• Defect correlations:

ffs

• Performance monitoring:

Ffs.

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.

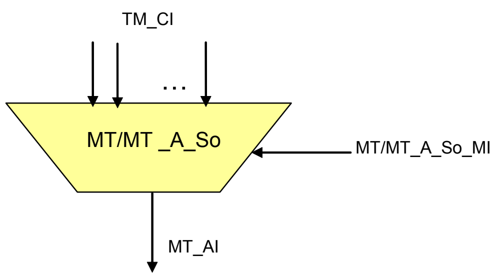


Figure 35/G.8121/Y.1381 – MT/MT_A_So function

• **Interfaces:**

Table 6/G.8121/Y.1381 – MT/MT_A_So interfaces

Inputs	Outputs
Each MT_CP: MT_CI_Data MT_CI_iPHB MT_CI_oPHB **MT/MT_A_So_MI:** MT/MT_A_So_MI_Admin_State MT/MT_A_So_MI_Label[1…M] MT/MT_A_So_MI_LSPType[1…M] MT/MT_A_So_MI_CoS[1…M] MT/MT_A_So_MI_PHB2TCMapping[1…M] MT/MT_A_So_MI_QoSEncodingMode[1…M] MT/MT_A_So_MI_LCK_Period[1…M] MT/MT_A_So_MI_LCK_CoS[1…M]	**MT_AP:** MT_AI_Data MT_AI_PHB

• **Processes:**

A process diagram of this function is shown in Figure 36.

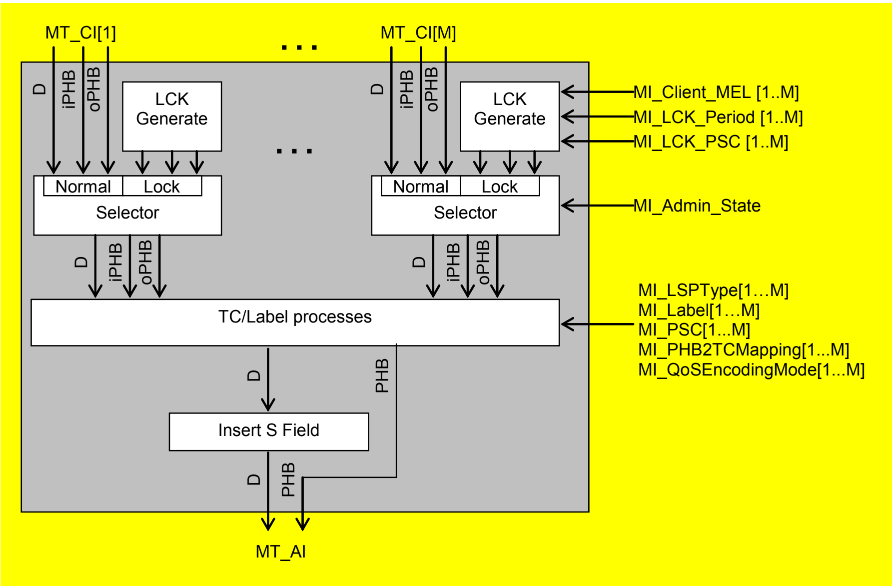


Figure 36/G.8121/Y.1381 – MT/MT_A_So process diagram
[~~PSC~~ CoS, remove clinet_MEL]

– *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.

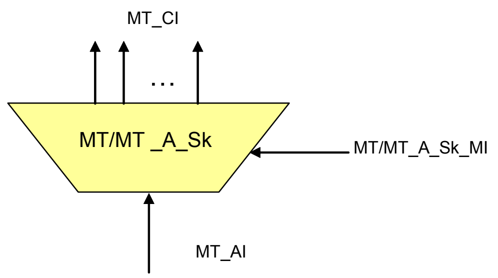


Figure 37/G.8121/Y.1381 – MT/MT_A_Sk function

• **Interfaces:**

Table 7/G.8121/Y.1381 – MT/MT_A_Sk interfaces

Inputs	Outputs
MT_AP: MT_AI_Data MT_AI_PHB MT_AI_TSF MT_AI_AIS MT_AI_LStack	**Each MT_CP:** MT_CI_Data MT_CI_iPHB MT_CI_oPHB MT_CI_SSF MT_AI_LStack

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]	

•

Processes:

A process diagram of this function is shown in Figure 38.

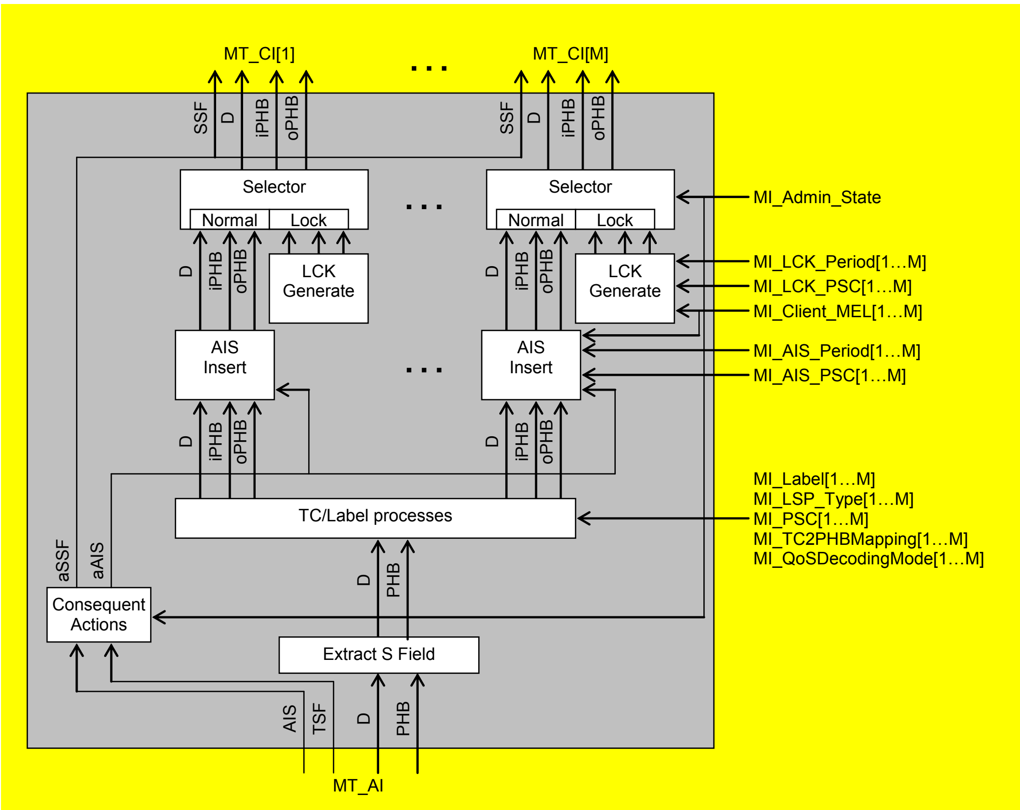


Figure zz/G.8121/Y.1381 – MT/MT_A_Sk process diagram
[add and update required (add LStack, rm clinet_MEL and PSC→CoS)]

Formatted: Highlight

Formatted: Highlight

– *Selector process:*

See 8.6.1. The normal CI is blocked if Admin_State = LOCKED.

– *LCK Generate process:*

See 8.6.3.

– *AISprocess:*

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 ← AI_TSF

aAIS ← AI_AIS • Defect correlations:

None.

• Performance monitoring:

None.

Formatted: Italian (Italy)

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-c1.

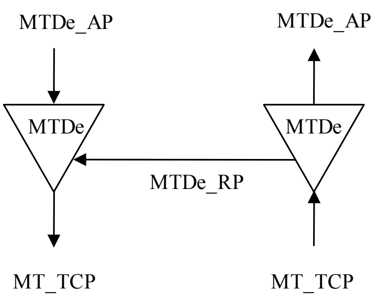


Figure 9-c1/G.8121/Y.1381 – MTDe_TT

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-c2.

Symbol

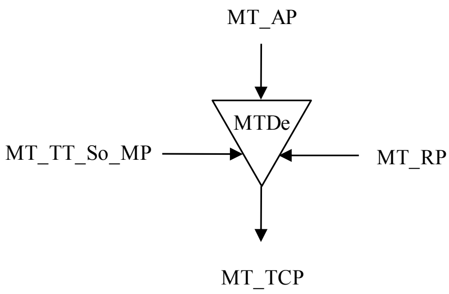


Figure 9-c2 – MTDe_TT_So symbol

Interfaces

Table 9-c1 – MTDe_TT_So interfaces

Input(s)	Output(s)
MTDe_AP: MTDe_AI_D MTDe_AI_oPHB MTDe_AI_iPHB MTDe_AI_LStack **MTDe_RP:** MTDe_RI_OAM_Info(D,CoS,DP) MTDe_RI_LMRo(TxFCf,RxFCf,TxFCb,RxFCl,CoS) MTDe_RI_DMRo(TxTimeStampf,RxTimeStampf, TxTimeStampb,RxTimeb,P,TestID)	**MT_CP:** MT_CI_D MT_CI_oPHB MT_CI_iPHB MTDe_CI_LStack **MT_RP:** **MTDe_TT_So_MP:** MTDe_TT_So_MI_CV_Series_Result(REC,ERR,OO) MTDe_TT_So_MI_RT_Result(<MIP_ID>,MEP_ID) MTDe_TT_So_MI_DT_Result(Sent, REC, CRC, BER, OO) MTDe_TT_So_MI_1DT_Result(Sent)

MTDe_RI_SLRo(TxFCf,TxFCb,rTestID) **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,DP,N,Length,Period) MTDe_TT_So_MI_RT_OAM_Tool MTDe_TT_So_MI_RT(Target_MEP/MIP,TTL,P) MTDe_TT_So_MI_DT_OAM_Tool MTDe_TT_So_MI_DT_Start (CoS,DP,Pattern,Length,Period) MTDe_TT_So_MI_DT_Terminate MTDe_TT_So_MI_1DT_OAM_Tool MTDe_TT_So_MI_1DT_Start (CoS,DP,Pattern,Length,Period) MTDe_TT_So_MI_1DT_Terminate MTDe_TT_So_MI_ LMo_OAM_Tool MTDe_TT_So_MI_LMo_Start(CoS,Period) [1...M_{LMo}] MTDe_FT_So_MI_LMo_Terminate[1...M_{LMo}] MTDe_TT_So_MI_ DMo_OAM_Tool MTDe_TT_So_MI_DMo_Start (CoS,Test_ID,Length,Period)[1...M_{DMo}] MTDe_TT_So_MI_DMo_Terminate[1...M_{DMo}] MTDe_TT_So_MI_ 1DMo_OAM_Tool MTDe_TT_So_MI_1DMo_Start (CoS,Test_ID,Length,Period)[1...M_{1DMo}] MTDe_TT_So_MI_1DMo_Terminate[1...M_{1DMo}] MTDe_TT_So_MI_ SLo_OAM_Tool MTDe_TT_So_MI_SLo_Start (CoS,Test_ID,Length,Period)[1...M_{SLo}] MTDe_TT_So_MI_SLo_Terminate[1...M_{SLo}] **MTDe_TT_So_TP:** MTDe_TT_So_TI_ TimeStampl	MTDe_TT_So_MI_LMo_Result(N_TF,N_LF,F_TF,F_LF)[1...M_{LMo}] MTDe_TT_So_MI_DMo_Result(count,B_FD[],F_FD[],N_FD[])[1...M_{DMo}] MTDe_TT_So_MI_SLo_Result(N_TF,N_LF,F_TF,F_LF)[1...M_{SLo}]

Formatted: Font: (Asian) MS Mincho, (Asian) Japanese

Formatted: Font: (Asian) MS Mincho, (Asian) Japanese

Processes

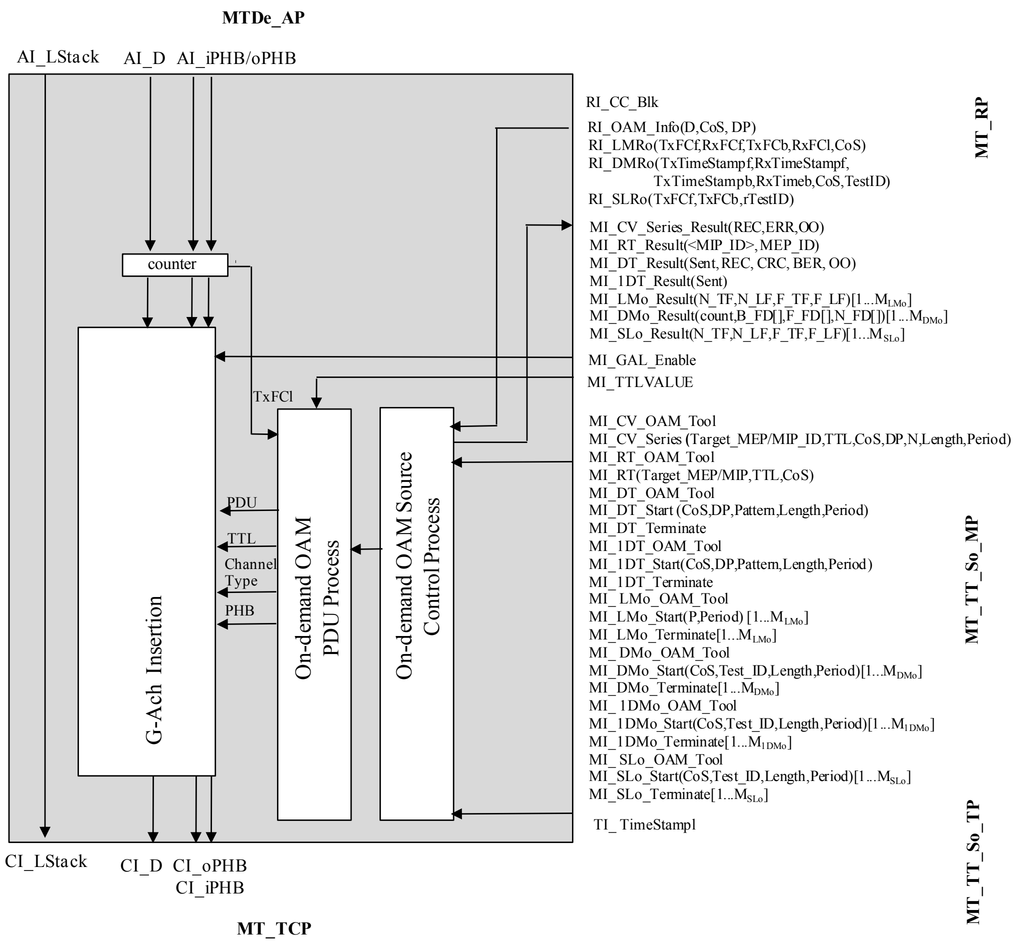


Figure 9-c3 – MTDe_FT_So Process

G-Ach/GAL Insertion process: See 8.1/G.8121.

On-demand 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

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

Symbol

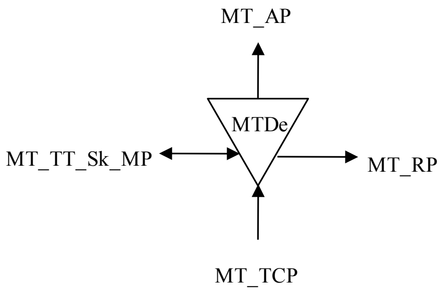


Figure 9-c4 – MTDe_TT_Sk symbol

Interfaces

Table 9-c2 – MTDe_TT_Sk interfaces

Input(s)	Output(s)
MT_TCP: MT_CI_D MT_CI_iPHB MT_CI_oPHB MT_CI_LStack **MT_RP:** **MTDe_TT_Sk_MP:** MTDe_TT_Sk_MI_ GAL_Enable MTDe_TT_Sk_MI_ MEG_ID MTDe_TT_Sk_MI_ PeerMEP_ID MTDe_TT_Sk_MI_CV_OAM_Tool MTDe_TT_Sk_MI_RT_OAM_Tool MTDe_TT_Sk_MI_DT_OAM_Tool MTDe_TT_Sk_MI_1DT_OAM_Tool MTDe_TT_Sk_MI_1DT_Start(Pattern) MTDe_TT_Sk_MI_1DT_Terminate MTDe_TT_Sk_MI_ LMo_OAM_Tool MTDe_TT_Sk_MI_ DMo_OAM_Tool MTDe_TT_Sk_MI_ 1DMo_OAM_Tool MTDe_TT_Sk_MI_1DMo_Start(Test_ID)[1...M_{1DMo}]	**MTDe_AP:** MTDe_AI_D MTDe_AI_oPHB MTDe_AI_iPHB MTDe_AI_LStack **MTDe_RP:** MTDe_RI_OAM_Info(D,CoS,DP) MTDe_RI_LMRo(TxFCf,RxFCf,TxFCb,RxFCl,CoS) MTDe_RI_DMRo(TxTimeStampf,RxTimeStampf,TxTimeStampb,RxTimeb,CoS,TestID) MTDe_RI_SLRo(TxFCf,TxFCb,rTestID) **MTDe_FT_Sk_MP:** MTDe_TT_Sk_MI_1DT_Result(REC,CRC,BER,OO) MTDe_TT_Sk_MI_1DMo_Result(count,N_FD[])[1...M_{DMo}]]

Input(s)	Output(s)
MTDe_TT_Sk_MI_1DMo_Terminate[1...M_{1DMo}] MTDe_TT_Sk_MI_ SLo_OAM_Tool **MTDe_TP:** MTDe_TT_Sk_TI_ TimeStampl	

Processes

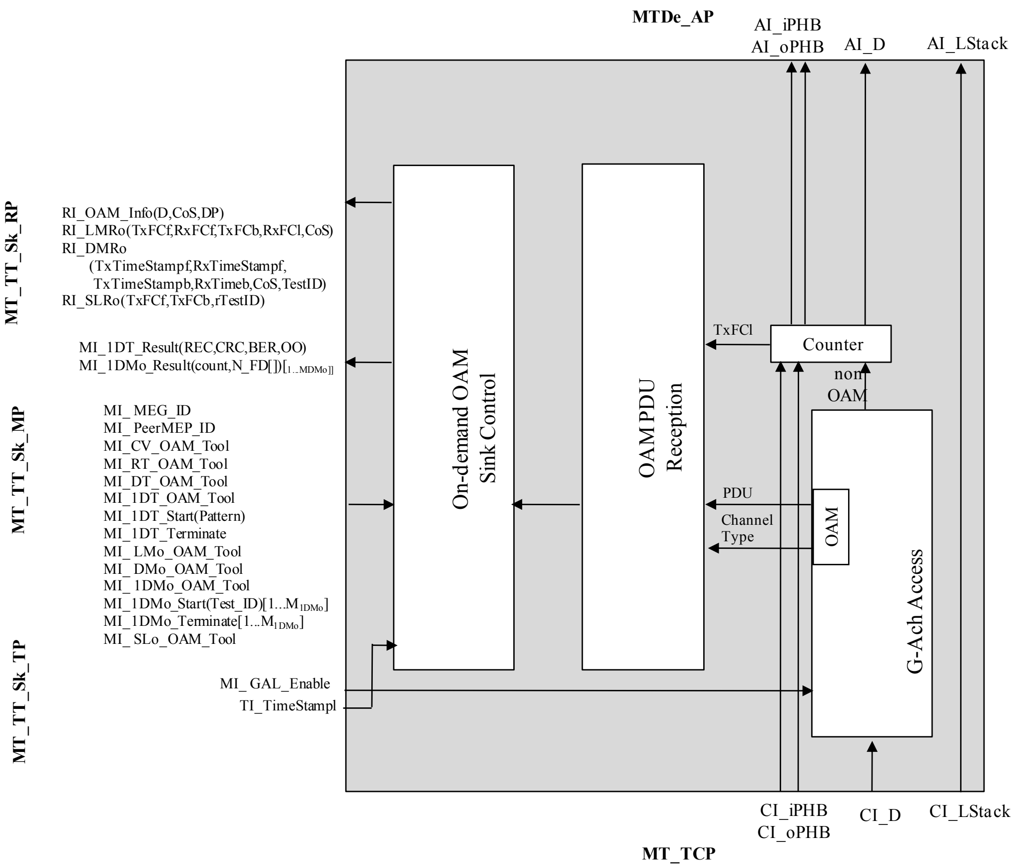


Figure 9-c5 – MTDe_TT_Sk Process

]

G-Ach/GAL Extraction Process: See 8.1/G.8121.

On-demand OAM Sink Control Process: See 8.8/G.8121

OAM PDU Reception Process: See 8.8/G.8121

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

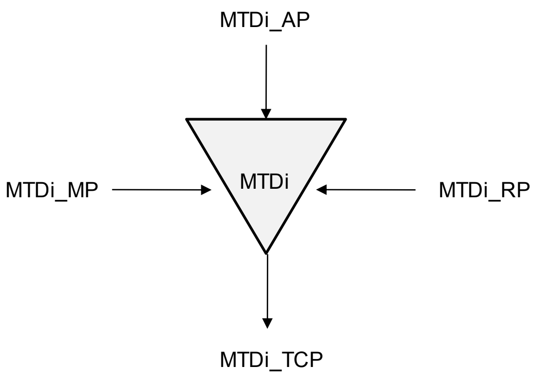


Figure 9-x1 – MTDi_TT_So symbol

Interfaces

Table 9-y1 – MTDi_TT_So interfaces

Inputs	Outputs
MTDi_AP MT_AI_D MT_AI_iPHB MT_AI_oPHB MT_AI_Lstack **MTDi_RP** MT_RI_OAM_Info (D, CoS, DP) **MTDi_TT_So_MP** MTDi_TT_So_MI_GAL_Enable MTDi_TT_So_MI_TTLVALUE MTDi_TT_So_MI_MIP_ID MTDi_TT_So_MI_CV_OAM_Tool MTDi_TT_So_MI_RT_OAM_Tool	**MTDi_TCP** MT_CI_D, MT_CI_iPHB, MT_CI_oPHB, MT_CI_LStack

Processes

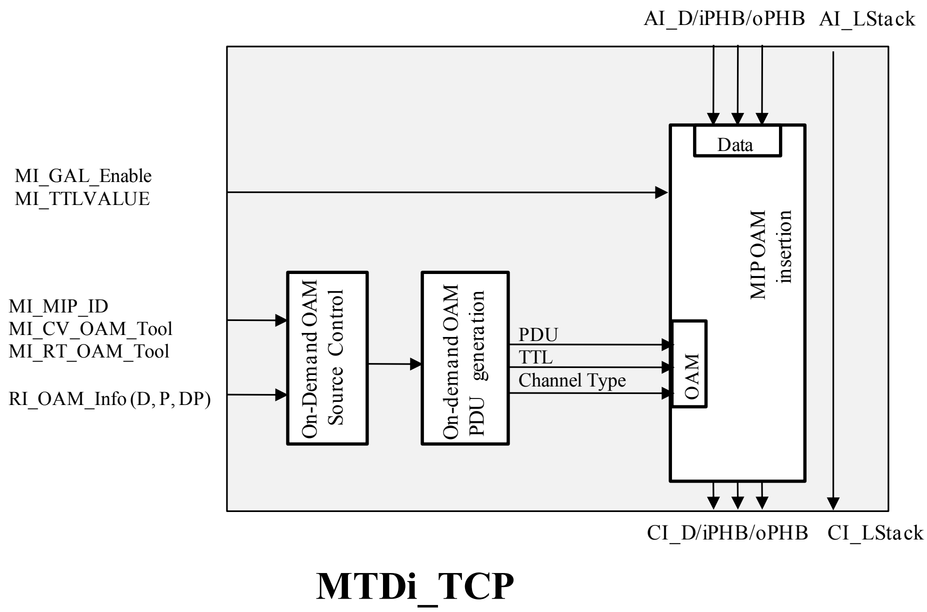


Figure 9-x2 – MTDi_TT_So Process

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

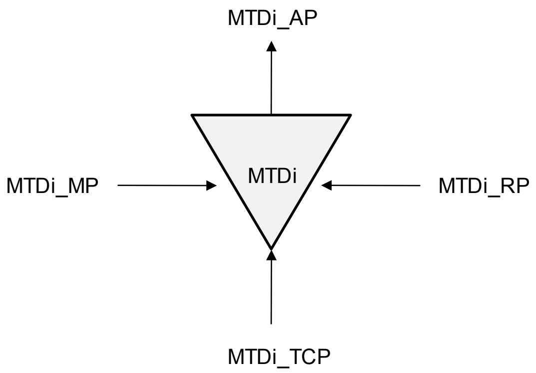


Figure 9-x3 – MTDi_TT_Sk symbol

Interfaces

Table 9-y2 – MTDi_TT_Sk interfaces

Inputs	Outputs
MTDi_TCP MT_CI_D MT_CI_iPHB MT_CI_oPHB MT_CI_LStack **MTDi_TT_Sk_MP** MTDi_TT_Sk_MI_GAL_Enable MTDi_TT_Sk_MI_MIP_ID MTDi_TT_Sk_MI_CV_OAM_Tool MTDi_TT_Sk_MI_RT_OAM_Tool	**MTDi_AP** MT_AI_D MT_AI_iPHB MT_AI_oPHB MT_AI_LStack **MTDi_RP** MT_RI_OAM_Info (D, CoS, DP)

Processes

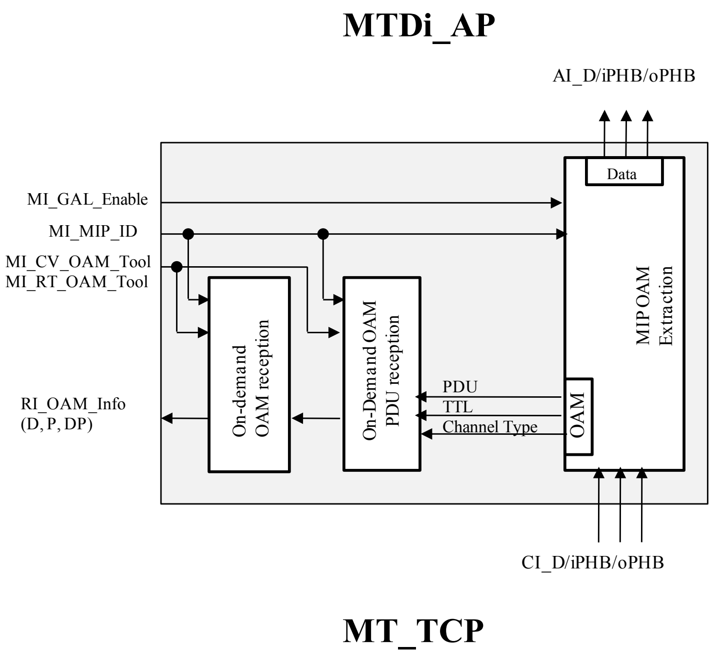


Figure 9-x4 – MTDi_TT_Sk Process

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.

Defects	None.
Consequent actions	None.
Defect correlations	None.
Performance monitoring	None**.**

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-xx.

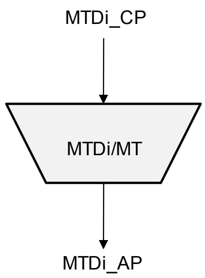


Figure 9-x5 – MTDi/MT_A_So symbol

Interfaces

Table 9-y3 – MTDi/MT_A_So interfaces

Inputs	Outputs
MT_CP: MT_CI_D MT_CI_iPHB MT_CI_oPHB MT_CI_LStack	**MT_AP:** MT_AI_D MT_AI_iPHB MT_AI_oPHB MT_AI_LStack

Processes

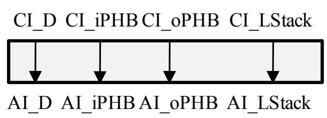


Figure 9-x6 – MTDi/MT_A_So Process

Formatted: Heading 4,h4,12,4, Indent: Before: 0 cm, First line: 0 cm, Tab stop Not at 2 cm

Formatted: Heading 5,h5,5, Indent: Before: 0 cm, First line: 0 cm, Tab stop Not at 2 cm

Formatted: German (Germany)

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)

Formatted: Heading 5,h5,5, Indent: Before: 0 cm, First line: 0 cm

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

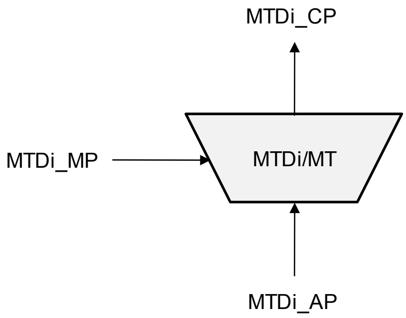


Figure 9-x7 – MTDi/MT_A_Sk symbol

Interfaces

Table 9-y4 –MTDi/MT_A_Sk interfaces

Inputs	Outputs
MT_AP: MT_AI_D MT_AI_iPHB MT_AI_oPHB MT_AI_LStack **MT_MP:** MI_DS_MP_Type	**MT_CP:** MT_AI_D MT_AI_iPHB MT_AI_oPHB MT_AI_LStack

Formatted: German (Germany)

Processes

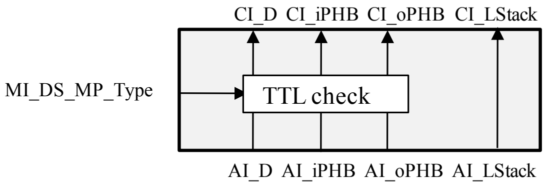


Figure 9-x8 – MTDi/MT_A_Sk Process

TTL check process:

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 39.

• Symbol:

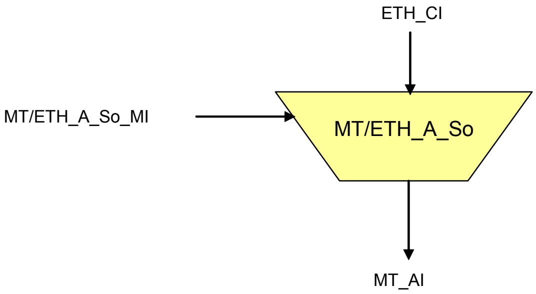


Figure 39/G.8121/Y.1381 – MT/ETH_A_So function

• **Interfaces:**

Table 8/G.8121/Y.1381 – MT/ETH_A_So Inputs and Outputs

Inputs	Outputs
ETH_FP: ETH_CI_Data ETH_CI_P ETH_CI_DE **MT/ETH_A_So_MP:** MT/ETH_A_So_MI_AdminState MT/ETH_A_So_MI_FCSEnable MT/ETH_A_So_MI_CWEnable MT/ETH_A_So_MI_SQUse MT/ETH_A_So_MI_PRI2PSCMapping MT/ETH_A_So_MI_MEP_MAC* MT/ETH_A_So_MI_Client_MEL* MT/ETH_A_So_MI_LCK_Period* MT/ETH_A_So_MI_LCK_Pri* MT/ETH_A_So_MI_MEL* * ETH OAM related	**MT_AP:** MT_AI_Data MT_AI_PHB

• **Processes:**

The processes associated with the MT/ETH_A_So function are as depicted in Figure 40.

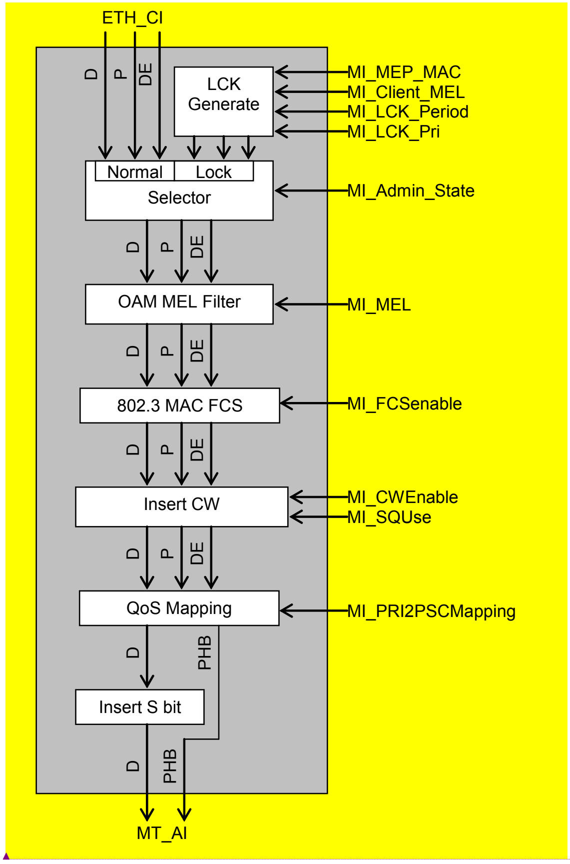


Figure 40/G.8121/Y.1381 – MT/ETH_A_So process diagram (PSC→CoS)

Formatted: Highlight

– *LCK Generate process:*

See 8.1.2/G.8021/Y.1341.

– *Selector process:*

See 8.1.3/G.8021/Y.1341. The normal CI is blocked if Admin_State = LOCKED.

– *OAM MEL Filter process:*

See 8.1.1/G.8021/Y.1341.

– *802.3 MAC FCS generation:*

See 8.8.1/G.8021/Y.1341. 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:

```
If CI_DE = True
     DP(AI_PHB) = Yellow
Else
     DP(AI_PHB) = Green
```

– *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 41.

• Symbol:

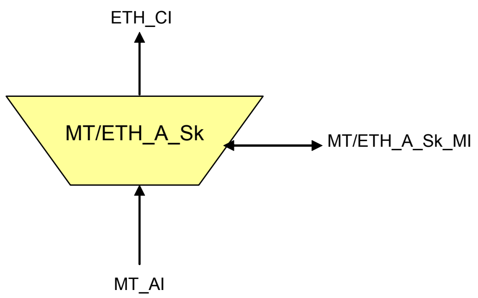


Figure 41/G.8121/Y.1381 – MT/ETH_A_Sk function

• Interfaces:

Table 9/G.8121/Y.1381 – MT/ETH_A_Sk Inputs and Outputs

Inputs	Outputs
Each MT_AP: MT_AI_Data MT_AI_PHB MT_AI_TSF **MT/ETH_A_Sk_MP:** MT/ETH_A_Sk_MI_FCSEnable MT/ETH_A_Sk_MI_CIIEnable MT/ETH_A_So_MI_SQUse MT/ETH_A_Sk_MI_CoS2PRIMapping MT/ETH_A_Sk_MI_Admin_State MT/ETH_A_Sk_MI_LCK_Period * MT/ETH_A_Sk_MI_LCK_Pri * MT/ETH_A_Sk_MI_Client_MEL * MT/ETH_A_Sk_MI_MEP_MAC * MT/ETH_A_Sk_MI_AIS_Pri * MT/ETH_A_Sk_MI_AIS_Period * * ETH OAM related	**ETH_FP:** ETH_CI_Data ETH_CI_P ETH_CI_DE ETH_CI_SSF

• **Processes:**

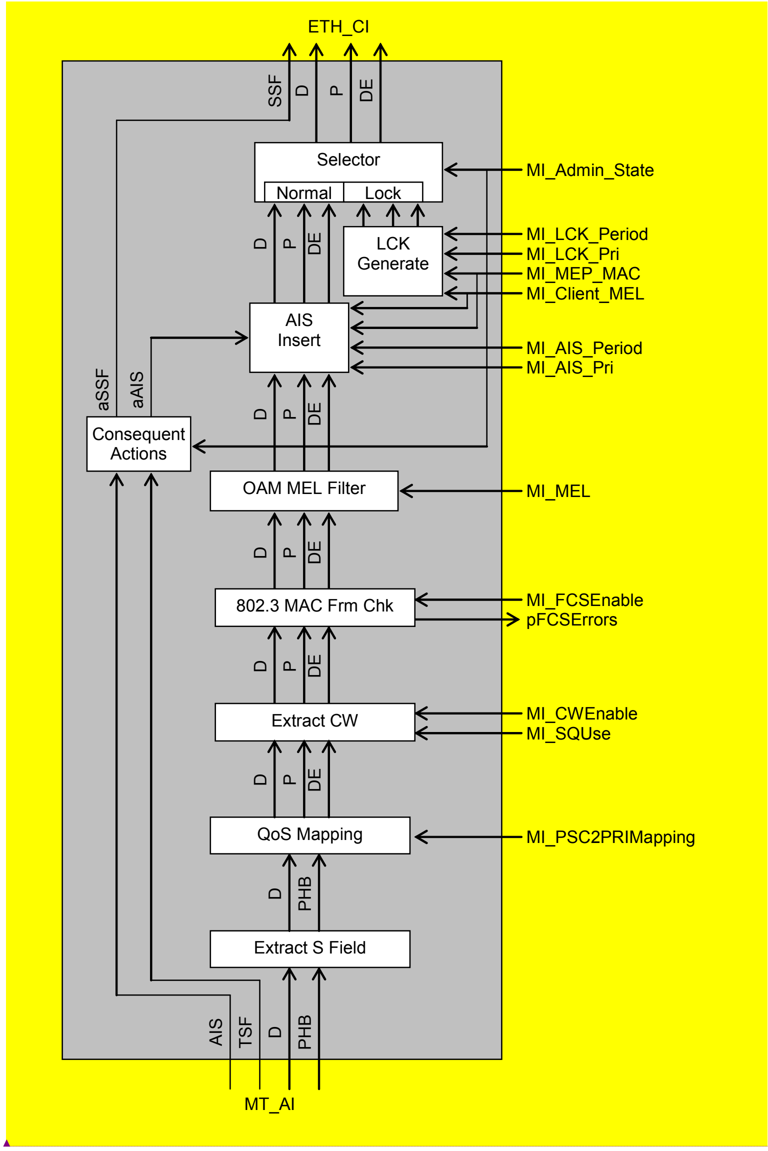


Figure 42/G.8121/Y.1381 – MT/ETH_A_Sk process diagram (PSC→CoS)

Formatted: Highlight

Formatted: Highlight

Formatted: Highlight

Formatted: Highlight

Formatted: Font: (Asian) MS Mincho, English (U.S.)

– *Selector process:*

See 8.1.3/G.8021/Y.1341. The normal CI is blocked if Admin_State = LOCKED.

– *LCK Generate process:*

See 8.1.2/G.8021/Y.1341.

– *AIS Insert process:*

See 8.1.4/G.8021/Y.1341.

– *OAM MEL Filter process:*

See 8.1.1/G.8021/Y.1341.

– *"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

```
If DP(AI_PHB) = Green
     CI_DE = False
Else
     CI_DE = 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.

Formatted: Not Highlight

Formatted: Not Highlight

Formatted: English (U.K.)

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-A.1 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.

It shall be noted that the MT/MCC adaptation function, the MT/SCC adaptation function, and the MT/MT adaptation function in Figure 10-A.1 are described as a single combined MT/MT adaptation function in [ITU-T G.8110.1].

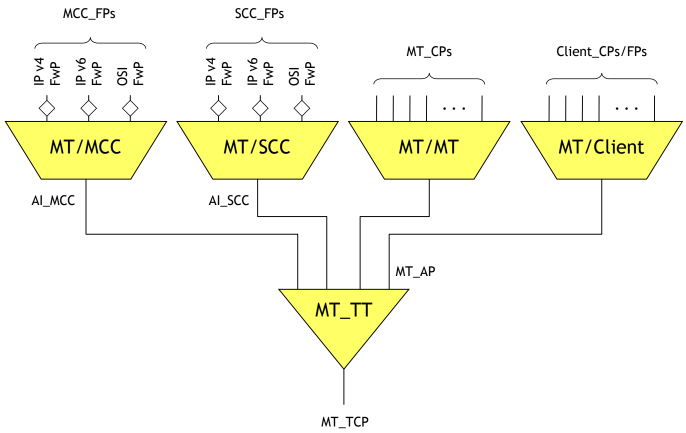


Figure 10-A.1 – MT/SCC_A function, MT/MCC_A function, MT/MT_A function, and MT/client_A function

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-A.2 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-A.2 and 10-A.3.

Symbol

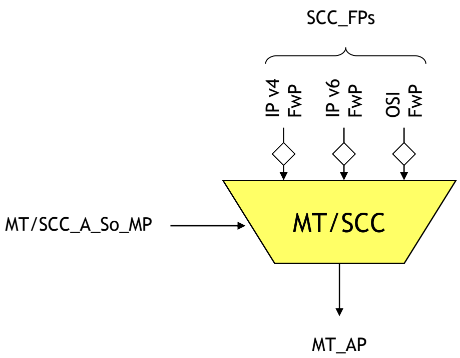


Figure 10-A.2 – MT/SCC_A_So function

Interfaces

Table 10-A.1 – MT/SCC_A_So inputs and outputs

Input(s)	Output(s)
SCC_FP: SCC_CI_D **MT/SCC_A_So_MP**: MT/SCC_A_So_MI_Active	**MT_AP**: MT_AI_SCC

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-A.3.

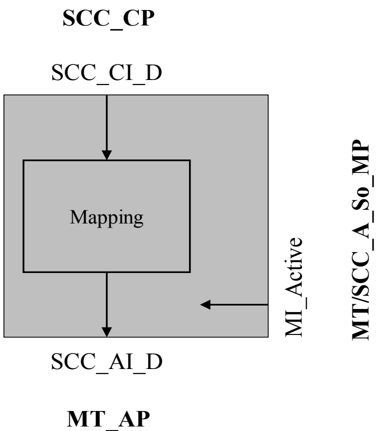


Figure 10-A.3 – MT/SCC_A_So processes

ECC Mapping process: See clause 8.7.1.1/G.8121

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-A.4 and 10-A.5.

Symbol

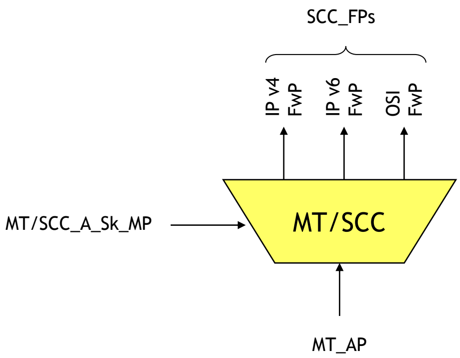


Figure 10-A.4 – MT/SCC_A_Sk function

Interfaces

Table 10-A.2 – MT/SCC_A_Sk inputs and outputs

Input(s)	Output(s)
MT_AP: MT_AI_SCC MT_AI_TSF **MT/SCC_A_Sk_MP**: MT/SCC_A_Sk_MI_Active	**SCC_FP**: SCC_CI_D SCC_CI_SSF

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-A.5.

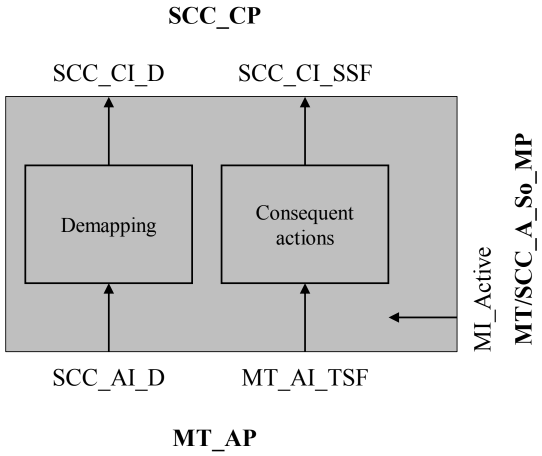


Figure 10-A.5 – MT/SCC_A_Sk processes

ECC Demapping process: See clause 8.7.1.2/G.8121

Defects: None.

Consequent actions

The function shall perform the following consequent actions:

aSSF ← 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-A.6 and 10-A.7.

Symbol

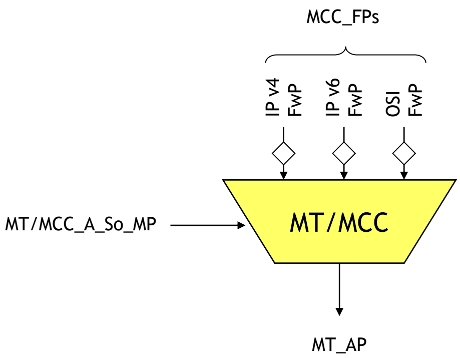


Figure 10-A.6 – MT/MCC_A_So function

Interfaces

Table 10-A.3 – MT/MCC_A_So inputs and outputs

Input(s)	Output(s)
MCC_FP: MCC_CI_D **MT/MCC_A_So_MP**: MT/MCC_A_So_MI_Active	**MT_AP**: MT_AI_MCC

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-A.7.

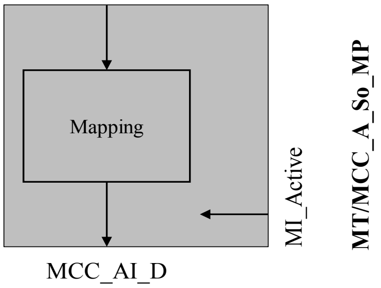


Figure 10-A.7 – MT/MCC_A_So processes

MCC Mapping process: See clause 8.7.1.1/G.8121

Defects: None.

Consequent actions: None.

Defect correlations: None.

Performance Monitoring: None.

10.2.2.2 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-A.8 and 10-A.9.

Symbol

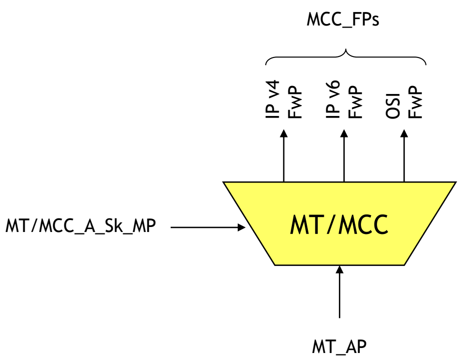


Figure 10-A.8 – MT/MCC_A_Sk function

Interfaces

Table 10-A.4 – MT/MCC_A_Sk inputs and outputs

Input(s)	Output(s)
MT_AP: MT_AI_MCC MT_AI_TSF **MT/MCC_A_Sk_MP**: MT/MCC_A_Sk_MI_Active	**MCC_FP**: MCC_CI_D MCC_CI_SSF

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-A.9.

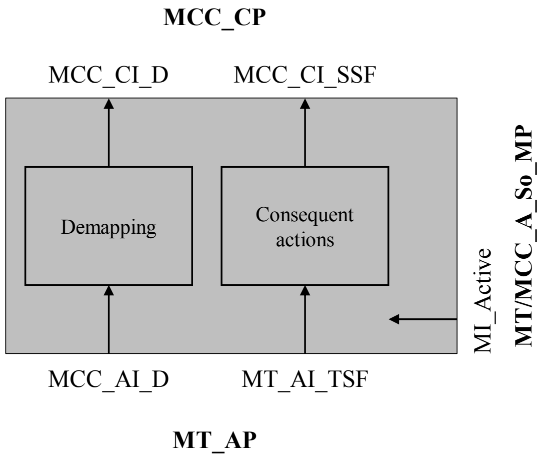


Figure 10-A.9 – MT/MCC_A_Sk processes

MCC Demapping process: See clause 8.7.1.2/G.8121

Defects: None.

Consequent actions

The function shall perform the following consequent actions:

aSSF ← AI_TSF or (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 (S*n*/MT_A; *n*=3, 3-X, 4, 4-X)

11.1.1.1 VC-n to MPLS-TP adaptation source function (S*n*/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 Rec. G.707/Y.1322, but with indeterminate POH bytes: J1, B3, G1.

• Symbol:

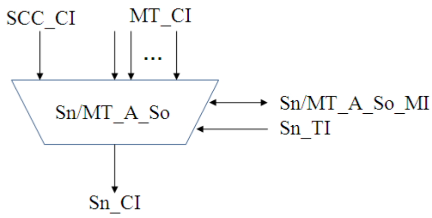


Figure 43/G.8121/Y.1381 – Sn/MT_A_So symbol

• Interfaces:

Table 10/G.8121/Y.1381 – Sn/MT_A_So interfaces

Inputs	Outputs
Each MT_CP: MT_CI_Data MT_CI_iPHB MT_CI_oPHB **SCC_CP:** SCC_CI_Data **Sn_TP:** Sn_TI_Clock Sn_TI_FrameStart **Sn/MT_A_So_MP:** Sn/MT_A_So_MI_SCCType Sn/MT_A_So_MI_Label[1…M] Sn/MT_A_So_MI_LSPType[1…M] Sn/MT_A_So_MI_CoS[1…M] Sn/MT_A_So_PHB2TCMapping[1…M] Sn/MT_A_So_MI_QoSEncodingMode[1…M]	**Sn_AP:** Sn_AI_Data Sn_AI_Clock Sn_AI_FrameStart

• Processes:

A process diagram of this function is shown in Figure 44.

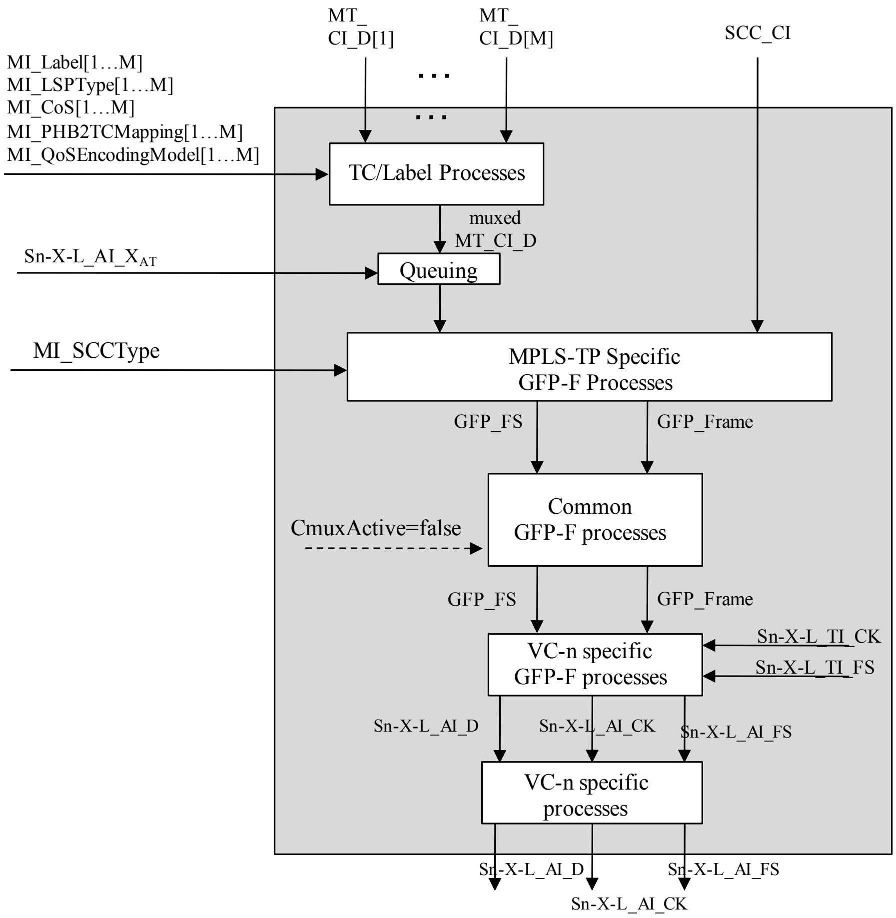


Figure 44/G.8121/Y.1381 – Sn/MT_A_So process diagram

– *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-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 (S*n*/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 Rec. G.707/Y.1322.

• Symbol:

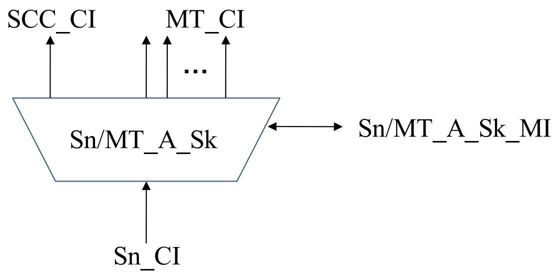


Figure 45/G.8121/Y.1381 – Sn/MT_A_Sk symbol

• **Interfaces:**

Table 11/G.8121/Y.1381 – Sn/MT_A_Sk interfaces

Inputs	Outputs
Sn_AP: Sn_AI_Data Sn_AI_ClocK Sn_AI_FrameStart Sn_AI_TSF **Sn/MT_A_Sk_MP:** Sn/MT_A_Sk_MI_SCCType Sn/MT_A_Sk_MI_Label[1…M] Sn/MT_A_Sk_MI_LSPType[1…M] Sn/MT_A_Sk_MI_CoS[1…M] Sn/MT_A_Sk_MI_TC2PHBMapping[1…M] Sn/MT_A_Sk_MI_QoSDecodingMode[1…M] Sn/MT_A_Sk_MI _LCK_Period[1…M] Sn/MT_A_Sk_MI _LCK_CoS[1…M] Sn/MT_A_Sk_MI _Admin_State Sn/MT_A_Sk_MI _AIS_Period[1…M] Sn/MT_A_Sk_MI _AIS_CoS[1…M]	**Each MT_CP:** MT_CI_Data MT_CI_iPHB MT_CI_oPHB MT_CI_SSF MT_CI_LStack **SCC_CP:** SCC_CI_Data SCC_CI_SSF **Sn/MT_A_Sk_MP:** Sn/MT_A_Sk_MI_AcSL Sn/MT_A_Sk_MI_AcEXI Sn/MT_A_Sk_MI_LastValidUPI Sn/MT_A_Sk_MI_cPLM Sn/MT_A_Sk_MI_cLFD Sn/MT_A_Sk_MI_cEXM Sn/MT_A_Sk_MI_cUPM

• Processes:

A process diagram of this function is shown in Figure 46.

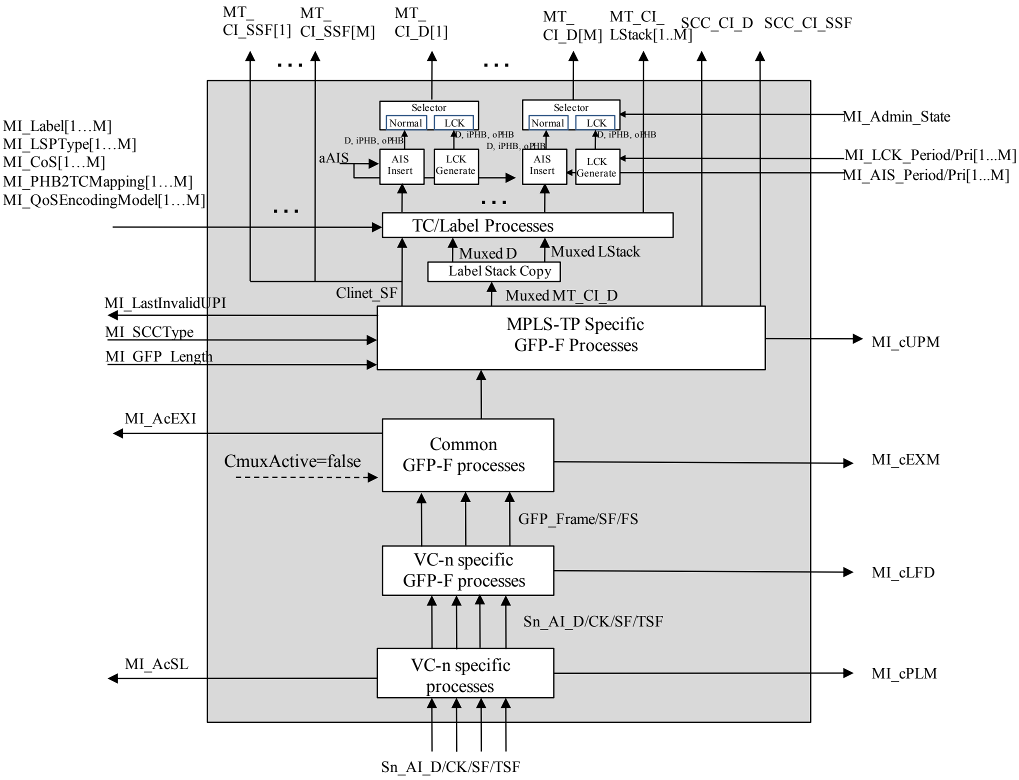


Figure 46/G.8121/Y.1381 – Sn/MT_A_Sk process diagram

– *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/G.806. GFP channel multiplexing is not supported (CMuxActive=false).

– *VC-n specific GFP sink process:*

See 8.5.2.2/G.806. The GFP frames are demapped from the VC-n payload area according to 10.6/G.707/Y.1322.

– *VC-n-specific sink process:*

C2: The signal label is recovered from the C2 byte as per 6.2.4.2/G.806. The signal label for "GFP mapping" in Table 9-11/G.707/Y.1322 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/G.806.

dLFD – See 6.2.5.2/G.806.

dEXM – See 6.2.4.4/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 (S*n*-X-L/MT_A; *n*=3, 4)

11.1.2.1 LCAS-capable VC-n to MPLS-TP adaptation source function (S*n*-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 Rec. G.707/Y.1322, but with indeterminate POH bytes: J1, B3, G1.

• Symbol:

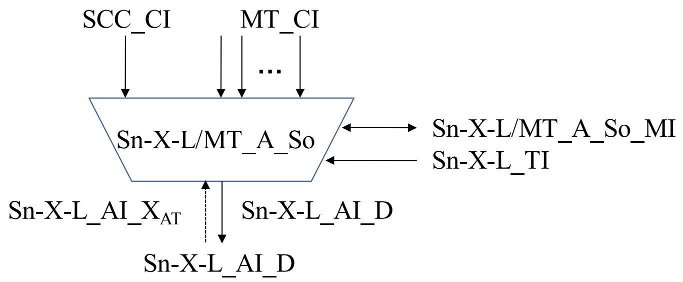


Figure 47/G.8121/Y.1381 – Sn-X-L/MT_A_So symbol

• Interfaces:

Table 12/G.8121/Y.1381 – Sn-X-L/MT_A_So interfaces

Inputs	Outputs
Each MT_CP: MT_CI_Data MT_CI_iPHB MT_CI_oPHB **SCC_CP:** SCC_CI_Data **Sn-X-L_AP:** Sn-X-L_AI_X_{AT} **Sn-X-L_TP:** Sn-X-L_TI_Clock Sn-X-L_TI_FrameStart **Sn-X-L/MT_A_So_MP:** Sn-X-L/MT_A_So_MI_SCCType Sn-X-L/MT_A_So_MI_Label[1…M] Sn-X-L/MT_A_So_MI_LSPType[1…M] Sn-X-L/MT_A_So_MI_CoS[1…M] Sn-X-L/MT_A_So_PHB2TCMapping[1…M] Sn-X-L/MT_A_So_MI_QoSEncodingMode[1…M]	**Sn-X-L_AP:** Sn-X-L_AI_Data Sn-X-L_AI_Clock Sn-X-L_AI_FrameStart

• Processes:

A process diagram of this function is shown in Figure 48.

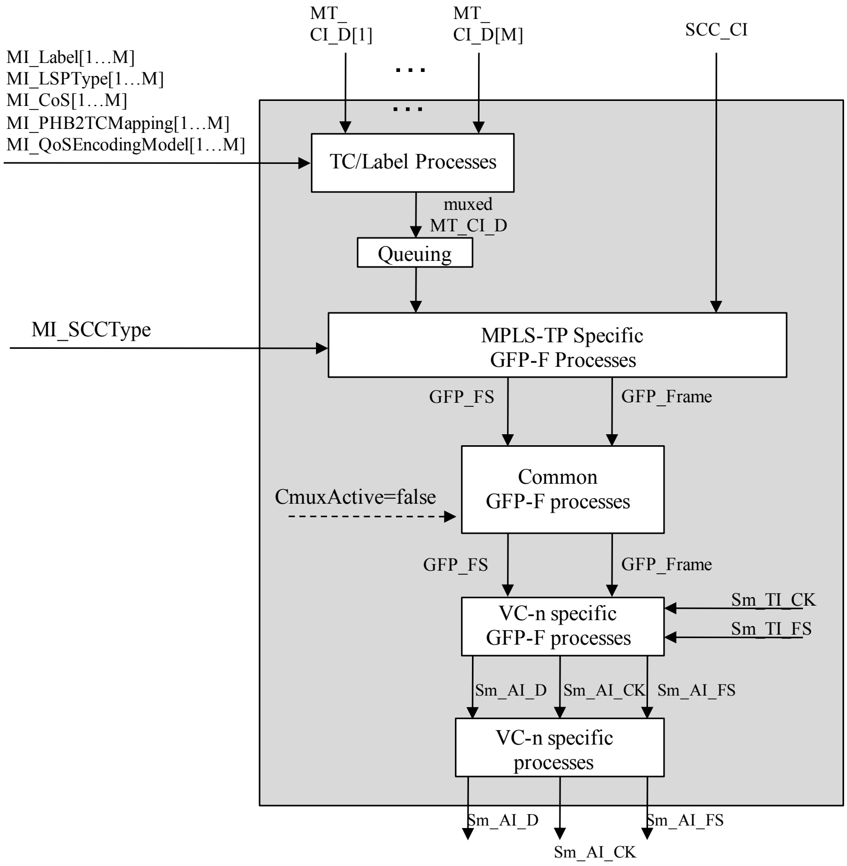


Figure 48/G.8121/Y.1381 – Sn-X-L/MT_A_So process diagram

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 (S*n*-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 Rec. G.707/Y.1322.

• Symbol:

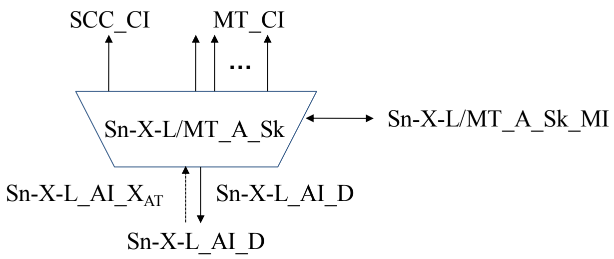


Figure 49/G.8121/Y.1381 – Sn-X-L/MT_A_Sk symbol

• Interfaces:

Table 13/G.8121/Y.1381 – Sn-X-L/MT_A_Sk interfaces

Inputs	Outputs
Sn-X-L_AP: Sn-X-L_AI_Data Sn-X-L_AI_ClocK Sn-X-L_AI_FrameStart Sn-X-L_AI_TSF Sn-X-L_AI_X$_{AR}$ **Sn-X-L/MT_A_Sk_MP:** Sn-X-L/MT_A_Sk_MI_SCCType Sn-X-L/MT_A_Sk_MI_Label[1…M] Sn-X-L/MT_A_Sk_MI_LSPType[1…M] Sn-X-L/MT_A_Sk_MI_CoS[1…M] Sn-X-L/MT_A_Sk_MI_TC2PHBMapping[1…M] Sn-X-L/MT_A_Sk_MI_QoSDecodingMode[1…M] Sn-X-L /MT_A_Sk_MI _LCK_Period[1…M] Sn-X-L /MT_A_Sk_MI _LCK_CoS[1…M] Sn-X-L /MT_A_Sk_MI _Admin_State Sn-X-L /MT_A_Sk_MI _AIS_Period[1…M] Sn-X-L /MT_A_Sk_MI _AIS_CoS [1…M]	**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 **Sn-X-L/MT_A_Sk_MP:** Sn-X-L/MT_A_Sk_MI_AcSL Sn-X-L/MT_A_Sk_MI_AcEXI Sn-X-L/MT_A_Sk_MI_LastValidUPI Sn-X-L/MT_A_Sk_MI_cPLM Sn-X-L/MT_A_Sk_MI_cLFD Sn-X-L/MT_A_Sk_MI_cEXM Sn-X-L/MT_A_Sk_MI_cUPM

• Processes:

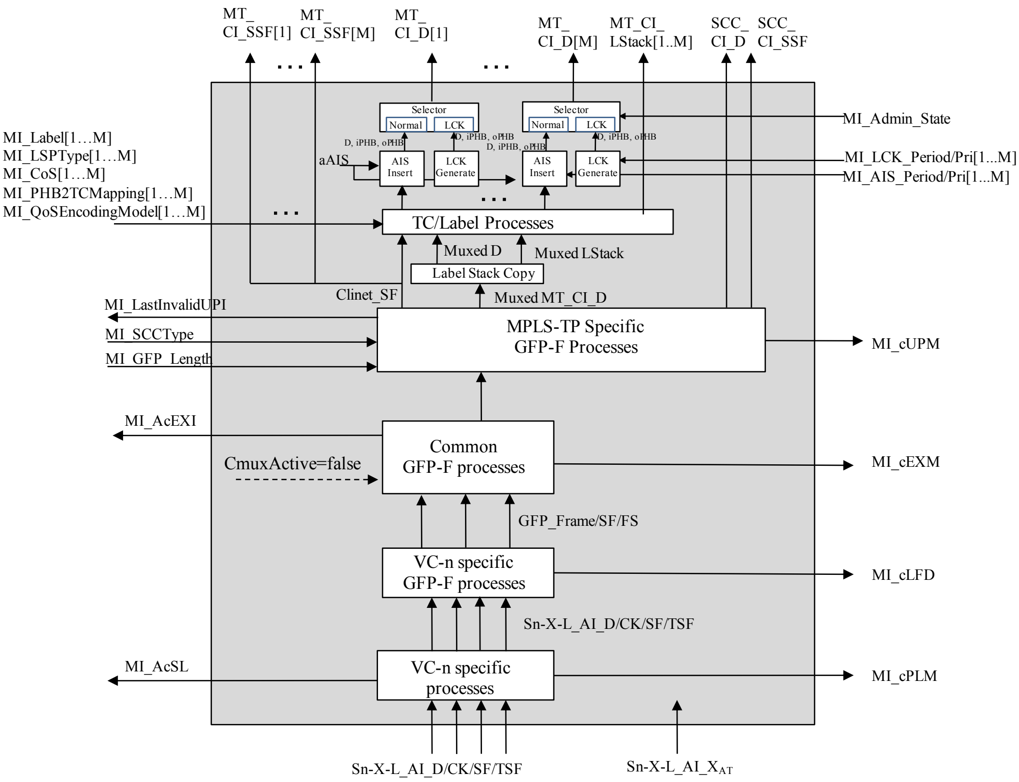


Figure 50/G.8121/Y.1381 – Sn-X-L/MT_A_Sk process diagram

See process diagram and process description in 11.1.1.2. The additional Sn-X-L_AI_X_{AR} interface is not connected to any of the internal processes.

• Defects:

dPLM – See 6.2.4.2/G.806.

dLFD – See 6.2.5.2/G.806.

dUPM – See 8.4.2.

dEXM – See 6.2.4.4/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.

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

11.1.3.1 VC-m to MPLS-TP adaptation source function (S*m*/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 Rec. G.707/Y.1322, but with indeterminate POH bytes: J2, V5[1-4], V5[8].

• **Symbol:**

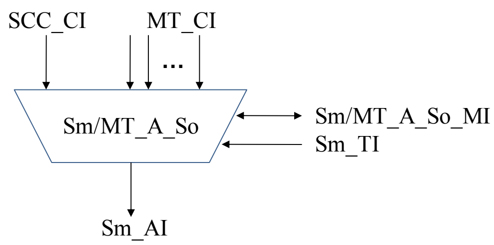


Figure 51/G.8121/Y.1381 – Sm/MT_A_So symbol

• Interfaces:

Table 14/G.8121/Y.1381 – Sm/MT_A_So interfaces

Inputs	Outputs
Each MT_CP: MT_CI_Data MT_CI_iPHB MT_CI_oPHB **SCC_CP:** SCC_CI_Data **Sm_TP:** Sm_TI_Clock Sm_TI_FrameStart **Sm/MT_A_So_MP:** Sm/MT_A_So_MI_SCCType Sm/MT_A_So_MI_Label[1…M] Sm/MT_A_So_MI_LSPType[1…M] Sm/MT_A_So_MI_CoS[1…M] Sm/MT_A_So_PHB2TCMapping[1…M] Sm/MT_A_So_MI_QoSEncodingMode[1…M]	**Sm_AP:** Sm_AI_Data Sm_AI_Clock Sm_AI_FrameStart

• Processes:

A process diagram of this function is shown in Figure 52.

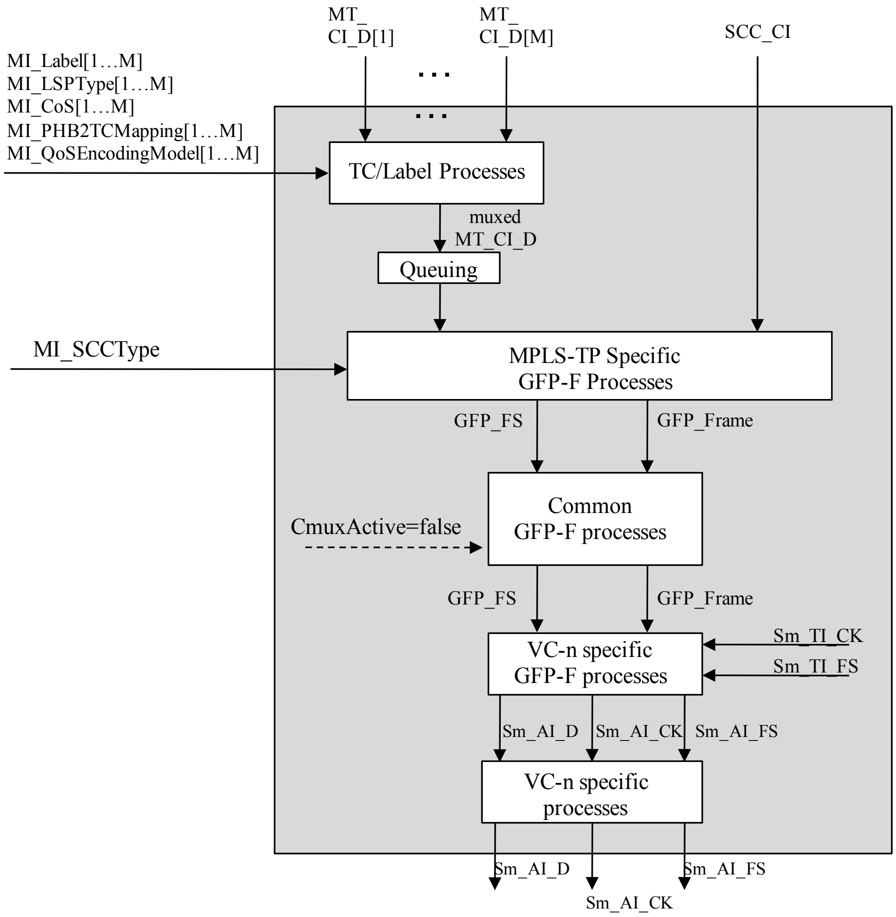


Figure 52/G.8121/Y.1381 – Sm/MT_A_So process diagram

– *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 Rec. G.707/Y.1322.

• Symbol:

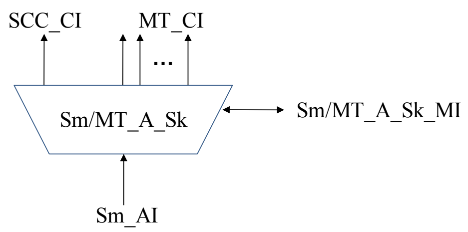


Figure 53/G.8121/Y.1381 – Sm/MT_A_Sk symbol

• Interfaces:

Table 15/G.8121/Y.1381 – Sm/MT_A_Sk interfaces

Inputs	Outputs
Sm_AP: Sm_AI_Data Sm_AI_ClocK Sm_AI_FrameStart Sm_AI_TSF **Sm/MT_A_Sk_MP:** Sm/MT_A_Sk_MI_SCCType Sm/MT_A_Sk_MI_Label[1…M] Sm/MT_A_Sk_MI_LSPType[1…M] Sm/MT_A_Sk_MI_CoS[1…M] Sm/MT_A_Sk_MI_TC2PHBMapping[1…M] Sm/MT_A_Sk_MI_QoSDecodingMode[1…M] Sm/MT_A_Sk_MI _LCK_Period[1…M] Sm/MT_A_Sk_MI _LCK_P[1…M] Sm/MT_A_Sk_MI _Admin_State Sm/MT_A_Sk_MI _AIS_Period[1…M] Sm/MT_A_Sk_MI _AIS_P[1…M]	**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 54.

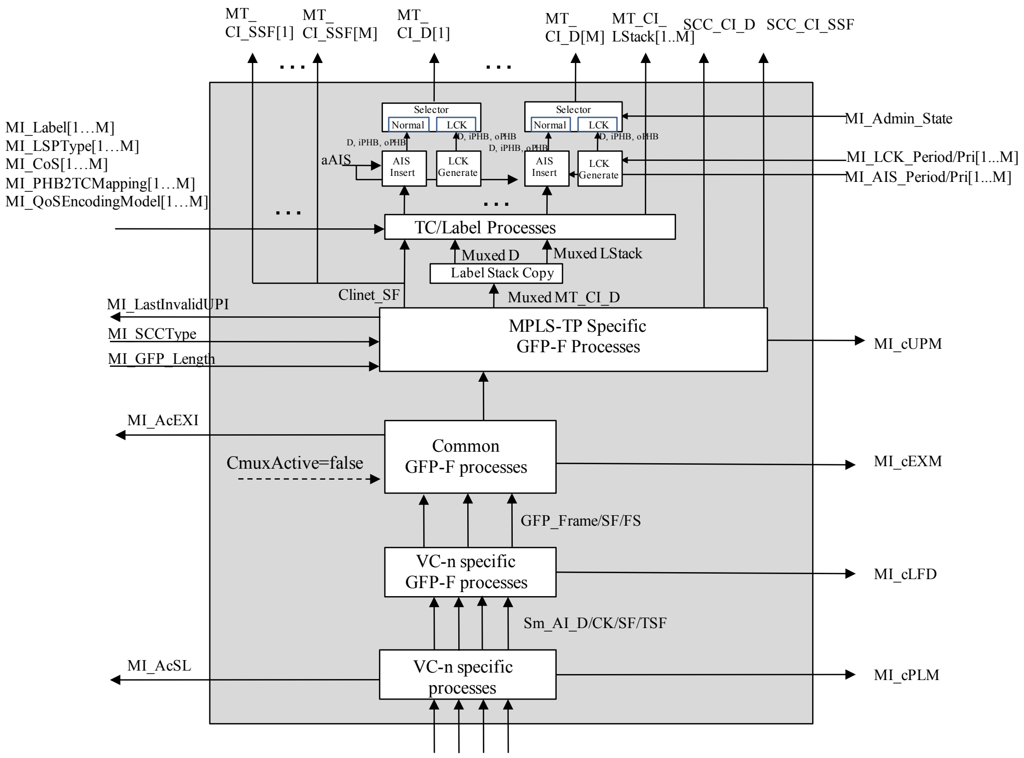


Figure 54/G.8121/Y.1381 – Sm/MT_A_Sk process diagram

– *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/G.806. GFP channel multiplexing is not supported (CMuxActive=false).

– *VC-m-specific GFP sink process:*

See 8.5.2.2/G.806. The GFP frames are demapped from the VC-m payload area according to 10.6/G.707/Y.1322.

– *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/G.783 and 6.2.4.2/G.806. The signal label for "GFP mapping" in Table 9-13/G.707/Y.1322 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/G.806.

dLFD – See 6.2.5.2/G.806.

dUPM – See 8.4.2.

dEXM – See 6.2.4.4/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 Rec. G.707/Y.1322, but with indeterminate POH bytes: J2, V5[1-4], V5[8].

• Symbol:

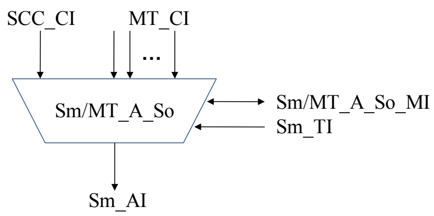


Figure 55/G.8121/Y.1381 – Sm-X-L/MT_A_So symbol

• Interfaces:

Table 16/G.8121/Y.1381 – Sm-X-L/MT_A_So interfaces

Inputs	Outputs
Each MT_CP: MT_CI_Data MT_CI_iPHB MT_CI_oPHB **SCC_CP:** SCC_CI_Data **Sm-X-L_AP:** Sm-X-L_AI_X_{AT} **Sm-X-L_TP:** Sm-X-L_TI_Clock Sm-X-L_TI_FrameStart **Sm-X-L/MT_A_So_MP:** Sm-X-L/MT_A_So_MI_SCCType Sm-X-L/MT_A_So_MI_Label[1…M] Sm-X-L/MT_A_So_MI_LSPType[1…M] Sm-X-L/MT_A_So_MI_CoS[1…M] Sm-X-L/MT_A_So_PHB2TCMapping[1…M] Sm-X-L/MT_A_So_MI_QoSEncodingMode[1…M]	**Sm-X-L_AP:** Sm-X-L_AI_Data Sm-X-L_AI_Clock Sm-X-L_AI_FrameStart

• Processes:

A process diagram of this function is shown in Figure 56.

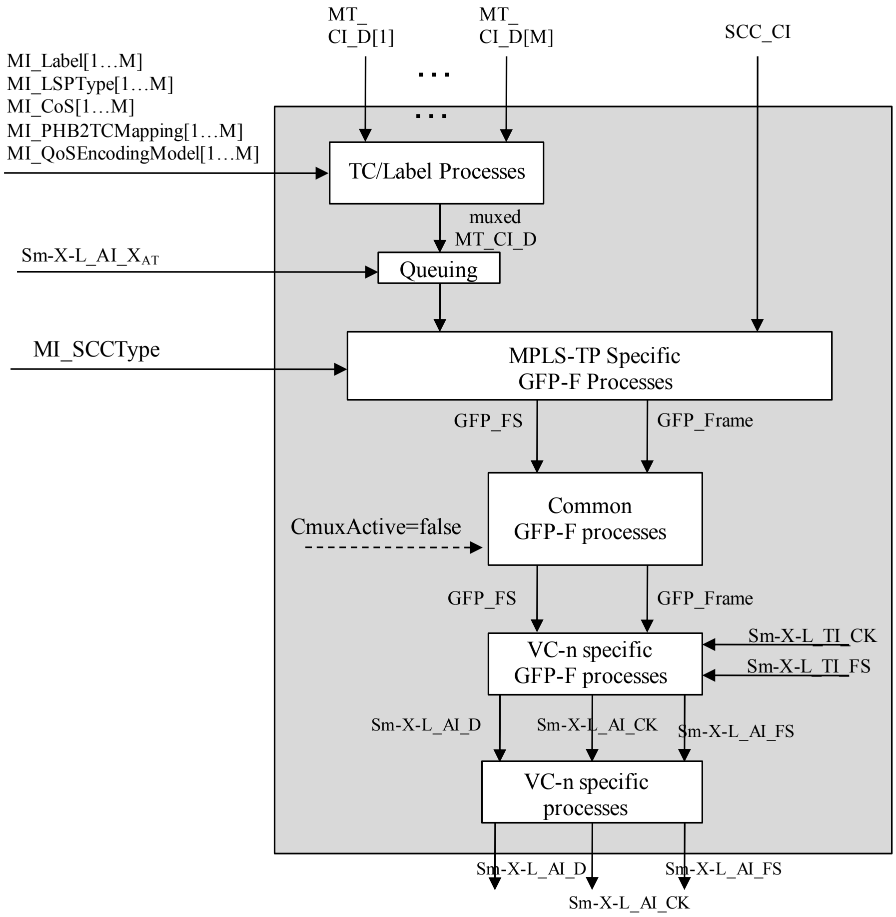


Figure 56/G.8121/Y.1381 – Sm-X-L/MT_A_So process diagram

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 Rec. G.707/Y.1322.

• Symbol:

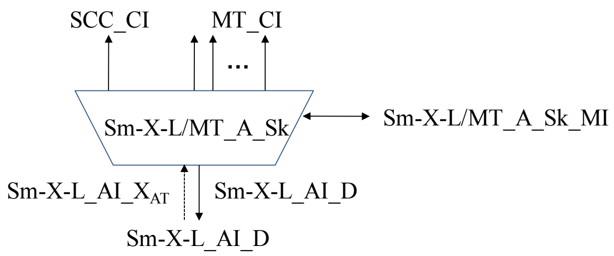


Figure 57/G.8121/Y.1381 – Sm-X-L/MT_A_Sk symbol

• Interfaces:

Table 17/G.8121/Y.1381 – Sm-X-L/MT_A_Sk interfaces

Inputs	Outputs
Sm-X-L_AP: Sm-X-L_AI_Data Sm-X-L_AI_ClocK Sm-X-L_AI_FrameStart Sm-X-L_AI_TSF Sm-X-L_AI_X$_{AR}$ **Sm-X-L/MT_A_Sk_MP:** Sm-X-L/MT_A_Sk_MI_SCCType Sm-X-L/MT_A_Sk_MI_Label[1…M] Sm-X-L/MT_A_Sk_MI_LSPType[1…M] Sm-X-L/MT_A_Sk_MI_CoS[1…M] Sm-X-L/MT_A_Sk_MI_TC2PHBMapping[1…M] Sm-X-L/MT_A_Sk_MI_QoSDecodingMode[1…M] Sm-X-L/MT_A_Sk_MI _LCK_Period[1…M] Sm-X-L/MT_A_Sk_MI _LCK_P[1…M] Sm-X-L/MT_A_Sk_MI _Admin_State Sm-X-L/MT_A_Sk_MI _AIS_Period[1…M] Sm-X-L/MT_A_Sk_MI _AIS_P[1…M]	**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-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:**

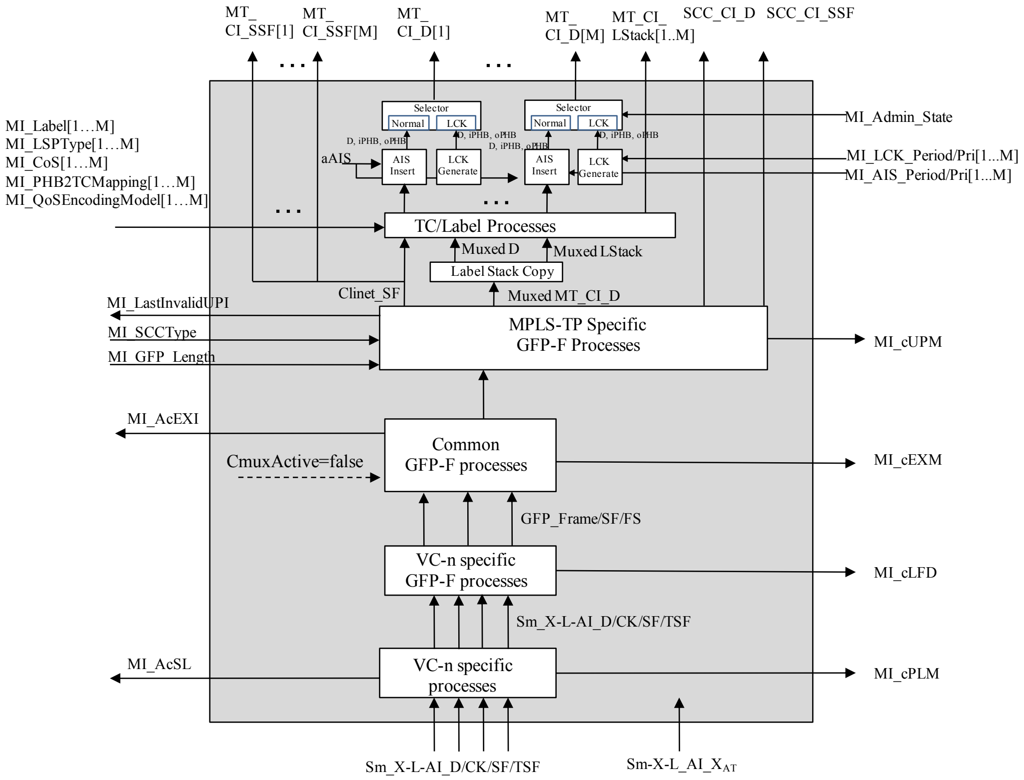


Figure 58/G.8121/Y.1381 – Sm-X-L/MT_A_Sk process diagram

See process diagram and process description in 11..1.1.2. The additional Sm-X-L_AI_X_{AR} interface is not connected to any of the internal processes.

• **Defects:**

dPLM – See 6.2.4.2/G.806.

dLFD – See 6.2.5.2/G.806.

dUPM – See 8.4.2.

dEXM – See 6.2.4.4/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_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.

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

11.2.1 ODU*k* 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:

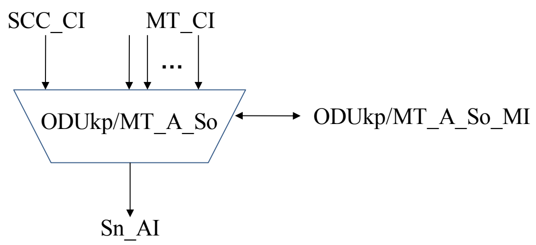


Figure 59/G.8121/Y.1381 – ODUkP/MT_A_So symbol

Interfaces:

Table 10/G.8121/Y.1381 – ODUkP/MT_A_So interfaces

Inputs	Outputs
Each MT_CP: MT_CI_Data MT_CI_iPHB MT_CI_oPHB **SCC_CP:** SCC_CI_Data **ODUkP/MT_A_So_MP:** ODUkP/MT_A_So_MI_Active ODUkP/MT_A_So_MI_SCCType	**ODUkP_AP:** ODUkP_AI_Data ODUkP_AI_Clock ODUkP_AI_FrameStart ODUkP_AI_MultiFrameStart

ODUkP/MT_A_So_MI_Label[1…M] ODUkP/MT_A_So_MI_LSPType[1…M] ODUkP/MT_A_So_MI_CoS[1…M] ODUkP/MT_A_So_PHB2TCMapping[1…M] ODUkP/MT_A_So_MI_QoSEncodingMode[1…M]	

Processes:

A process diagram of this function is shown in Figure 44.

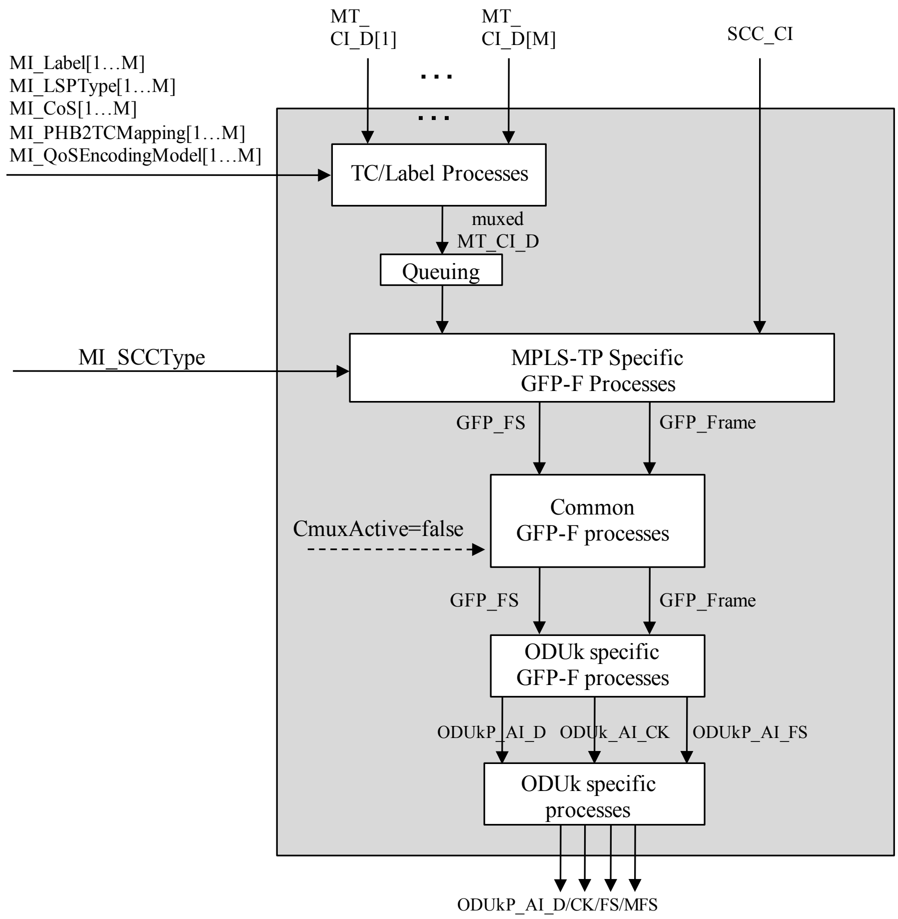


Figure 44/G.8121/Y.1381 – ODUkP/MT_A_So process diagram

– *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).

– *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/G.709/Y.1331.

– *ODUk specific source process:*

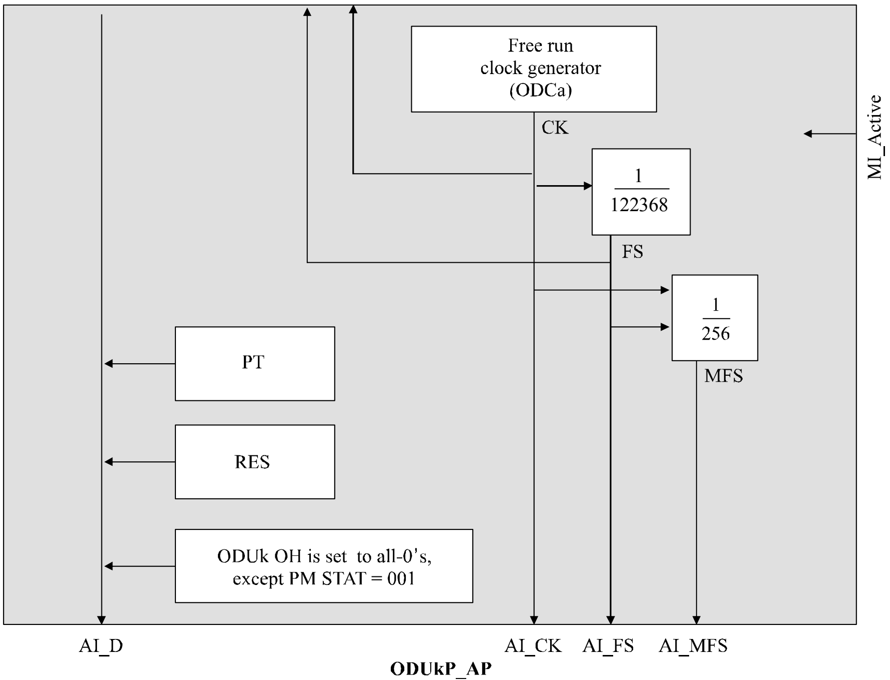


Figure 11-2/G.8121/Y.1381 – ODUkP specific source processes

Clock and (Multi)Frame Start signal generation: The function shall generate a local ODUk clock (ODUkP_AI_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/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.

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/G.709/Y.1331.

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:

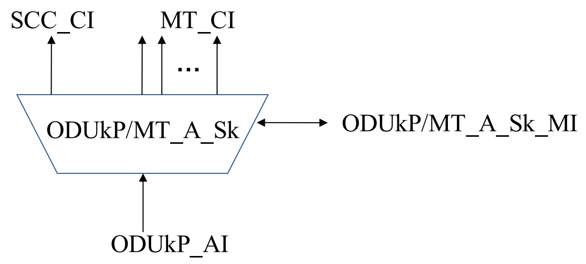


Figure 45/G.8121/Y.1381 – ODUkP/MT_A_Sk symbol

Interfaces:

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

Inputs	Outputs
ODUkP_AP: ODUkP_AI_Data ODUkP_AI_ClocK ODUkP_AI_FrameStart ODUkP_AI_MultiFrameStart ODUkP_AI_TSF **ODUkP/MT_A_Sk_MP:** ODUkP/MT_A_Sk_MI_Active ODUkP/MT_A_Sk_MI_SCCType ODUkP/MT_A_Sk_MI_Label[1…M] ODUkP/MT_A_Sk_MI_LSPType[1…M] ODUkP/MT_A_Sk_MI_CoS[1…M] ODUkP/MT_A_Sk_MI_TC2PHBMapping[1…M] ODUkP/MT_A_Sk_MI_QoSDecodingMode[1…M] ODUkP/MT _A_Sk_MI _LCK_Period[1…M] ODUkP/MT _A_Sk_MI _LCK_P[1…M] ODUkP/MT _A_Sk_MI _Admin_State ODUkP/MT _A_Sk_MI _AIS_Period[1…M] ODUkP/MT _A_Sk_MI _AIS_P[1…M]	**Each MT_CP:** MT_CI_Data MT_CI_iPHB MT_CI_oPHB MT_CI_SSF MT_CI_LStack **SCC_CP:** SCC_CI_Data SCC_CI_SSF **ODUkP/MT_A_Sk_MP:** ODUkP/MT_A_Sk_MI_AcPT ODUkP/MT_A_Sk_MI_AcEXI ODUkP/MT_A_Sk_MI_LastValidUPI ODUkP/MT_A_Sk_MI_cPLM ODUkP/MT_A_Sk_MI_cLFD ODUkP/MT_A_Sk_MI_cEXM ODUkP/MT_A_Sk_MI_cUPM

Processes:

A process diagram of this function is shown in Figure 46.

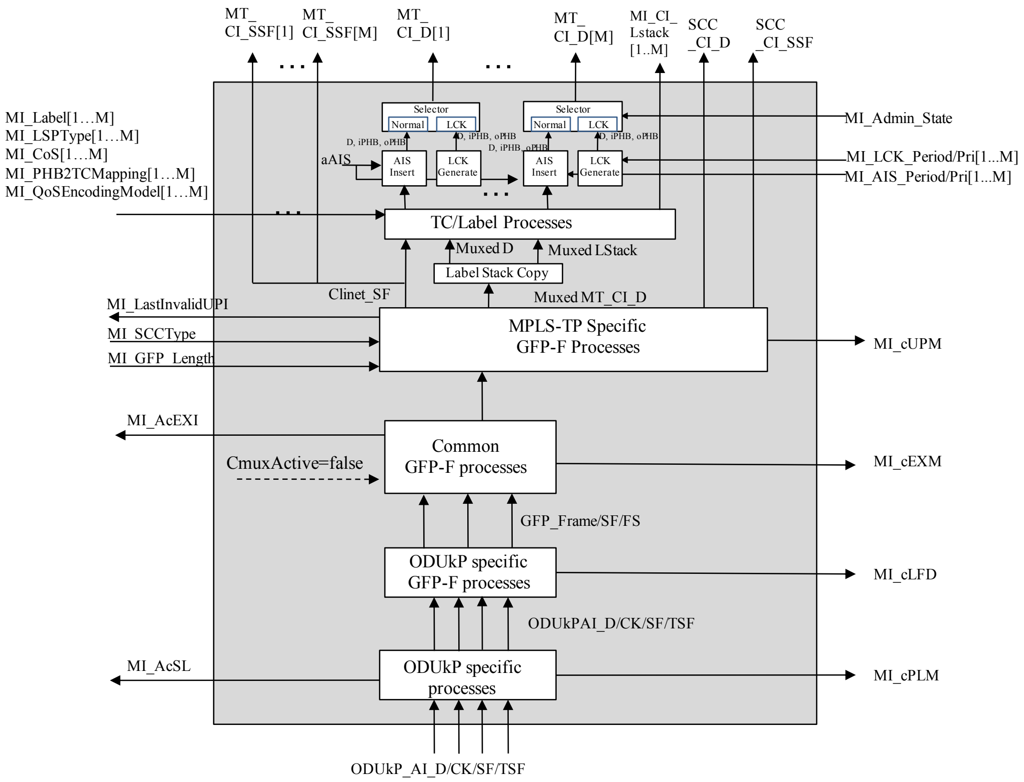


Figure 46/G.8121/Y.1381 – ODUkP/MT_A_Sk process diagram

– *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/G.806. GFP channel multiplexing is not supported (CMuxActive=false).

– *ODUk specific GFP sink process:*

See 8.5.2.2/G.806. The GFP frames are demapped from the ODUk payload area according to 17.3/G.709/Y.1331.

– *ODUk-specific sink process:*

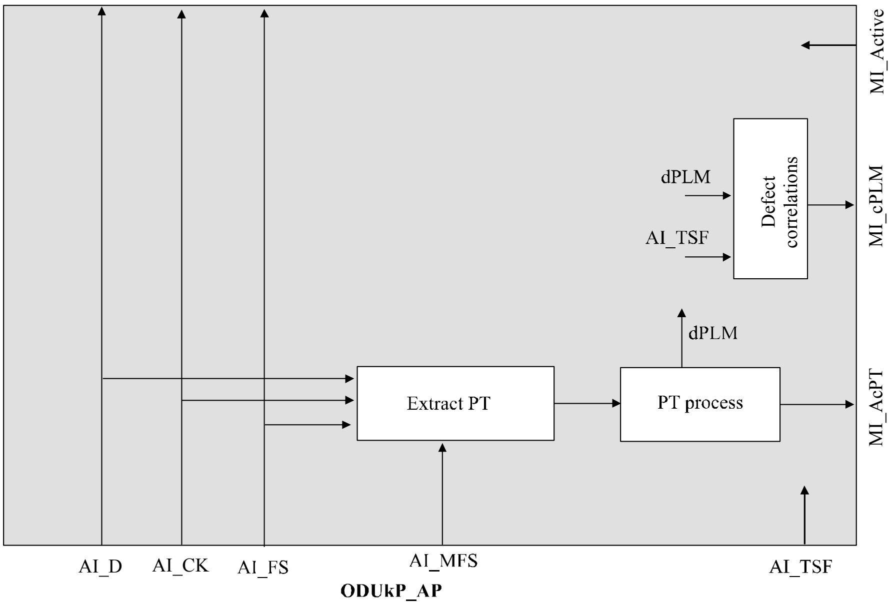


Figure 11-4/G.8121/Y.1381 – ODUkP specific sink processes

PT: The function shall extract the PT byte from the PSI overhead as defined in 8.7.1/G.798. The payload type value for "GFP mapping" in 15.9.2.1.1/G.709/Y.1331 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/G.798.

dLFD – See 6.2.5.2/G.806.

dEXM – See 6.2.4.4/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.

11.2.2 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 ODUk to MPLS-TP adaptation source function (ODUkP-X-L/MT_A_So)

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

Symbol:

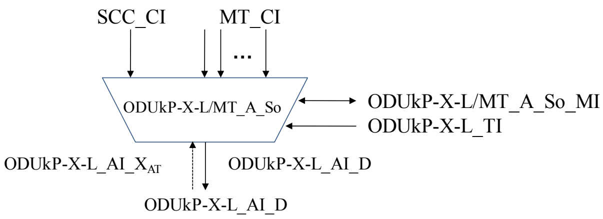


Figure 47/G.8121/Y.1381 – ODUkP-X-L/MT_A_So symbol

Interfaces:

Table 12/G.8121/Y.1381 – ODUkP-X-L/MT_A_So interfaces

Inputs	Outputs
Each MT_CP: MT_CI_Data MT_CI_iPHB	**ODUkP-X-L_AP:** ODUkP-X-L_AI_Data ODUkP-X-L_AI_Clock

MT_CI_oPHB **SCC_CP:** SCC_CI_Data **ODUkP-X-L_AP:** ODUkP-X-L_AI_X_{AT} **ODUkP-X-L/MT_A_So_MP:** ODUkP-X-L/MT_A_So_MI_Active ODUkP-X-L/MT_A_So_MI_SCCType ODUkP-X-L/MT_A_So_MI_Label[1…M] ODUkP-X-L/MT_A_So_MI_LSPType[1…M] ODUkP-X-L/MT_A_So_MI_CoS[1…M] ODUkP-X-L/MT_A_So_PHB2TCMapping[1…M] ODUkP-X-L/MT_A_So_MI_QoSEncodingMode[1…M]	ODUkP-X-L_AI_FrameStart ODUkP-X-L_AI_MultiFrameStart

Processes:

A process diagram of this function is shown in Figure 48.

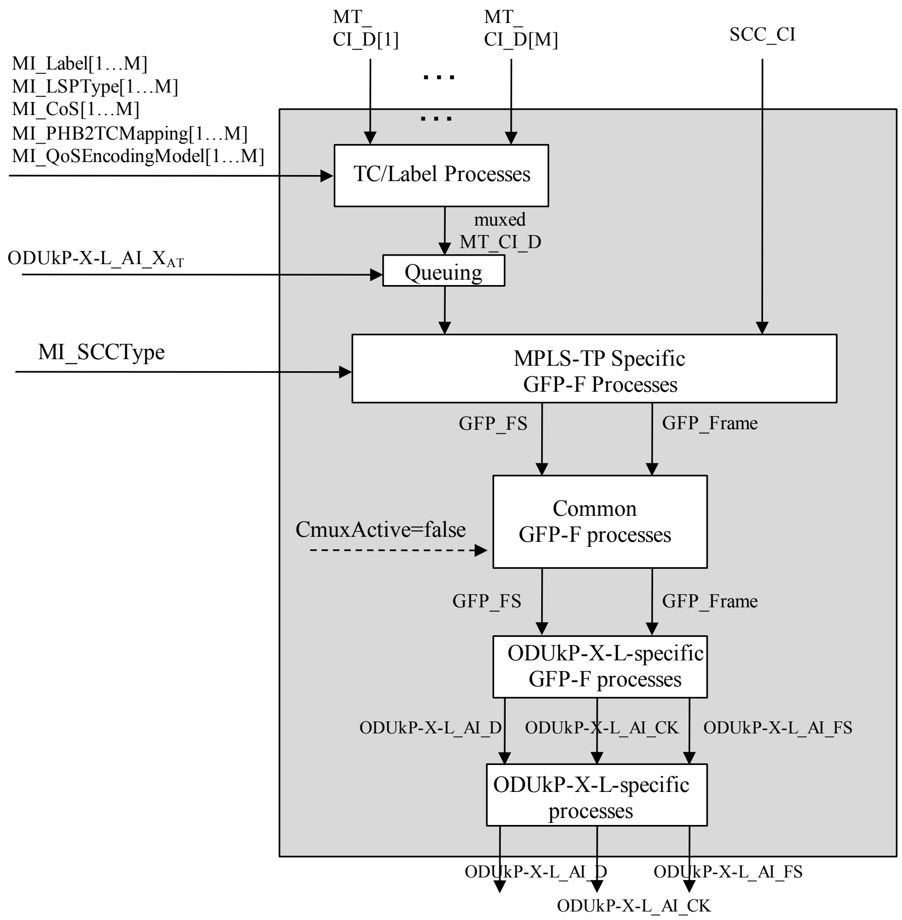


Figure 48/G.8121/Y.1381 – ODUkP-X-L/MT_A_So process diagram

The processes have the same definition as in 11..2.1.1.

ODUkP-X-L specific source process:

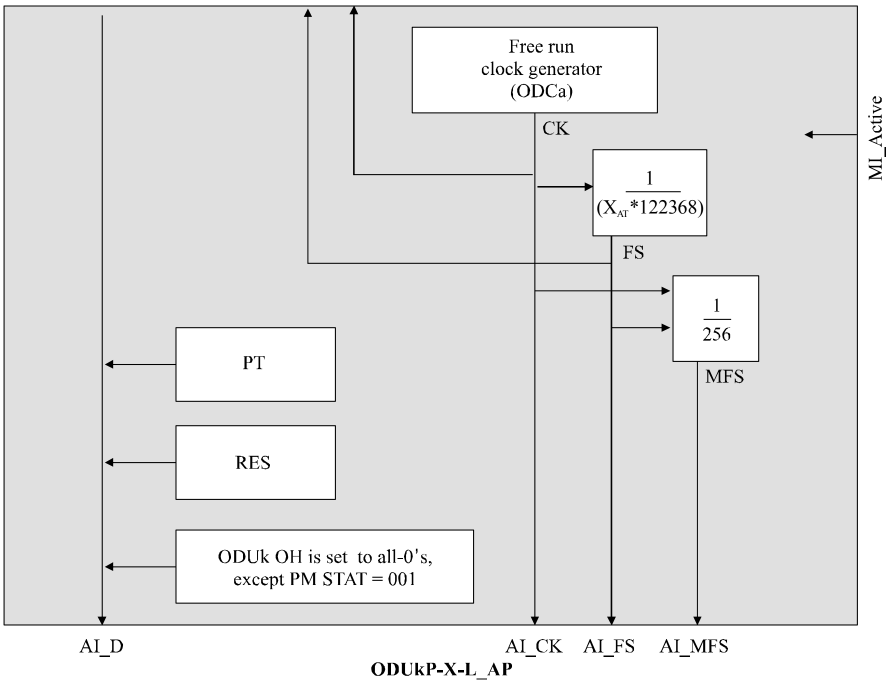


Figure 11-2/G.8121/Y.1381 – ODUkP-X-L specific source processes

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/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/G.709/Y.1331.

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

Defects:

None.

Consequent actions:

None.

Defect correlations:

None.

Performance monitoring:

Ffs.

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.

<u>**Symbol:**</u>

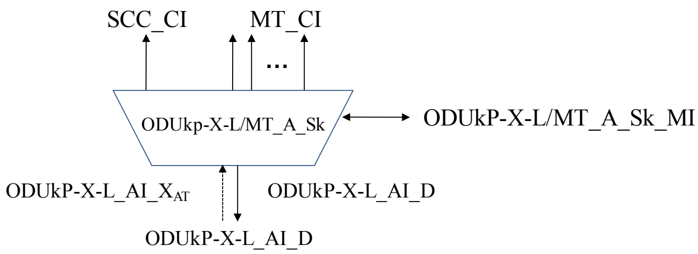


Figure 49/G.8121/Y.1381 – ODUkP-X-L/MT_A_Sk symbol

Interfaces:

Table 13/G.8121/Y.1381 – ODUkP-X-L/MT_A_Sk interfaces

Inputs	Outputs
ODUkP-X-L_AP: ODUkP-X-L_AI_Data ODUkP-X-L_AI_ClocK ODUkP-X-L_AI_FrameStart ODUkP-X-L_AI_MultiFrameStart ODUkP-X-L_AI_TSF ODUkP-X-L_AI_X_{AR} **ODUkP-X-L/MT_A_Sk_MP:** ODUkP-X-L/MT_A_Sk_MI_Active ODUkP-X-L/MT_A_Sk_MI_SCCType ODUkP-X-L/MT_A_Sk_MI_Label[1…M] ODUkP-X-L/MT_A_Sk_MI_LSPType[1…M] ODUkP-X-L/MT_A_Sk_MI_CoS[1…M] ODUkP-X-L/MT_A_Sk_MI_TC2PHBMapping[1…M] ODUkP-X-L/MT_A_Sk_MI_QoSDecodingMode[1…M] ODUkP-X-L/MT _A_Sk_MI _LCK_Period[1…M]	**Each MT_CP:** MT_CI_Data MT_CI_iPHB MT_CI_oPHB MT_CI_SSF MT_CI_LStack **SCC_CP:** SCC_CI_Data SCC_CI_SSF **ODUkP-X-L/MT_A_Sk_MP:** ODUkP-X-L/MT_A_Sk_MI_AcVcPT ODUkP-X-L/MT_A_Sk_MI_AcEXI ODUkP-X-L/MT_A_Sk_MI_LastValidUPI ODUkP-X-L/MT_A_Sk_MI_cVcPLM ODUkP-X-L/MT_A_Sk_MI_cLFD ODUkP-X-L/MT_A_Sk_MI_cEXM

ODUkP-X-L/MT _A_Sk_MI _LCK_P[1…M] ODUkP-X-L/MT _A_Sk_MI _Admin_State ODUkP-X-L/MT _A_Sk_MI _AIS_Period[1…M] ODUkP-X-L/MT _A_Sk_MI _AIS_P[1…M]	ODUkP-X-L/MT_A_Sk_MI_cUPM

Processes:

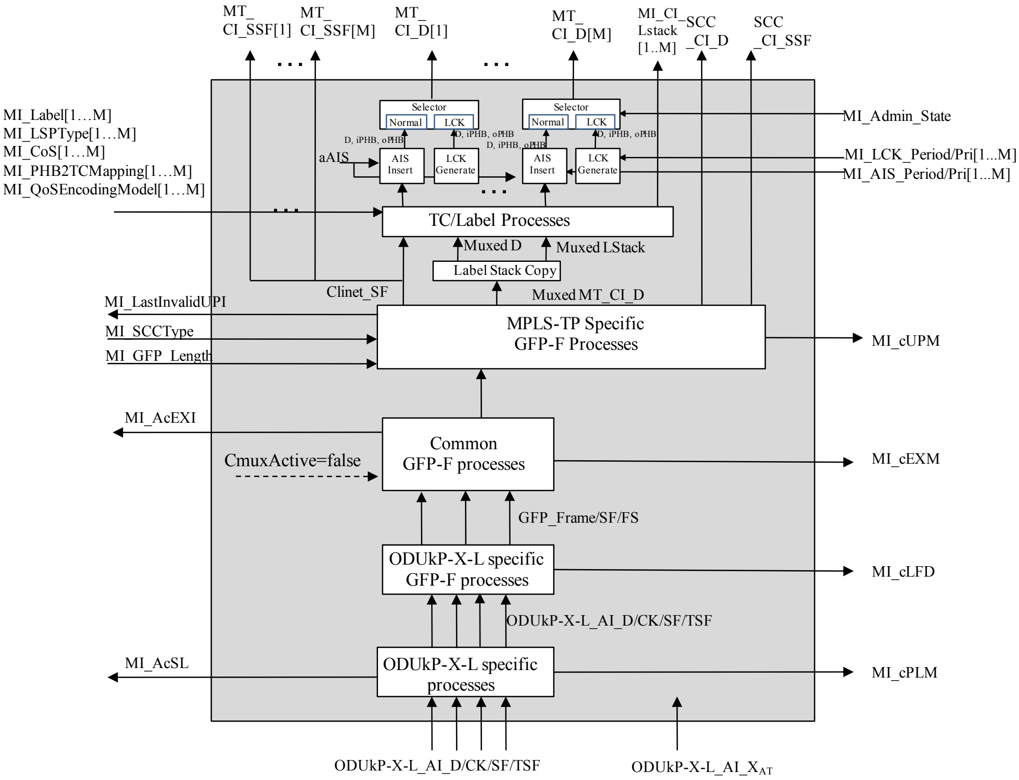


Figure 50/G.8121/Y.1381 – ODUkP-X-L/MT_A_Sk process diagram

See process diagram and process description in 11..2.1.2. The additional ODUkP-X-L_AI_X$_{AR}$ interface is not connected to any of the internal processes.

ODUkP-X-L specific sink process:

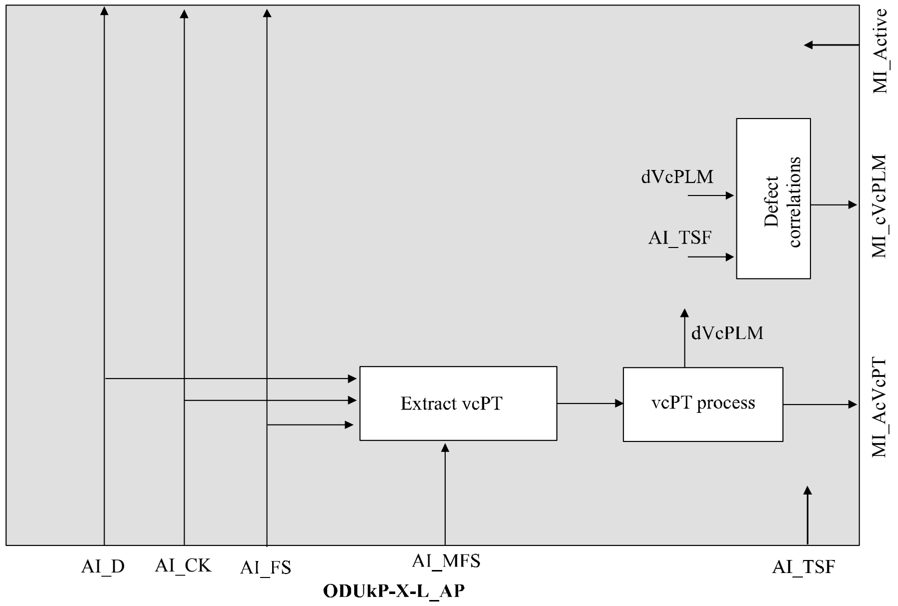


Figure 11-4/G.8121/Y.1341 – ODUkP-X-L specific sink processes

PT: The function shall extract the vcPT byte from the PSI overhead as defined in 8.7.3/G.798. The payload type value for "GFP mapping" in 18.1.2.2/G.709/Y.1331 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/G.798.

dLFD – See 6.2.5.2/G.806.

dUPM – See 8.4.2.

dEXM – See 6.2.4.4/G.806.

Consequent actions:

The function shall perform the following consequent actions:

aSSF ← AI_TSF or dVcPLM or dLFD or dUPM or dEXM

aAIS ← 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/G.806). This fault cause shall be reported to the EMF.

cVcPLM ← dVcPLM and (not AI_TSF)

cLFD ← 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:

Ffs.

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

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

11.3.1.1 P*q* to MPLS-TP Adaptation Source function (P*q*/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/Y.1343 with a value of N=1. The VLI byte is reserved and not used for payload data.

Symbol

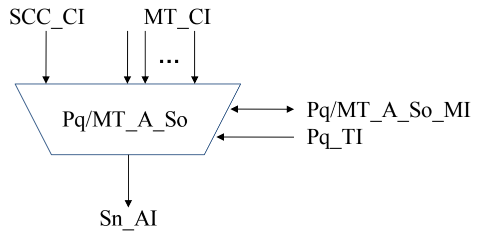


Figure 1/G.8121/Y.1381 – Pq/MT_A_So symbol

Interfaces

Table 1/G.8121/Y.1381: Pq/MT_A_So interfaces

Inputs	Outputs
Each MT_CP: MT_CI_Data MT_CI_iPHB MT_CI_oPHB SCC_CP: SCC_CI_Data Pq_TP: Pq_TI_Clock Pq_TI_FrameStart Pq/MT_A_So_MP: Pq/MT_A_So_MI_SCCType Pq/MT_A_So_MI_Label[1…M] Pq/MT_A_So_MI_LSPType[1…M] Pq/MT_A_So_MI_CoS[1…M] Pq/MT_A_So_PHB2TCMapping[1…M] Pq/MT_A_So_MI_QoSEncodingMode[1…M]	Pq_AP: Pq_AI_Data Pq_AI_Clock Pq_AI_FrameStart

Processes

A process diagram of this function is shown in Figure 20.

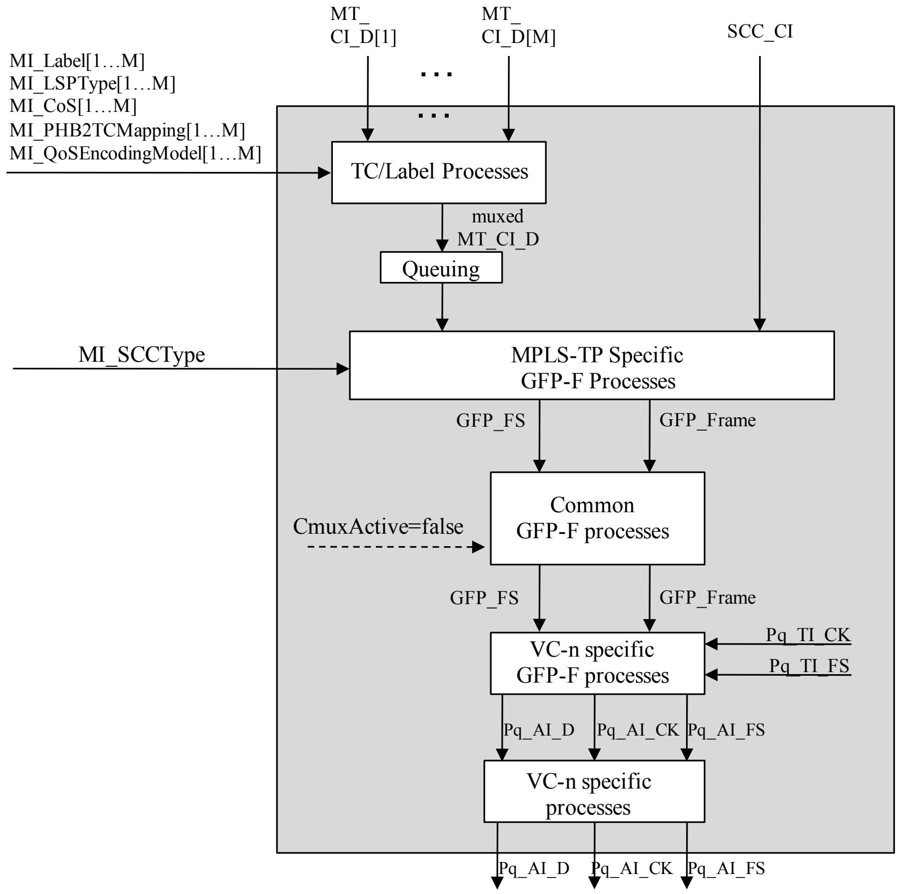


Figure 20/G.8121/Y.1381 – Pq/MT_A_So process diagram

TC/Label processes:

See 8.2.1/G.8121/Y.1381.

Queuing process:

See 8.3/G.8121/Y.1381.

MPLS-TP-specific GFP-F source process:

See 8.4.1/G.8121/Y.1381.

Common GFP source process:

See 8.5.3.1/G.806. GFP channel multiplexing is not supported (CMuxActive=false).

Pq specific GFP source process:

See 8.5.2.1/G.806. The GFP frames are mapped into the Pq payload area according to G.8040/Y.1340.

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 P*q* to MPLS-TP Adaptation Sink function (P*q*/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/Y.1343 with a value of N=1. The VLI byte is reserved and not used for payload data.

Symbol

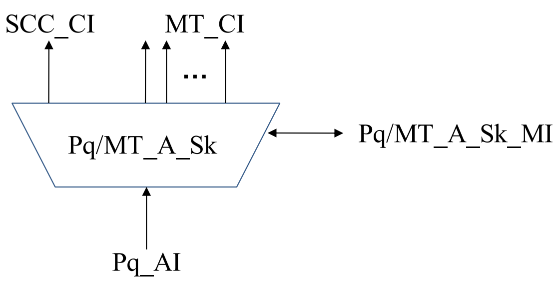


Figure 2/G.8121/Y.1381 – Pq/MT_A_Sk symbol

Interfaces

Table 2/G.8121/Y.1381: Pq/MT_A_Sk interfaces

Inputs	Outputs
Pq_AP: Pq_AI_Data Pq_AI_ClocK Pq_AI_FrameStart Pq_AI_TSF Pq/MT_A_Sk_MP: Pq/MT_A_Sk_MI_SCCType Pq/MT_A_Sk_MI_Label[1…M] Pq/MT_A_Sk_MI_LSPType[1…M] Pq/MT_A_Sk_MI_CoS[1…M] Pq/MT_A_Sk_MI_TC2PHBMapping[1…M] Pq/MT_A_Sk_MI_QoSDecodingMode[1…M] Pq/MT _A_Sk_MI _LCK_Period[1…M] Pq/MT _A_Sk_MI _LCK_P[1…M] Pq/MT _A_Sk_MI _Admin_State Pq/MT _A_Sk_MI _AIS_Period[1…M] Pq/MT _A_Sk_MI _AIS_P[1…M]	Each MT_CP: MT_CI_Data MT_CI_iPHB MT_CI_oPHB MT_CI_SSF MT_CI_LStack SCC_CP: SCC_CI_Data SCC_CI_SSF Pq/MT_A_Sk_MP: Pq/MT_A_Sk_MI_AcSL Pq/MT_A_Sk_MI_AcEXI Pq/MT_A_Sk_MI_LastValidUPI Pq/MT_A_Sk_MI_cPLM Pq/MT_A_Sk_MI_cLFD Pq/MT_A_Sk_MI_cEXM Pq/MT_A_Sk_MI_cUPM

Processes

A process diagram of this function is shown in Figure 22.

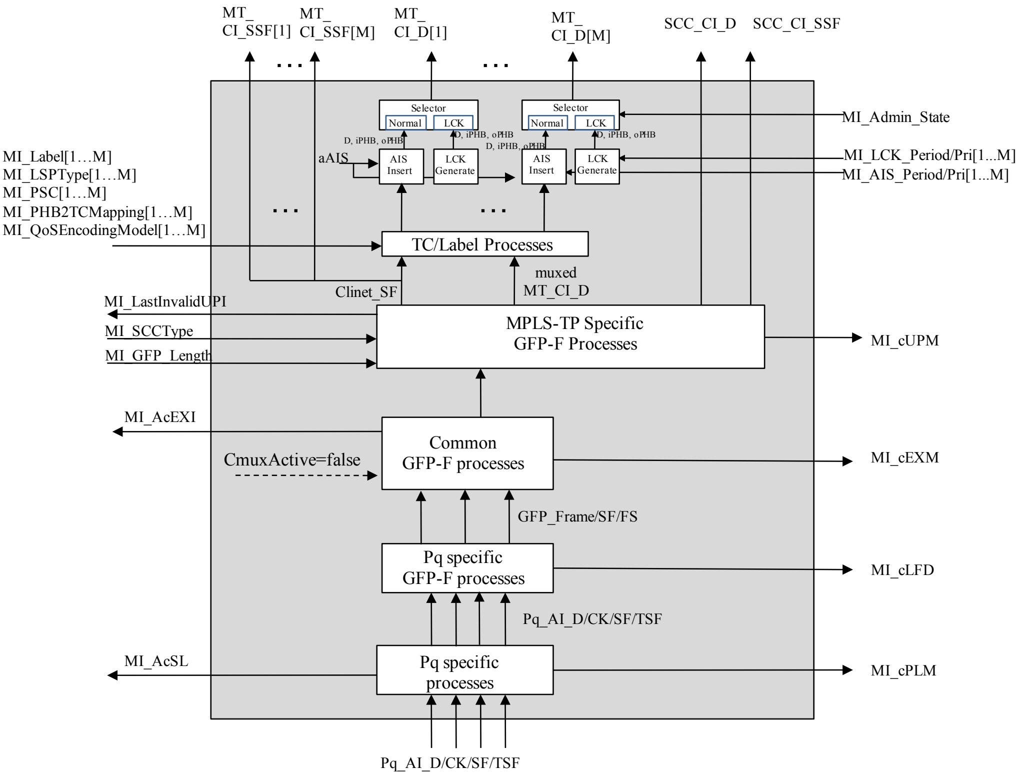


Figure 22/G.8121/Y.1381 – Pq/MT_A_Sk process diagram

[WD23 proposes show MT_CI_LStack as an output.]]

– *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/G.806. The signal label for "GFP mapping" in clause 2.1/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/G.806.

dLFD – See 6.2.5.2/G.806.

dUPM - See 8.4.2/G.8121/Y.1381

dEXM – See 6.2.4.4/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 P*q* to MPLS-TP Adaptation functions (P*q*-X-L/MT_A; *q*=11s, 12s, 31s, 32e)

11.3.2.1 LCAS-capable P*q* to MPLS-TP Adaptation Source function (P*q*-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/Y.1343.

Symbol

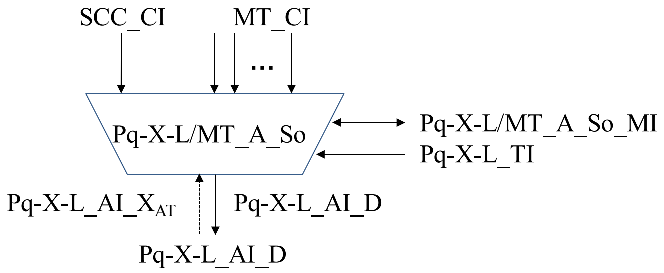


Figure 3/G.8121/Y.1381 – Pq-X-L/MT_A_So symbol

Interfaces

Table 3/G.8121/Y.1381: Pq-X-L/MT_A_So interfaces

Inputs	Outputs
Each MT_CP: MT_CI_Data MT_CI_iPHB MT_CI_oPHB SCC_CP: SCC_CI_Data Pq-X-L_AP: Pq-X-L_AI_X_{AT} Pq-X-L_TP: Pq-X-L_TI_Clock Pq-X-L_TI_FrameStart Pq-X-L/MT_A_So_MP: Pq-X-L/MT_A_So_MI_SCCType Pq-X-L/MT_A_So_MI_Label[1…M] Pq-X-L/MT_A_So_MI_LSPType[1…M] Pq-X-L/MT_A_So_MI_CoS[1…M] Pq-X-L/MT_A_So_PHB2TCMapping[1…M] Pq-X-L/MT_A_So_MI_QoSEncodingMode[1…M]	Pq-X-L_AP: Pq-X-L_AI_Data Pq-X-L_AI_Clock Pq-X-L_AI_FrameStart

Processes

A process diagram of this function is shown in Figure 24.

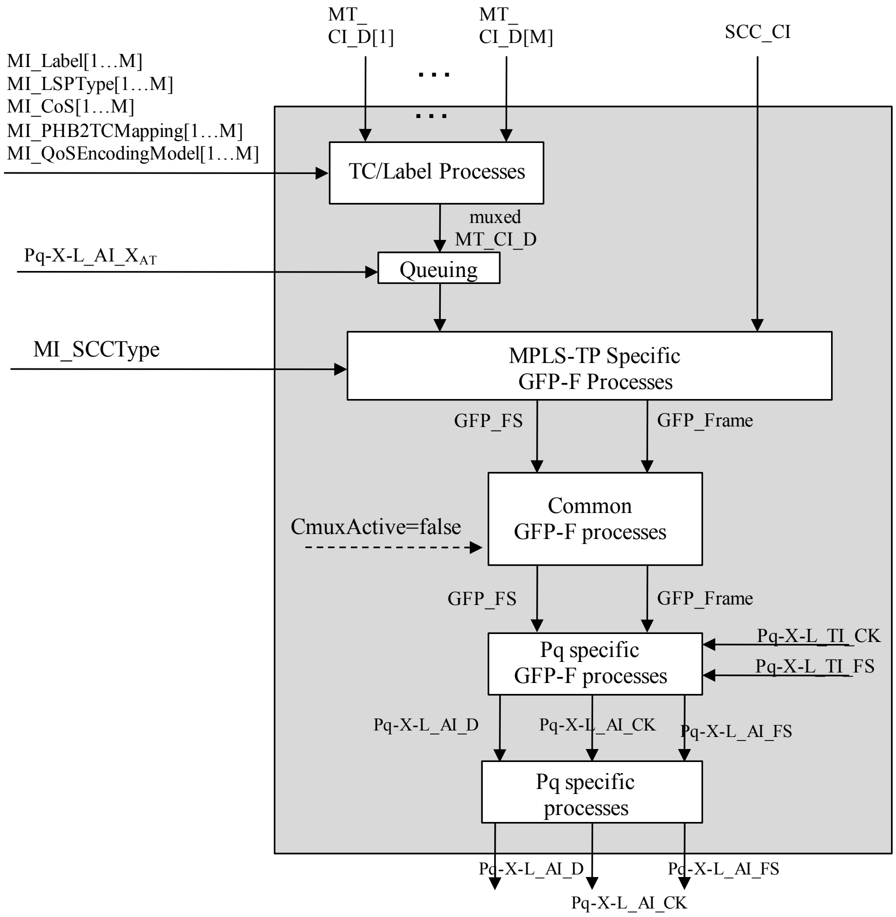


Figure 24/G.8121/Y.1381 – Pq-X-L/MT_A_So process diagram

The processes have the same definition as in 11..1.1.1/G.8121/Y.1381.

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/Y.1343.

Symbol

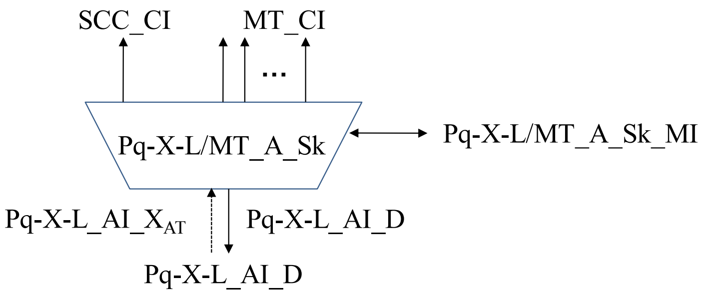


Figure 4/G.8121/Y.1381 – Pq-X-L/MT_A_Sk symbol

Interfaces

Table 4/G.8121/Y.1381: Pq-X-L/MT_A_Sk interfaces

Inputs	Outputs
<u>Pq-X-L_AP:</u> Pq-X-L_AI_Data Pq-X-L_AI_ClocK Pq-X-L_AI_FrameStart Pq-X-L_AI_TSF Pq-X-L_AI_X_{AR} <u>Pq-X-L/MT_A_Sk_MP:</u> Pq-X-L/MT_A_Sk_MI_SCCType Pq-X-L/MT_A_Sk_MI_Label[1…M] Pq-X-L/MT_A_Sk_MI_LSPType[1…M] Pq-X-L/MT_A_Sk_MI_CoS[1…M] Pq-X-L/MT_A_Sk_MI_TC2PHBMapping[1…M] Pq-X-L/MT_A_Sk_MI_QoSDecodingMode[1…M] Pq-X-L//MT _A_Sk_MI _LCK_Period[1…M] Pq-X-L//MT _A_Sk_MI _LCK_P[1…M] Pq-X-L//MT _A_Sk_MI _Admin_State Pq-X-L//MT _A_Sk_MI _AIS_Period[1…M] Pq-X-L//MT _A_Sk_MI _AIS_P[1…M]	<u>Each MT_CP:</u> MT_CI_Data MT_CI_iPHB MT_CI_oPHB MT_CI_SSF MT_CI_LStack <u>SCC_CP:</u> SCC_CI_Data SCC_CI_SSF <u>Pq-X-L/MT_A_Sk_MP:</u> Pq-X-L/MT_A_Sk_MI_AcSL Pq-X-L/MT_A_Sk_MI_AcEXI Pq-X-L/MT_A_Sk_MI_LastValidUPI Pq-X-L/MT_A_Sk_MI_cPLM Pq-X-L/MT_A_Sk_MI_cLFD Pq-X-L/MT_A_Sk_MI_cEXM Pq-X-L/MT_A_Sk_MI_cUPM

Processes

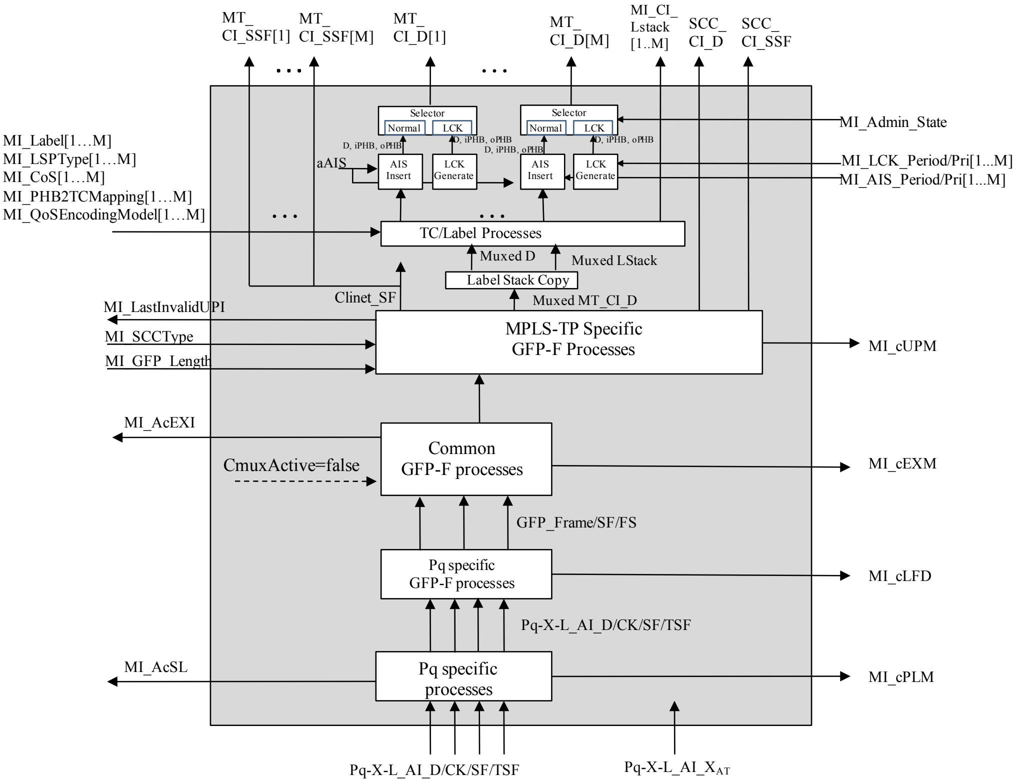


Figure 5/G.8121/Y.1381 – Pq-X-L/MT_A_Sk process diagram

See process diagram and process description in 11..1.1.2/G.8121/Y.1381. The additional Pq-X-L_AI_X_{AR} interface is not connected to any of the internal processes.

Defects

dPLM – See 6.2.4.2/G.806.

dLFD – See 6.2.5.2/G.806.

dUPM - See 8.4.2/G.8121/Y.1381

dEXM – See 6.2.4.4/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.

11.4 Ethernet to MPLS-TP adaptation function

11.4.1 ETY to MPLS-TP adaptation function (ETY/MT_A)

11.4.1.1 ETY to MPLS-TP adaptation function (ETY/MT_A)

Symbol

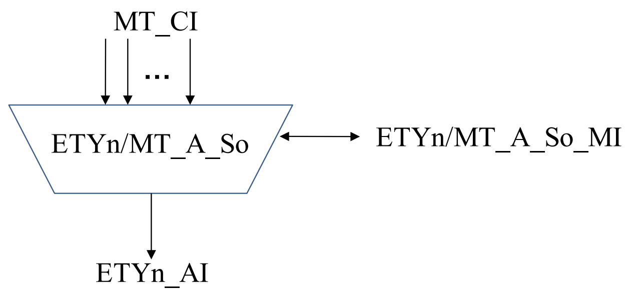


Figure 11.-6 – ETYn/ETH_A_So symbol

Interfaces

Table 11.-4: ETYn/ETH_A_So interfaces

Inputs	Outputs
Each MT_CP: MT_CI_Data MT_CI_iPHB MT_CI_oPHB **<u>ETH_FP:</u>** ETH_CI_ESMC ETH_CI_PauseTrigger **<u>ETH_TP:</u>** ETH_TI_Clock **ETYn/MT_A_So_MP:** ETYn/MT_A_So_MI_SCCType ETYn/MT_A_So_MI_Label[1…M] ETYn/MT_A_So_MI_LSPType[1…M] ETYn/MT_A_So_MI_CoS[1…M] ETYn/MT_A_So_PHB2TCMapping[1…M] ETYn/MT_A_So_MI_QoSEncodingMode[1…M] ETYn/MT_A_So_MI_TxPauseEnable	**<u>ETYn_AP:</u>** ETYn_AI_Data ETYn_AI_Clock ETYn_AI_SSF ETYn_AI_SSFrdi ETYn_AI_SSFfdi **<u>ETYn/MT_A_So_MP:</u>** ETYn/MT_A_So_MI_pFramesTransmittedOK ETYn/MT_A_So_MI_pOctetsTransmittedOK

Processes

A process diagram of this function is shown in Figure 11.-7.

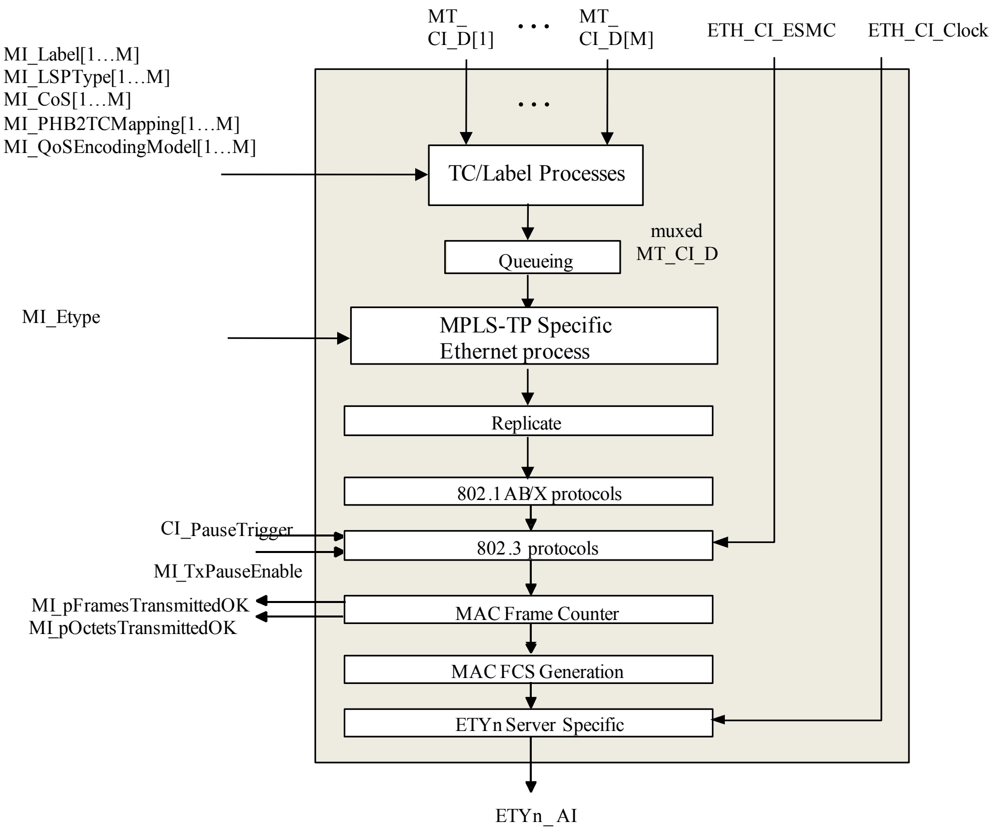


Figure x – ETYn/MT_A_So process

The "Replicate," "802.3 protocols" "802.1AB/X protocols" and "MAC FCS Generate" processes are defined in clause 8/G.8021 ("Generic processes").

The "ETYn Server Specific" source process pads frames shorter than the minimum frame size (of 64 octets) to the minimum frame size according to clause 3.2.8 of [IEEE 802.3].

MAC Frame counting process location is For Further Study.

MPLS-TP process inserts the Ethertype for MPLS-TP packets according to [RFC5332]

TC/Label processing and Queuing are defined in clause 8.2 and clause 8.3

Defects	None.
Consequent actions	None.
Defect correlations	None.
Performance monitoring	For Further Study.

11.4.1.2 ETY to MPLS-TP adaptation function (ETY/MT_A)

Symbol

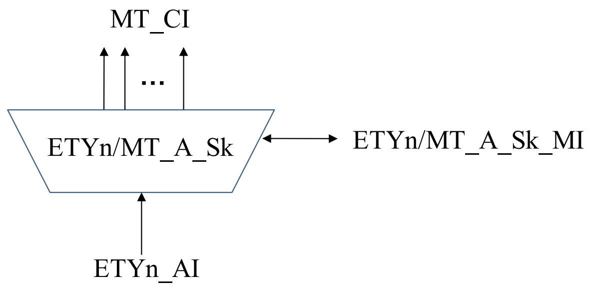


Figure 11.-8 – ETYn/ETH_A_Sk symbol

Interfaces

Table 11.-5: ETYn/MT_A_Sk interfaces

Inputs	Outputs
ETYn_AP: ETYn_AI_Data ETYn_AI_Clock ETYn_AI_TSF ETYn_AI_TSFrdi ETYn_AI_TSFfdi ETYn/MT_A_Sk_MP: ETYn/MT_A_Sk_MI_FilterConfig ETYn/MT_A_Sk_MI_Frame_Type_Config ETYn/MT_A_Sk_MI_Etype ETYn/MT_A_Sk_MI_MAC_Length ETYn/MT_A_Sk_MI _LCK_Period[1…M] ETYn/MT_A_Sk_MI _LCK_CoS[1…M] ETYn/MT_A_Sk_MI _Admin_State ETYn/MT_A_Sk_MI _AIS_Period[1…M] ETYn/MT_A_Sk_MI _AIS_CoS[1…M]	**Each MT_CP:** MT_CI_Data MT_CI_iPHB MT_CI_oPHB MI_CI_LStack **ETH_FP:** ETH_CI_ESMC **ETH_TP:** ETH_TI_Clock ETYn/MT_A_Sk_MP: ETYn/MT_A_Sk_MI_pErrors ETYn/MT_A_Sk_MI_pFramesReceivedOK ETYn/MT_A_Sk_MI_pOctetsReceivedOK

Processes

A process diagram of this function is shown in Figure 11.-9.

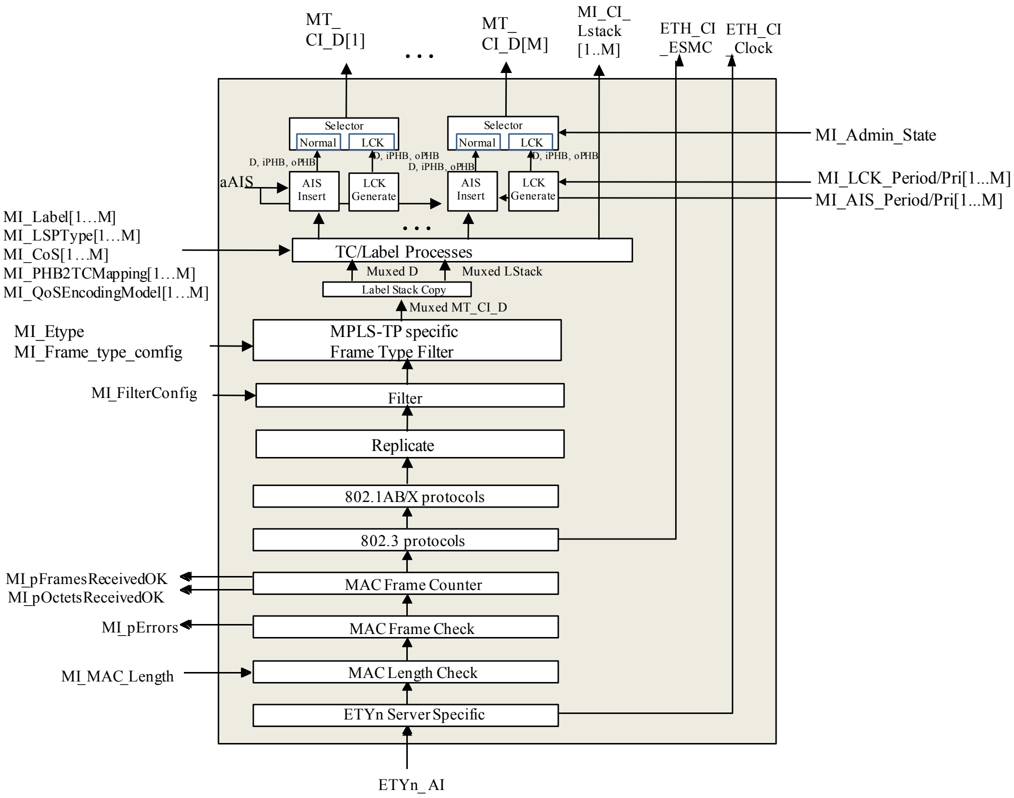


Figure 11.-9 – ETYn/ETH_A_Sk process

The "Filter," "Replicate," "802.3 protocols", "802.1AB/X protocols", MAC Frame Counting, "MAC FCS Check" and "MAC Length Check" processes are defined in clause 8 ("Generic processes"). Filter process is for the reception process of the Ethertype for MPLS-TP packets according to [RFC5332]

The "ETYn Server Specific" sink process is a null process.

MAC Frame Counting: For Further Study.

TC/Label processing, Label Stack Copy process:

and Queuing are defined in clause 8.2.2, 8.2.3 and 8.3 respectively

Defects	None.
Consequent actions	FFS
Defect correlations	None.
Performance monitoring	For Further Study.

Appendix I
[Note: Need to be reviewed, contribution invited]

Formatted: Highlight

Formatted: Font: 11 pt, Not Bold, Comp
Script Font: 11 pt

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:

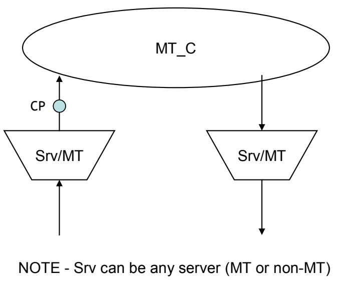


Figure I.1 – Intermediate node with no MIPs

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:

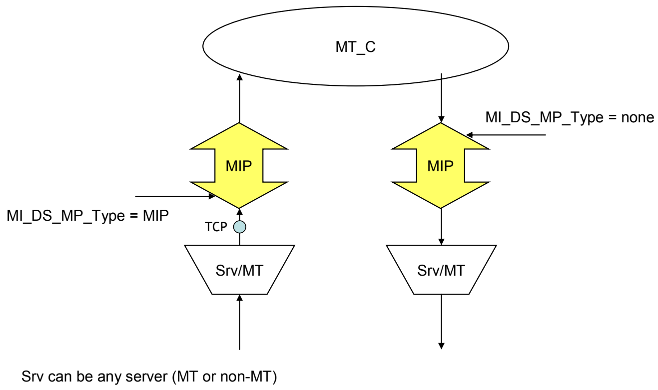


Figure I.2 – Intermediate node with per-interface MIPs

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:

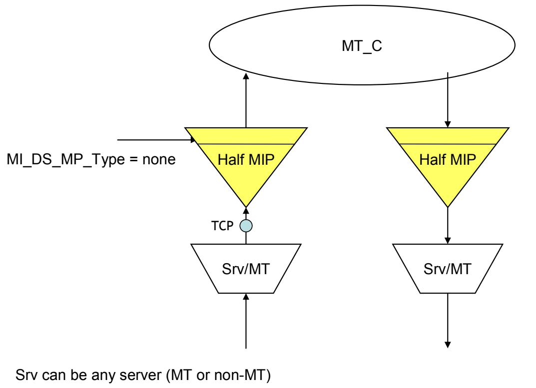


Figure I.3 – Intermediate node with a per-node MIP

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 half-MIP.

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

The TTL check process in the MTDi/MT_A_Sk, 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 (user data or OAM) that are received with an expired TTL.

Figure I.4 describes the behavior of a terminating node with a Down MEP or a per-node MEP using the atomic functions defined in this Recommendation. These two cases are modeled in the same way:

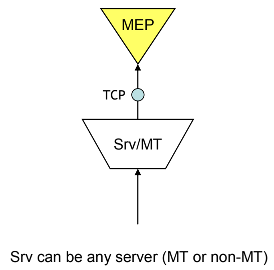


Figure I.4 – Terminating node with a down MEP or node MEP

The Server/MT_A_Sk is connected to MEP 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 MEP.

The MEP terminates the MPLS-TP trail and processes all the MPLS-TP packets it receives regardless of whether the TTL has expired or not.

Figure I.5 describes the behavior of a terminating node with an Up MEP, and therefore a per-interface ingress MIP, using the atomic functions defined in this Recommendation:

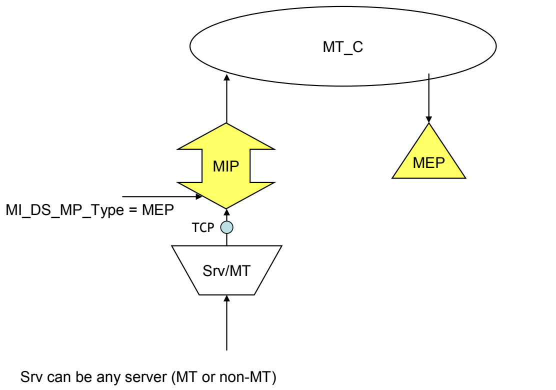


Figure I.5 – Terminating node with an Up MEP (and a per-interface MIP)

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=MEP) to forward all the MPLS-TP packets (user data or OAM) that are 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 Up MEP that terminates the MPLS-TP trail and processes all the MPLS-TP packets it receives regardless of whether the TTL has expired or not.

Formatted: English (U.K.)

Bibliography

draft-ietf-ccamp-rsvp-te-mpls-tp-oam-ext

draft-xxx-pwe3-mpls-tp-ldp-oam-config

ITU-T Y-SERIES RECOMMENDATIONS

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

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

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

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	**Transmission systems and media, digital systems and networks**
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	**Global information infrastructure, Internet protocol aspects and next-generation networks**
Series Z	Languages and general software aspects for telecommunication systems