

International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

G.806

(02/2012)

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

Digital networks – General aspects

**Characteristics of transport equipment –
Description methodology and generic
functionality**

Recommendation ITU-T G.806



ITU-T G-SERIES RECOMMENDATIONS

TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS

INTERNATIONAL TELEPHONE CONNECTIONS AND CIRCUITS	G.100–G.199
GENERAL CHARACTERISTICS COMMON TO ALL ANALOGUE CARRIER-TRANSMISSION SYSTEMS	G.200–G.299
INDIVIDUAL CHARACTERISTICS OF INTERNATIONAL CARRIER TELEPHONE SYSTEMS ON METALLIC LINES	G.300–G.399
GENERAL CHARACTERISTICS OF INTERNATIONAL CARRIER TELEPHONE SYSTEMS ON RADIO-RELAY OR SATELLITE LINKS AND INTERCONNECTION WITH METALLIC LINES	G.400–G.449
COORDINATION OF RADIOTELEPHONY AND LINE TELEPHONY	G.450–G.499
TRANSMISSION MEDIA AND OPTICAL SYSTEMS CHARACTERISTICS	G.600–G.699
DIGITAL TERMINAL EQUIPMENTS	G.700–G.799
DIGITAL NETWORKS	G.800–G.899
General aspects	G.800–G.809
Design objectives for digital networks	G.810–G.819
Quality and availability targets	G.820–G.829
Network capabilities and functions	G.830–G.839
SDH network characteristics	G.840–G.849
Management of transport network	G.850–G.859
SDH radio and satellite systems integration	G.860–G.869
Optical transport networks	G.870–G.879
DIGITAL SECTIONS AND DIGITAL LINE SYSTEM	G.900–G.999
MULTIMEDIA QUALITY OF SERVICE AND PERFORMANCE – GENERIC AND USER-RELATED ASPECTS	G.1000–G.1999
TRANSMISSION MEDIA CHARACTERISTICS	G.6000–G.6999
DATA OVER TRANSPORT – GENERIC ASPECTS	G.7000–G.7999
PACKET OVER TRANSPORT ASPECTS	G.8000–G.8999
ACCESS NETWORKS	G.9000–G.9999

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

Recommendation ITU-T G.806

Characteristics of transport equipment – Description methodology and generic functionality

Summary

Recommendation ITU-T G.806 specifies the methodology, generic functionality and components that should be used in order to specify transport network functionality of network elements; it does not specify individual transport network equipment as such. It is the baseline Recommendation for other standards that specify the characteristics of equipment for specific transport networks (e.g., synchronous digital hierarchy (SDH), plesiochronous digital hierarchy (PDH)).

History

Edition	Recommendation	Approval	Study Group
1.0	ITU-T G.806	2000-10-06	15
1.1	ITU-T G.806 (2000) Amd. 1	2003-03-16	15
2.0	ITU-T G.806	2004-02-06	15
2.1	ITU-T G.806 (2004) Amd. 1	2004-06-13	15
2.2	ITU-T G.806 (2004) Cor. 1	2004-08-22	15
2.3	ITU-T G.806 (2004) Cor. 2	2005-01-13	15
3.0	ITU-T G.806	2006-03-29	15
3.1	ITU-T G.806 (2006) Amd. 1	2008-11-22	15
4.0	ITU-T G.806	2009-01-13	15
4.1	ITU-T G.806 (2009) Amd. 1	2011-06-22	15
5.0	ITU-T G.806	2012-02-13	15

FOREWORD

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

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

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

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

NOTE

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

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

INTELLECTUAL PROPERTY RIGHTS

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

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at <http://www.itu.int/ITU-T/ipr/>.

© ITU 2012

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

Table of Contents

		Page
1	Scope	1
2	References.....	1
3	Definitions	3
	3.1 Terms defined elsewhere.....	3
	3.2 Terms defined in this Recommendation.....	3
4	Abbreviations and acronyms	6
5	Methodology.....	14
	5.1 Basic methodology.....	14
	5.2 Transmission layer naming.....	16
	5.3 Atomic function naming and diagrammatic conventions.....	17
	5.4 Reference point naming.....	21
	5.5 Reference point information naming.....	23
	5.6 Atomic function process allocation.....	24
	5.7 Combination rules	30
	5.8 Fault management and performance monitoring naming.....	34
	5.9 Fault management and performance monitoring specification techniques	34
6	Supervision	35
	6.1 Trail termination point mode and port mode.....	37
	6.2 Defect filter.....	37
	6.3 Consequent actions.....	54
	6.4 Defect correlations.....	60
	6.5 One-second performance monitoring filters.....	61
7	Information flow (XXX_MI) across the XXX_MP reference points.....	63
8	Generic processes	65
	8.1 Line coding and scrambling processes	65
	8.2 Alignment processes.....	65
	8.3 Performance supervision process	66
	8.4 BIP correction.....	69
	8.5 GFP processes	69
9	Performance and reliability	83
	9.1 Transit delay	83
	9.2 Response times	83
	9.3 Availability and reliability.....	83
	9.4 Laser safety.....	84
10	Generic equipment functions.....	84
	10.1 LCAS-capable virtual concatenated path layer functions P-Xv-L ($X \geq 1$)	84

Annex A – Assignment and use of signal label, payload type and user payload identifier codes	107
A.1 Experimental code	107
A.2 Proprietary codes	107
A.3 Request for standardized codes	107
Annex B – P-Xv/P-X-L_A_Sk: Calculation of RI_MST_gen for _LCASActive = true	108
Appendix I – Connection matrix examples.....	109
I.1 Connection matrix example for full connectivity.....	109
I.2 Connection matrix example for 2-port groups	110
I.3 Connection matrix example for 3-port groups type I	110
I.4 Connection matrix example for 3-port groups type II.....	111
I.5 Connection matrix example for 4-port groups type I	111
I.6 Connection matrix example for 4-port groups type II.....	112
I.7 Example of a provisioned connection matrix	112
I.8 Connection matrix example for 3-port groups (address slot group switching)	113
I.9 Connection matrix example for 4-port groups (address slot group switching)	113
Appendix II – Example of remote indication operation	115
II.1 Remote defect indication (RDI)	115
II.2 Remote error indication (REI).....	116
Appendix III – Alarm indication signal (AIS).....	118
Appendix IV – Signal fail (SF) and signal degrade (SD)	120
IV.1 Server signal fail (SSF) signal	120
IV.2 Server signal degrade (SSD) signal	120
IV.3 Trail signal fail (TSF) signal	120
IV.4 Trail signal degrade (TSD) signal	120
Appendix V – Description of error detection code (EDC) terminology $N \times BIP-m$	121
Appendix VI – Calculations leading to BIP saturation results in Tables 6-4 and 6-5	124
VI.1 Introduction	124
VI.2 Calculations and results.....	124
Appendix VII – Examples for the operation of the processes within LCAS-capable adaptation functions.....	126
VII.1 Basic configuration.....	126
VII.2 LCAS-enabled So and Sk functions	126
VII.3 LCAS-enabled So and LCAS-disabled Sk functions	150
VII.4 LCAS-disabled So and LCAS-enabled Sk functions	150
VII.5 Non-virtual-concatenation So and LCAS-capable virtual-concatenation Sk functions.....	150

	Page
Appendix VIII – Consequent actions for signals with no AIS/FDI defined.....	151
Appendix IX Types of flow forwarding processes	152
Bibliography.....	155

Recommendation ITU-T G.806

Characteristics of transport equipment – Description methodology and generic functionality

1 Scope

This Recommendation specifies a methodology for describing equipment for transport networks based on the transport processing functions and architectural entities defined in [ITU-T G.805]. It defines the set of generic atomic and compound functions and the set of rules for how to combine them. The detailed characteristics of equipment functional blocks of specific transport networks (e.g., SDH, OTN) will be defined in follow-up Recommendations based on this methodology. Equipment can then be described by an equipment functional specification (EFS) which lists the atomic functions and their interconnection.

In addition, generic functionality, processes and overall performance objectives for transport networks are defined in this Recommendation.

The internal structure of the implementation of this functionality (equipment design) need not be identical to the structure of the functional model, as long as all the details of the externally observable behaviour comply with the EFS.

Equipment which is normally stated to be compliant with this Recommendation may not fulfil all the requirements in the case that it is interworking with old equipment that is not compliant with this Recommendation.

2 References

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

- [ITU-T E.862] Recommendation ITU-T E.862 (1992), *Dependability planning of telecommunication networks*.
- [ITU-T G.664] Recommendation ITU-T G.664 (2006), *Optical safety procedures and requirements for optical transport systems*.
- [ITU-T G.703] Recommendation ITU-T G.703 (2001), *Physical/electrical characteristics of hierarchical digital interfaces*.
- [ITU-T G.704] Recommendation ITU-T G.704 (1998), *Synchronous frame structures used at 1544, 6312, 2048, 8448 and 44 736 kbit/s hierarchical levels*.
- [ITU-T G.705] Recommendation ITU-T G.705 (2000), *Characteristics of plesiochronous digital hierarchy (PDH) equipment functional blocks*.
- [ITU-T G.707] Recommendation ITU-T G.707/Y.1322 (2003), *Network node interface for the synchronous digital hierarchy (SDH)*.
- [ITU-T G.709] Recommendation ITU-T G.709/Y.1331 (2009), *Interfaces for the Optical Transport Network (OTN)*.

- [ITU-T G.775] Recommendation ITU-T G.775 (1998), *Loss of Signal (LOS), Alarm Indication Signal (AIS) and Remote Defect Indication (RDI) defect detection and clearance criteria for PDH signals.*
- [ITU-T G.781] Recommendation ITU-T G.781 (2008), *Synchronization layer functions.*
- [ITU-T G.783] Recommendation ITU-T G.783 (2006), *Characteristics of synchronous digital hierarchy (SDH) equipment functional blocks.*
- [ITU-T G.784] Recommendation ITU-T G.784 (1999), *Synchronous digital hierarchy (SDH) management.*
- [ITU-T G.798] Recommendation ITU-T G.798 (2010), *Characteristics of optical transport network hierarchy equipment functional blocks.*
- [ITU-T G.803] Recommendation ITU-T G.803 (2000), *Architecture of transport networks based on the synchronous digital hierarchy (SDH).*
- [ITU-T G.805] Recommendation ITU-T G.805 (2000), *Generic functional architecture of transport networks.*
- [ITU-T G.808.1] Recommendation ITU-T G.808.1 (2010), *Generic protection switching – Linear trail and subnetwork protection.*
- [ITU-T G.826] Recommendation ITU-T G.826 (2002), *End-to-end error performance parameters and objectives for international, constant bit-rate digital paths and connections.*
- [ITU-T G.831] Recommendation ITU-T G.831 (2000), *Management capabilities of transport networks based on the synchronous digital hierarchy (SDH).*
- [ITU-T G.832] Recommendation ITU-T G.832 (1998), *Transport of SDH elements on PDH networks – Frame and multiplexing structures.*
- [ITU-T G.842] Recommendation ITU-T G.842 (1997), *Interworking of SDH network protection architectures.*
- [ITU-T G.911] Recommendation ITU-T G.911 (1997), *Parameters and calculation methodologies for reliability and availability of fibre optic systems.*
- [ITU-T G.7041] Recommendation ITU-T G.7041/Y.1303 (2011), *Generic framing procedure.*
- [ITU-T G.7042] Recommendation ITU-T G.7042/Y.1305 (2006), *Link capacity adjustment scheme (LCAS) for virtual concatenated signals.*
- [ITU-T G.7710] Recommendation ITU-T G.7710/Y.1701 (2010), *Common equipment management function requirements.*
- [ITU-T G.8001] Recommendation ITU-T G.8001/Y.1354 (2011), *Terms and definitions for Ethernet frames over transport.*
- [ITU-T G.8010] Recommendation ITU-T G.8010/Y.1306 (2004), *Architecture of Ethernet layer networks.*
- [ITU-T G.8021] Recommendation ITU-T G.8021/Y.1341 (2010), *Characteristics of Ethernet transport network equipment functional blocks.*
- [ITU-T I.732] Recommendation ITU-T I.732 (2000), *Functional characteristics of ATM equipment.*
- [ITU-T M.20] Recommendation ITU-T M.20 (1992), *Maintenance philosophy for telecommunication networks.*
- [ITU-T M.125] Recommendation ITU-T M.125 (1988), *Digital loopback mechanisms.*

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

- 3.1.1 **access point (AP)**: [ITU-T G.805].
- 3.1.2 **access point identifier (API)**: [ITU-T G.831].
- 3.1.3 **automatic laser shutdown (ALS)**: [ITU-T G.664].
- 3.1.4 **automatic power shutdown (APSD)**: [ITU-T G.664].
- 3.1.5 **channel identifier (CID)**: [ITU-T G.7041].
- 3.1.6 **client data frame**: [ITU-T G.7041].
- 3.1.7 **client management frame**: [ITU-T G.7041].
- 3.1.8 **connection**: [ITU-T G.805].
- 3.1.9 **flow termination**: [ITU-T G.8021].
- 3.1.10 **frame-mapped generic framing procedure (GFP)**: [ITU-T G.7041].
- 3.1.11 **layer network**: [ITU-T G.805].
- 3.1.12 **maintenance entity**: [ITU-T G.8001].
- 3.1.13 **maintenance entity group**: [ITU-T G.8001].
- 3.1.14 **MEG end point (MEP)**: [ITU-T G.8001].
- 3.1.15 **MEG intermediate point (MIP)**: [ITU-T G.8001].
- 3.1.16 **member**: [ITU-T G.7042].
- 3.1.17 **network connection (NC)**: [ITU-T G.805].
- 3.1.18 **subnetwork connection (SNC)**: [ITU-T G.805].
- 3.1.19 **superblock**: [ITU-T G.7041].
- 3.1.20 **termination flow point**: [ITU-T G.8021].
- 3.1.21 **traffic conditioning function**: [ITU-T G.8001].
- 3.1.22 **trail**: [ITU-T G.805].
- 3.1.23 **transparent generic framing procedure (GFP)**: [ITU-T G.7041].
- 3.1.24 **virtual concatenation group (VCG)**: [ITU-T G.7042].

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

- 3.2.1 **adaptation function (A)**: An atomic function that performs the adaptation between client and server layer network.
- 3.2.2 **adapted information (AI)**: The information passing across an AP (see also [ITU-T G.805]).
- 3.2.3 **alarm**: A human-observable indication that draws attention to a failure (detected fault) usually giving an indication of the severity of the fault.
- 3.2.4 **all-ones**: The entire capacity of the adapted or characteristic information is set to logic "1".

3.2.5 anomaly: The smallest discrepancy which can be observed between the actual and desired characteristics of an item. The occurrence of a single anomaly does not constitute an interruption in the ability to perform a required function. Anomalies are used as the input for the performance monitoring (PM) process and for the detection of defects.

3.2.6 atomic function: A function that, if divided into simpler functions, would cease to be uniquely defined for digital transmission hierarchies. It is, therefore, indivisible from a network point of view.

3.2.7 bidirectional trail/connection type: A two-way trail/connection through a transport network.

3.2.8 broadcast connection type: An input connection point is connected to more than one output connection point.

3.2.9 characteristic information (CI): The information passing across a connection point or termination connection point (see also [ITU-T G.805]).

3.2.10 client/server layer: Any two adjacent network layers are associated in a client/server relationship. Each transport network layer provides transport to the layer above and uses transport from the layers below. The layer providing transport is termed a server; the layer using transport is termed client.

3.2.11 compound function: A function that represents a collection of atomic functions within one or more layer(s).

3.2.12 connection function (C): An atomic function within a layer which, if connectivity exists, relays a collection of items of information between groups of atomic functions. It does not modify the members of this collection of items of information, although it may terminate any switching protocol information and act upon it. Any connectivity restrictions between inputs and outputs shall be stated.

3.2.13 connection matrix (CM): A connection matrix is a matrix of appropriate dimensions which describes the connection pattern for assigning VC-ns on one side of a lower order path connection or higher order path connection function to VC-n capacities on the other side and vice versa.

3.2.14 connection point (CP): A reference point where the output of a trail termination source or a connection is bound to the input of another connection, or where the output of a connection is bound to the input of a trail termination sink or another connection.

3.2.15 consolidation: The allocation of server layer trails to client layer connections which ensures that each server layer trail is full before the next is allocated. Consolidation minimizes the number of partially filled server layer trails. It therefore maximizes the fill factor (e.g., a number of partially filled VC-4 paths may be consolidated into a single, fully filled VC-4).

3.2.16 defect: The density of anomalies has reached a level where the ability to perform a required function has been interrupted. Defects are used as input for performance monitoring, the control of consequent actions and for the determination of fault causes.

3.2.17 failure: The fault cause persisted long enough to consider the ability of an item to perform a required function to be terminated. The item may be considered as failed; a fault has now been detected.

3.2.18 fault: A fault is the inability of a function to perform a required action. This does not include an inability due to preventive maintenance, lack of external resources or planned actions.

3.2.19 fault cause: A single disturbance or fault may lead to the detection of multiple defects. A fault cause is the result of a correlation process which is intended to identify the defect that is representative of the disturbance or fault that is causing the problem.

3.2.20 function: A process defined for digital transmission hierarchies (e.g., plesiochronous digital hierarchy (PDH), synchronous digital hierarchy (SDH)) which acts on a collection of input information to produce a collection of output information. A function is distinguished by the way in which characteristics of the collection of output information differ from the collection of input information.

3.2.21 grooming: The allocation of server layer trails to client layer connections which groups together client layer connections whose characteristics are similar or related (e.g., it is possible to groom VC-12 paths by service type, by destination or by protection category into particular VC-4 paths which can then be managed accordingly).

3.2.22 layer network interworking function: An atomic function that provides interworking of characteristic information between two layer networks.

3.2.23 management information (MI): The signal passing across an access point.

3.2.24 management point (MP): A reference point where the output of an atomic function is bound to the input of the element management function, or where the output of the element management function is bound to the input of an atomic function. The MP is not the TMN Q3 interface.

3.2.25 MST_Range: The highest possible sequence number value in a particular path-layer technology. This parameter depends only upon the path-layer technology being used and is independent of any implementation details. This parameter is used to define how many member status signals (MSTs) are available in a particular technology. As an example, this parameter equals 255 for SDH Sn-layer and OTN OPUk-layer functions, 63 for SDH Sm-layer.

3.2.26 path: A trail in a path layer.

3.2.27 process: A generic term for an action or a collection of actions.

3.2.28 protection information (PI): The information passing across a protection point.

3.2.29 protection point (PP): A reference point where the output of a trail termination sink function of a bidirectional trail termination is bound to the input of its trail termination source function for the purpose of conveying information to the protection end.

3.2.30 reference point: The delimiter of a function.

3.2.31 remote defect indication (RDI): A signal that conveys the defect status of the characteristic information received by the trail termination sink function back to the network element which originated the characteristic information.

3.2.32 remote error indication (REI): A signal which conveys either the exact or truncated number of error detection code violations of the characteristic information as detected by the trail termination sink function back to the network element which originated the characteristic information.

3.2.33 remote information (RI): The information passing across a remote point; e.g., remote defect indication (RDI) and remote error indication (REI).

3.2.34 remote point (RP): A reference point where the output of a trail termination sink function of a bidirectional trail termination is bound to the input of its trail termination source function for the purpose of conveying information to the remote end.

3.2.35 replication information (PI): The information delivered across a replication point.

3.2.36 replication point (PP): A reference point between an adaptation source and sink. Characteristic information from a source flow point is replicated and delivered across a flow replication point to a termination flow point.

3.2.37 section: A trail in a section layer.

- 3.2.38 server signal degrade (SSD):** A signal degrade indication output at the connection point of an adaptation function.
- 3.2.39 server signal fail (SSF):** A signal fail indication output at the connection point of an adaptation function.
- 3.2.40 signal degrade (SD):** A signal indicating that the associated data has degraded in the sense that a degraded defect (dDEG) condition is active.
- 3.2.41 signal fail (SF):** A signal indicating that the associated data has failed in the sense that a near-end defect condition (not being the degraded defect) is active.
- 3.2.42 termination connection point (TCP):** A special case of a connection point where a trail termination function is bound to an adaptation function or a connection function. In the information model, the termination connection point is called the trail termination point (TTP).
- 3.2.43 timing information (TI):** The information passing across a timing point.
- 3.2.44 timing point (TP):** A reference point where an output of the synchronization distribution layer is bound to the input of an adaptation source or connection function, or where the output of an adaptation sink function is bound to an input of the synchronization distribution layer.
- 3.2.45 trail signal degrade (TSD):** A signal degrade indication output at the access point of a termination function.
- 3.2.46 trail signal fail (TSF):** A signal fail indication output at the access point of a termination function.
- 3.2.47 trail termination function (TT):** An atomic function within a layer that generates, adds and monitors information concerning the integrity and supervision of adapted information.
- 3.2.48 transit delay:** Transit delay is defined as the period of time taken for an information bit arriving at a network element input port to reappear at an output port on the same network element via a defect free trail.
- 3.2.49 undefined bit:** If a bit is undefined, its value is set to a logical "0" or a logical "1". See regional standards for further specifications of the value of undefined bits.
- 3.2.50 undefined byte:** If a byte is undefined, it contains eight undefined bits.
- 3.2.51 unidirectional trail/connection type:** A one-way trail/connection through a transport network.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

A	Adaptation function
AcCID	Accepted Channel Identifier
AcEXI	Accepted Extension header Identifier
Ack	Acknowledge
AcPFI	Accepted Payload Frame check sequence Identifier
AcPTI	Accepted Payload Type Identifier
AcSL	Accepted Signal Label
AcSQ	Accepted Sequence indicator
AcTI	Accepted Trace Identifier
AcUPI	Accepted User Payload Identifier

AI	Adapted Information
AIS	Alarm Indication Signal
ALS	Automatic Laser Shutdown
AP	Access Point
API	Access Point Identifier
APS	Automatic Protection Switching
APSD	Automatic Power ShutDown
AU	Administrative Unit
AU-n	Administrative Unit, level n
BER	Bit Error Ratio
BIP	Bit Interleaved Parity
C	Connection function
cHEC	core Header Error Check
CI	Characteristic Information
CID	Channel Identifier
CK	Clock
CM	Connection Matrix
COMMS	Communication access
COMMS_AC	COMMS access function
CP	Connection Point
CRC	Cyclic Redundancy Check
CRC16Err	Cyclic Redundancy Check-16 Errors
CRC-n	Cyclic Redundancy Check, width N
CSF	Client Signal Fail
CTRL	Control field sent from source to sink
D	Data
D	Delay
D_FT	Diagnostic Flow Termination
D_TT	Diagnostic Trail Termination
DCC	Data Communication Channel
DCI	Defect Clear Indication
DE	Drop Eligible
DEG	Degraded
DEGTHR	Degraded Threshold
DMFI	relative ('Differential') delay in Multiframe Indicator units
DNU	Do Not Use
DS	Defect Second

EBC	Errored Block Count
ECenabled	Error Correction enabled
EDC	Error Detection Code
EDCV	Error Detection Code Violation
EFS	Equipment Functional Specification
eHEC	extension Header Error Check
EMF	Equipment Management Function
EOS	End Of Sequence
EQ	Equipment
Eq	PDH Electrical signal, bit rate order q
ES	Electrical Section
ES	Errored Second
ES1	Electrical Section, level 1
ETCn	Ethernet physical coding sublayer
ETH	Ethernet VLAN
ETYn	Ethernet physical
EXI	Extension header Identifier
EXM	Extension header Mismatch
ExSL	Expected Signal Label
ExSQ	Expected Sequence indicator
ExtCmd	External Command
ExTI	Expected Trace Identifier
F_B	Far-end Block
F_DS	Far-end Defect Second
F_EBC	Far-end Errored Block Count
FAS	Frame Alignment Signal
FDI	Forward Defect Indication
FDis	Frames Discarded
FIT	Failure In Time
FM	Fault Management
FOP	Failure Of Protocol
FOP-CM	Failure Of Protocol Configuration Mismatch
FOP-NR	Failure Of Protocol No Response
FOP-PM	Failure Of Protocol Provisioning Mismatch
FOPR	Failure Of Protocol, sink (Receive) direction
FOPT	Failure Of Protocol, source (Transmit) direction
FOP-TO	Failure Of Protocol Timeout

FP	Flow Point
FS	Frame Start
FT	Flow Termination
GFP	Generic Framing Procedure
GFP-F	Frame-mapped Generic Framing Procedure
GFP-T	Transparent Generic Framing Procedure
GID	Group Identifier
HEC	Header Error Check
HO	Higher Order
HO	Hold Off
I	layer network Interworking function
ID	Identifier
IEC	Incoming Error Count
IF	In Frame state
IM	In Multiframe state
LBC	Lost Block Count
LC	Link Connection
LCAS	Link Capacity Adjustment Scheme
LCK	Locked
LCS	Loss of Character Synchronization
LFD	Loss of Frame Delineation
LO	Lower Order
LOA	Loss Of Alignment; also generic for loss of frame, loss of multiframe and loss of pointer
LOC	Loss Of Continuity
LOF	Loss Of Frame
LOM	Loss Of Multiframe
LOP	Loss Of Pointer
LOS	Loss Of Signal
ME	Maintenance Entity
MEG	Maintenance Entity Group
MEP	MEG End Point
MFI	Multiframe Indicator
MI	Management Information
MIP	MEG Intermediate Point
MMG	Mismerge
MND	Member Not Deskewable

MON	Monitored
MP	Management Point
MS	Multiplex Section
MSB	Most Significant Bit
MSn	Multiplex Section layer, level n
MSnP	Multiplex Section trail Protection layer, level n
MSOH	Multiplex Section Overhead
MST	Member Status (signal)
MSU	Member Signal Unavailable
MSU_L	Member Signal Unavailable, LCAS-enabled criteria
MTBF	Mean Time Between Failures
MTU	Maximum Transmission Unit
N_B	Near-end Block
N_DS	Near-end Defect Second
N_EBC	Near-end Errored Block Count
NCM	Network Connection Monitoring
NE	Network Element
NMON	Not Monitored
NNI	Network Node Interface
OAM	Operation, Administration and Maintenance
OCh	OTN optical Channel
ODI	Outgoing Defect Indication
ODU	OTN optical channel Data Unit
OEI	Outgoing Error Indication
OMSn	OTN optical Multiplex Section
OOF	Out Of Frame state
OOM	Out Of Multiframe state
OPSn	OTN optical Physical Section
OPUk	Optical channel Payload Unit-k
OS	Operation System
OS	Optical Section
OSn	Optical Section layer, level n
OTSn	OTN optical Transmission Section
OTU	OTN optical channel Transport Unit
P	Priority
P0x	64 kbit/s layer (transparent)
P11x	1544 kbit/s layer (transparent)

P12s	2048 kbit/s PDH path layer with synchronous 125 μ s frame structure
P12x	2048 kbit/s layer (transparent)
P21x	6312 kbit/s layer (transparent)
P22e	8448 kbit/s PDH path layer with 4 plesiochronous 2048 kbit/s
P22x	8448 kbit/s layer (transparent)
P31e	34 368 kbit/s PDH path layer with 4 plesiochronous 8448 kbit/s
P31s	34 368 kbit/s PDH path layer with synchronous 125 μ s frame
P31x	34 368 kbit/s layer (transparent)
P32x	44 736 kbit/s layer (transparent)
P4a	139 264 kbit/s PDH path layer with 3 plesiochronous 44 736 kbit/s
P4e	139 264 kbit/s PDH path layer with 4 plesiochronous 34 368 kbit/s
P4s	139 264 kbit/s PDH path layer with synchronous 125 μ s frame structure
P4x	139 264 kbit/s layer (transparent)
PC	Payload-Carrying
PDH	Plesiochronous Digital Hierarchy
pFCS	payload Frame Check Sequence
PFI	Payload Frame check sequence Indication
PI	Protection reference point Information
PI	Replication Information
PLCR	Partial Loss of Capacity Receive
PLCT	Partial Loss of Capacity Transmit
PLM	Payload Mismatch
PM	Performance Monitoring
POH	Path Overhead
PP	Protection reference Point
PP	Replication Point
Pq	PDH path layer, bit rate order q
Prov	Provisioned
ProvM	Provisioned Member
PTI	Payload Type Identifier
PTR	Pointer
P-Xv-L	LCAS-capable virtual concatenated path layer
RDI	Remote Defect Indication
rec	received
REI	Remote Error Indication
RI	Remote Information

RP	Remote Point
RS	Regenerator Section
RS	Re-Sequence
RS-Ack	Re-Sequence Acknowledge
RSn	Regenerator Section layer, level n
RSOH	Regenerator Section OverHead
RxSL	Received Signal Label
RxTI	Received Trace Identifier
S11	VC-11 path layer
S11D	VC-11 tandem connection sublayer
S11P	VC-11 path protection sublayer
S12	VC-12 path layer
S12D	VC-12 tandem connection sublayer
S12P	VC-12 path protection sublayer
S2	VC-2 path layer
S2D	VC-2 tandem connection sublayer
S2P	VC-2 path protection sublayer
S3	VC-3 path layer
S3D	VC-3 tandem connection sublayer using TCM option 2
S3P	VC-3 path protection sublayer
S3T	VC-3 tandem connection sublayer using TCM option 1
S4	VC-4 path layer
S4D	VC-4 tandem connection sublayer using TCM option 2
S4P	VC-4 path protection sublayer
S4T	VC-4 tandem connection sublayer using TCM option 1
SD	Signal Degrade
SDH	Synchronous Digital Hierarchy
SF	Signal Fail
Sk	Sink
Sn	higher order VC-n layer
SNC	Subnetwork Connection
SNC/I	Inherently monitored Subnetwork Connection protection
SNC/N	Non-intrusively monitored Subnetwork Connection protection
SNC/S	Sublayer (tandem connection) monitored Subnetwork Connection protection
So	Source
SOH	Section Overhead
SQ	Sequence indicator

SQM	Sequence indicator Mismatch
SQv	validated Sequence indication
SSD	Server Signal Degrade
SSF	Server Signal Fail
SSM	Synchronization Status Message
STM	Synchronous Transport Module
TCM	Tandem Connection Monitor
TCP	Termination Connection Point
TCS	Traffic Conditioning and Shaping
TDM	Time Division Multiplexing
TF	Transmit Fail
TFAS	trail Trace identifier Frame Alignment Signal
TFP	Termination Flow Point
tHEC	type Header Error Check
TI	Timing Information
TIM	Trace Identifier Mismatch
TLCR	Total Loss of Capacity Receive
TLCT	Total Loss of Capacity Transmit
TP	Timing Point
TPmode	Termination Point mode
TS	Time Slot
TSD	Trail Signal Degrade
TSF	Trail Signal Fail
TSL	Trail Signal Label
TSx	Trail Signal status, x = F (fail) or D (degraded)
TT	Trail Termination function
TTI	Trail Trace Identifier
TTP	Trail Termination Point
TU	Tributary Unit
TUG	Tributary Unit Group
TU-m	Tributary Unit, level m
TxSL	Transmitted Signal Label
TxSQ	Transmitted Sequence number
TxTI	Transmitted Trace Identifier
UMST	(persistent) Unexpected Member Status (signal)
UNEQ	UNEQuipped
UNP	Unexpected Priority

UNPr	Unexpected Period
UPI	User Payload Identifier
UPM	User Payload Mismatch
VC	ATM Virtual Channel
VC	Virtual Container
VCAT	Virtual Concatenation
VCG	Virtual Concatenation Group
VC-n	Virtual Container, level n
VLI	Virtual concatenation/Link capacity adjustment scheme (VCAT/LCAS) Information
VP	ATM Virtual Path
WDM	Wavelength Division Multiplexing
WTR	Wait to Restore

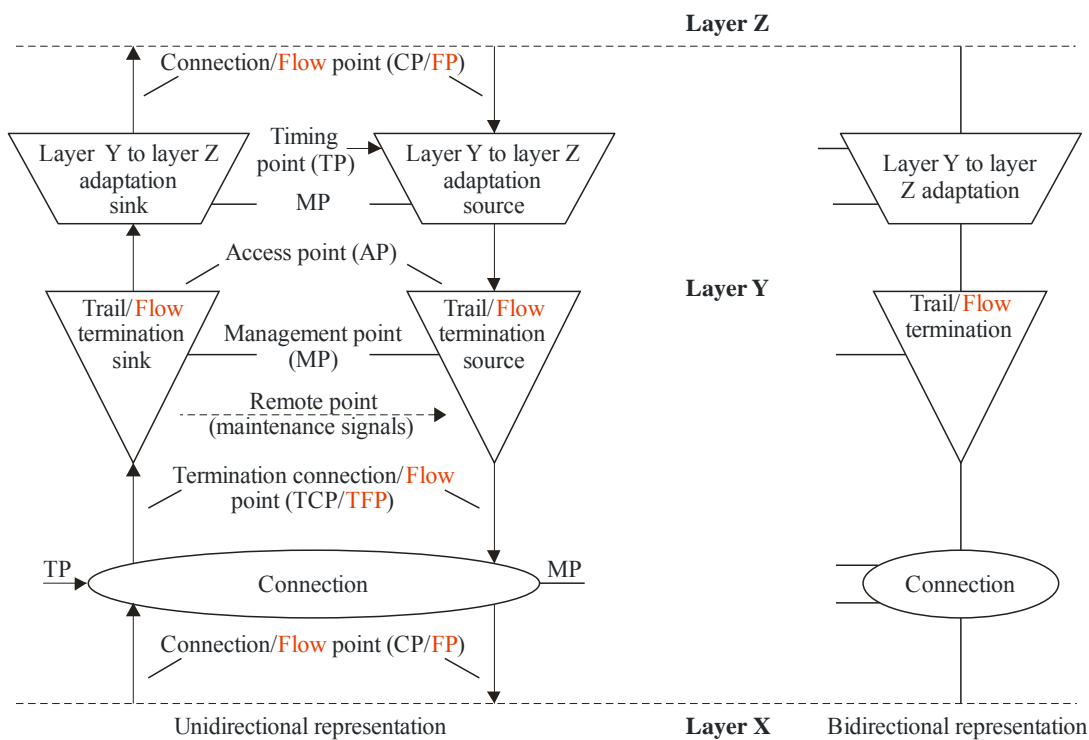
5 Methodology

5.1 Basic methodology

The methodology used to describe transport network functionality of network elements is based on the generic functional architecture of transport networks, the architectural entities and transport processing functions defined in [ITU-T G.805].

The functionality of transport processing functions within network elements is represented by atomic functions for each layer of the transport network and a set of combination rules for these functions. The basic set of atomic functions of a layer is shown in Figure 5-1 and consists of:

- trail and flow termination functions
- adaptation functions
- connection functions.



G.806(12)_F5-1

Figure 5-1 – Atomic functions and reference points

The interworking atomic function shown in Figure 5-2 is used for the special application of interworking between two network layers with similar characteristic information.

The diagrammatic convention for a communication access (COMMS) atomic function is shown in Figure 5-3b. The COMMS function provides access at intermediate connection points along a connection to general purpose communication channels.

The diagrammatic convention for a traffic conditioning and shaping (TCS) atomic function is shown in Figure 5-3c. The TCS sink function provides the traffic conditioning process. The TCS source function provides the shaping process.

An atomic function is described by the processes within the function, its reference points and the information flow via these reference points.

Within a network element, transport processing functions may interact with equipment management functions (EMFs) for fault, performance and configuration management. For the common EMF requirements, see [ITU-T G.7710].

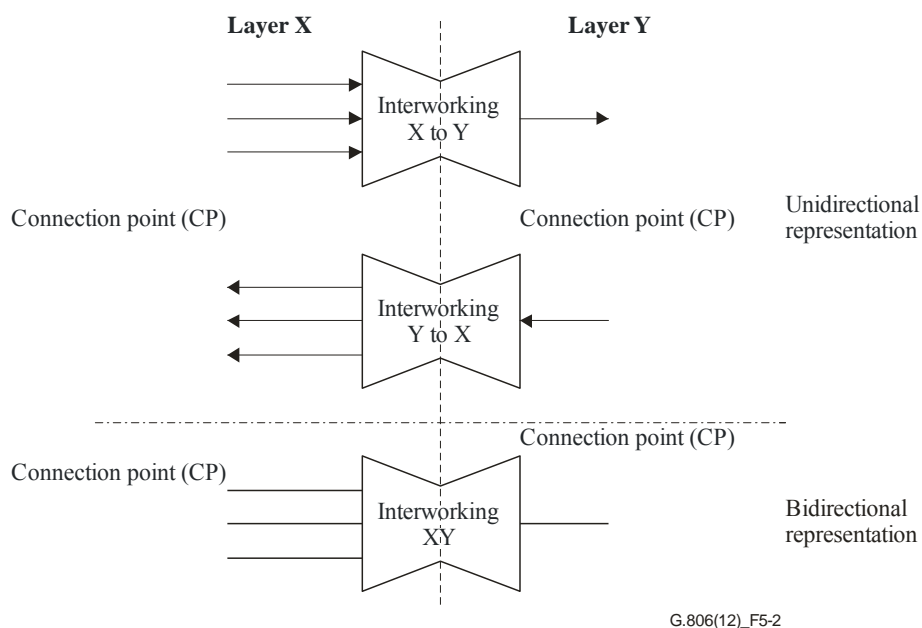


Figure 5-2 – Layer network interworking function

An atomic function may have several transmission reference points as input or output as shown in Figure 5-2.

5.2 Transmission layer naming

In order to identify the numerous transmission layers of the transport network hierarchies, a specific naming scheme is defined. The naming scheme consists of:

- one or more letters to identify the hierarchy and/or, if necessary, a specific layer type;
- a number or a number/letter combination that indicates the hierarchy level;
- one or more letters for further details on layer, sublayer or specific frame structure.

Table 5-1 shows the currently defined layer names.

Table 5-1 – Transmission layers

Name	Layer	Defined in
OSn	STM-n optical section	[ITU-T G.783]
ES1	STM-1 electrical section	[ITU-T G.783]
RSn	STM-n regenerator section	[ITU-T G.783]
MSn	STM-n multiplex section	[ITU-T G.783]
MSnP	STM-n multiplex section protection sublayer	[ITU-T G.783]
Sn	SDH VC-n path layer	[ITU-T G.783]
SnP	SDH VC-n trail protection sublayer	[ITU-T G.783]
SnD	SDH VC-n TCM option 2 sublayer	[ITU-T G.783]
SnT	SDH VC-n TCM option 1 sublayer	[ITU-T G.783]
Eq	PDH electrical section	[ITU-T G.705]
Pqe	Plesiochronous framed PDH layer	[ITU-T G.705]
Pqs	Synchronous framed PDH layer	[ITU-T G.705]
Pqx	Unframed PDH layer	[ITU-T G.705]

Table 5-1 – Transmission layers

Name	Layer	Defined in
NS	Network synchronization layer	[ITU-T G.781]
SD	Synchronization distribution layer	[ITU-T G.781]
VC	ATM virtual channel	[ITU-T I.732]
VP	ATM virtual path	[ITU-T I.732]
ODU	OTN optical channel data unit	[ITU-T G.798]
OTU	OTN optical channel transport unit	[ITU-T G.798]
OCh	OTN optical channel	[ITU-T G.798]
OMSn	OTN optical multiplex section	[ITU-T G.798]
OTSn	OTN optical transmission section	[ITU-T G.798]
OPSn	OTN optical physical section	[ITU-T G.798]
ETH	Ethernet VLAN	[ITU-T G.8021]
ETYn	Ethernet physical	[ITU-T G.8021]
ETCn	Ethernet physical coding sublayer	[ITU-T G.8021]

5.3 Atomic function naming and diagrammatic conventions

The naming of adaptation, trail and flow termination and connection functions follow the following rules:

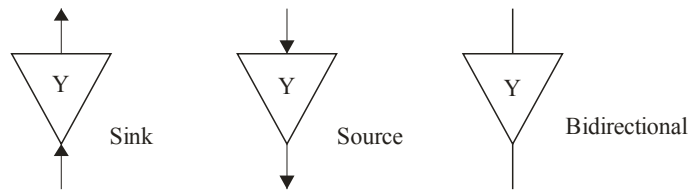
Adaptation function	<layer>/<client layer>_A[_<direction>]
Trail termination function	<layer>_TT[_<direction>]
Flow termination function	<layer>_FT[_<direction>]
Tandem connection trail termination function	<layer>T_TT[_<direction>]
Diagnostic trail termination function	<layer>D_TT[_<direction>]
Diagnostic flow termination function	<layer>D_FT[_<direction>]
Connection function	<layer>_C
Layer network interworking function	<layer>[</>/</><layer>[(set of accepted client layers X)]_I
Communication access function	<layer>/COMMS_AC[_<direction>]
Traffic conditioning and shaping function	<layer>_TCS[_<direction>]

Examples are: MS1/S4_A, S12/P12s_A_So, S4_TT, RS16_TT_Sk, S3_C, ETH_FT, ETY/ETH_A.

The diagrammatic conventions and nomenclature for adaptation, termination, connection layer interworking, communication access, and traffic conditioning functions (used to describe the atomic functions) are shown in Figures 5-3.



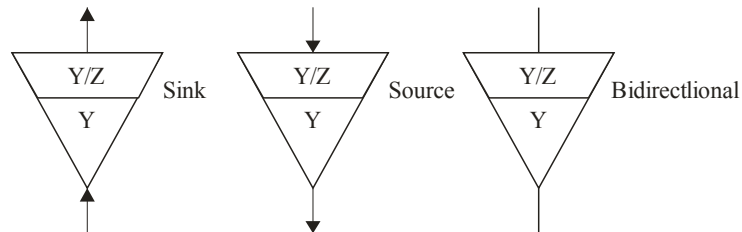
a) Adaptation functions from server layer Y to client layer Z



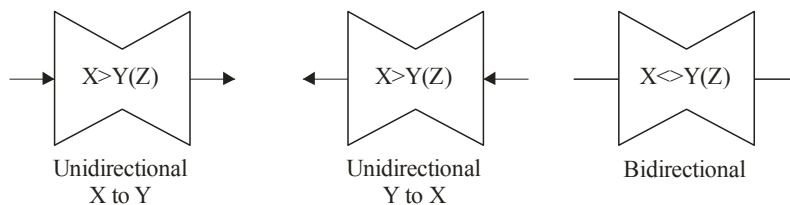
b) Trail/Flow termination functions in layer Y



c) Connection functions in layer Y



d) Trail/Flow termination function in layer Y and adaptation function to layer Z



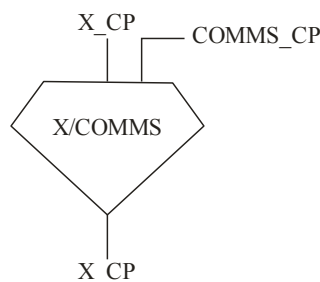
e) Interworking function between layer X and layer Y

G.806(12)_F5-3

NOTE 1 – If the above symbols are used for generic figures, i.e., not for specific layers, the layer references Y and Z may be omitted. Alternatively, the references may be to the type of function or layer, e.g., supervision, protection.

NOTE 2 – The order of the layers in the name of an interworking function can be changed (e.g., X>Y is identical to Y<X).

Figure 5-3a – Symbols and diagrammatic conventions



G.806(12)_F5-3a

Figure 5-3b – Diagrammatic convention for communication access function

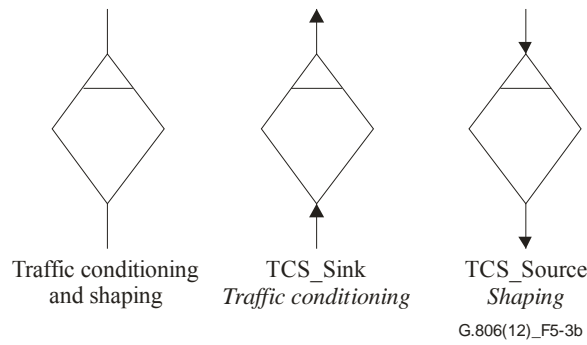


Figure 5-3c – Diagrammatic convention for traffic conditioning and shaping function

As an example of the use of this diagrammatic nomenclature, Figure 5-4 shows a unidirectional VC-4 path in an SDH network.

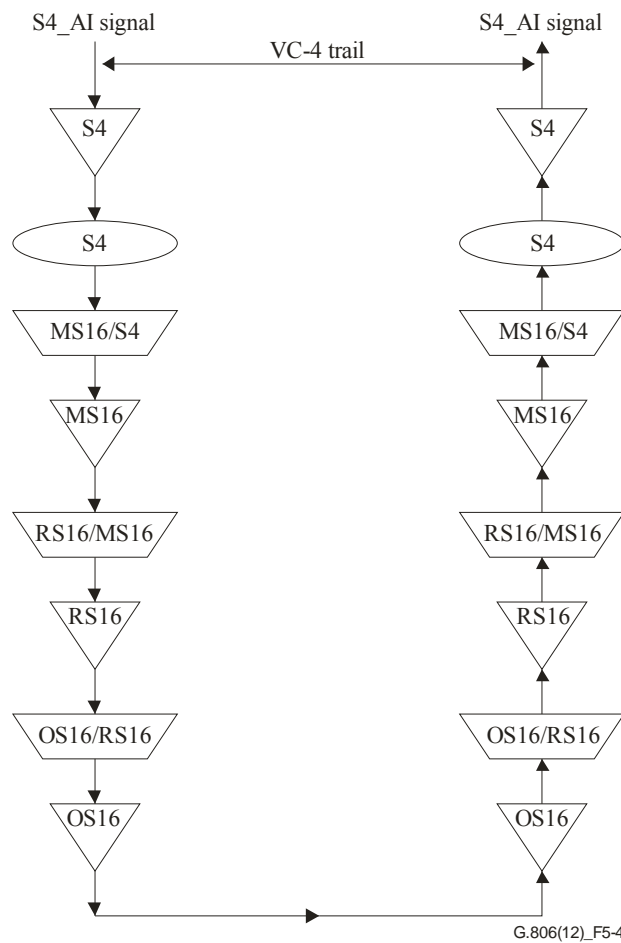
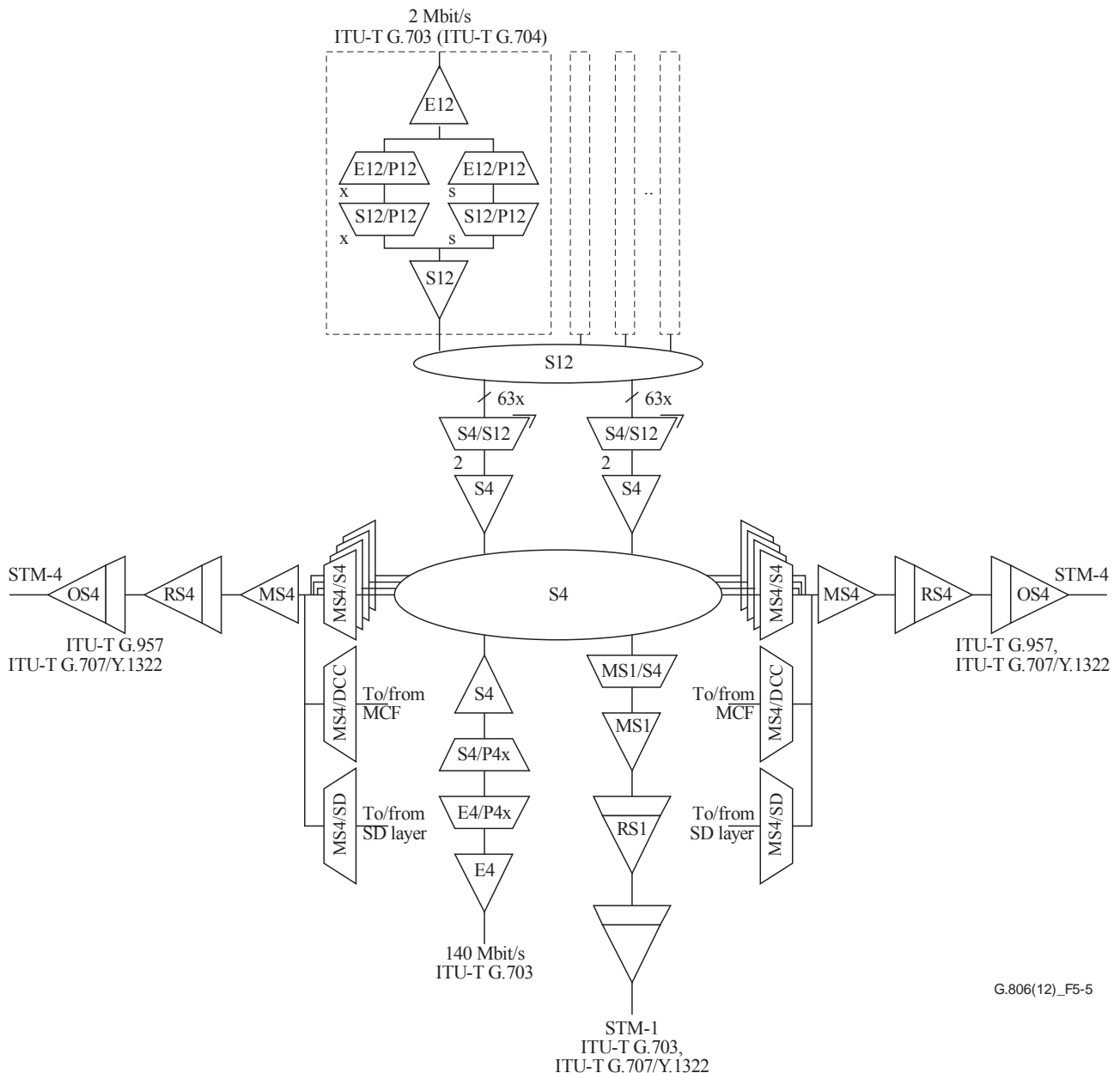


Figure 5-4 – Example of a unidirectional VC-4 path in an SDH network

As an example of the use of this diagrammatic nomenclature, Figure 5-5 shows an example of a transport level fragment of an equipment functional specification (EFS).



G.806(12)_F5-5

Figure 5-5 – Example of an SDH equipment functional specification

The equipment represented by the EFS supports the following interfaces: two optical STM-4, one electrical STM-1, one 140 Mbit/s, a number of 2 Mbit/s.

The STM-4 interfaces contain the MS-DCC signal and SSM signal. The STM-4 interfaces can contribute to the synchronization reference selection process in synchronization layers.

NOTE 1 – RS-DCC, RS-USER, RS-OW and MS-OW signals are not supported by the STM-4 interfaces.

NOTE 2 – RS-DCC, RS-USER, RS-OW, MS-DCC, MS-OW and contribution to the synchronization reference selection process are not supported by the STM-1 interface. Nor is the SSM supported on the output STM-1 signal.

The 140 Mbit/s signal is asynchronous mapped into a VC-4.

NOTE 3 – VC4-USER signals are not supported by the VC-4 processing.

The 2 Mbit/s signal is either asynchronous or byte synchronous mapped into the VC-12.

The VC-4 matrix contains twelve inputs and outputs: three towards a VC-4 termination function and the other nine to MSn to VC-4 adaptation functions.

NOTE 4 – Connectivity restrictions related to the VC-4 connection function are not represented in this presentation of the EFS. If applicable, connectivity restrictions can be presented in a further decomposed connection function representation, or by means of connectivity tables as shown in Appendix I.

NOTE 5 – The VC-4 connection function can support SNC protection switching. Such can be represented by means of a "rounded box" around the ellipse, as defined in [ITU-T G.803].

Two VC-4 signals can be terminated when they contain a TUG structure with sixty-three TU-12s. The resulting one hundred and twenty six VC-12 signals are connected to the VC-12 connection function that is also connected to a number of VC-12 termination functions.

NOTE 6 – Connectivity restrictions related to the VC-12 connection function are not represented in this presentation of the EFS. If applicable, connectivity restrictions can be presented in a further decomposed connection function representation, or by means of connectivity tables as shown in Appendix I.

NOTE 7 – The VC-12 connection function can support SNC protection switching. Such can be represented by means of a "rounded box" around the ellipse, as defined in [ITU-T G.803].

Examples of possible connectivity are:

- a VC-4 from an STM-4 interface can be passed through to the other STM-4 interface, with or without time slot interchange;
- a VC-4 from an STM-4 interface can be passed through (or dropped) to the STM-1 interface;
- a VC-4 from an STM-4 interface can be terminated, making the 140 Mbit/s payload available at the 140 Mbit/s interface;
- a VC-4 from an STM-4 interface can be terminated, making the TUG payload accessible for further processing;
- a VC-12 from an STM-4 interface can be passed through to the other STM-4 interface, with or without time slot interchange between the VC-4 server signals;
- a VC-12 from an STM-4 or the STM-1 interface can be terminated (after VC-4 termination), making the 2 Mbit/s payload available at a 2 Mbit/s interface. Either asynchronous or byte synchronous mapping into the VC-12 is supported;
- a VC-12 from an STM-4 interface can be passed through (dropped) to the STM-1 interface (after VC-4 termination), with or without time slot interchange between the VC-4 server signals;
- VC-4 SNC/I protection could be supported between e.g., two VC-4s within the two STM-4 signals, or between a VC-4 within an STM-4 signal and the VC-4 in the STM-1 signal;
- VC-12 SNC/I protection could be supported between two VC-12s within the two TUG structured terminated VC-4 signals. These two VC-4 signals can come from the two STM-4 signals or one STM-4 signal and the STM-1 signal.

5.4 Reference point naming

The atomic functions are defined between fixed reference points at which defined information is assumed to be present. That is, at a given reference point, specific types of information can always be assumed to be present. There are several different types of reference points within the functional model, including reference points for:

- transmission signals
- management information
- timing references
- remote information
- protection information
- replication information.

5.4.1 Transmission reference points

Because they are so numerous, and their detailed characteristics are so important to the functional model, transmission reference points are designated with a more complex naming convention. A transmission reference point name is formed by a transmission layer designation, followed by an underscore character, followed by either AP or CP or FP, depending on whether that reference point is an access point (AP) or a connection/flow point (CP/FP). As described in [ITU-T G.805], the information at an access point is a signal into which the client signal(s) has been mapped, but which does not include the full complement of overhead or OAM information for the given layer. The information at a connection/flow point is a signal that includes the full complement of overhead or OAM information. The access point is at the server side of adaptation functions and the client side of termination functions. The connection/flow point is at the client side of adaptation functions and the server side of termination functions (Figure 5-1). Thus, a transmission reference point name is formed according to the syntax:

<TransmissionReferencePointName> = <LayerName>_<AP or CP or FP>

<TransmissionReferencePointNameDiagnostic> = <LayerName>D_<AP>

<TransmissionReferencePointNameTandem> = <LayerName>T_<AP>

Connection and flow points represent the same type of reference points in two types of layer networks. The term connection point is used in layer networks which support unidirectional and bidirectional 2-port point-to-point and n-port (n>2) unidirectional point-to-multipoint connections in which the forwarding of information applied at the input port is performed by means of broadcasting/flooding that information towards all output ports with the exception of the output port associated with the input port. The term flow point is used in layer networks which support unidirectional and bidirectional 2-port point-to-point and n-port (n>2) unidirectional point-to-multipoint connections, as well as supporting bidirectional n-port (n>2) rooted-multipoint and n-port (n>2) multipoint-to-multipoint connections in which the forwarding of information applied at one of the input ports is performed by means of either unicasting towards a single output port, multicasting towards a subset of the output ports, or broadcasting/flooding that information towards all output ports (with the exception of the output port associated with the input port).

5.4.2 Management reference points

Management reference points are also quite numerous and are, therefore, named directly after the name of the associated function according to the syntax:

<ManagementReferencePointName> = <FunctionName>_MP

Thus, for example, the management reference point for the OS_TT function is named OS_TT_MP.

5.4.3 Timing reference points

Timing reference points are named directly after the name of the associated layer according to the syntax:

<TimingReferencePointName> = <LayerName>_TP

Thus, for example, the timing reference point for the VC-4 layer is named S4_TP.

5.4.4 Remote reference points

Remote reference points are named directly after the name of the associated layer according to the syntax:

<RemoteReferencePointName> = <LayerName>_RP

Thus, for example, the remote reference point for the VC-12 layer is named S12_RP.

5.4.5 Protection reference points

Protection reference points are named directly after the name of the associated layer according to the syntax:

$$\langle \text{ProtectionReferencePointName} \rangle = \langle \text{LayerName} \rangle_PP$$

Thus, for example, the protection reference point for the ODU layer is named ODU_PP. The protection reference point is used to transport automatic protection switching (APS) information between a layer's connection function and its server adaptation function in the case of compound link subnetwork connection group protection with inherent monitoring (CL-SNCG/I).

5.4.6 Replication reference points

Replication reference points are named directly after the name of the associated layer according to the syntax:

$$\langle \text{ReplicationReferencePointName} \rangle = \langle \text{LayerName} \rangle_PP$$

5.5 Reference point information naming

The information passing a CP is called characteristic information (CI), the information passing an AP is called adapted information (AI), the information passing a MP is called management information (MI), and the information passing a TP is called timing information (TI).

5.5.1 Transmission reference point information naming

The coding of the characteristic information (CI) and adapted information (AI) in the model follows the following rules:

$$\langle \text{layer} \rangle_ \langle \text{information type} \rangle_ \langle \text{signal type} \rangle[/ \langle \text{number} \rangle].$$

[...] optional term

$\langle \text{layer} \rangle$ represents one of the layer names (e.g., RS1)

$\langle \text{information type} \rangle$ CI or AI

$\langle \text{signal type} \rangle$ CK (clock), or
D (data), or FS (frame start), or
SSF (server signal fail), or
TSF (trail signal fail), or
SSD (server signal degrade), or
TSD (trail signal degrade), or
APS (automatic protection switching), or
P (priority), or
DE (drop eligible)

$\langle \text{number} \rangle$ indication of multiplex/inverse multiplex number; e.g., (1,1,1) for the case of a TU-12 within a VC-4

AI and CI coding examples are: MS1_CI_D, RS16_AI_CK, P12x_AI_D, S2_AI_So_D(2,3,0).

Within the network each access point is uniquely identified by means of its access point identifier (API) (see [ITU-T G.831]). The termination connection/flow point (TCP/TFP) (see Figure 5-1), can be uniquely identified by means of the same API. The connection/flow point (CP/FP) (see Figure 5-1), can be uniquely identified by the API extended with the multiplex number, e.g., the AU or TU number, the VPI or VCI, or the VLAN ID, the VPI or VCI, or the VLAN ID.

Example: a VC12 CP (S12_CP) can be identified by means of the API of the S4_AP, extended with the TU12 TUG number (K,L,M) and an ETH FP (ETH_FP) can be identified by means of the API of the ETH_AP, extended with the VLAN Identifier (VID).

5.5.2 Management reference point information naming

The coding of the MI signals follows the following rule:

<atomic function>_MI_<MI signal type>

5.5.3 Timing reference point information naming

The coding of the TI signals follows the following rule:

<layer>_TI_<TI signal type: CK or FS>

5.5.4 Remote reference point information naming

The coding of the RI signals follows the following rule:

<layer>_RI_<RI signal type: RDI, REI, ODI or OEI>

5.5.5 Protection reference point information naming

The coding of the PI signals follows the following rule:

<layer>_PI_<PI signal type: APS>

5.5.6 Replication reference point information naming

The coding of the PI signals follows the following rule:

<layer>_PI_<PI signal type: D, DE, P>

5.6 Atomic function process allocation

5.6.1 Connection function

The connection function provides flexibility within a layer. It may be used by the network operator to provide routing, grooming, protection and restoration.

The model describes the connection function as a space switch that provides connectivity between its inputs and outputs. Connections might be set up, modified or torn down based on management commands via the MI interface. A connection is supported by a flow forwarding (FF) process. Multiple types of flow forwarding processes are illustrated in Appendix IX.

The connectivity between inputs and outputs of the connection function might be limited due to implementation constraints. Several examples are given in Appendix I.

NOTE – The connection function's flexibility process is modelled as a timing-transparent switch, also referred to as "space switch". In case of time division multiplexing, the switch matrix type may be either a "space switch" or a combination of "space and time switches". If a time switch is involved, the adaptation source functionality that performs the alignment to a common time base (clock) shall be located at the input of the switch matrix (connection function) rather than at the output (as in the functional model).

For the case of SDH, the location of the adaptation source functionality (i.e., elastic store and pointer generator) with respect to the connection functionality (i.e., switch matrix) is observable at the STM-N interface when the matrix connection is changed (e.g., due to SNC protection switching). A pointer with "enabled NDF" is generated when the adaptation source functionality is located at the output of the connection functionality. A pointer without "enabled NDF" is generated when the adaptation source functionality is located at the input of the connection functionality.

5.6.2 Trail termination function

The trail termination function performs the signal integrity supervision of the layer. This includes:

- connectivity supervision;
- continuity supervision;

- signal quality supervision;
- processing of maintenance information (forward/backward indications).

In the source direction, it generates and adds some or all of the following:

- error detection code or forward error indication (e.g., bit interleaved parity (BIP), cyclic redundancy check (CRC), incoming error count);
- frame or packet loss measurement information;
- synthetic frame or packet loss measurement information;
- frame or packet delay and delay variation measurement information;
- trail trace identifier (i.e., source address);
- maintenance entity group identifier;
- maintenance entity group end point identifier;

It conveys back the following remote information:

- remote error indicator signal (e.g., REI, OEI, E-bit), containing the number of detected error detection code violations in the received signal;
- remote defect indicator signal (e.g., RDI, ODI, A-bit), representing the defect status of the received signal;
- remote frame or packet loss measurement information;
- remote synthetic frame or packet loss measurement information;
- remote frame or packet delay and delay variation measurement information.

In the sink direction, it monitors for some or all of the following:

- signal quality (e.g., bit errors);
- (mis-)connection;
- near-end performance;
- far-end performance;
- server signal fail (i.e., alarm indication signal (AIS) instead of data);
- signal loss (disconnection, idle signal, unequipped signal).

NOTE – Functionality is reduced in the physical section layer termination functions, which can only monitor the signal loss. In addition, the physical section termination source function performs logical/optical or logical/electrical conversion. In addition, the physical section termination sink function performs optical/logical or electrical/logical conversion.

Bit errors are detectable via line code violations, parity violations or CRC violations; i.e., error detection code violations. Frame or packet loss is detectable via comparison of transmitted frame or packet count and received frame or packet count information. Synthetic frame or packet loss is detectable via comparison of transmitted synthetic frame or packet identifiers and received synthetic frame or packet identifiers. Frame or packet delay is detectable via comparison of transmitted frame or packet timestamps and received frame or packet timestamps.

To monitor the provisioning of flexibility within a network, access points (APs) will be identified (named/numbered). The API is inserted in the signal, by the trail termination source function, in the trail trace identifier (TTI). The trail termination sink function checks the received name/number with the expected one (provisioned by the network manager).

To enable single-ended maintenance, the defect status and number of error detection code violations detected at the sink trail termination are conveyed back to the source trail termination; the defect status via the remote defect indicator (RDI) signal and the number of error detection code violations via the remote error indicator (REI) signal. The RDI and REI signals are part of the trail overhead.

Degradation of the signal results in the detection of anomalies and defects. As a consequent action of the detection of certain near-end defects, the signal is replaced by the all-ones (AIS) signal and RDI is inserted in the return direction. The defects are reported to the fault management process.

The number of near-end block errors¹ per second is counted. The number of far-end block errors² per second is counted. A second is indicated as a near-end defect second in cases where a signal fail condition was detected in that second. A second is indicated as a far-end defect second in cases where a RDI defect was detected in that second.

Refer to the supervision process description (see clause 6) for detailed descriptions.

5.6.3 Adaptation function

An adaptation function represents the conversion process between server and client layers. One or more of the following processes may be present in an adaptation function:

- scrambling/descrambling
- encoding/decoding
- alignment (framing, pointer interpretation, FAS/PTR generation)
- bit rate adaptation
- frequency justification
- time slot/wavelength assignment/access
- multiplexing/demultiplexing
- timing recovery
- smoothing
- payload type identification
- payload composition selection
- administrative locking
- forward error correction
- port identification.

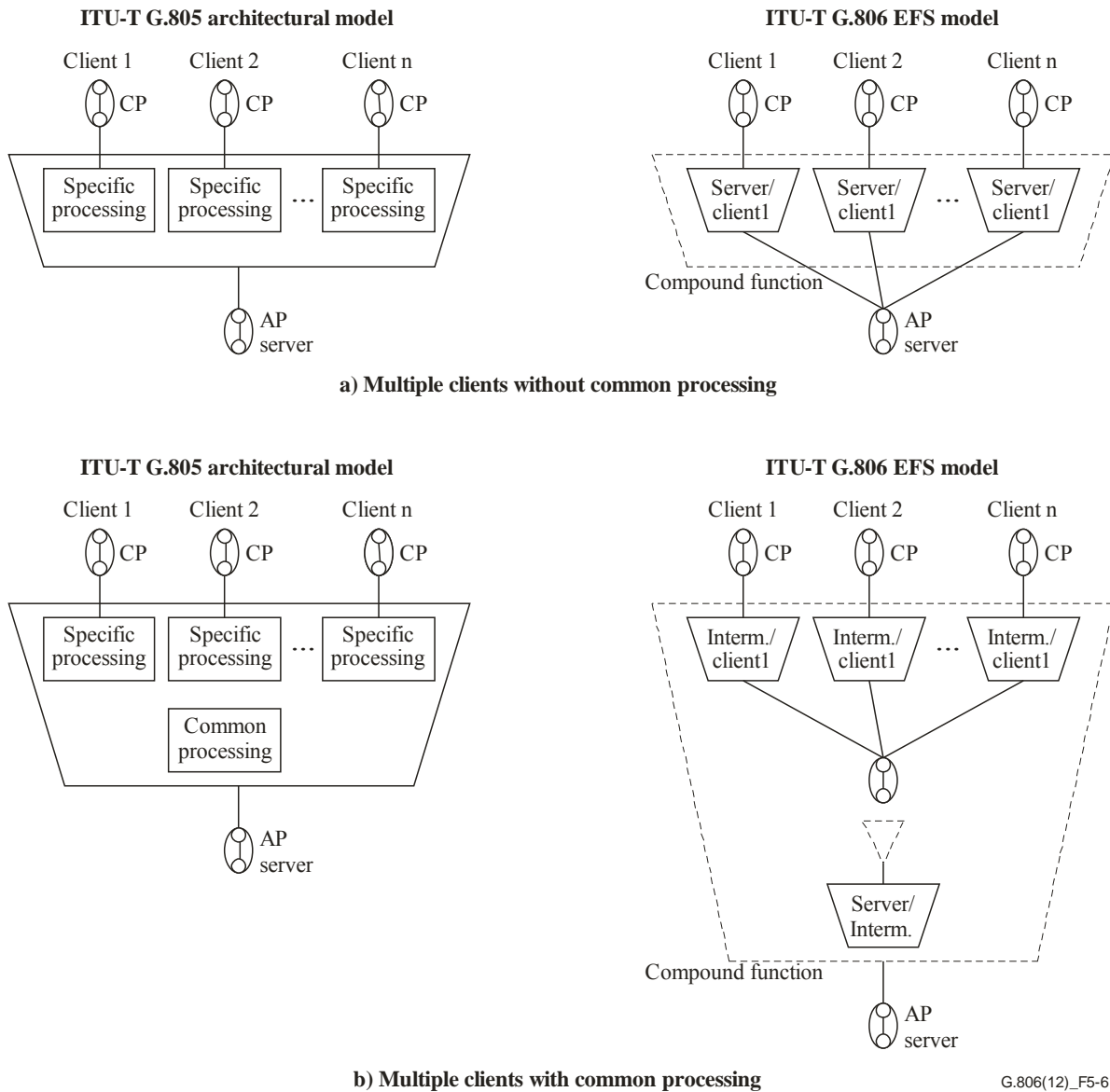
A server layer may provide transport for several client layer signals in parallel (e.g., n VC-4 in an STM- n signal), which is referred to as multiplexing. These client layer signals could be of different layer network types (e.g., mixture of VC-11/12/2/3 within a VC-4, DCCM, EOW, VC-4s in an STM- N multiplex section). According to [ITU-T G.805], this is represented in the functional model by one adaptation function that includes specific processes for each client layer signal. In addition, common processes for all or a set of client signals could be part of the adaptation function. For the equipment functional specification, a different approach is used which provides more flexibility. An adaptation function is defined for each client/server combination. This adaptation function performs the specific processing for this client/server relationship, including the time slot/wavelength assignment/access required for the multiplexing/demultiplexing. The individual adaptation functions are then connected to one AP as shown in Figure 5-6a. This can be viewed in the source direction as each adaptation function sends its AI on a different time slot/wavelength, and the AP just combines this information. In the sink direction, the full AI is distributed to all adaptation functions and each accesses only its specific time slot/wavelength.

¹ Detected by means of error detection code violation monitoring.

² Received via REI.

In the case of common processes, an intermediate signal is defined between the specific and common processes. The specific adaptation functions are between the client and the intermediate signal, and the common adaptation function is between the server and the intermediate signal, as shown in Figure 5-6b. The dotted trail termination function might be used due to historical reasons when a sublayer approach was used for this kind of modelling.

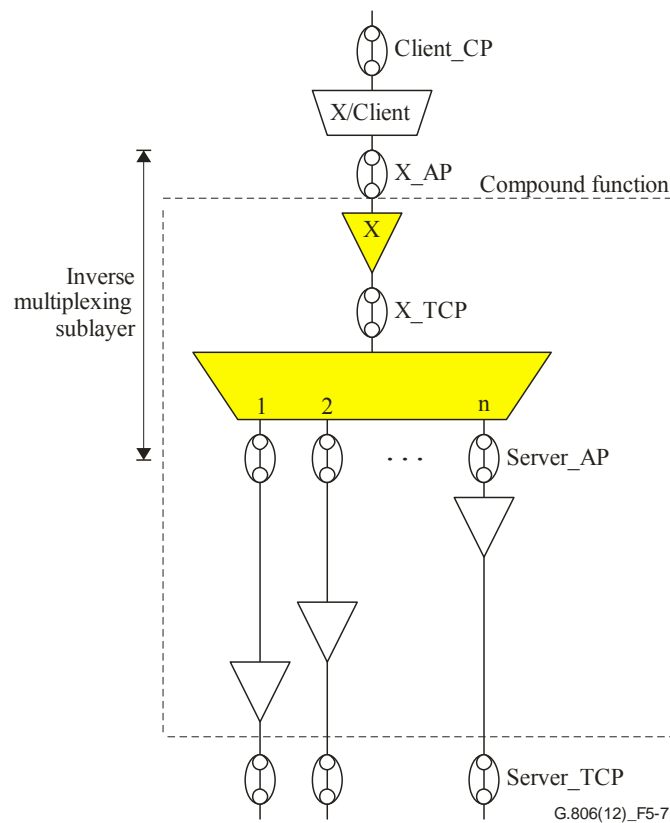
Note that the individual adaptation functions could be combined into a compound function as defined in clause 5.7.7.



Intern. Intermediate signal

Figure 5-6 – Comparison with the ITU-T G.805 multiplexing model

A client layer signal might be distributed via several server layer signals; this is referred to as inverse multiplexing. According to [ITU-T G.805] this is done by creating an inverse multiplexing sublayer with an adaptation function to the set of server layers as shown in Figure 5-7.



G.806(12)_F5-7

Figure 5-7 – Inverse multiplexing

The **scrambling** process alters digital data in a predefined way to ensure the resulting bit stream has a sufficient density of $0 \rightarrow 1$ and $1 \rightarrow 0$ transitions to allow bit clock recovery from it. The **descrambling** process recovers the original digital data from the scrambled bit stream.

NOTE 1 – The scrambling/descrambling process would be an adaptation process. The historical definition of signals in existing standards causes a violation of this process allocation, hence the scrambling/descrambling processes are often located in the trail termination functions. Refer to the individual atomic functions for details.

The **encoding/decoding** process adapts a digital data stream to the characteristics of the physical medium over which it is meant to be transported. The **decoding** process recovers the original digital data from the medium-specific form in which it is received.

The **alignment** process locates the first bit/byte of the framed signal (frame start (FS)) by means of a search for the frame alignment signal (FAS) or the interpretation of the pointer (PTR). If the FAS cannot be found, or the PTR is corrupted for a specific period, an alignment defect is detected (LOF, LOP). The alignment defect may be the result of the reception of the all-ones (AIS) signal. If so, the AIS defect is detected also. The defects are reported to the fault management layer/process.

NOTE 2 – The insertion of a frame alignment signal would be an A_{So} process. The (historical) definition of the many signals in existing standards causes a violation of this process allocation, hence the frame alignment insertion process is often located in the TT_{So} function. Refer to the individual atomic functions for details.

A second kind of alignment process aligns several input signals to a common frame start, as is the case for inverse multiplexing.

The **bit-rate adaptation** process accepts input information at a certain bit rate and outputs that same information at a different bit rate. In the source direction, this process creates gaps in which other functions can add their signals. An example is the S12/P12s_A_So function; the 2 Mbit/s signal input to this function is output at a higher bit rate. The created gaps will be filled with the VC-12 POH.

The **frequency justification** process accepts an input information at a certain frequency and outputs that same information either at the same or at a different frequency. In the source direction, in order to accommodate any frequency (and/or phase) differences between input and output signals, this process may write data into a specific "justification" bit/byte in the outgoing frame structure when the elastic store (buffer) is going to overflow. It will skip data writing when the elastic store is going to underflow. Examples are the S4/S12_A_So and P4e/P31e_A_So functions.

NOTE 3 – The commonly used terms "mapping" and "demapping" are covered by bit-rate adaptation and frequency justification processes.

The **time slot/wavelength assignment/access** process assigns the adapted client layer information to specific time slots/wavelength of the server layer in source direction. In the sink direction, the process provides access to the specific time slot/wavelength of the server layer. Time slots are used in TDM systems. Wavelengths are used in WDM systems. The specific time slot/wavelength is normally fixed for the adaptation function and indicated by an index numbering.

NOTE 4 – Variable connection of client signals to different time slots/wavelengths can be provided by the client layer connection function.

The **multiplexing/demultiplexing** process is modelled by means of multiple adaptation functions, connected to one AP as described above.

In the case where multiple adaptation functions are connected to the same AP and accessing the same time slots (bits/bytes), a **selection** process controls the actual access to the AP. In the atomic functions, this is modelled via the activation/deactivation signal (MI_Active). In the case where only one adaptation function is present, it is selected. Control is not required.

The **timing recovery** process extracts a clock signal, the "recovered clock", from the incoming data signal. The timing recovery process is performed in the adaptation sink function in the physical section layer; e.g., in OS16/RS16_A_Sk.

The **smoothing** process filters the phase step of "gapped input signals". The smoothing process is performed in the adaptation sink functions; e.g., in Sm/Xm_A_Sk, Pn/Pm_A_Sk.

Many layers are able to transport a variety of client signals applied to the layer via different adaptation functions. To monitor the provisioning process, the source adaptation inserts the appropriate code in the trail signal label (TSL). The sink adaptation will check the **composition of the payload**, comparing the received TSL number with its own.

The **administrative locking** process controls the forwarding of the characteristic information signals in an adaptation function depending on the administrative state (refer to [b-ITU-T X.731]). The forwarding of characteristic information is blocked when the administrative state is LOCKED and forwarding is enabled when the administrative state is UNLOCKED. In a locked condition, the LCK maintenance signal replaces the client's characteristic information.

The **forward error correction** process is a coding process which adds redundancy to the transmitted characteristic information using a predetermined algorithm such that the redundancy added by the coding allows the decoding to detect and correct a limited number of bit errors.

The **port identification** process adds source and destination port identifiers (typically referred to as addresses) to the adapted characteristic information to control delivery of this information to the intended subset of ports on a multipoint connection.

5.6.4 Layer network interworking function

A layer network interworking function represents the semantically transparent conversion of characteristic information between two layer networks. The conversion process maintains the integrity of the end-to-end supervision of the trail. Conversion of the adapted information may also be required. The integrity of the client layer characteristic information has to be maintained in this case. The interworking function may be limited to a set of client layer signals.

The process is specific for the interworked layers and may include processes from the adaptation and termination function.

5.7 Combination rules

5.7.1 General

In general, any functions that share the same characteristic, or adapted information, may be combined.

5.7.2 Binding at connection/flow points

The connection/flow point input (output) of an adaptation function may be bound to the connection/flow point output (input) of either a connection function, layer network interworking function or an adaptation function. The connection/flow point of a layer network interworking function may be bound to the connection/flow point of either a connection function or an adaptation function, as shown in Figure 5-8.

Example: An S12_CP of an S12_C function may be connected to an S12_CP of an S4/S12_A function.

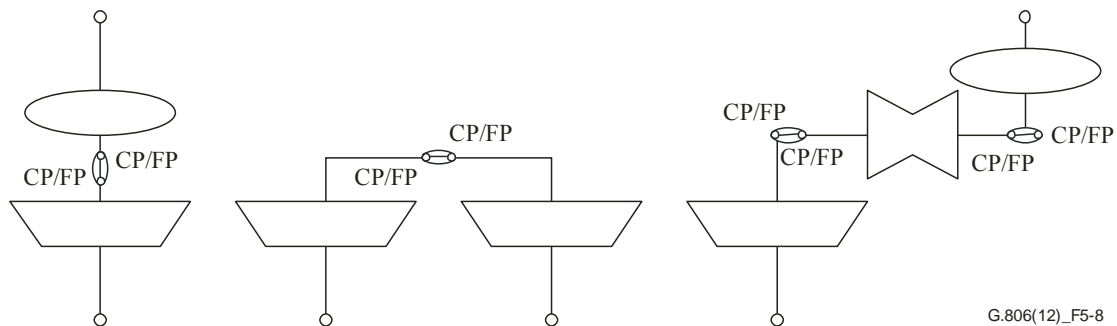


Figure 5-8 – Binding of connection/flow points (CP-CP binding)

5.7.3 Binding at (termination) connection/flow points

The termination connection/flow point output (input) of a trail termination function may be bound to the connection/flow point input (output) of either an adaptation function, layer network interworking function or a connection function or the termination connection/flow point input (output) of a trail termination function, as shown in Figure 5-9.

NOTE – Once bound, the CP/FP and TCP/TFP are referred to as a termination connection/flow point.

Example: An S12_TCP of an S12_TT function may be connected to an S12_CP of an S12_C function.

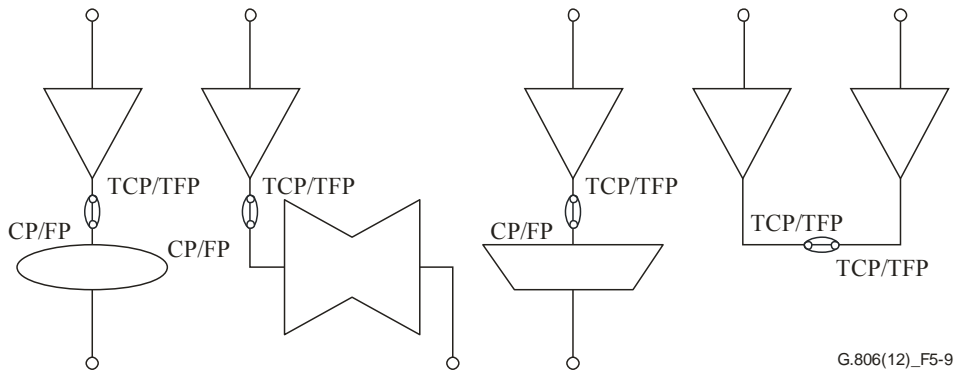


Figure 5-9 – Binding involving a termination of connection/flow points (TCP-CP/TFP-FP and TCP-TCP/TFP-FP binding)

5.7.4 Binding at access points

The AP input (output) of a trail termination function may be bound to the AP output (input) of an adaptation function as shown in Figure 5-10.

Example: An S4_AP of an S4/S12_A function may be connected to an S4_AP of an S4_TT function.

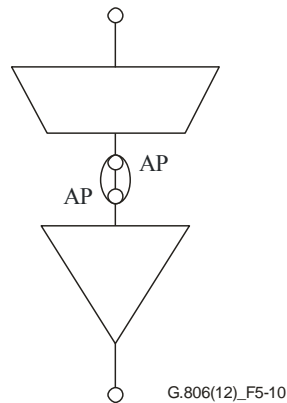


Figure 5-10 – Binding of access points (AP-AP binding)

5.7.5 Alternative binding representations

The binding at reference points can continue, according to the above rules, and create a path such as the one shown in Figures 5-4 and 5-5.

NOTE – The binding at reference points may also be represented as illustrated in Figure 5-11. In an equipment functional specification, the explicit reference to the reference points is not required if the atomic functions are named. In such a case, the names of the reference points are obvious.

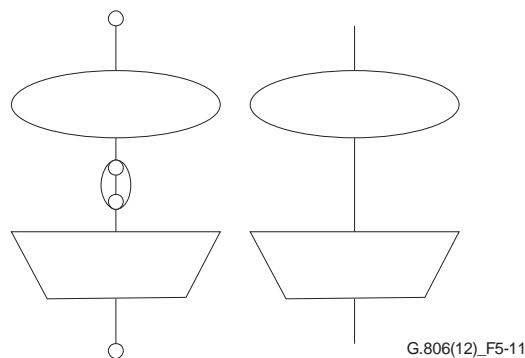


Figure 5-11 – Alternative binding representation

5.7.6 Directionality

Atomic functions are normally defined with unidirectional functionality, except for certain connection functions. The directionality of trail termination and adaptation functions is identified by the directionality identifier sink/source. The directionality of layer network interworking functions is identified by the direction of the arrow (>).

Two unidirectional atomic functions with opposite directionality may be associated as a bidirectional pair (when a function is referred to without the directionality qualifier, it can be taken to be bidirectional). In the case of trail termination functions, their remote information reference points are connected together in this case.

Bidirectional servers may support bidirectional or unidirectional clients, but unidirectional servers may only support unidirectional clients.

5.7.7 Compound functions

Combinations of atomic functions in one or more layer(s) may be identified by a special symbol, a compound function. Three examples are shown in Figures 5-12, 5-13 and 5-14.

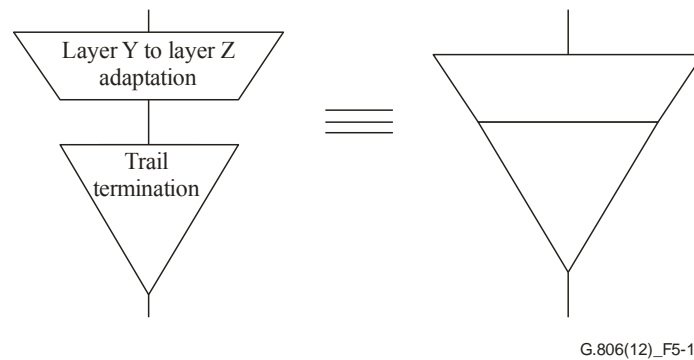


Figure 5-12 – Compound termination/adaptation function

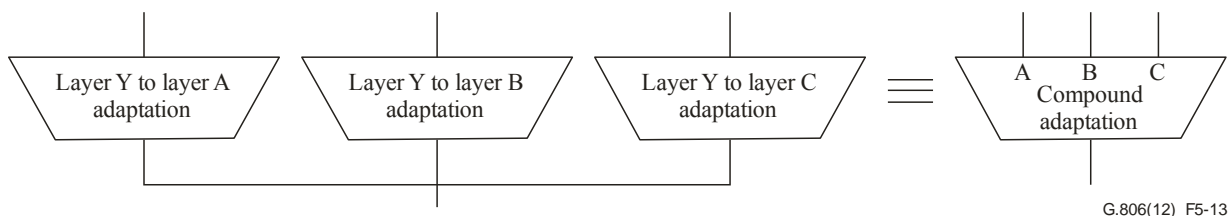


Figure 5-13 – Compound adaptation function

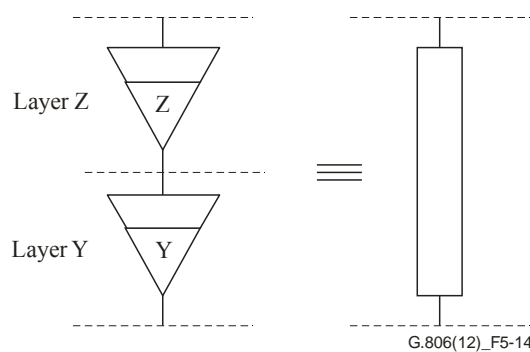


Figure 5-14 – Compound function spanning multiple layers

The diagrammatic conventions for NCM MEG end point (NCM MEP) compound functions are shown in Figure 5-15. NCM MEP functions may either have a single termination connection/flow port and access port (left) or multiple termination connection/flow ports and multiple access ports (right).

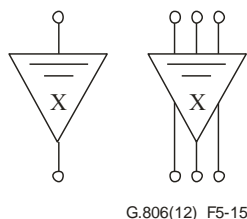


Figure 5-15 – Diagrammatic convention for NCM MEP compound functions

The diagrammatic conventions for TCM MEG end point (TCM MEP) compound functions are shown in Figure 5-16. TCM MEP functions may either have a single termination connection/flow port and access port (left) or multiple termination connection/flow ports and multiple access ports (right).

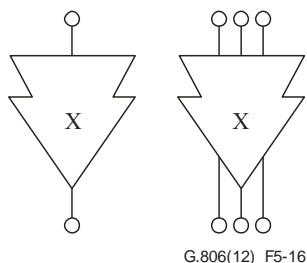


Figure 5-16 – Diagrammatic convention for TCM MEP compound functions

The diagrammatic conventions for MEG intermediate point (MIP) compound functions are shown in Figure 5-17. The MIP compound functions consist of two pairs of the diagnostic adaptation and trail/flow termination functions, each facing in opposite directions. MIP functions may either have a single connection/flow port (left) or multiple connection/flow ports (right).

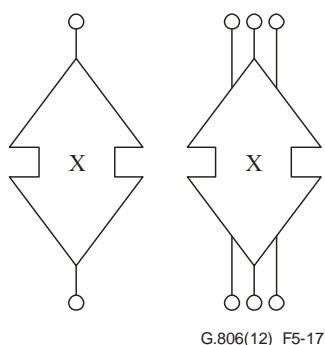


Figure 5-17 – Diagrammatic convention for MEG intermediate point (MIP) compound functions

A variant of the MIP compound function is the half MIP compound function, which consists of a single pair of the diagnostic adaptation and trail/flow termination functions (Figure 5-18). Half MIP functions may either have a single connection/flow port (left) or multiple connection/flow ports (right).

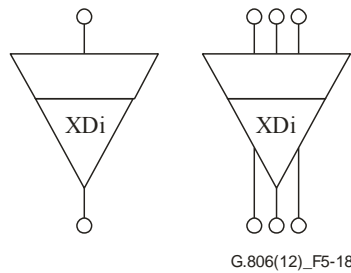


Figure 5-18 – Half MIP compound functions

5.8 Fault management and performance monitoring naming

The naming of supervision variables (see also Figures 6-1 and 6-2) are as follows.

The supervision variables are defined as "yZZZ", with:

y	defect:	y = d
	fault cause (i.e., correlated defect):	y = c
	consequent action request:	y = a
	performance parameter:	y = p
	anomaly:	y = n
ZZZ	kind of defect, fault cause, failure, consequent action, performance parameter or command	

dZZZ and cZZZ represent Boolean variables with states TRUE or FALSE. pZZZ represents an integer variable. aZZZ, except aREI, represents a Boolean variable; aREI represents an integer variable.

5.9 Fault management and performance monitoring specification techniques

The defect correlation and consequent action specifications make use of the following supervision equation techniques:

aX ← A or B or C

cY ← D and (not E) and (not F) and G

pZ ← H or J

"aX" represents the control of consequent action "X". The associated consequent action will be performed if the Boolean equation "A or B or C" is true. Otherwise, if the equation is false, the consequent action will not be performed. Consequent actions are, e.g.,: insertion of all-ones (AIS) signal, insertion of RDI signal, insertion of REI signal, activation of signal fail or signal degrade signals.

"cY" represents the fault cause "Y" which is (will be) declared if the Boolean expression "D and (not E) and (not F) and G" is true. Otherwise (expression is false), the fault cause is (will be) cleared. MON will often be a term in this equation (see clause 6.1).

"pZ" represents the performance monitoring primitive "Z", a value which, at the end of a one-second period, represents the number of errored blocks (or error detection code violations) or the occurrence of a defect in that second.

"A" to "J" represent either defects (e.g., dLOS), reporting control parameters (e.g., AIS_Reported), consequent actions (e.g., aTSF), or the number of errored blocks over a one-second period (e.g., Σ nN_B).

NOTE – Hardware faults causing signal transfer interruption is represented by "dEQ". Such faults contribute to the near-end performance monitoring primitive pN_DS.

6 Supervision

Transmission and equipment supervision processes are concerned with the management of the transmission resources in the network and they are only interested in the functionality that is being provided by a network element (NE). They require a functional representation of an NE that is implementation independent.

The supervision process describes the way in which the actual occurrence of a disturbance or fault is analysed with the purpose of providing an appropriate indication of performance and/or detected fault condition to maintenance personnel. The following terms are used to describe the supervision process: anomaly, defect, consequent action, fault cause, failure and alarm.

Any equipment faults are represented by the unavailability of the affected functions because the transmission management has no knowledge of the equipment as such. Most functions monitor the signals they are processing for certain characteristics and provide performance information or alarm conditions based on these characteristics. Therefore, transmission supervision processing provides information on the external interface signals that are processed by an NE.

The following basic supervision functions are defined:

- continuity supervision (trail termination)
- connectivity supervision (trail termination)
- signal quality supervision (trail termination)
- payload type supervision (adaptation)
- alignment supervision (adaptation)
- maintenance signal processing (trail termination, adaptation)
- protocol supervision (connection).

The supervision processes and their interrelationships within atomic functions are depicted in Figures 6-1 and 6-2. The interrelations between the supervision processes in atomic functions, and the equipment management function are defined in [ITU-T G.7710] and the corresponding technology-specific Recommendations.

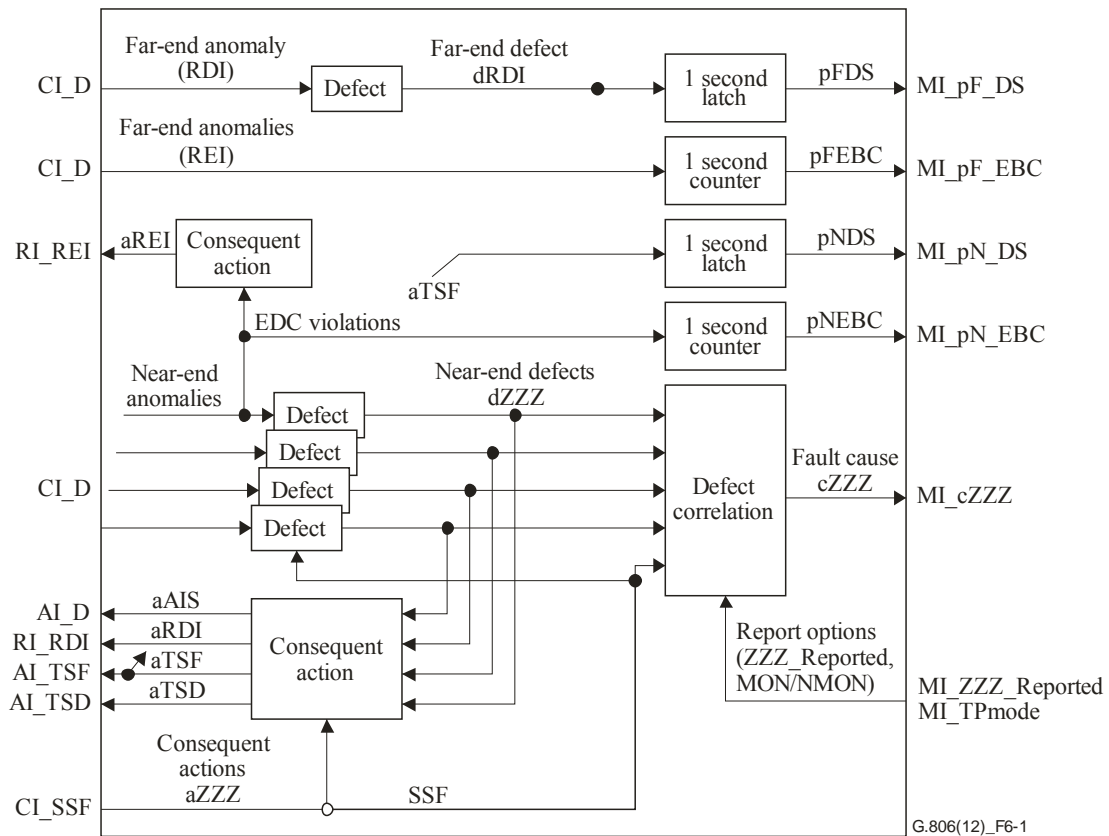


Figure 6-1 – Supervision process within trail termination functions

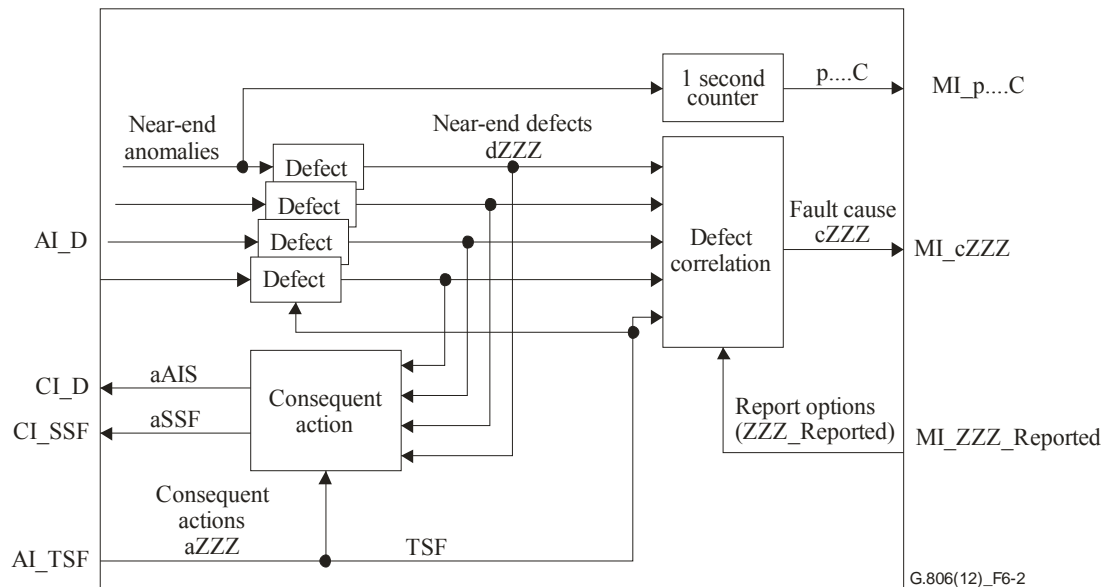


Figure 6-2 – Supervision process within adaptation functions

The filtering functions provide a data reduction mechanism within atomic functions on the anomalies and defects before being presented at the XXX_MP reference points. Four types of techniques can be distinguished:

- trail termination point and port modes
- one second integration

- defect detection
- fault management and performance monitoring correlations.

6.1 Trail termination point mode and port mode

To prevent alarms from being raised and failures being reported during trail provisioning actions, trail termination functions shall have the ability to enable and disable fault cause declaration. This shall be controlled via their termination point mode or port mode parameter.

The termination point mode (see Figure 6-3) shall be either "monitored (MON) or "not monitored" (NMON). The state shall be MON if the termination function is part of a trail and provides service, and NMON if the termination function is not part of a trail or is part of a trail that is in the process of set-up, breakdown or re-arrangement.

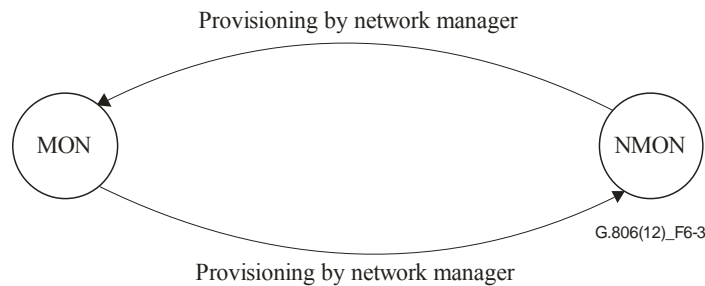


Figure 6-3 – Trail termination point modes

In physical section layers, the termination point mode is called the port mode. It has three modes (Figure 6-4): MON, AUTO and NMON. The AUTO mode is like the NMON mode with one exception: if the LOS defect clears, the port mode is automatically changed to MON. This allows for alarm-free installation without the burden of using a management system to change the monitor mode. The AUTO mode is optional. When it is supported, it shall be the default mode; otherwise, NMON shall be the default mode.

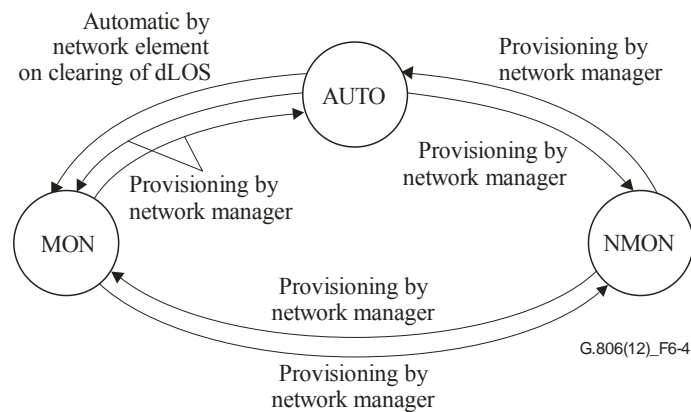


Figure 6-4 – Port modes

6.2 Defect filter

The (anomaly to) defect filter will provide a persistency check on the anomalies that are detected while monitoring the data stream; when passed, the defect is being detected.

Generic defect filters are defined below. Specific defect filter definitions can be found in the Recommendations for the specific hierarchies.

6.2.1 Continuity supervision

6.2.1.1 Generic behaviour

Continuity supervision monitors the integrity of the continuity of a trail. This is done by monitoring the presence/absence of the CI. The monitoring process can check for the whole CI (e.g., LOS at the physical layer) or a specific mandatory part of it (e.g., multiframe indication for SDH TCM). At path layer networks, a replacement signal might be generated by an open connection matrix (e.g., unequipped signal for SDH). The detection of this replacement signal is then an indication of loss of continuity.

Note that a server layer defect will result in a loss of continuity for client layers. This is normally detected via maintenance signalling (AIS, SSF, TSF) at the client layer and reported as an SSF alarm for the client layer (see clause 6.3)

6.2.1.2 Loss of signal defect (dLOS)

LOS signal supervision is used at the physical layer. For the specific detection processes, refer to the Recommendations for the specific hierarchies ([ITU-T G.783], [ITU-T G.705], [ITU-T G.781], [ITU-T G.798], and [ITU-T G.8021]).

6.2.1.3 Unequipped defect (dUNEQ)

Basic function sink direction

The unequipped overhead is recovered from the CP.

The unequipped defect (dUNEQ) shall be detected if z consecutive frames contain the unequipped activation pattern in the unequipped overhead. The dUNEQ defect shall be cleared if, in z consecutive frames, the unequipped deactivation pattern is detected in the unequipped overhead. Details for the UNEQ defect are provided in Table 6-1.

dUNEQ shall be cleared during SSF conditions. A new evaluation period for dUNEQ shall start after SSF is cleared.

NOTE – Some regional standards require a burst-proof algorithm of the UNEQ defect.

Table 6-1 – UNEQ defect details

Hierarchy	Layer	Unequipped overhead	Unequipped activation pattern	Unequipped deactivation pattern	z (Note)
SDH	S3/4 (VC-3/4)	C2 byte	"00000000"	≠ "00000000"	5
	S11/12/2 (VC-11/12/2)	V5, bits 5 to 7	"000"	≠ "000"	5
	S3D/S4D (VC-3/4 TCM option 2)	N1	"00000000"	≠ "00000000"	5
	S11D/S12D/S2D (VC-11/12/2 TCM)	N2	"00000000"	≠ "00000000"	5
PDH with SDH frame	P4s/3s (140/34 Mbit/s)	MA, bits 3 to 5	"000"	≠ "000"	3 to 5
	P4sD/P3sD (140/34 Mbit/s TCM)	NR	"00000000"	≠ "00000000"	5

NOTE – z is not configurable.

6.2.1.4 Loss of tandem connection defect (dLTC)

The function shall detect for the presence/absence of the tandem connection overhead in the TCM overhead by evaluating the multiframe alignment signal in the TCM multiframe overhead. The loss of tandem connection defect (dLTC) shall be detected if the multiframe alignment process is in the OOM state. The dLTC shall be cleared if the multiframe alignment process is in the IM state. For details on the alignment process, refer to Table 6-2, clause 8.2 and the specific equipment functional Recommendations ([ITU-T G.783] and [ITU-T G.705]).

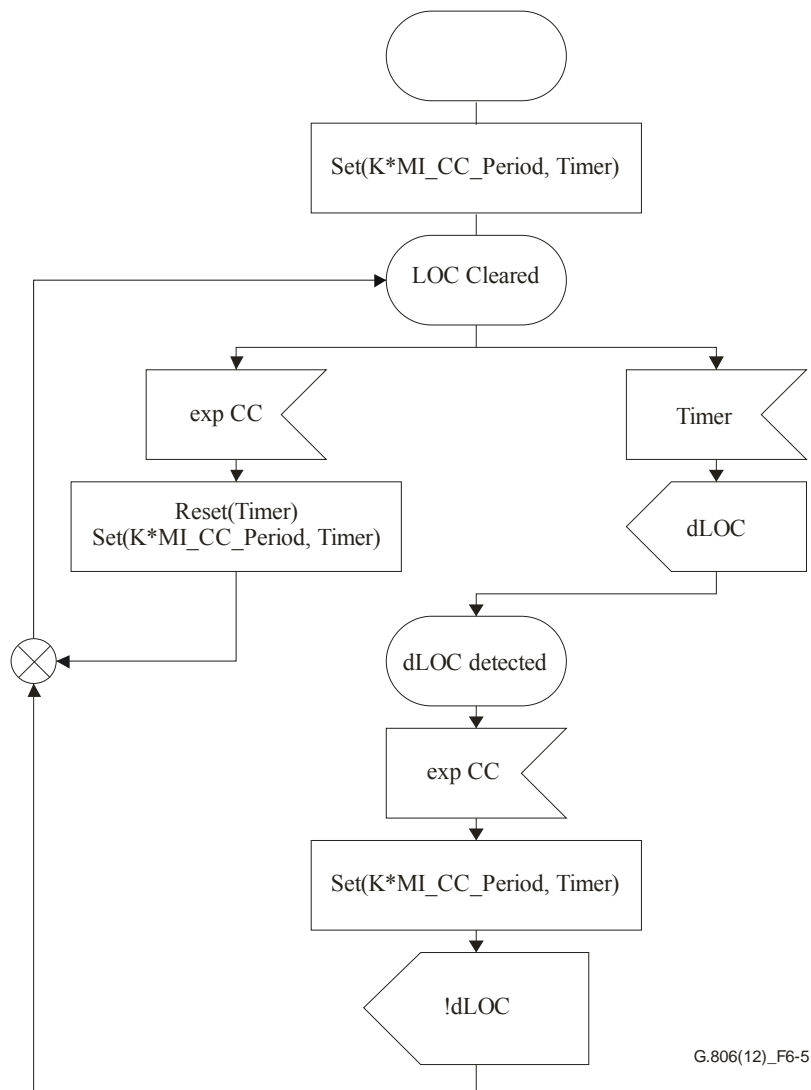
Table 6-2 – LTC defect details

Hierarchy	Layer	TCM multiframe overhead
SDH	S3D/S4D (VC-3/4 TCM option 2)	N1, bits 7 to 8
	S11D/S12D/S2D (VC-11/12/2 TCM)	N2, bits 7 to 8
PDH with SDH frame	P4sD/3sD (140/34 Mbit/s TCM)	NR, bits 7 to 8

6.2.1.5 Loss of continuity defect (dLOC)

The loss of continuity defect is calculated at the network layer. It monitors the presence of continuity in trails.

Its detection and clearance are defined in Figure 6-5. The 'period' in Figure 6-5 is the period as carried in the CC (continuity check) frame triggering the expCC event that means reception of a CC frame.



G.806(12)_F6-5

Figure 6-5 – dLOC detection and clearance process

6.2.2 Connectivity supervision

6.2.2.1 Generic behaviour

Connectivity supervision monitors the integrity of the routing of the trail between sink and source. Connectivity is normally only required if the layer provides flexible connectivity, both automatically (e.g., cross-connects controlled by the TMN) or manually (e.g., fibre distribution frame). The connectivity is supervised by attaching a unique identifier at the source. If the received identifier does not match this expected identifier, a connectivity defect has occurred.

6.2.2.2 Trail trace identifier processing and trace identifier mismatch defect (dTIM)

Basic function source direction

The generation of trail trace identifier (TTI) is optional and in the province of regional standards.

If TTI generation is not required, the content of the TTI overhead is not configurable.

If TTI generation is required, the TTI information derived from the management reference point (MI_TxTI) is placed in the TTI overhead position.

Basic function sink direction

The TTI overhead is recovered from the CP.

The detection of a trace identifier mismatch defect (dTIM) is optional and in the province of the regional standards.

In the case where dTIM detection is not required, the receiver shall be able to ignore the received TTI overhead values, and dTIM is considered "false".

In the case where dTIM detection is required, the following applies: the detection of dTIM is based on a comparison between the expected TTI, configured via the management reference point (MI_ExTI), and the accepted TTI (AcTI). If dTIM detection is disabled via an input ("Set") command (MI_TIMdis) at the management reference point, then dTIM is considered "false".

NOTE 1 – Acceptance criteria and defect specification for the TTI are for further study to ensure integrity, and robustness to errors for TIM.

NOTE 2 – A mismatch in the CRC-7 or TFAS signal of the 16-byte trace identifier results in the detection of the dTIM defect.

The accepted TTI shall be reported via the management point (MI_AcTI) to the EMF. The query of the AcTI shall be independent of the dTIM detection process.

NOTE 3 – Some equipment developed prior to the 04/97 revision of [ITU-T G.783] may not support this query in the event that trace identifier mismatch detection is disabled.

dTIM shall be cleared during SSF conditions. A new evaluation period for dTIM shall start after SSF is cleared.

Details on the TIM defect are provided in Table 6-3.

Table 6-3 – TIM defect details

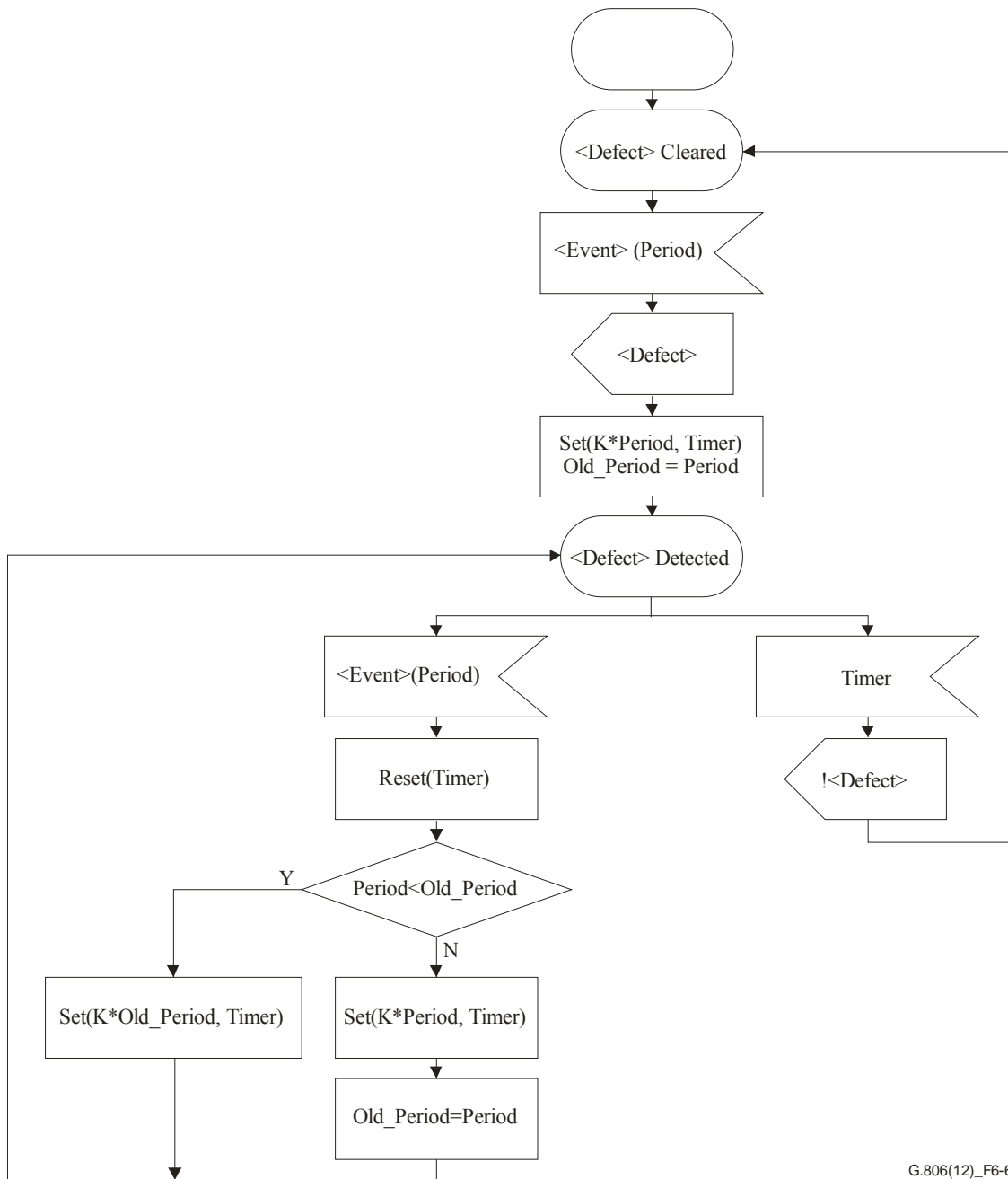
Hierarchy	Layer	TTI overhead	TTI format
SDH	RSn	J0 byte	1/16 byte (see [ITU-T G.707])
	S3/4 (VC-3/4) (see Note)	J1 byte	16/64 byte (see [ITU-T G.707])
	S3D/S4D (VC-3/4 TCM option 2)	N1, bits 7 to 8, frame 9 to 72	16 byte (see [ITU-T G.707])
	S11/12/2 (VC-11/12/2) (see Note)	J2	16 byte (see [ITU-T G.707])
	S11D/S12D/S2D (VC-11/12/2 TCM)	N2, bits 7 to 8, frame 9 to 72	16 byte (see [ITU-T G.707])
PDH with SDH frame	P4s/3s (140/34 Mbit/s)	TR	16 byte (see [ITU-T G.831] and [ITU-T G.832])
	P4sD/3sD (140/34 Mbit/s TCM)	NR, bits 7 to 8, frame 9 to 72	16 byte (see [ITU-T G.831] and [ITU-T G.832])

NOTE – In order to distinguish between unequipped and supervisory unequipped, the fixed code 00000000 in J1/J2 should not be used in the supervisory unequipped termination source function.

6.2.2.3 Mismatch defect (dMMG)

The mismatch defect is calculated at the layer where the MEG (maintenance entity group) and MEP (MEG end point) are defined. It monitors the connectivity in a maintenance entity group.

Its detection and clearance are defined in Figure 6-6. The <Defect> in Figure 6-6 is dMMG. The <Event> in Figure 6-6 is the Mismatch defect event as generated by the continuous check frame reception process and the Period is the Period carried in the frame that triggered the event, unless an earlier continuous check frame triggering an MMG event carried a greater period.



G.806(12)_F6-6

Figure 6-6 – Defect detection and clearance process

6.2.2.4 Unexpected MEP defect (dUNM)

The unexpected MEP (maintenance entity group (MEG) end point) defect is calculated at the layer where the MEG and MEP are defined. It monitors the connectivity in a maintenance entity group.

Its detection and clearance are defined in Figure 6-6. The <Defect> in Figure 6-6 is dUNM. The <Event> in Figure 6-6 is the Unexpected MEP event (as generated by the continuous check frame reception process) and the Period is the Period carried in the frame that triggered the event, unless an earlier continuous check frame triggering an Unexpected MEP event carried a greater period.

6.2.2.5 Degraded signal defect (dDEG)

See clause 6.2.3.1.

6.2.3 Signal quality supervision

6.2.3.1 Generic behaviour

Signal quality supervision, in general, monitors the performance of a trail. If the performance falls below a certain threshold, this might activate a defect. For the generic performance monitoring process see clause 8.3.

For networks where the network operator assumes a Poisson distribution of errors, an excessive error defect and a degraded signal defect are to be detected.

For networks where the operator assumes a bursty distribution of errors, a degraded signal defect is to be detected. The excessive error defect, for this case, is assumed to be false.

For networks where the network operator assumes bad second on frame calculation, only a degraded signal defect is to be detected.

The applicability of the two is in the province of the regional standards.

6.2.3.1.1 Excessive error (dEXC) and degraded signal defects (dDEG) assuming Poisson distribution of errors

Excessive error and degraded signal defects are to be detected according to the following process:

An excessive error defect (dEXC) shall be detected if the equivalent BER exceeds a preset threshold of 10^{-x} , $x = 3, 4$ or 5 . The excessive error defect shall be cleared if the equivalent BER is better than $10^{-(x+1)}$.

With $BER \geq 10^{-x}$ the probability of defect detection within the measuring time shall be ≥ 0.99 .

With $BER < 10^{-(x+1)}$ the probability of defect detection within the measuring time shall be $\leq 10^{-6}$.

With $BER \geq 10^{-x}$ the probability of defect clearing within the measuring time shall be $\leq 10^{-6}$.

With $BER < 10^{-(x+1)}$ the probability of defect clearing within the measuring time shall be ≥ 0.99 .

A degraded signal defect (dDEG) shall be detected if the equivalent BER exceeds a preset threshold of 10^{-x} , $x = 5, 6, 7, 8$ or 9 . The degraded signal defect shall be cleared if the equivalent BER is better than $10^{-(x+1)}$.

With $BER \geq 10^{-x}$ the probability of defect detection within the measuring time shall be ≥ 0.99 .

With $BER \leq 10^{-(x+1)}$ the probability of defect detection within the measuring time shall be $\leq 10^{-6}$.

With $BER \geq 10^{-x}$ the probability of defect clearing within the measuring time shall be $\leq 10^{-6}$.

With $BER \leq 10^{-(x+1)}$ the probability of defect clearing within the measuring time shall be ≥ 0.99 .

Maximum detection and clearing time requirements for the BER calculations for SDH are listed in Tables 6-4, 6-5 and 6-6. For all other signals, these values are for further study.

NOTE – The specification in the 01/94 revision of [ITU-T G.783] could have been interpreted as listed in Table 6-7.

dEXC and dDEG shall be cleared during SSF conditions. A new evaluation period for dEXC and dDEG shall start after SSF is cleared.

Table 6-4 – Maximum detection time requirements for multiplex section, VC-4-Xc, VC-4 and VC-3

Detector threshold	Actual BER						
	$\geq 10^{-3}$	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}
10^{-3} (see Note 1)	10 ms						
10^{-4} (see Note 2)	10 ms	100 ms					
10^{-5} (see Note 3)	10 ms	100 ms	1 s				
10^{-6} (see Note 4)	10 ms	100 ms	1 s	10 s			
10^{-7}	10 ms	100 ms	1 s	10 s	100 s		
10^{-8}	10 ms	100 ms	1 s	10 s	100 s	1 000 s	
10^{-9}	10 ms	100 ms	1 s	10 s	100 s	1 000 s	10 000 s

NOTE 1 – For VC-4, VC-4-4c, VC-4-16c, VC-4-64c, VC-4-256c, the BIP has saturated and the detection is not reliable (see Appendix VI for details).
 NOTE 2 – For VC-4-4c, VC-4-16c, VC-4-64c, VC-4-256c, the BIP has saturated and the detection is not reliable (see Appendix VI for details).
 NOTE 3 – For VC-4-64c, VC-4-256c, the BIP has saturated and the detection is not reliable (see Appendix VI for details).
 NOTE 4 – For VC-4-256c the BIP has saturated and the detection is not reliable (see Appendix VI for details).

Table 6-5 – Maximum detection time requirements for VC-2, VC-12 and VC-11

Detector threshold	Actual BER					
	$\geq 10^{-3}$	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}
10^{-3} (see Note)	40 ms					
10^{-4}	40 ms	400 ms				
10^{-5}	40 ms	400 ms	4 s			
10^{-6}	40 ms	400 ms	4 s	40 s		
10^{-7}	40 ms	400 ms	4 s	40 s	400 s	
10^{-8}	40 ms	400 ms	4 s	40 s	400 s	4 000 s

NOTE – For VC-2, the BIP has saturated and the detection is not reliable (see Appendix VI for details).

Table 6-6 – Clearing time requirements

Detector threshold	Set/clear values associated with detector threshold	STM-N multiplex section VC-4-Xc VC-4 VC-3	VC-2 VC-12 VC-11
10^{-3}	$10^{-3}/10^{-4}$	10 ms	40 ms
10^{-4}	$10^{-4}/10^{-5}$	100 ms	400 ms
10^{-5}	$10^{-5}/10^{-6}$	1 s	4 s
10^{-6}	$10^{-6}/10^{-7}$	10 s	40 s
10^{-7}	$10^{-7}/10^{-8}$	100 s	400 s
10^{-8}	$10^{-8}/10^{-9}$	1000 s	4000 s
10^{-9}	$10^{-9}/10^{-10}$	10 000 s	

NOTE – The values in this table for the clearing times are upper bounds. For STM-N and VC-4-Xc, it is possible to reduce the maximum clearing times in column 3 by a factor between 1 and N (for STM-N multiplex section) or between 1 and X (for VC-4-Xc), respectively (but note that clearing times below 10 ms are not recommended).

Table 6-7 – Alternative interpretation of maximum detection and clearing time requirements in the 1994 revision of Recommendation ITU-T G.783

Detector threshold	Multiplex section VC-4 VC-3	VC-2 VC-12 VC-11
10^{-3}	10 ms	40 ms
10^{-4}	100 ms	400 ms
10^{-5}	1 s	4 s
10^{-6}	10 s	40 s
10^{-7}	100 s	400 s
10^{-8}	1000 s	4000 s
10^{-9}	10 000 s	

6.2.3.1.2 Excessive error (dEXC) and degraded signal defects (dDEG) assuming bursty distribution of errors

The excessive error defect is not defined, and dEXC is assumed to be false.

The degraded signal defect (dDEG) shall be declared if DEGM consecutive bad intervals (interval is the 1-second period used for performance monitoring) are detected. An interval is declared bad if the percentage of detected errored blocks in that interval, or the number of errored blocks in that interval \geq degraded threshold (DEGTHR).

NOTE 1 – For the case of dDEG in the MSn layer, the errored block is equal to a BIP violation.

The degraded signal defect shall be cleared if DEGM consecutive good intervals are detected. An interval shall be declared good if the percentage of detected errored blocks in that interval, or the number of errored blocks in that interval $<$ DEGTHR.

The parameter DEGM shall be provisionable in the range 2 to 10.

The parameter DEGTHR shall be provisioned either as a percentage or as a number of errored blocks. When based on a percentage, it shall be in the range $0 < \text{DEGTHR} \leq 100\%$. When based on a number of errored blocks, it shall be in the range $0 < \text{DEGTHR} \leq \text{number of blocks in the interval}$.

NOTE 2 – When using percentage, for higher rate interfaces, 1% is equal to a large number of blocks. For example, in an STM-16 interface, 1% is equal to a step of 30720 blocks in the interval for the multiplex section.

dDEG shall be cleared during SSF conditions. A new evaluation period for dDEG shall start after SSF is cleared.

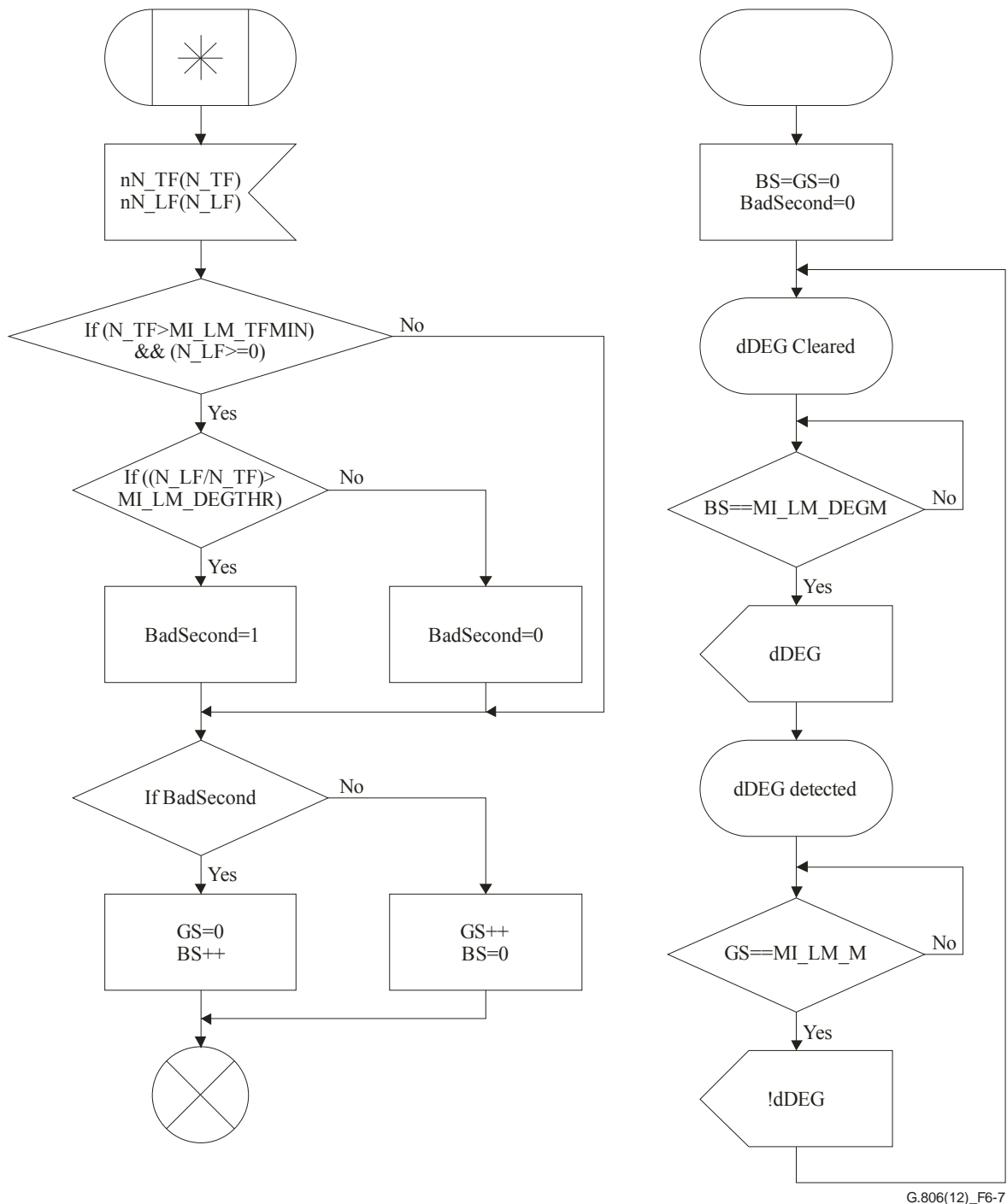
6.2.3.1.3 Degraded signal defects (dDEG) assuming bad second on frame calculation

The degraded signal defect is calculated by a frame loss measurement mechanism at a network layer. It monitors the connectivity of a trail.

Its detection and clearance are defined in Figure 6-7.

Every second the state machine receives the one-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 (MI_LM_TFMIN). Furthermore, if the frame loss ratio (lost frames/transmitted frames) is greater than MI_LM_DEGTHR, a bad second is declared.



G.806(12)_F6-7

Figure 6-7 – dDEG detection and clearance process

6.2.4 Payload type supervision

6.2.4.1 Generic behaviour

Payload type supervision checks that compatible adaptation functions are used at the source and the sink. This is normally done by adding a signal type identifier at the source adaptation function and comparing it with the expected identifier at the sink. If they do not match, a payload mismatch is detected.

For assignment of payload types, see Annex A.

6.2.4.2 Payload composition and payload mismatch defect (dPLM)

The signal label identifies the presence of a payload and the signal type carried in the payload.

Basic function source direction

The generation of the payload identifier in the signal label is required. The value is bound to and represents the selected (activated) adaptation function.

The payload identifier is inserted in the signal label overhead.

Basic function sink direction

The signal label overhead (TSL) is recovered from the AP.

The detection of dPLM is based on a comparison between the expected TSL, representing the selected/activated adaptation function, and the accepted TSL.

A new signal label code value shall be accepted if the signal label overhead carries the same code value in m consecutive (multi)frames with $3 \leq m \leq 10$.

The payload label mismatch defect (dPLM) shall be detected if the "accepted TSL" code does not match the "expected TSL" code. If the "accepted TSL" is "equipped non-specific", the mismatch is not detected.

In case of a PLM condition, the dPLM defect shall be cleared if the "accepted TSL" code matches the "expected SL" code or if the "accepted TSL" code is "equipped non-specific".

The dPLM shall be detected within a maximum period of 100 ms in the absence of bit errors.

The dPLM shall be cleared within a maximum period of 100 ms in the absence of bit errors.

dPLM shall be cleared during TSF conditions. A new evaluation period for dPLM shall start after TSF is cleared.

The value of the signal label passed to the management system should be an accepted value rather than the received value.

Details on the PLM defect are provided in Table 6-8.

NOTE – An "expected TSL" code of "equipped non-specific" is no longer applicable according to [ITU-T G.707].

Table 6-8 – PLM defect details

Hierarchy	Layer	Signal label overhead	Signal label values
SDH	S3/4 (VC-3/4) (see Note 1)	C2 byte	See [ITU-T G.707]
	S11/12/2 (VC-11/12/2) (see Note 1)	V5, bits 5 to 7 K4, bit 1 (see Note 2)	See [ITU-T G.707]
PDH with SDH frame	P4s/3s (140/34 Mbit/s)	MA, bits 3 to 5	See [ITU-T G.832]

NOTE 1 – In order to distinguish between unequipped and supervisory unequipped, the fixed code 00000000 in J1/J2 should not be used in the supervisory unequipped termination source function.
NOTE 2 – K4, bit 1, is used for an extended signal label in a multiframe manner. The signal label overhead is located in frames 12 to 19 of the multiframe (see [ITU-T G.707]). If the multiframe cannot be recovered, it results in a PLM defect.

6.2.4.3 GFP user payload mismatch defect (dUPM)

GFP user payload mismatch (dUPM) is raised when the accepted UPI (AcUPI, see clause 8.5.1.4) is different from the expected UPI. dUPM is cleared when AcUPI matches the expected UPI or GFP_SF is active.

6.2.4.4 GFP extension header mismatch defect (dEXM)

GFP extension header mismatch (dEXM) is raised when the accepted EXI (AcEXI, see clause 8.5.1.3) is different from the expected EXI. dEXM is cleared when AcEXI matches the expected EXI or GFP_SF is active.

6.2.5 Alignment supervision

6.2.5.1 Generic behaviour

Alignment supervision checks that the client layer frame and frame start can be correctly recovered. The specific processes depend on the signal/frame structure and may include:

- (multi)frame alignment;
- pointer processing;
- alignment of several independent frames to a common frame start in case of inverse multiplexing.

If one of these processes fails, a related loss of alignment defect (dLOA) shall be activated. The defect detection process shall be normally tolerant to single frame slips, but should detect for continuous frame slips.

NOTE – dLOA is the generic defect term. Specific defects are loss of frame (dLOF), loss of multiframe (dLOM) or loss of pointer (dLOP).

For generic alignment processes refer to clause 8.2. For the specific detection processes, refer to the specific equipment functional Recommendations ([ITU-T G.783] and [ITU-T G.705]) or those indicated below.

6.2.5.2 GFP loss of frame delineation defect (dLFD)

GFP loss of frame delineation (dLFD) is raised when the frame delineation process (clause 6.3.1 of [ITU-T G.7041]) is not in the "SYNC" state. dLFD is cleared when the frame delineation process is in the "SYNC" state.

6.2.5.3 Loss Of frame defect (dLOF)

A loss of frame (LOF) defect shall be declared if the frame alignment process is in the out-of-frame (OOF) state for a certain period. dLOF shall be cleared when the frame alignment process enters in the in-frame (IF) state and/or the state persists for a certain time.

In SDH and OTUk, dLOF shall be cleared when the IF state persists for 3 ms. (See [ITU-T G.783] and [ITU-T G.798].)

6.2.5.4 Loss Of multiframe defect (dLOM)

A loss of multiframe (LOM) defect shall be declared if the multiframe alignment process is in the out-of-frame (OOF) state for a certain period. dLOM shall be cleared when the multiframe alignment process is in the in-multiframe (IM) state.

6.2.6 Maintenance signal supervision

6.2.6.1 Generic behaviour

Maintenance signal supervision is concerned with the detection of maintenance indications in the signal. For the use and generation of maintenance signals see clause 6.3.

6.2.6.2 Alarm indication signal defect (dAIS)

An alarm indication signal (AIS) defect is calculated at the network layer. It monitors the presence of an AIS maintenance signal. For AIS generation see clause 6.3.1.

Basic function sink direction

If z consecutive frames contain the AIS activation pattern in the AIS overhead or the frame indicating AIS is received, an AIS defect shall be detected. The dAIS defect shall be cleared if z consecutive frames contain the AIS deactivation pattern in the AIS overhead or the frame indicating AIS is not received within a certain period.

Details on the AIS defect by activation/deactivation are provided in Table 6-9.

Table 6-9 – AIS defect details

Hierarchy	Layer	Type	AIS overhead	AIS activation pattern	AIS deactivation pattern	z (Note 1)
SDH	MSn	MS-AIS	K2, bits 6 to 8	"111"	≠ "111"	3
	S3/4 (VC-3/4)	AU-AIS	H1, H2	See Annex A of [ITU-T G.783]		
		VC-AIS (Notes 2, 3)	C2 byte	"11111111"	≠ "11111111"	5
	S3D/4D (VC-3/4 TCM)	IncAIS	N1, bits 1 to 4	"1110"	≠ "1110"	5
	S11/12/2 (VC-11/12/2)	TU-AIS	V1, V2	See Annex A of ITU-T G.783]		
		VC-AIS (Notes 2, 3)	V5, bits 5 to 7	"111"	≠ "111"	5
S11D/12D/2D (VC-11/12/2 TCM)	IncAIS	N2, bit 4	"1"	"0"	5	
PDH with SDH frame	P4s/3s (140/34 Mbit/s)	AIS	MA, bits 3 to 5	"111"	≠ "111"	5
	P4sD/3sD (140/34 Mbit/s TCM)	IncAIS	NR, bits 1 to 4	"1110"	≠ "1110"	5
PDH	P11s, P12s, P22e, P31e, P32e, P4e, P4a	AIS	See [ITU-T G.775]			

NOTE 1 – z is not configurable.

NOTE 2 – Equipment designed prior to this Recommendation may be able to perform VC-AIS detection either as specified above with "frames" being replaced by "samples (not necessarily frames)", or by a comparison of the accepted signal label with the all-ones pattern. If the accepted signal label is not equal to all-ones, the VC-AIS defect is cleared.

NOTE 3 – In networks that do not support/allow the transport of VC-n/VC-m signals with tandem connection overhead, the VC-AIS defect is not defined and the VC-AIS defect is assumed to be false.

6.2.6.2.1 dAIS for CBR client signals (generic AIS)

For the CBR dAIS detection, the reverse PN-11 process is applied to the data signal as shown in Figure 6-8. At the output of this process (OUT) an all-zeroes pattern will occur if the input data (IN) is the PN-11 Generic AIS sequence. Note that an all-zeroes output pattern will also occur in case of an all-zeroes input pattern. Both the output (OUT) and input (IN) signals are constantly checked over an 8192-bit interval for the number of non-zero bits (= one bits). If the number of one bits per interval at OUT is less than 256 and the number of one bits per interval at IN is above or equal to 256 in 3 consecutive intervals, dAIS is raised. If the number of one bits at OUT is above or equal to 256, or the number of one bits at IN is below 256 in 3 consecutive intervals, dAIS is cleared.

NOTE – Generic AIS forwarded to SDH interfaces will lead to LOF in OSn/RSn_A_Sk functions not capable of detecting this AIS signal. In the case where an SDH input interface is connected to an STM-N output signal of a network-element terminating the OTN transport where this AIS signal is inserted, a dLOF defect could be interpreted as an AIS indication.

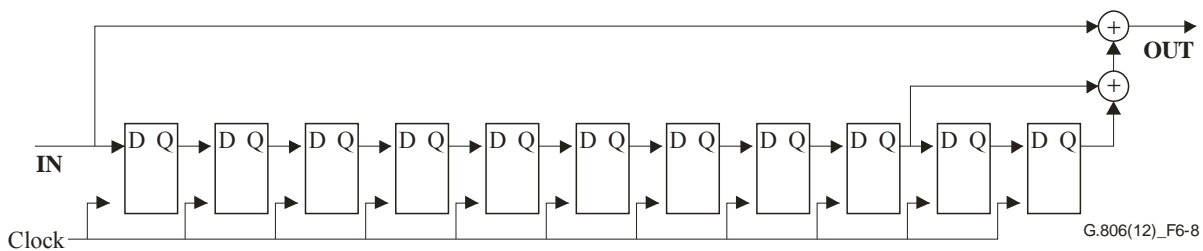


Figure 6-8 – Inverse PN-11 process for generic AIS detection

Table 6-10 – Generic AIS defect details

Hierarchy	Layer	Type
SDH	STM-N	STM-AIS

6.2.6.3 Remote/outgoing defect indication defect (dRDI/ODI)

Basic function source direction

The generation of RDI/ODI is required for bidirectional trail/flow termination functions. For RDI/ODI generation, see clause 6.3.2. The value inserted is the value received via RI_RDI/ODI from the associated basic sink function. The RDI/ODI value is inserted into the RDI/ODI overhead or the frame indicating RDI/ODI.

NOTE – For unidirectional trail termination functions not being paired with a termination sink function, the RDI/ODI signal output should be inactive, but can be undefined in old equipment not explicitly supporting unidirectional transport.

Basic function sink direction

The RDI/ODI overhead or the frame is recovered from the CP.

If z consecutive frames contain the RDI/ODI activation pattern in the RDI/ODI overhead or the frame indicating RDI/ODI is received, a dRDI/ODI defect shall be detected. The dRDI/ODI defect shall be cleared if z consecutive frames contain the RDI/ODI deactivation pattern in the RDI/ODI overhead or the frame indicating RDI/ODI is not received within a certain period.

dRDI/ODI shall be cleared during SSF conditions. A new evaluation period for dRDI/ODI shall start after SSF is cleared.

Details on the RDI/ODI defects by activation/deactivation pattern are provided in Table 6-11.

Table 6-11 – RDI/ODI defect details

Hierarchy	Layer	Type	RDI/ODI overhead	RDI/ODI activation pattern	RDI/ODI deactivation pattern	z (Note 1)
SDH	MSn	RDI	K2, bits 6 to 8	"110"	≠ "110"	3 to 5
	S3/4 (VC-3/4) (Note 2)	RDI	G1, bit 5	"1"	"0"	3, 5 or 10
	S3D/4D (VC-3/4 TCM option 2)	RDI	N1, bit 8, frame 73	"1"	"0"	5
		ODI	N1, bit 7, frame 74	"1"	"0"	5
	S11/12/2 (VC-11/12/2)	RDI	V5, bit 8	"1"	"0"	3, 5 or 10
	S11D/12D/2D (VC-11/12/2 TCM)	RDI	N2, bit 8, frame 73	"1"	"0"	5
		ODI	N2, bit 7, frame 74	"1"	"0"	5
PDH with SDH frame	P4s/3s (140/34 Mbit/s)	RDI	MA, bit 1	"1"	"0"	5
	P4sD/3sD (140/34 Mbit/s TCM)	RDI	NR, bit 8, frame 73	"1"	"0"	5
		ODI	NR, bit 7, frame 74	"1"	"0"	5
PDH	P12s	RDI	See [ITU-T G.775]			
	P22e,31e,4e	RDI	See [ITU-T G.775]			
	P32e	RDI	X	"11"	"00"	1
NOTE 1 – z is not configurable.						
NOTE 2 – Enhanced RDI processing is for further study.						

6.2.6.4 Client signal fail defect (dCSF)

Client signal fail (dCSF) is raised when a signal that identifies client signal fail is received. dCSF defines the following defects:

- dCSF-LOS (client loss of signal)
- dCSF-FDI (client forward defect indication)
- dCSF-RDI (client reverse defect indication)

dCSF is raised when a frame indicating dCSF is received. dCSF is cleared when

- no such frame is received in a certain period; or
- a frame indicating CSF defect clear indication is received.

6.2.6.4.1 GFP client signal fail defect

GFP client signal fail (dCSF) is raised when a GFP frame with correct tHEC, with aPTI = "100" and a valid and supported UPI code is received. dCSF is cleared when:

- no such GFP client management frame is received in $N \times 1000$ ms (a value of 3 is suggested for N); or

- a valid GFP client data frame is received; or
- a GFP[cmf] with UPI = DCI is received.

6.2.6.4.2 OPU client signal fail defect

dCSF shall be declared if the CSF bit in the OPU_k PSI overhead is "1" for X consecutive 256 frame multi-frames. dCSF shall be cleared if the CSF bit is "0" for consecutive 256 frame multi-frames. X shall be 3.

6.2.6.5 Locked defect (dLCK)

The locked defect is calculated at the network layer. It monitors the presence of a locked maintenance signal.

If z consecutive frames contain the LCK activation pattern in the LCK overhead or a frame indicating LCK is received, an AIS defect shall be detected. The dLCK defect shall be cleared if z consecutive frames contain the LCK deactivation pattern in the LCK overhead or a frame indicating LCK is not received within a certain period.

6.2.7 Protocol supervision

6.2.7.1 Generic behaviour

Protocol supervision detects failures in the sequence of a protocol exchange.

6.2.7.2 Failure of protocol defect (dFOP)

The dFOP defect indicates a failure in the automatic protection switching protocol. The detailed behaviour is defined at the specific atomic functions.

6.2.7.2.1 Failure of protocol provisioning mismatch defect (dFOP-PM)

The failure of protocol provisioning mismatch defect is calculated at the network layer. It monitors provisioning mismatch of the transmitted and received APS protocols.

6.2.7.2.2 Failure of protocol no response defect (dFOP-NR)

The failure of protocol no response defect is calculated at the network layer. It monitors incompleteness of protection switching by comparing the transmitted "Requested Signal" values and the received "Requested Signal" in the APS protocol.

6.2.7.2.3 Failure of protocol configuration mismatch defect (dFOP-CM)

The failure of protocol configuration mismatch defect is calculated at the network layer. It monitors working and protection configuration mismatch by detecting the reception of an APS protocol from the working transport entity.

6.2.7.2.4 Failure of protocol timeout defect (dFOP-TO)

The failure of protocol timeout defect is calculated at the network layer. It monitors the timeout defect by detecting a prolonged absence of expected APS protocol information.

6.2.7.3 Unexpected periodicity defect (dUNP)

The unexpected periodicity defect is calculated at the network layer where the MEG and MEP are defined. It detects the configuration of different periods of continuity check frames at different MEPs belonging to the same MEG.

Its detection and clearance are defined in Figure 6-6. The <Defect> in Figure 6-6 is dUNP. The <Event> in Figure 6-6 is the unexpPeriod event as generated by the continuity check frame reception process and the period is the period carried in the continuity check frame that triggered the event, unless an earlier continuity check frame triggering an unexpPeriod event carried a greater period.

6.2.7.4 Unexpected priority defect (dUNPr)

The unexpected priority defect is calculated at the network layer. It detects the configuration of different priorities continuity check frames at different MEPs belonging to the same MEG.

Its detection and clearance are defined in Figure 6-6. The <Defect> in Figure 6-6 is dUNPr. The <Event> in Figure 6-6 is the unexpPriority event as generated by the continuity check frame reception process and the period is the period carried in the continuity check frame that triggered the event, unless an earlier continuity check frame triggering an unexpPriority event carried a greater period.

6.3 Consequent actions

This clause presents in generic terms the generation and control of the set of consequent actions. Specific details are presented in each atomic function.

After a defect or anomaly is detected, one or more of the following consequent actions may be requested:

- all-ones (AIS) insertion
- RDI insertion
- REI insertion
- ODI insertion
- OEI insertion
- unequipped signal insertion
- generation of "server signal fail (SSF)" signal
- generation of "trail signal fail (TSF)" signal
- generation of "trail signal degrade (TSD)" signal.

Figure 6-9 shows how the aAIS, aRDI and aREI consequent action request signals control the associated consequent actions: insertion of all-ones, insertion of RDI code and insertion of REI value. Figure 6-9 also shows the location of aSSF, aTSF and aTSD consequent action requests.

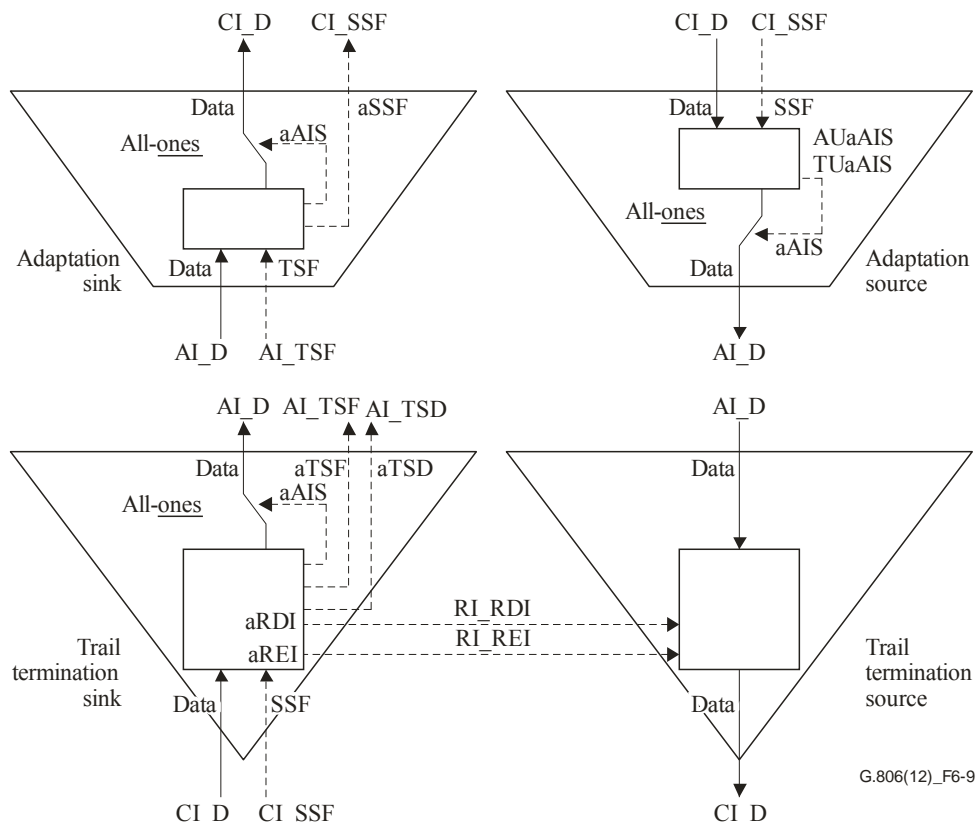


Figure 6-9 – Consequent action control: AIS, RDI and REI

Certain detected near-end defects cause the insertion of the all-ones signal in trail termination sink functions. Detected defects cause the insertion of the all-ones signal in adaptation sink functions. The reception of a server signal fail (SSF) indication causes the insertion of all-ones in the adaptation source.

In cases where the all-ones signal is inserted, either in a trail termination sink or in the previous adaptation sink function, the RDI code is inserted in the associated trail termination source signal. That is, the RDI code is inserted on detected defects or on the reception of a SSF indication in a trail termination sink function (aRDI).

Every frame, the number of detected EDC violations (aREI) in the trail termination sink function are inserted in the REI bits in the associated trail termination source signal.

A connection function inserts the unequipped VC signal at one of its outputs if that output is not connected to one of its inputs.

6.3.1 Alarm indication signal (AIS)

The all-ones (AIS) signal replaces the received signal under certain detected near-end defect conditions in order to prevent downstream failures being declared and alarms being raised. See Appendix III for a description of the application and the insertion control.

Specific details with respect to all-ones (AIS) insertion are defined in the individual atomic functions. Generically, the logic equations and the time requirement for the all-ones (aAIS) insertion request are:

Adaptation sink functions: $aAIS \leftarrow dPLM \text{ or } dAIS/AI_TSF \text{ or } dLOA$

NOTE 1 – dLOA represents either dLOF or dLOM or dLOP, whichever is applicable in the atomic function.

NOTE 2 – Certain adaptation sink functions do not detect dAIS. To ensure that the adaptation sink function is aware of the reception of the all-ones signal, the termination sink function (which inserted the all-ones signal on detected defect conditions) informs the adaptation sink about this condition by means of the AI_TSF signal. In such a case, the dAIS term in the aAIS expression is replaced by AI_TSF.

NOTE 3 – In case of a 45 Mbit/s interface, the AIS signal is defined in [ITU-T M.20] and [ITU-T G.704].

Termination sink functions: $aAIS \leftarrow dAIS$ or $dUNEQ/dLOS$ or $(dTIM$ and not $TIM AISdis)$

Some national networks allow to enable/disable AIS/TSF activation on detection of dTIM, while others always activate AIS/TSF on detection of dTIM. In the latter case, TIMAISdis is always false and not configurable via the management interface.

NOTE 4 – The term dAIS is applicable for the MS_TT function. The term dLOS is applicable for physical section layer termination functions while dUNEQ represents a similar condition for the (SDH) path layers.

Adaptation source functions: $aAIS \leftarrow CI_SSF$

The termination sink, and adaptation sink and source functions shall insert the all-ones (AIS) signal within two (multi)frames after AIS request generation (aAIS) and cease the insertion within two (multi)frames after the AIS request has cleared.

6.3.1.1 Generic AIS generation and detection

Generic AIS including OTUk AIS and STM-AIS is a PN-11 pseudo-random pattern as defined in [ITU-T G.709]. The pattern is generated by a pseudo-random generator. For the detection of generic AIS, the reverse process as shown in Figure 6-10 is used. As the flip-flops of the detector circuit are fed with the same data as the flip-flops of the generator circuit, data at point D1 are the same as data at G1 with a delay of 11 clock cycles. As the G1 data appear at the output of the generator (G_{out}) and as such also at the input of the detector (D_{in}) with a delay of 11 clock cycles, D1 and D_{in} data are the same for each clock cycle. A PN-11 generic AIS pattern at the input of the detector (D_{in}) should therefore result in an all-zeroes pattern at point D2. The only other input pattern that will result in an all-zeroes pattern at D2 is an all-zeroes input pattern.

The detection of an all-zeroes pattern at D2 and a non-all-zeroes pattern at D_{in} are criteria for the generic AIS defect. For the specific detection process, see clause 6.2.6.2.1.

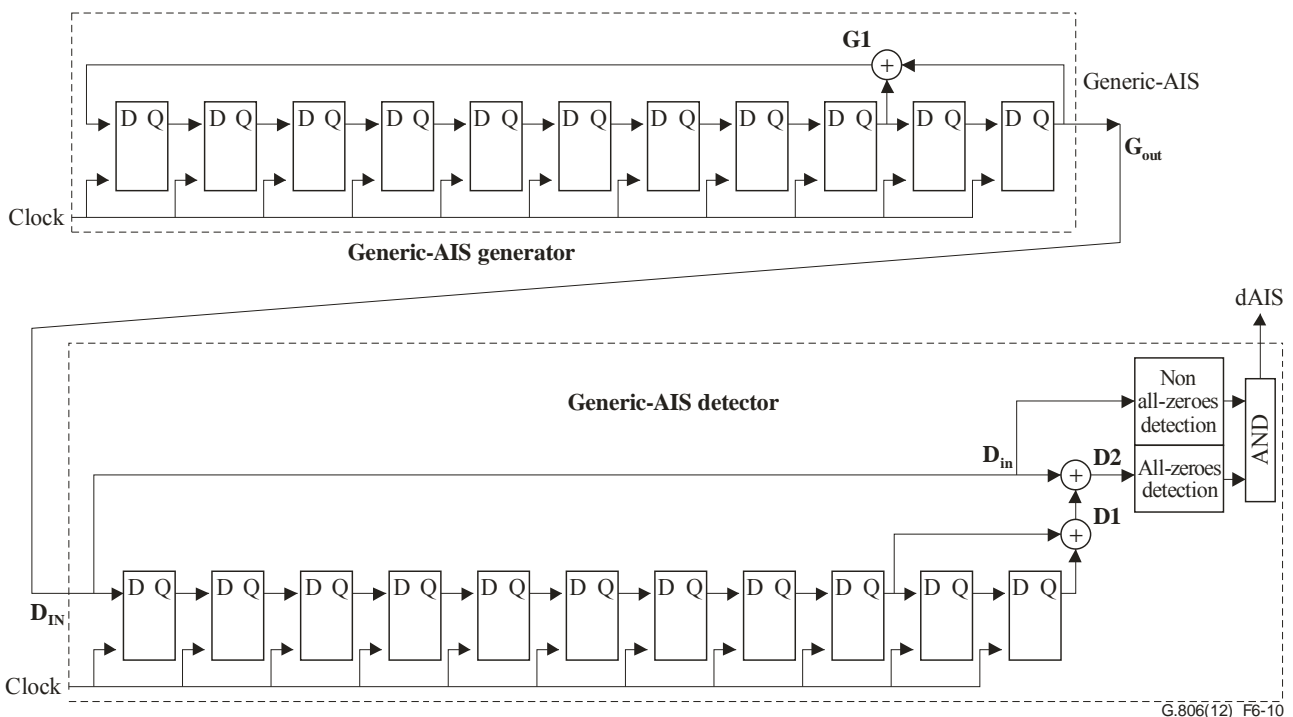


Figure 6-10 – Generic AIS generation and detection

6.3.2 Remote defect indication (RDI)

If the all-ones signal is inserted, either in a trail termination sink or in the previous adaptation sink function, the RDI code is inserted in the associated trail termination source signal. See Appendix II for a description of the RDI application and the insertion control.

Specific details with respect to RDI insertion are defined in the individual atomic functions. Generically, the logic equations and the time requirement for the RDI insertion is:

Termination sink functions: $aRDI \leftarrow dAIS/CI_SSF \text{ or } dUNEQ \text{ or } dTIM$

Supervisory termination sink functions: $aRDI \leftarrow CI_SSF \text{ or } dTIM$

NOTE 1 – Some trail termination functions do not detect dAIS. To ensure that the trail termination function is aware of the reception of the all-ones signal, the server layer (which inserted the all-ones signal on detected defect conditions) informs the client layer about this condition by means of the CI_SSF signal. In such a case, the dAIS term in the aRDI expression is replaced by CI_SSF.

NOTE 2 – In the case of supervisory-unequipped termination functions, dUNEQ cannot be used to activate aRDI; an expected supervisory-unequipped VC signal will have the signal label set to all-zeroes, causing a continuous detection of dUNEQ. If an unequipped VC signal is received, dTIM will be activated and can serve as a trigger for aRDI instead of dUNEQ.

Upon the declaration/clearing of aRDI at the termination sink function, the trail termination source function shall have inserted/removed the RDI code within the following time limits:

- MSn_TT: 1 ms.
- S4_TT, S3_TT, S4s_TT, S3s_TT: 1 ms.
- S2_TT, S12_TT, S11_TT, S2s_TT, S12s_TT, S11s_TT: 4 ms.
- S4D_TT, S3D_TT: 20 ms.
- S2D_TT, S12D_TT, S11D_TT: 80 ms.

NOTE 3 – RDI is undefined and should be ignored by the receiver (TT_Sk) in the case of a unidirectional trail.

6.3.3 Remote error indication (REI)

At every frame, the number of detected EDC violations in the trail termination sink function is inserted in the REI bits in the signal generated by the associated trail termination. See Appendix II for a description of the REI application and the insertion control.

Specific details with respect to REI insertion are defined in the individual atomic functions. Generically, the logic equations and the time requirement for the REI insertion is:

Termination sink function: $aREI \leftarrow \text{"number of error detection code violations"}$

Upon the detection of a number of errors at the termination sink function, the trail termination source function shall have inserted that value in the REI bit(s) within the following time limits:

- MSn_TT: 1 ms.
- S4_TT, S3_TT, S4s_TT, S3s_TT: 1 ms.
- S2_TT, S12_TT, S11_TT, S2s_TT, S12s_TT, S11s_TT: 4 ms.
- S4D_TT, S3D_TT: 20 ms.
- S2D_TT, S12D_TT, S11D_TT: 80 ms.

NOTE – REI is undefined and should be ignored by the receiver (TT_Sk) in the case of a unidirectional trail.

6.3.4 Server signal fail (SSF)

SSF signals are used to forward the defect condition of the server to the client in the next (sub)layer, to:

- prevent defect detection in layers without incoming AIS detectors in trail termination sink functions (e.g., S4_TT, S12_TT);
- report the server signal fail condition in layers without incoming AIS detectors in trail termination sink functions;
- control the link connection AIS (e.g., AU-AIS) insertion in adaptation source functions;
- initiate protection switching/restoration in the (protection-)connection function.

Specific details with respect to SSF generation are defined in the individual atomic functions. Generically, the logic equations and the time requirement for the SSF generation is:

Adaptation sink function: $aSSF \leftarrow dPDM \text{ or } dAIS/AI_TSF \text{ or } dLOA$

NOTE 1 – In case the adaptation function does not detect the AIS defect, the dAIS term will be replaced by AI_TSF generated by the previous TT_Sk.

NOTE 2 – The term dLOA is the general indication for dLOF, dLOM or dLOP, whichever is applicable.

Upon the declaration of aSSF, the function shall activate CI_SSF (CI_SSF = true) and deactivate CI_SSF (CI_SSF = false) after the SSF request has cleared.

6.3.5 Trail signal fail (TSF)

TSF signals are used to forward the defect condition of the trail to the:

- adaptation sink function, to control all-ones (AIS) insertion in the function, when the function does not perform AIS defect detection; e.g., in S12/P12x_A_Sk.

Specific details with respect to TSF generation are defined in the individual atomic functions. Generically, the logic equations and the time requirement for the TSF generation is:

Termination sink function: $aTSF \leftarrow dAIS/CI_SSF \text{ or } dUNEQ/dLOS \text{ or } (dTIM \text{ and not } TIMAISdis)$

Supervisory termination sink function: $aTSF \leftarrow CI_SSF \text{ or } (dTIM \text{ and not } TIMAISdis)$

Some national networks allow to enable/disable AIS/TSF activation on detection of dTIM, while others always activate AIS/TSF on detection of dTIM. In the latter case, TIMAISdis is always false and not configurable via the management interface.

NOTE 1 – Some trail termination functions do not detect dAIS. To ensure that the trail termination function is aware of the reception of the all-ones signal, the server layer (which inserted the all-ones signal on detected defect conditions) informs the client layer about this condition by means of the SSF signal. In such a case, the dAIS term, in the aTSF expression, is replaced by CI_SSF.

NOTE 2 – In the case of supervisory-unequipped termination functions, dUNEQ cannot be used to activate aTSF; an expected supervisory-unequipped VC signal will have the signal label set to all-zeroes, causing a continuous detection of dUNEQ. If an unequipped VC signal is received, dTIM will be activated and can serve as a trigger for aTSF instead of dUNEQ.

Upon the declaration of aTSF, the function shall activate AI_TSF (AI_TSF = true), and deactivate AI_TSF (AI_TSF = false) after the TSF request has cleared.

6.3.6 Trail signal fail protection (TSFprot)

TSFprot signals are used to forward the defect condition of the trail to the:

- protection connection function in the trail protection sublayer, to initiate trail protection switching in that function;
- connection function in the same layer which performs a non-intrusively monitored SNC (SNC/N) protection scheme, to initiate SNC protection switching in that function.

Specific details with respect to TSFprot generation are defined in the individual atomic functions. Generically, the logic equations and the time requirement for the TSF generation is:

Termination sink function: $aTSFprot \leftarrow aTSF \text{ or } dEXC$

NOTE – aTSFprot and aTSF will be identical for network elements that support error defects assuming bursty distribution of errors. For such networks, dEXC is assumed to be permanently false (see clause 6.2.3.1.2).

Upon the declaration of aTSFprot, the function shall activate AI_TSFprot (AI_TSFprot = true) and deactivate AI_TSFprot (AI_TSFprot = false) after the TSFprot request has cleared.

6.3.7 Trail signal degrade (TSD)

TSD signals are used to forward the signal degrade defect condition of the trail to the:

- protection connection function in the trail protection sublayer, to initiate trail protection switching in that function;
- connection function in the layer to initiate subnetwork connection protection switching in that function for the case of a non-intrusive monitored SNC (SNC/N) protection scheme.

Specific details with respect to TSD generation are defined in the individual atomic functions. Generically, the logic equations and the time requirement for the TSD generation is:

Termination sink function: $aTSD \leftarrow dDEG$

Upon the declaration of aTSD, the function shall activate AI_TSD (AI_TSD = true) and deactivate AI_TSD (AI_TSD = false) after the TSD request has cleared.

6.3.8 Outgoing defect indication (ODI)

Specific details with respect to ODI insertion are defined in the individual atomic functions. Generically, the logic equations and the time requirement for the ODI insertion is:

Termination sink functions: $aODI \leftarrow CI_SSF \text{ or } dUNEQ \text{ or } dTIM \text{ or } dIncAIS \text{ or } dLTC$

Upon the declaration/clearing of aODI at the termination sink function, the trail termination source function shall have inserted/removed the ODI code within the following time limits:

- S4D_TT, S3D_TT: 20 ms.
- S2D_TT, S12D_TT, S11D_TT: 80 ms.

NOTE – ODI is undefined and should be ignored by the receiver (TT_Sk) in the case of a unidirectional TC trail.

6.3.9 Outgoing error indication (OEI)

Every frame, the number of detected EDC violations in the VC signal in the TC trail termination sink function is inserted in the OEI bit in the signal generated by the associated TC trail termination.

Specific details with respect to OEI insertion are defined in the individual atomic functions. Generically, the logic equations and the time requirement for the OEI insertion is:

TC termination sink function: $aOEI \leftarrow \text{"number of error detection code violations in the VC"}$

Upon the detection of a number of errors at the termination sink function, the trail termination source function shall have inserted that value in the OEI bit within the following time limits:

- S4D_TT, S3D_TT: 20 ms.
- S2D_TT, S12D_TT, S11D_TT: 80 ms.

NOTE – OEI is undefined and should be ignored by the receiver (TT_Sk) in the case of a unidirectional TC trail.

6.3.10 Unequipped signal

Unequipped indicating signals are generated by connection functions.

If the output of a connection function is not connected to an input of that connection function, the CI originates at that connection function. In this case, an unequipped CI shall be generated by the connection function.

6.4 Defect correlations

This clause presents, in generic terms, the defect correlations within trail termination, adaptation and connection functions. Specific details are presented in each atomic function. See clause 5.9 for a description of the applied specification technique.

Since all of the defects will appear at the input of the defect correlation filter (Figures 6-1 and 6-2), it provides correlation to reduce the amount of information offered to the EMF.

A fault may cause multiple defect detectors to be activated. To determine, from the activated defects, which fault is present, the activated defects are correlated to obtain the fault cause.

The cZZZ fault causes (correlated defects) shall be activated if the expression is true. cZZZ shall be deactivated if the expression is false.

6.4.1 Termination sink functions

<i>Trail termination sink:</i>	cUNEQ	←	dUNEQ and MON
<i>Supervisory trail termination sink:</i>	cUNEQ	←	dUNEQ and dTIM and (AcTI = all-zeroes) and MON
<i>Trail termination sink:</i>	cTIM	←	dTIM and (not dUNEQ) and MON
<i>Supervisory trail termination sink:</i>	cTIM	←	dTIM and not (dUNEQ and AcTI = all-zeroes) and MON
	cDEG	←	dDEG and (not dTIM) and MON
cRDI	←	dRDI and (not dUNEQ/LTC) and (not dTIM) and RDI_Reported and MON	
cODI	←	dODI and (not dUNEQ/LTC) and (not dTIM) and ODI_Reported and MON	
cSSF	←	CI_SSF/dAIS and MON and SSF_Reported	
cLOS	←	dLOS and MON	
cAIS	←	dAIS and AIS_Reported and MON	

The reporting of the following defects is provisionable: AIS, SSF, RDI, ODI. These defects are "secondary defects" in that they are the result of a consequent action on a "primary defect" in another network element.

Example: A single STM-16 LOS defect (dLOS) may cause a few thousand AIS defects (e.g., AU4dAIS s, TU12dAIS s) to be detected in the network and about one thousand RDI defects (e.g., MS16dRDI, VC4dRDI s, VC12dRDI s).

It shall, therefore, be provisionable to report AIS, SSF, RDI or ODI as a fault cause. This is controlled by means of the parameters AIS_Reported, SSF_Reported, RDI_Reported and ODI_Reported, respectively. The default for these parameters is "false".

NOTE 1 – dUNEQ, dTIM, dDEG, dEXC, dPLM and dRDI/ODI are cleared during a SSF/TSF condition.

NOTE 2 – In the MS_TT function, defects of the server layer are detected by dAIS from the K2 byte and not through SSF.

NOTE 3 – By default, AIS, as such, is not reported. Instead, trail terminations shall report (as an option) that the server (layer) failed to pass the signal (server signal fail) if they receive the all-ones (AIS) signal. This reduces the declaration of "AIS failures" to one failure (SSF) at the trail termination NE. No failures are generated at intermediate nodes in the (long) trail.

NOTE 4 – Refer to clause 6.1 for a MON description.

NOTE 5 – The detection of an unequipped VC signal is possible in a termination supervisory sink function despite both the supervisory-unequipped VC signal and the unequipped VC signal having signal label code "0". A trace identifier mismatch will be detected with the accepted trace identifier being all-zeroes. This combination is the signature of the reception of an unequipped VC.

6.4.2 Adaptation sink function

cPLM ← dPLM and (not AI_TSF)

cAIS ← dAIS and (not AI_TSF) and (not dPLM) and AIS_Reported

cLOA ← dLOA and (not dAIS) and (not dPLM)

It shall be provisionable to report AIS as a fault cause. This is controlled by means of the parameter AIS_Reported. The default shall be AIS_Reported = false.

NOTE 1 – dLOA represents dLOF, dLOP or dLOM, whichever is applicable.

NOTE 2 – The specification of the pointer interpreter algorithm is such that either dAIS or dLOP can be declared, not both at the same time. Refer to Annex A of [ITU-T G.783].

NOTE 3 – dPLM is cleared during a TSF condition.

6.4.3 Connection function

cFOP ← dFOP and (not CI_SSF)

6.5 One-second performance monitoring filters

The one-second filters perform a simple integration of reported anomalies and defects by counting during a one-second interval. At the end of each one-second interval the contents of the counters is made available to the performance monitoring processes within the EMF for further processing ([see ITU-T G.7710]). Generically, the following (superset of) counter outputs will be provided:

- near-end/far-end errored or lost block counts
- near-end/far-end transmitted block counts
- near-end/far-end defect seconds
- pointer justification counts (see [ITU-T G.783]).

This clause presents in generic terms the performance monitoring primitive generation within atomic functions. Specific details are presented in each atomic function.

NOTE – Near-end/far-end processing includes also outgoing near-end/far-end processing.

6.5.1 Near-end errored block count (pN_EBC) and near-end lost block count (PN_LBC)

Every second, the number of errored near-end blocks (N_Bs) within that second is counted as the near-end error block count (pN_EBC) for the circuit layer. For the packet layer, the number of lost near-end blocks within that second is counted as the near-end lost block count (pN_LBC).

The errored near-end blocks (N_Bs) are defined in Table 6-12.

Table 6-12 – Near-end errored blocks definition

Hierarchy	Layer	Errored block definition
SDH	RS1	One or more errors in the STM-1 frame detected by BIP-8
	RSn (n ≥ 4)	For further study
	MS1/4/16/64	Number of errors in the STM-n frame detected by BIP-24×n
	MSn (n ≥ 256)	For further study
	S4/3	One or more errors in the VC frame detected by BIP-8 (see Note 2)
	S2/12/11	One or more errors in the VC frame detected by BIP-2 (see Note 2)
	S4D/3D	One or more errors in the VC frame detected by IEC
	S2D/12D/11D	One or more errors in the VC frame detected by BIP-2
PDH with SDH frame	S4T/3T	One or more errors in the VC frame detected by IEC
	P4s/3s	One or more errors in the frame
PDH	P12s	One or more errors in the frame detected by CRC-4 or one or more errors detected in the frame alignment word
	P4e/31e/32e/22e	One or more errors detected in the frame alignment word
NOTE 1 – For error detection refer to clause 8.3 and the specific equipment functional Recommendations ([ITU-T G.783] and [ITU-T G.705]).		
NOTE 2 – For backward compatibility, the specification is as follows: every second, the number of errors is counted and "translated" into the pN_EBC according to Annex C of [ITU-T G.826].		

6.5.2 Near-end defect second (pN_DS)

Every second with at least one occurrence of aTSF (e.g., CI_SSF, dAIS, dTIM, dUNEQ) or dEQ shall be indicated as a near-end defect second (pN_DS).

$pN_DS \leftarrow aTSF \text{ or } dEQ$

6.5.3 Far-end errored block count (pF_EBC) and far-end lost block count (pF_LBC)

Every second, the number of errored far-end blocks (F_Bs) within that second is counted as the far-end error block count (pF_EBC) for the circuit layer. For the packet layer, the number of lost near-end blocks within that second is counted as the far-end lost block count (pF_LBC).

The errored far-end blocks (F_Bs) are defined in Table 6-13.

Table 6-13 – Far-end errored blocks definition

Hierarchy	Layer	Errored block definition
SDH	MS1/4/16	Number of errors indicated by REI in the STM-n frame
	MSn (n ≥ 64)	For further study
	S4/3/2/12/11	One or more errors indicated by REI in the VC frame (see Note 1)
	S4D/3D/2D/12D/11D	One or more errors indicated by REI in the VC frame
	S4T/3T	One or more errors indicated by REI in the VC frame
PDH with SDH frame	P4s/3s	One or more errors indicated by REI in the VC frame
PDH	P12s (see Note 2)	One or more errors indicated by REI in the VC frame
NOTE 1 – For backward compatibility, the specification is as follows: every second, the number of errors is counted and "translated" into the pF_EBC according to Annex C of [ITU-T G.826].		
NOTE 2 – REI and far-end errored blocks are only supported if a CRC EDC is used.		

6.5.4 Far-end defect second (pF_DS)

Every second with at least one occurrence of dRDI shall be indicated as a far-end defect second (pF_DS).

pF_DS ← dRDI

6.5.5 Near-end transmitted block count (pN_TBC)

Every second, the number of transmitted near-end blocks (N_TBs) within that second is counted as the near-end transmitted block count (pN_TBC).

6.5.6 Far-end transmitted block count (pF_TBC)

Every second, the number of transmitted far-end blocks (F_TBs) within that second is counted as the far-end transmitted block count (pF_TBC).

7 Information flow (XXX_MI) across the XXX_MP reference points

Table 7-1 summarizes the generic (superset) of configuration, provisioning and reporting information (MI) that is passed across the XXX_MP reference points for the three types of atomic functions. The information listed under input ("Set") in this table refers to configuration and provisioning data that is passed from the EMF to the other functional blocks. The information listed under output ("Get") refers to (autonomous) status reports to the EMF from the atomic functions.

NOTE – The configuration, provisioning and reporting information for a specific atomic function is listed in the I/O table in the atomic function description itself.

As an example, we may consider the SDH higher order path trace. The SDH higher order path termination sink function may be provisioned for the HO path trace for what it should expect by a "MI_ExTI" command received from the manager. If the HO path trace that is received does not match the expected HO path trace, this will give rise to a report of a mismatch of the HO path trace across the Sn_TT_MP reference point (MI_cTIM). Having received this mismatch indication, the relevant managed object may then decide to request a report of the HO path trace ID that has been received by a "MI_AcTI" report.

Table 7-1 – Generic command, configuration, provisioning and reporting information flow over the XXX_MP reference points

Management point	Process within atomic function	Input ("Set")	Output ("Get")	
TT_So_MP	Trace identifier	Transmitted trail trace identifier (MI_TxTI) value		
TT_Sk_MP	Termination point/port mode	Termination point mode control (MI_TPmode: MON, <u>NMON</u>) Port mode control (MI_Portmode: MON, (<u>AUTO</u>), <u>NMON</u>)		
	Continuity supervision		Signal loss fault cause (MI_cLOS, MI_cUNEQ, MI_cLTC)	
	Connectivity supervision	Expected trail trace identifier (MI_ExTI) value Misconnected traffic defect detection control (MI_TIMdis: <u>true</u> , <u>false</u>) Enable/disable AIS insertion on dTIM detection (MI_TIMAISdis: <u>true</u> , <u>false</u>)	Accepted (received) trail trace identifier value (MI_AcTI) Misconnected traffic fault cause (MI_cTIM)	
	Signal quality supervision	Poisson-based excessive defect threshold selection (MI_EXC_X: <u>10⁻³</u> , <u>10⁻⁴</u> , <u>10⁻⁵</u>) Poisson-based degraded defect threshold selection (MI_DEG_X: <u>10⁻⁵</u> , <u>10⁻⁶</u> , <u>10⁻⁷</u> , <u>10⁻⁸</u> , <u>10⁻⁹</u>)		Poisson-based excessive errors fault cause (MI_cEXC) Poisson-based degraded errors fault cause (MI_cDEG)
		Burst-based degraded defect interval threshold selection (MI_DEGTHR: 0.. <u>(30)</u> ..100% or 0...N) Burst-based degraded defect monitor period selection (MI_DEGM: 2.. <u>10</u>)		Burst-based degraded errors fault cause (MI_cDEG)
	Maintenance signals processing	AIS fault cause reporting control (MI_AIS_Reported: true, <u>false</u>)		AIS fault cause (MI_cAIS, MI_cIncAIS)
		SSF fault cause reporting control (MI_SSF_Reported: true, <u>false</u>)		SSF fault cause (MI_cSSF)
		RDI fault cause reporting control (MI_RDI_Reported: true, <u>false</u>)		RDI fault cause (MI_cRDI)
		ODI fault cause reporting control (MI_ODI_Reported: true, <u>false</u>)		ODI fault cause (MI_cODI)
	Performance monitoring	1 second period indications (MI_1second)		Performance monitoring primitives (MI_pN_EBC, MI_pN_DS, MI_pF_EBC, MI_pF_DS, ...)
A_So_MP	Selection	Payload composition selection (MI_Active: true, <u>false</u>)		
	Performance monitoring		Performance monitoring justification actions (MI_pPJC+, MI_pPJC-)	

Table 7-1 – Generic command, configuration, provisioning and reporting information flow over the XXX_MP reference points

Management point	Process within atomic function	Input ("Set")	Output ("Get")
A_Sk_MP	Selection	Payload composition selection (MI_Active: true, <u>false</u>)	
	Maintenance signal processing	AIS fault cause reporting control (MI_AIS_Reported: true, <u>false</u>)	AIS fault cause (MI_cAIS)
	Payload type supervision		Accepted (received) payload type value (MI_AcSL) Miscomposed traffic fault cause (MI_cPLM)
	Alignment supervision		Alignment loss fault cause (MI_cLOF, MI_cLOM, MI_cLOP)
C_MP	Connection management	Matrix connection selection	
	Protection	Protection group selection (set of connection points, protection architecture: 1+1/1:n/m:n, switching type: uni/bidirectional, operation type: revertive/non-revertive, APS usage: true/false, extra traffic: true/false) External switch commands (MI_ExtCmd: LO, FS, MS, EXER, CLR) External control command (LOW) Hold off time value (MI_HOtime) WaitToRestore value (MI_WTRtime: 0.. <u>5</u> ..12 minutes)	Protocol fault cause (MI_cFOP) Protection status (for further study)
NOTE – Underlined values are suggested defaults.			

8 Generic processes

8.1 Line coding and scrambling processes

For the transmission of a digital signal via a physical media, special conditioning of the signal is required in order to:

- have sufficient signal changes for clock recovery
- avoid a DC level for the transmission.

Line coding or scrambling can be used for this task. Refer to the specific equipment functional Recommendations ([ITU-T G.783] and [ITU-T G.705]) for details.

8.2 Alignment processes

Alignment processes:

- recover the (multi)frame start of a client signal within the server signal
- recover the (multi)frame start of overhead information
- realign individual signals to a common frame phase.

For (multi)frame start recovery, two different processes can be used, frame alignment signal processing and pointer processing.

In case of frame alignment signal processing, a distinct bit pattern (the frame alignment signal (FAS)) is part of the frame that has to be recovered as shown in Figure 8-1. The FAS indicates a position within the frame, normally the frame start. Note that the pattern might be distributed over the frame. The FAS is inserted at the source. The sink searches for the FAS pattern and recovers the frame start based on it. If frame alignment cannot be established, this is indicated by the out of frame (OOF) condition. If frame alignment is established, this is indicated by the in frame (IF) condition. Based on these conditions, a loss of alignment defect (LOA) is generated. For details, refer to the specific equipment functional Recommendations ([ITU-T G.783] and [ITU-T G.705]).

NOTE – In case of multiframe alignment, the terms out of multiframe (OOM) and in multiframe (IM) might be used.

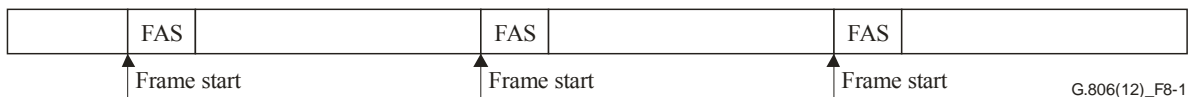


Figure 8-1 – Frame alignment signal

In case of pointer processing, the position of the client layer frame start within the server layer frame is indicated by a position indicator (the pointer) which is part of the server layer overhead as shown in Figure 8-2. The source generates the pointer based on the position of the client signal within the server frame. The sink recovers the pointer and identifies the client frame start, based on the pointer. If the pointer cannot be recovered correctly, a loss of pointer defect (LOP) shall be declared. For details, refer to the specific equipment functional Recommendation ([ITU-T G.783]).

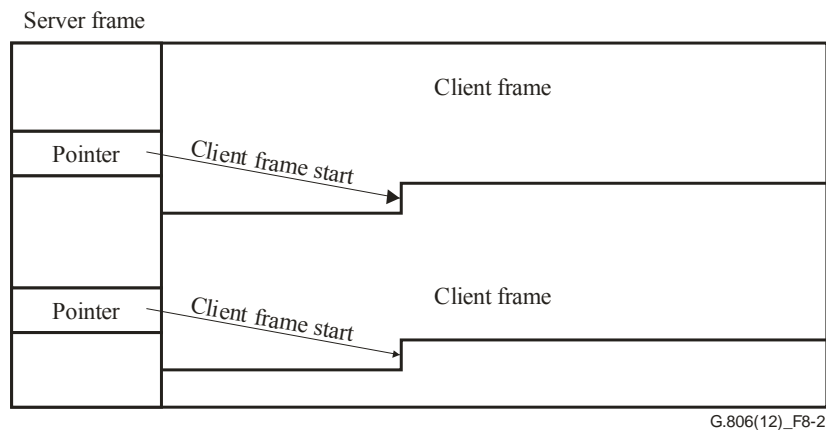


Figure 8-2 – Pointer

For other specific alignment processes, refer to the specific equipment functional Recommendations ([ITU-T G.783] and [ITU-T G.705]).

8.3 Performance supervision process

The performance supervision process monitors the quality of the trail between the source and the sink. For a digital signal, the process will provide information on bit errors and depends on some kind of error detection code (EDC). Different kinds of supervision processes are possible.

Figure 8-3 shows a pattern-based signal quality supervision. A known pattern (e.g., framing pattern) is inserted at the source. The sink extracts this pattern and compares it with the expected one. Any difference between the expected and received pattern is an indication of errors. Note that this kind

of error monitoring detects only errors in the supervised pattern and not in the whole signal. It is assumed that the rest of the signal is affected by errors in the same way as the supervised pattern.

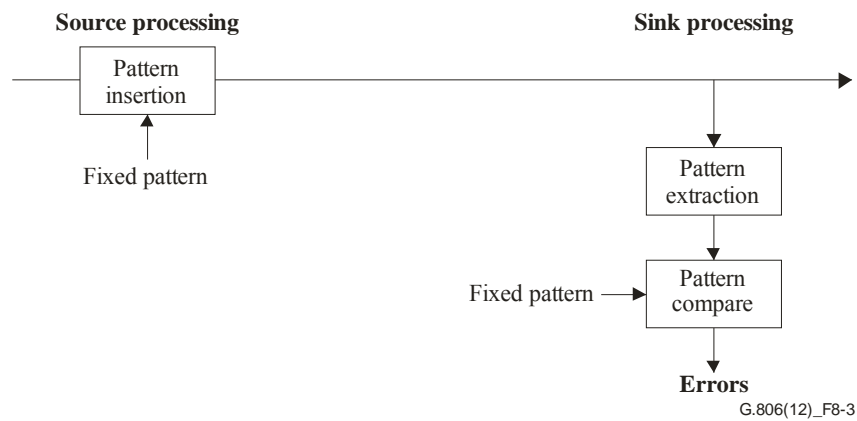


Figure 8-3 – Pattern-based signal quality supervision

Figure 8-4 shows a signature-based signal quality supervision. The signature is calculated over the signal or parts of the signal at the source and inserted into the signal. At the sink, the signature is calculated again and compared with the received signature. Any difference between the calculated and received signature indicates an error. Popular signatures are cyclic redundancy check (CRC) and bit interleaved parity (BIP). Note that the signature itself might be part of the next signature calculation as shown by the dotted lines in Figure 8-4. The signature is calculated over the signal frame and transmitted in the following frame as shown in Figure 8-5. Which part of the frame is included into the calculation depends on the specific layer network.

Refer to [ITU-T G.707] for a definition of BIP-N.

Refer to [ITU-T G.704] for a definition of CRC-4.

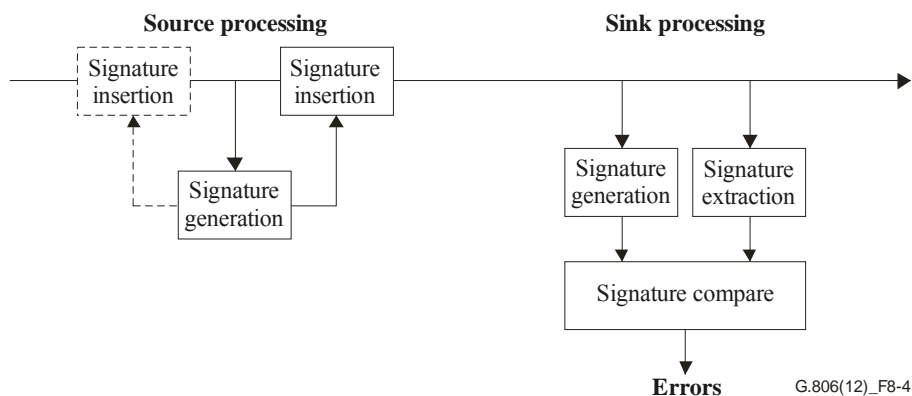


Figure 8-4 – Signature-based signal quality supervision

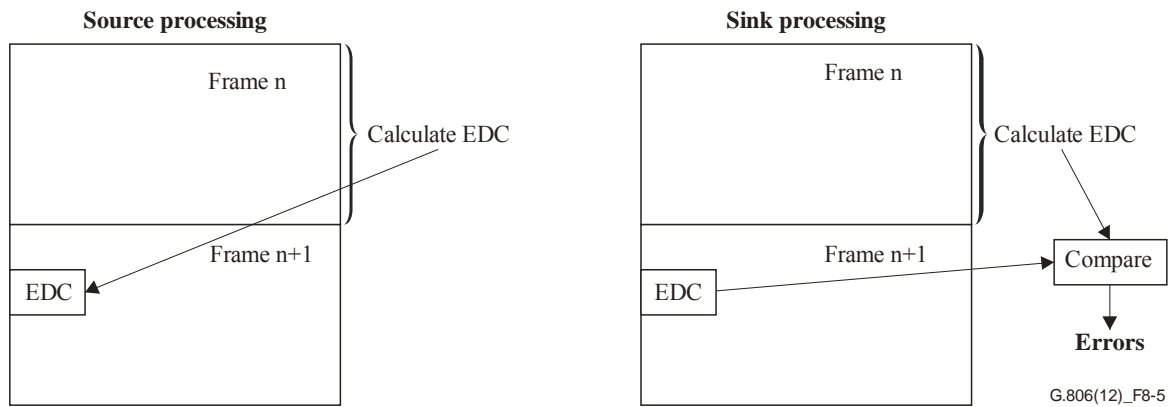


Figure 8-5 – Signature-based signal quality supervision example

If an EDC already exists in the signal (e.g., sublayer supervision) and it can differentiate between different amounts of errors, it can be used for the error supervision as shown in Figure 8-6. At the source, the errors are calculated based on the existing EDC. The result is the incoming error count (IEC) which is sent to the sink. At the sink, the errors are again calculated based on the existing EDC and compared with the received IEC. Any difference between local errors and received IEC indicates errors between source and sink. Figure 8-7 shows an example for an IEC-based signal quality supervision with a BIP EDC. As this kind of supervision depends on an incoming EDC, the behaviour for the case that this incoming EDC is missing has to be carefully defined.

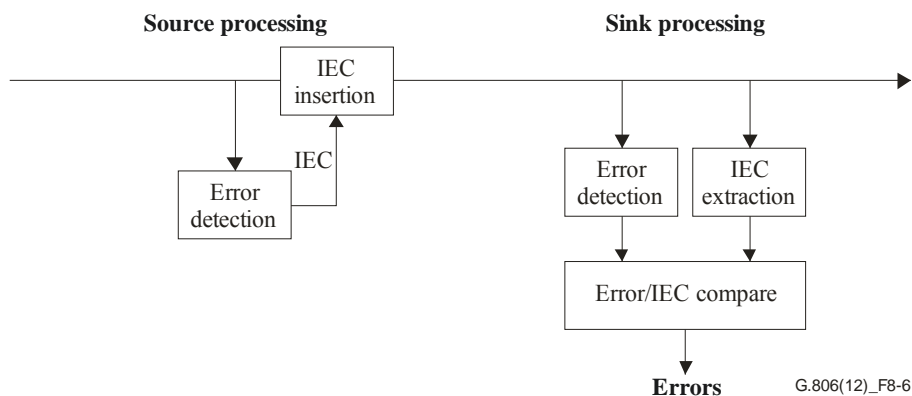


Figure 8-6 – IEC-based signal quality supervision

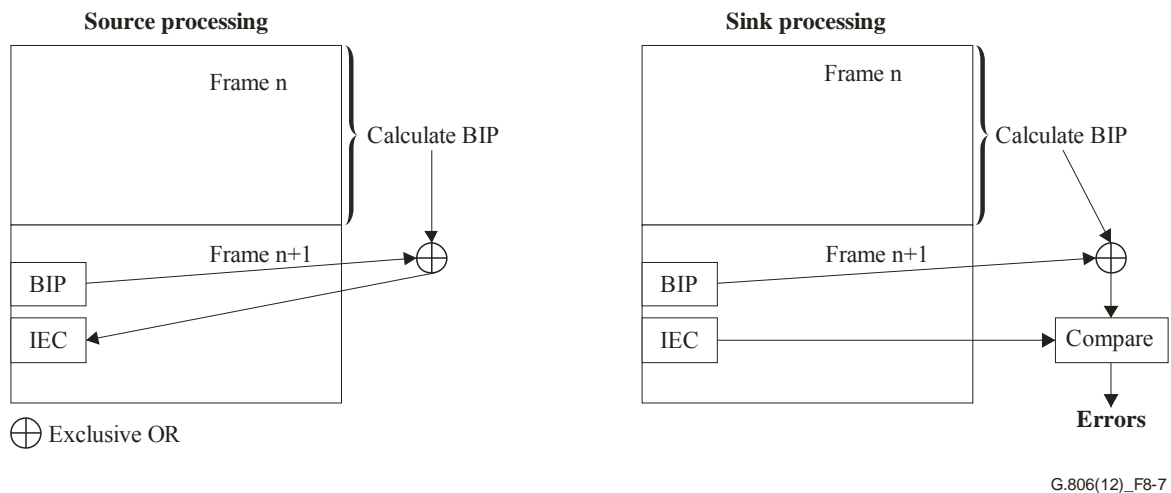


Figure 8-7 – IEC-based signal quality supervision example

8.4 BIP correction

In some cases, the overhead of the signal is overwritten along the trail (e.g., sublayer supervision). If this overhead is part of the EDC signature calculation, the signature has to be corrected accordingly in order to avoid the detection of errors at the sink. For a BIP type signature, the correction can be performed as shown in Figure 8-8. The BIP is calculated before and after the overhead insertion. Both results and the related incoming BIP overhead (which is usually transported in the following frame) are combined via an exclusive OR and form the new BIP overhead for the outgoing signal. The related processes are shown in Figure 8-9.

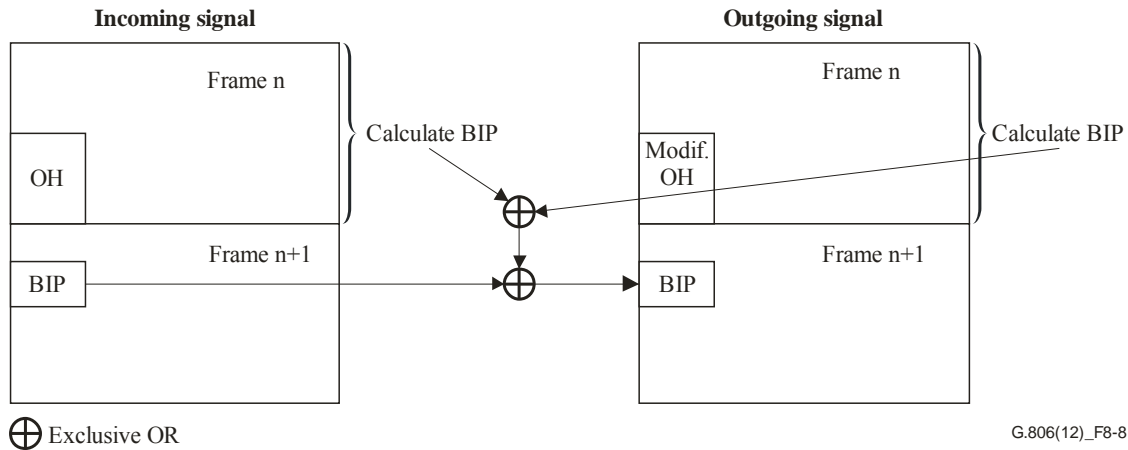


Figure 8-8 – BIP correction; functionality

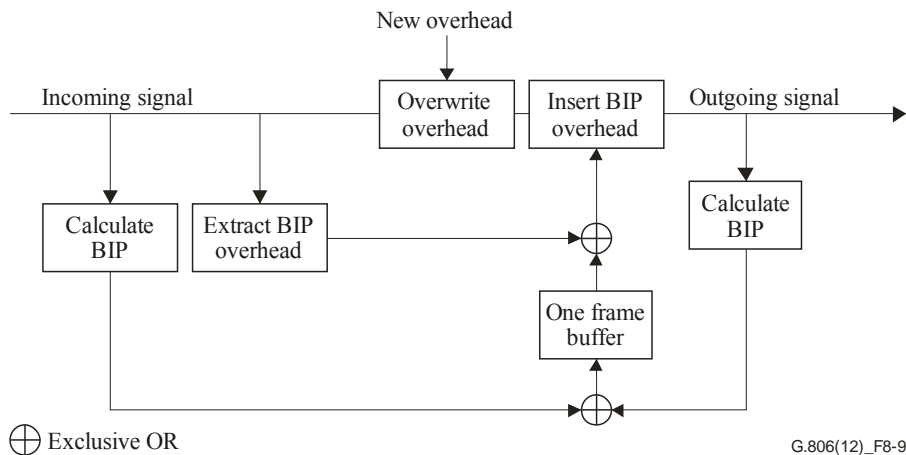


Figure 8-9 – BIP correction; processes

8.5 GFP processes

The generic framing procedure (GFP) provides a generic mechanism to adapt data client signals into SDH and OTN path layer networks as defined in [ITU-T G.7041]. The GFP processes are part of SDH VC or OTN ODU server layer to data client layer adaptations. The adaptation process can be separated into 3 generic blocks as shown in Figure 8-10: the server layer-specific processes with a GFP part, the common GFP processes and one or more instances of client-specific processes with GFP parts. In this clause, only the GFP-related functionality of the adaptation functions is defined. The adaptation functions themselves are described in technology-specific equipment Recommendations.

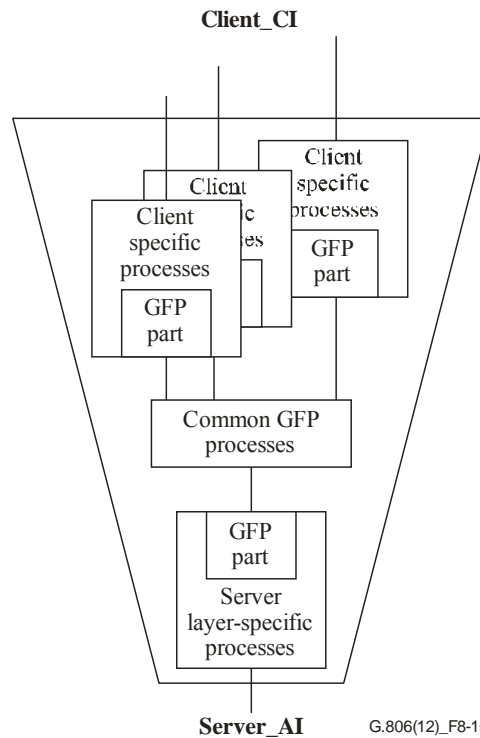


Figure 8-10 – GFP-based server/client adaptation generic blocks

8.5.1 Acceptance processes

8.5.1.1 PTI acceptance process

A PTI value is accepted (AcPTI) when a GFP frame with correct tHEC is received. AcPTI is set to the value of the PTI field of the type header of this frame.

8.5.1.2 PFI acceptance process

A PFI value is accepted (AcPFI) when a GFP frame with correct tHEC is received. AcPFI is set to the value of the PFI field of the type header of this frame.

8.5.1.3 EXI acceptance process

An EXI value is accepted (AcEXI) when a GFP frame with correct tHEC is received. AcEXI is set to the value of the EXI field of the type header of this frame.

8.5.1.4 UPI acceptance process

A UPI value is accepted (AcUPI) when a GFP frame with correct tHEC is received. AcUPI is set to the value of the UPI field of the type header of this frame.

8.5.1.5 CID acceptance process

A CID value is accepted (AcCID) when a GFP frame with linear extension header and correct eHEC is received. AcCID is set to the value of the CID field of the linear extension header of this frame.

8.5.2 Server layer-specific GFP processes

8.5.2.1 Server layer-specific GFP source processes

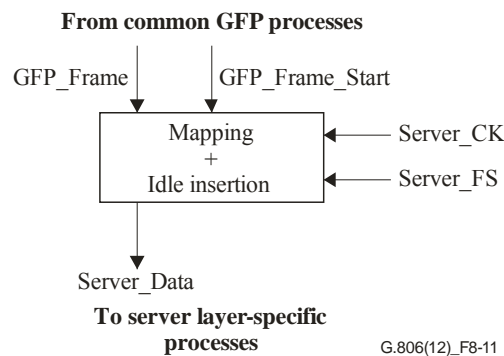


Figure 8-11 – Server layer-specific GFP source processes

Figure 8-11 shows the server layer-specific GFP source processes. The input to the processes is the GFP frame (GFP_Frame) together with the GFP frame start (GFP_Frame_Start) that indicates the availability of a new GFP frame. In addition, the server layer clock (Server_CK) and frame start (Server_FS) define the server layer frame.

The basic functionality is described below. Server layer-specific deviations or extensions to the processes might be defined in the adaptation functions of the technology-specific equipment Recommendations.

Mapping and idle insertion: The mapping process maps a GFP frame, if available, into the payload area of the server frame (Server_Data). If no GFP frame is available, a GFP Idle frame as defined in clause 6.2.1 of [ITU-T G.7041] is inserted. If the GFP frame rate exceeds the server payload capacity, GFP frames are discarded. An octet mapping is performed.

NOTE – In normal operation, it should not happen that the GFP frame rate exceeds the server layer payload capacity. The client layer or client layer-specific processing should implement the appropriate measures (e.g., traffic policing, traffic shaping) to prevent this.

Defects: None.

Consequent actions: None.

Defect correlations: None.

Performance monitoring: None.

8.5.2.2 Server layer-specific GFP sink processes

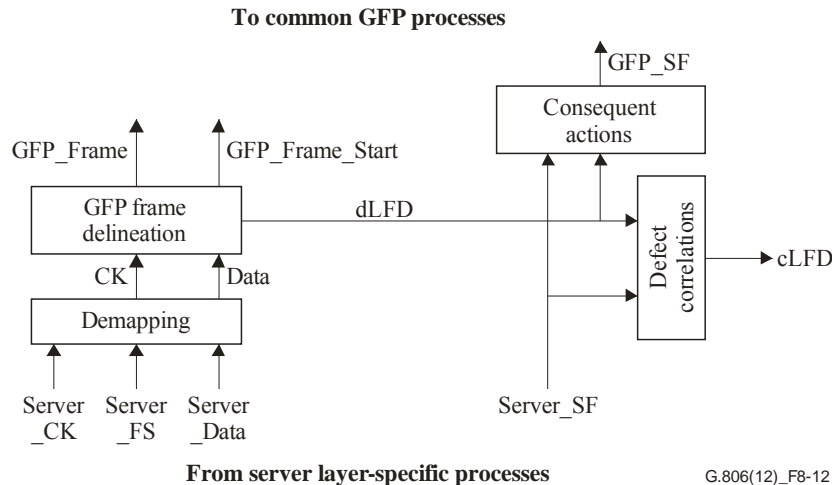


Figure 8-12 – Server layer-specific GFP sink processes

Figure 8-12 shows the server layer-specific GFP sink processes. The input to the processes is the server layer data (*Server_Data*), clock (*Server_CK*) and frame start (*Server_FS*).

The basic functionality is described below. Server layer-specific deviations or extensions to the processes might be defined in the adaptation functions of the technology-specific equipment Recommendations.

Demapping: The demapping process extracts the GFP data from the payload area of the server layer frame (*Server_Data*). The server layer frame is defined by the server layer clock (*Server_CK*) and frame start (*Server_FS*). An octet demapping is performed.

Frame delineation: GFP frame delineation is performed as defined in clause 6.3.1 of [ITU-T G.7041]. Frame delineation is assumed to be achieved when the process is in the "SYNC" state. Frame delineation is assumed to be lost when the process is not in the "SYNC" state. Idle GFP frames participate in the delineation process and are then discarded.

In the "HUNT" state searching for a correctly formatted core header includes the core header descrambling (clause 6.1.1.3 of [ITU-T G.7041]). In the "PRESYNC" and "SNYC" state, the core header descrambler is applied to the assumed core header positions.

Defects:

The function shall detect for the dLFD defect.

dLFD: see clause 6.2.5.2.

Consequent actions:

The function shall perform the following consequent actions:

$aGFP_SF \leftarrow Server_SF \text{ or } dLFD$

NOTE – *Server_SF* is the combination of *Server_AI_TSF* and server layer-specific adaptation defects (e.g., dPLM).

Defect correlations:

The function shall perform the following defect correlations to determine the most probable fault cause. This fault cause shall be reported to the EMF.

$cLFD \leftarrow dLFD \text{ and (not } Server_SF)$

Performance monitoring:

The generation of PM data in case of a LFD defect is for further study.

8.5.3 Common GFP processes

8.5.3.1 Common GFP source processes

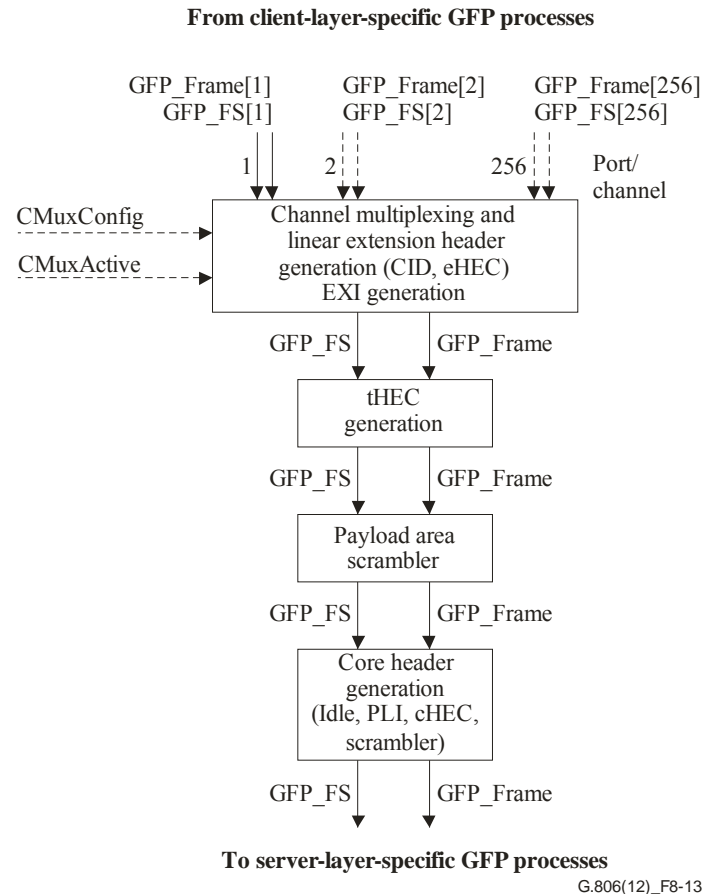


Figure 8-13 – Common GFP source processes

Figure 8-13 shows the common GFP source processes. The processes are performed on a frame-per-frame basis.

Channel multiplexing, linear extension header generation, EXI generation: In the case where the GFP channel multiplexing is supported and active ($CMuxActive = true$), the frames from up to 256 channels are extended with the linear extension header and multiplexed together on a frame-per-frame basis. The choice of scheduling algorithms is outside the scope of this Recommendation. The CID field of the linear extension header (see clause 6.1.2.1.3.2.1 of [ITU-T G.7041]) is set to (channel 1), where channel number corresponds to the port at which the frame is received. The spare field is set to all-zeroes and the eHEC is generated as defined in clause 6.1.2.1.4 of [ITU-T G.7041]. The EXI field of the type header is set to "0001" as defined in clause 6.1.2.1.1.3 of [ITU-T G.7041]. The number of supported channels is implementation specific. It might be fixed or configurable ($CMuxConfig$).

In the case where GFP channel multiplexing is not supported or not active ($CMuxActive = false$), the GFP frames from a single channel (channel 1) are forwarded. No extension header is added and the EXI field of the type header is set to "0000" as defined in clause 6.1.2.1.1.3 of [ITU-T G.7041].

Support of GFP channel multiplexing is optional.

tHEC generation: The tHEC of the payload header is generated as defined in clause 6.1.2.1.2 of [ITU-T G.7041].

Payload area scrambler: The GFP payload area is scrambled as defined in clause 6.1.2.3 of [ITU-T G.7041].

Core header generation: The length of the GFP payload area is calculated in octets and the value is inserted in the PLI field of the core header as defined in clause 6.1.1.1 of [ITU-T G.7041]. The cHEC for the core header is generated as defined in clause 6.1.1.2 of [ITU-T G.7041] and the core header is scrambled as defined in clause 6.1.1.3 of [ITU-T G.7041]. If the length of the GFP payload area exceeds 65535 octets, the frame is dropped.

NOTE – In normal operation, the length of the GFP payload area should not exceed 65535 octets. The client layers should implement the appropriate measures (e.g., adjusting the maximum transmission unit, MTU) to prevent this.

Defects: None.

Consequent actions: None.

Defect correlations: None.

Performance monitoring: None.

8.5.3.2 Common GFP sink processes

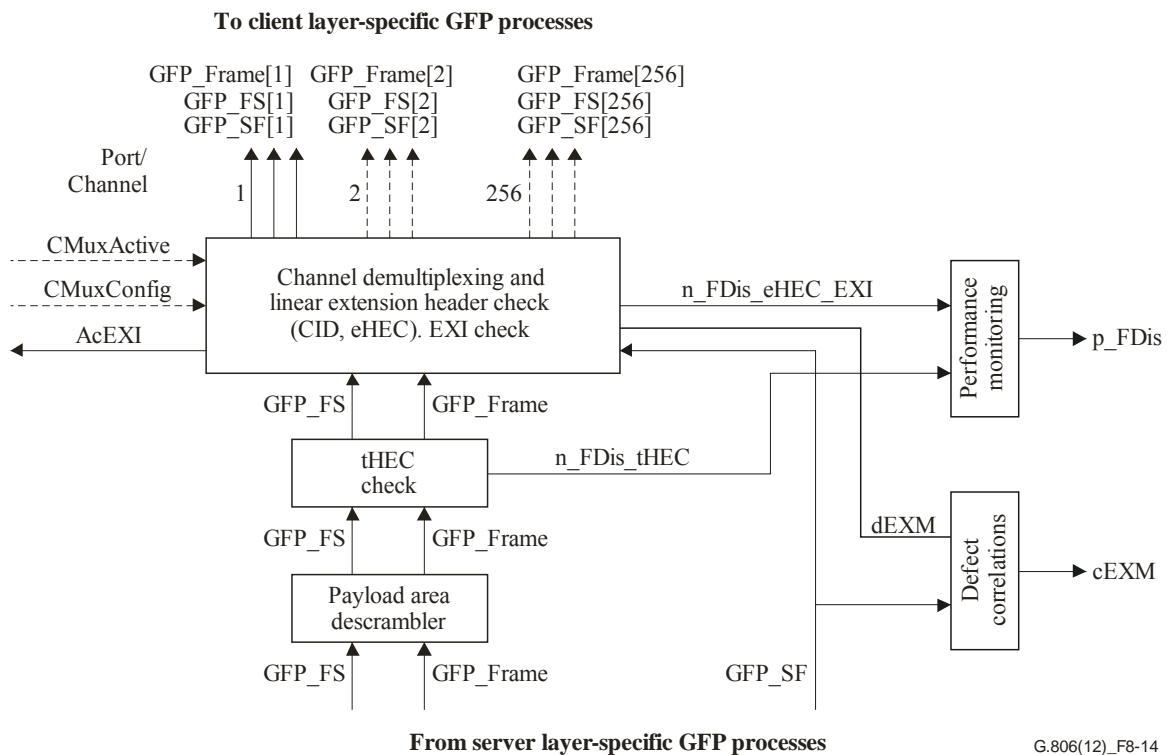


Figure 8-14 – Common GFP sink processes

Figure 8-14 shows the common GFP sink processes. The processes are performed on a frame-per-frame basis.

Payload area descrambler: The GFP payload area is descrambled as defined in clause 6.1.2.3 of [ITU-T G.7041].

tHEC check: The tHEC is checked as defined in clause 6.1.2.1.2 of [ITU-T G.7041]. Single bit error correction on all the fields protected by the tHEC (type field) shall be performed. In case of multiple errors, the frame is discarded and it is indicated by n_FDis_tHEC.

Channel demultiplexing, linear extension header check, EXI check: In the case where the GFP channel multiplexing is supported and active ($CMuxActive = true$), the accepted EXI ($AcEXI$, see clause 8.5.1.3) is compared with the value "0001". If it has a different value, the frame is discarded. Otherwise, the eHEC of the linear extension header is checked, as defined in clause 6.1.2.1.4 of [ITU-T G.7041]. Single bit error correction on the extension header may be performed. In case of multiple errors, or a single error when error correction is not used, the frame is discarded. All discarded frames are indicated by $n_FDis_eHEC_EXI$.

The frames are demultiplexed according to the value of the accepted CID value ($AcCID$) of the linear extension header. The frame is assigned to channel number ($AcCID+1$) where channel number corresponds to the port at which the frame is transmitted. Frames with channel numbers that are not active are discarded. The number of active channels is implementation specific. It might be fixed or configurable ($CMuxConfig$). The generation of a defect in case of frames with unexpected channel numbers is for further study.

The spare field of the linear extension header is ignored.

In case GFP channel multiplexing is not supported or not active ($CMuxActive = false$), the accepted EXI ($AcEXI$) is compared with the value "0000". If it has a different value, the frame is discarded. All discarded frames are indicated by $n_FDis_eHEC_EXI$.

The accepted EXI ($AcEXI$) is reported to the management ($AcEXI$).

Support of GFP channel multiplexing is optional.

Defects:

The function shall detect for dEXM defect.

dEXM: see clause 6.2.4.4.

NOTE 1 – The generation of a defect in the case of frames with unexpected channel numbers is for further study.

Consequent actions:

The function shall perform the following consequent actions:

$aGFP_SF[x] \leftarrow (aGFP_SF \text{ and } (\text{channel } x \text{ active based on ChannelMuxConfig})) \text{ or dEXM}$

Defect correlations:

$cEXM \leftarrow dEXM \text{ and } (\text{not GFP_SF})$

NOTE 2 – A defect for unexpected channel numbers is for further study.

Performance monitoring:

The function shall perform the following performance monitoring primitives processing. The performance monitoring primitives shall be reported to the EMF.

$p_FDis \leftarrow \sum (n_FDis_tHEC + n_FDis_eHEC_EXI)$

8.5.4 Client-specific GFP processes

The client-specific GFP processes perform the mapping between client data and the GFP frame. The basic processing is different for frame-mapped GFP (GFP-F) and transparent mapping of 8B/10B clients into GFP (GFP-T). Further details will be provided in the definitions of the client-specific adaptation functions.

8.5.4.1 Client-specific GFP-F processes

8.5.4.1.1 Client-specific GFP-F source processes

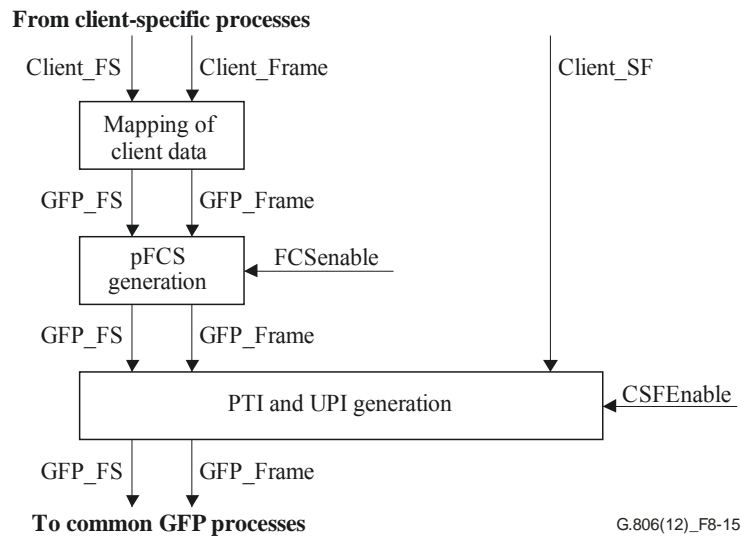


Figure 8-15 – Client-specific GFP-F source processes

Figure 8-15 shows the client-specific GFP-F source processes. The processes are performed on a frame-per-frame basis.

The basic functionality is described below. Client layer-specific deviations or extensions to the processes might be defined in the adaptation functions of the technology-specific equipment Recommendations.

Mapping of client data: The client frame is inserted into the client payload information field of the GFP frame. One client frame results in one GFP frame. The mapping for the different client signals is defined in clause 7 of [ITU-T G.7041].

pFCS generation: In the case where pFCS generation is enabled ($FCSenable = true$), the FCS is calculated over the payload information field of a frame and inserted into the pFCS fields of the frame as defined in clause 6.1.2.2.1 of [ITU-T G.7041]. The PFI field of the type header is set to "1".

In the case where pFCS generation is disabled ($FCSenable = false$), no pFCS field is added to the frame. The PFI field of the type header is set to "0".

NOTE 1 – For some client signals, FCS generation is not supported. This will be defined in the client-specific adaptation functions.

PTI and UPI generation: The PTI field of the GFP type header of an incoming GFP client data frame is set to "000". The UPI field of the GFP data frame type header is set according to the specific client signal and mapping. The UPI codes are defined in Table 6-3 of [ITU-T G.7041].

In the case where $Client_SF$ and $CSFenable$ are true, GFP client management frames are inserted instead of GFP client data frames. The PTI field of the GFP type header of the GFP client management frame is set to "100". The UPI field is set as per UPI codes defined in Table 6-4 of [ITU-T G.7041]. These GFP client management frames have no payload information field. They are generated as defined in clause 6.3.3 of [ITU-T G.7041].

Defects: None.

Consequent actions: None.

Defect correlations: None.

Performance monitoring: None.

8.5.4.1.2 Client-specific GFP-F sink processes

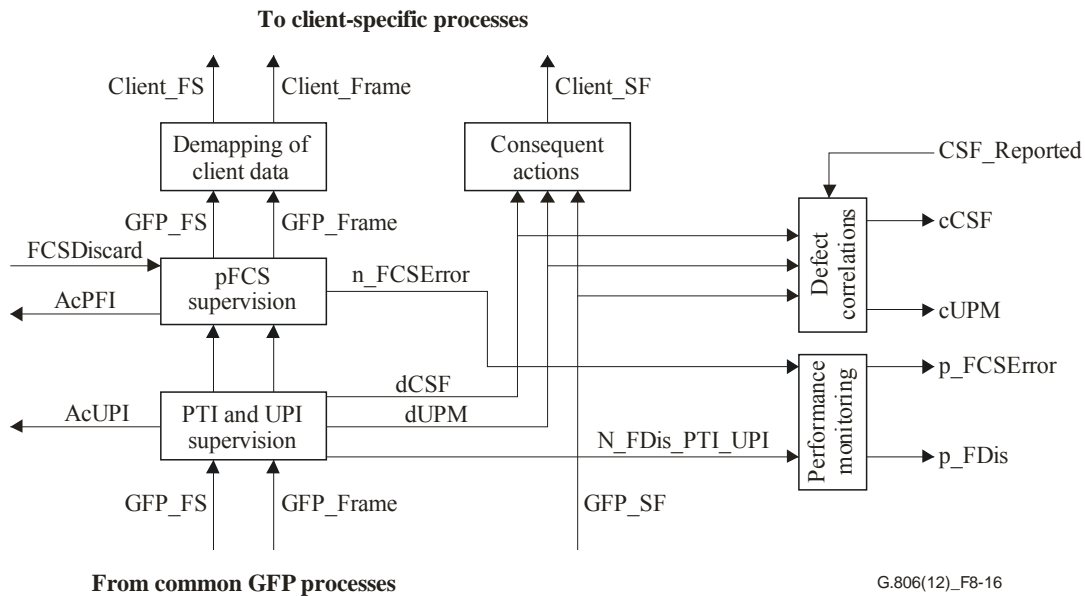


Figure 8-16 – Client-specific GFP-F sink processes

Figure 8-16 shows the client-specific GFP-F sink processes. The processes are performed on a frame-per-frame basis.

The basic functionality is described below. Client layer-specific deviations or extensions to the processes might be defined in the adaptation functions of the technology-specific equipment Recommendations.

PTI and UPI supervision: GFP frames with an accepted PTI (AcPTI, see clause 8.5.1.1) of "000" are client data frames. If the accepted UPI value (AcUPI, see clause 8.5.1.4) of these client data frames equals the expected value for the specific client signal and mapping, they are forwarded to the demapping process, otherwise they are discarded. The UPI codes are defined in Table 6-3 of [ITU-T G.7041].

GFP frames with an accepted PTI (AcPTI, see clause 8.5.1.1) value of "100" are client management frames (CMF). Received CMFs are checked for a valid and supported UPI value, as per the UPI codes defined in Table 6-4 of [ITU-T G.7041], processed based upon that UPI, and then discarded. UPI values, other than defect clear indication (0000 0011) cause dCSF to be detected. The UPI value client defect clear indication (0000 0011) causes an existing dCSF to clear (see clause 6.2.6.4).

NOTE 1 – For applications which do not utilize CMFs, the application-specific CMF process is to discard any received CMFs.

All GFP frames with an accepted PTI (AcPTI, see clause 8.5.1.1) value other than "000" or "100" shall be discarded. All discarded frames are indicated by n_FDis_PTI_UPI.

The accepted UPI (AcUPI) of client data frames is reported to the management (AcUPI).

pFCS supervision: In the case where the accepted PFI value (AcPFI, see clause 8.5.1.2) is set to "1", the pFCS of the frame is checked as defined in clause 6.1.2.2.1 of [ITU-T G.7041]. In case errors are detected and FCSDiscard is true, the frame is discarded. Errored frames are indicated by n_FCSError. The accepted PFI (AcPFI) is available to the management.

NOTE 2 – Discard of frames with FCS errors may depend on the client signal (e.g., disabled for clients that have error correction capabilities) or could be configurable by the management. The specific behaviour will be defined in the client-specific equipment functions.

NOTE 3 – For some client signals, FCS check is not supported. This will be defined in the client-specific adaptation functions. However the PFI acceptance process should always be supported in order to correctly extract the client data from GFP frames with and without FCS.

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

Defects:

The function shall detect for dUPM defect.

dUPM: see clause 6.2.4.3.

dCSF: see clause 6.2.6.4.

Consequent actions:

The function shall perform the following consequent actions:

$aClient_SF \leftarrow GFP_SF \text{ or } dUPM \text{ or } dCSF$

Defect correlations:

The function shall perform the following defect correlations to determine the most probable fault cause. This fault cause shall be reported to the EMF.

$cUPM \leftarrow dUPM \text{ and } (not \ GFP_SF)$

$cCSF \leftarrow dCSF \text{ and } (not \ dUPM) \text{ and } (not \ GFP_SF) \text{ and } CSF_Reported$

Performance monitoring:

The function shall perform the following performance monitoring primitives processing. The performance monitoring primitives shall be reported to the EMF.

$p_FDis \leftarrow \sum n_FDis_PTI_UPI$

$p_FCSError \leftarrow \sum n_FCSError$

8.5.4.2 Client-specific GFP-T processes

8.5.4.2.1 Client-specific GFP-T source processes

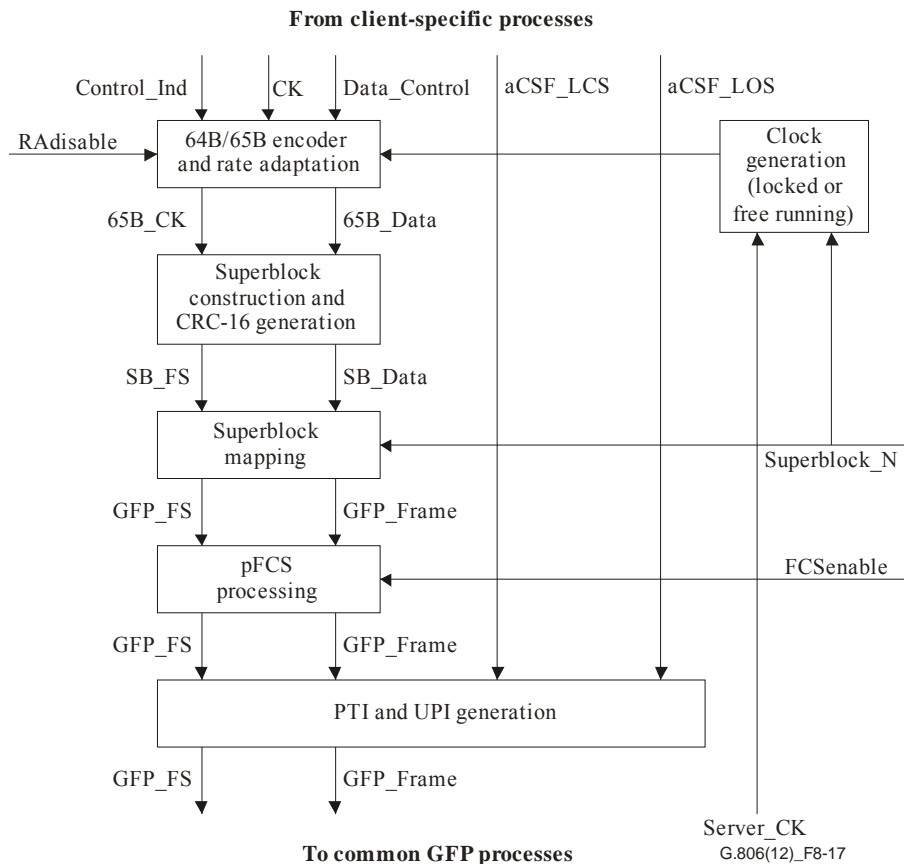


Figure 8-17 – Client-specific GFP-T source processes

Figure 8-17 shows the client-specific GFP-T source processes. The input to the process is a stream of data and control octets (Data_Control), an indication that the current octet is a control octet (Control_Ind), a clock (CK) and a loss of signal (CSF_LOS) and loss of character synchronization (CSF_LCS) indication from the server layer. The basic functionality is described below. Client layer-specific deviations or extensions to the processes might be defined in the adaptation functions of the technology-specific equipment Recommendations.

Clock generation: The process generates the clock for the generation of the GFP frames. The clock rate has to be such that client data can be accommodated at its maximum rate. The clock is locked to the server layer clock (Server_CK). Optionally, a free-running clock can be used. In the latter case, an additional rate adaptation to the server layer will be performed at the server layer-specific GFP processes using GFP idle frames.

64B/65B encoder and rate adaptation: The process constructs a 64B/65B code word from eight consecutive received data or control words as defined in clause 8.1.1 of [ITU-T G.7041]. If no data or control word is available and rate adaptation is enabled (RAisable=false), a 65B_PAD character is inserted instead as defined in clause 8.1.1.2 of [ITU-T G.7041].

NOTE – RAisable is implicitly set to false in equipment designed to earlier versions of this Recommendation.

Superblock construction and CRC-16 generation: The process constructs a GFP-T superblock from eight received 65B data words as defined in clause 8.1.2 of [ITU-T G.7041]. The leading flag bits of each 65B word are grouped together at the end of the 8x8 octet data or control fields. A CRC-16 is calculated over the superblock and inserted at the end of the superblock as defined in clause 8.1.2.1 of [ITU-T G.7041].

Superblock mapping: N superblocks are grouped together in the client payload information field of the GFP frame. N depends on the client bit rate and server layer capacity. It might be fixed or configurable (Superblock_N).

pFCS generation: In the case where pFCS generation is enabled (FCSenable = true) the FCS is calculated over the payload information field of a frame and inserted into the pFCS fields of the frame as defined in clause 6.1.2.2.1 of [ITU-T G.7041]. The PFI field of the type header is set to "1".

In the case where pFCS generation is disabled (FCSenable = false), no pFCS field is added to the frame. The PFI field of the type header is set to "0".

NOTE 1 – For some client signals, FCS generation is not supported. This will be defined in the client-specific adaptation functions.

NOTE 2 – FCS is not generated for the currently defined client management frames.

PTI and UPI generation: The PTI field of the GFP-type header of an incoming GFP client data frame is set to "000". The UPI field of the GFP-type header is set according to the specific client signal and mapping. The UPI codes are defined in Table 6-3 of [ITU-T G.7041]. In case aCSF_LOS or aCSF_LCS are active, GFP client management frames are inserted instead of GFP client data frames. The PTI field of the GFP-type header of the GFP client management frames is set to "100". The UPI is set to "0000 0001" in the case where aCSF_LOS is active and set to "0000 0010" in the case where aCSF_LCS is active. These GFP client management frames have no payload information field. They are generated as defined in clause 6.3.3 of [ITU-T G.7041].

Defects: None.

Consequent actions: None.

Defect correlations: None.

Performance monitoring: None.

8.5.4.2.2 Client-specific GFP-T sink processes

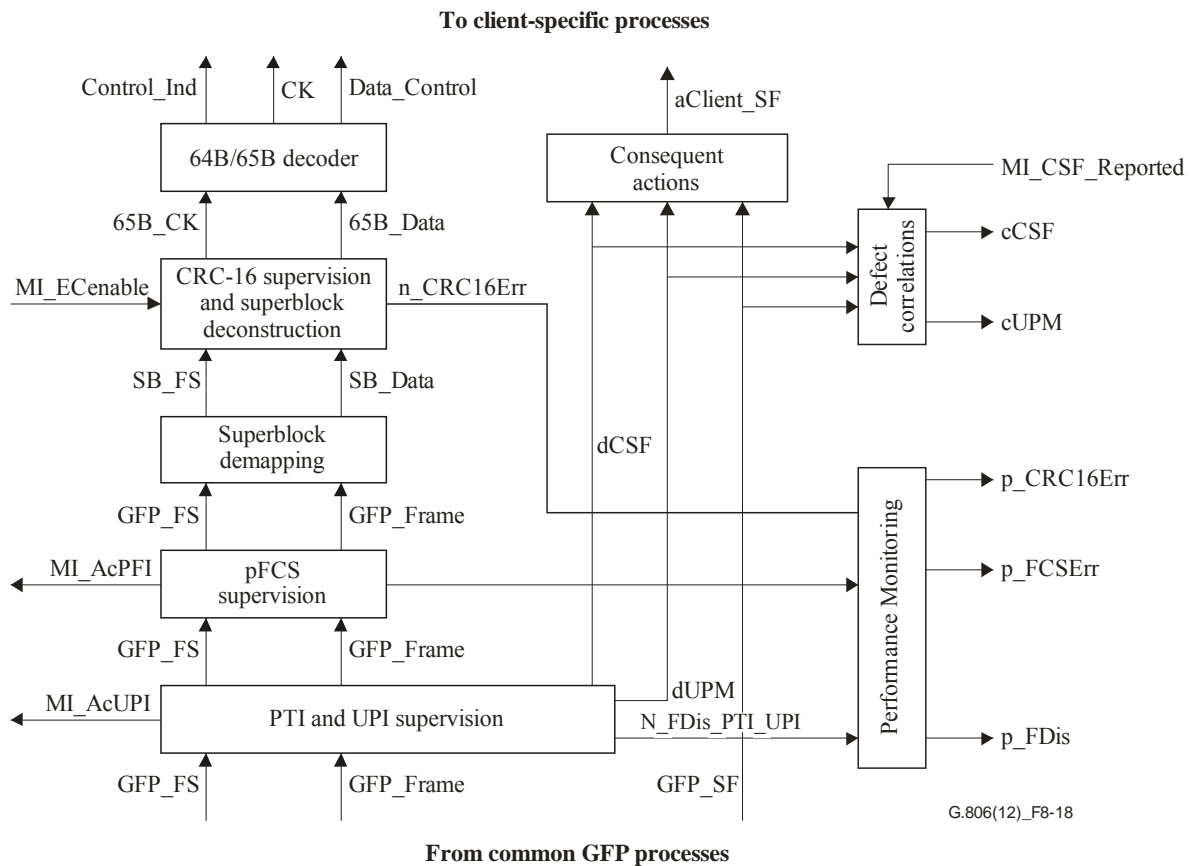


Figure 8-18 – Client-specific GFP-T sink processes

Figure 8-18 shows the client-specific GFP-T sink processes.

The basic functionality is described below. Client layer-specific deviations or extensions to the processes might be defined in the adaptation functions of the technology-specific equipment Recommendations. The output of the function is a stream of data and control octets (*Data_Control*), an indication that the current octet is a control octet (*Control_Ind*) and a (gapped) clock (*CK*).

PTI and UPI supervision: GFP frames with an accepted PTI value (*AcPTI*, see clause 8.5.1.1) of "000" are client data frames. If the accepted UPI value (*AcUPI*, see clause 8.5.1.4) of these frames equals the expected value for the specific client signal and mapping, they are forwarded to the demapping process. The UPI codes are defined in Table 6-3 of [ITU-T G.7041]. GFP frames with an *AcPTI* value of "100" are client management frames. The UPI value of these frames is checked for "0000 0001" and "0000 0010" for the *dCSF* defect detection and then are discarded. All other frames are discarded.

All discarded frames, except the discarded client management frames, are indicated by *n_FDIs_PTI_UPI*.

The accepted UPI (*AcUPI*) of the client data frames is reported to the management (*AcUPI*).

pFCS supervision: In the case where the accepted PFI value (*AcPFI*, see clause 8.5.1.2) is set to "1", the pFCS of the client data frame is checked, as defined in clause 6.1.2.2.1 of [ITU-T G.7041]. Errored frames are indicated by *n_FCSErr*. The accepted PFI (*AcPFI*) is available to the management.

NOTE 1 – Frames with FCS errors are not discarded in case of GFP-T mappings as the CRC-16 of the GFP-T superblock can correct single bit errors (see below).

NOTE 2 – FCS supervision is not performed for the currently defined client management frames.

NOTE 3 – For some client signals, the FCS check is not supported. This will be defined in the client-specific adaptation functions. However, the PFI acceptance process should always be supported in order to correctly extract the client data from GFP frames with and without FCS.

Superblock demapping: N superblocks are extracted from the client payload information field of the GFP frame. N is defined by the size of the GFP frame.

CRC-16 supervision and superblock deconstruction: This process checks the CRC-16 of a received superblock for errors. Single bit error correction may be performed on each superblock (ECenable). If error correction is not enabled and a single error is detected, or if multiple errors are detected, all 64 data octets of the superblock are replaced by 10B_ERR control words and the block is indicated as errored block by n_CRC16Err.

NOTE 4 – Since the sink adaptation process performs the CRC-16 check after the payload descrambling is performed, the error correction circuit should account for single bit errors as well as double errors spaced 43 bits apart coming out of the descrambler.

The eight 65B data words are extracted from the superblock as defined in clause 8.1.2.1 of [ITU-T G.7041].

64B/65B decoder: The process extracts 8 data or control words from the 65B code word as defined in clause 8.1.1 of [ITU-T G.7041]. A control word is indicated by an active Control_Ind. 65B_PAD characters are dropped from the data stream as defined in clause 8.1.1.2 of [ITU-T G.7041].

NOTE – RAdisable is implicitly set to false in equipment designed to earlier versions of this Recommendation.

Defects:

The function shall detect for dUPM and dCSF defects.

dUPM: see clause 6.2.4.3.

dCSF: see clause 6.2.6.4.

Consequent actions:

The function shall perform the following consequent actions:

aClient_SF ← GFP_SF or dUPM or dCSF

Defect correlations:

The function shall perform the following defect correlations to determine the most probable fault cause. This fault cause shall be reported to the EMF.

cUPM ← dUPM and (not GFP_SF)

cCSF ← dCSF and (not dUPM) and (not GFP_SF) and CSF_Reported

Performance monitoring:

The function shall perform the following performance monitoring primitives processing. The performance monitoring primitives shall be reported to the EMF.

p_FDis ← \sum n_FDis_PTI_UPI

p_FCSError ← \sum n_FCSError

p_CRC16Err ← \sum n_CRC16Err

9 Performance and reliability

9.1 Transit delay

To derive the total transit delay of a signal through a network element, all processes that could contribute non-negligible delay must be taken into account. Since it is only possible to measure transit delay from NNI to NNI, that value is the only one which must be derived.

The contributing processes which have been identified to date are:

- Pointer buffer processing. (A distinction could be made between pointer buffer threshold spacing and pointer adjustment processes).
- Fixed stuff processing. Overhead could be regarded as fixed stuffing for a particular signal.
- Processing which is implementation-dependent, e.g., internal interface processing.
- Connection processing.
- Mapping processing.
- Demapping processing.

Depending on NNI and processing levels, several of the above-mentioned processes must be taken into account. The total delay is then calculated as the sum of the processes involved. These values could be given as minimum, average or maximum values under normal operating conditions or in worst-case failure scenarios.

Another parameter associated with delay is the differential transit delay of path signals within the same server trail.

NOTE – Specifications of transit delay and differential transit delay are outside the scope of this Recommendation.

9.2 Response times

Matrix set-up delay is the time taken from the generation of a primitive within the EMF to the change of transport information at the NNI. It may be necessary to distinguish between preset configurations, subject to an execute primitive and a normal set.

Message processing delay is the time from the end of message at Q until the primitive is generated within the EMF; i.e., the message has been decoded to an actionable level.

NOTE – Specifications of response times are outside the scope of this Recommendation.

9.3 Availability and reliability

For a network provider, the reliability of network elements is of prime concern as it directly influences the availability of connections. However, the availability of a connection depends not only on the reliability of the network elements themselves, but also on the level of network redundancy. Furthermore, it depends on the restoration times of the equipment involved. The restoration times depend, to a great extent, on the operation, administration and management (OAM) philosophy of the network provider.

A manufacturer has, in most cases, requirements from several operators to take into account. Requirements from a certain network provider will depend on the level of economic development of the country concerned, the degree of market competition, customer requirements, the level of network redundancy, the level of maintenance support, etc.

The basis for determining the availability of a network element should be the analytical method for dependability as described in [ITU-T E.862].

The main point of the analytical method is that dependability aspects are taken into account as an economic factor. The level of availability is thus dimensioned according to cost-benefit analyses rather than by predetermined stated objectives.

The application of the method to network components is shown in [b-ITU-T Handbook].

Parameters and calculation methodologies for reliability and availability are defined in [ITU-T G.911].

NOTE – Availability and reliability specifications for network elements and trail/connections are outside the scope of this Recommendation.

9.4 Laser safety

For safety considerations, it may be necessary to provide for an automatic power shutdown (APSD) or automatic laser shutdown (ALS) facility of the laser in case of cable break. Refer to [ITU-T G.664].

10 Generic equipment functions

This clause includes generic definitions for some transport network functions that are independent of the transport network technology used (e.g., SDH, OTN). Wherever technology-specific aspects are required in the definition of these functions, this is noted in the definitions.

For any of these functions to be defined for a particular transport network technology, the technology-specific equipment Recommendations (e.g., [ITU-T G.783] for SDH and [ITU-T G.798] for OTN) must include definitions for it, which also must include any technology-specific aspects.

10.1 LCAS-capable virtual concatenated path layer functions P-Xv-L ($X \geq 1$)

In the equipment function definitions within this clause, the prefix "P" is used to denote the particular (network technology-specific) path layer that these functions are being instantiated in. For example, for SDH, "P" will represent "Sn" for the VC-3/4 path layer and "Sm" for the VC-1/2 path layer. For OTN, "P" will generally represent the OPUk path layer.

10.1.1 LCAS-capable virtual concatenated path layer trail termination function P-Xv-L_TT

The P-Xv-L_TT function is further decomposed as defined in [ITU-T G.805] and shown in Figure 10-1.

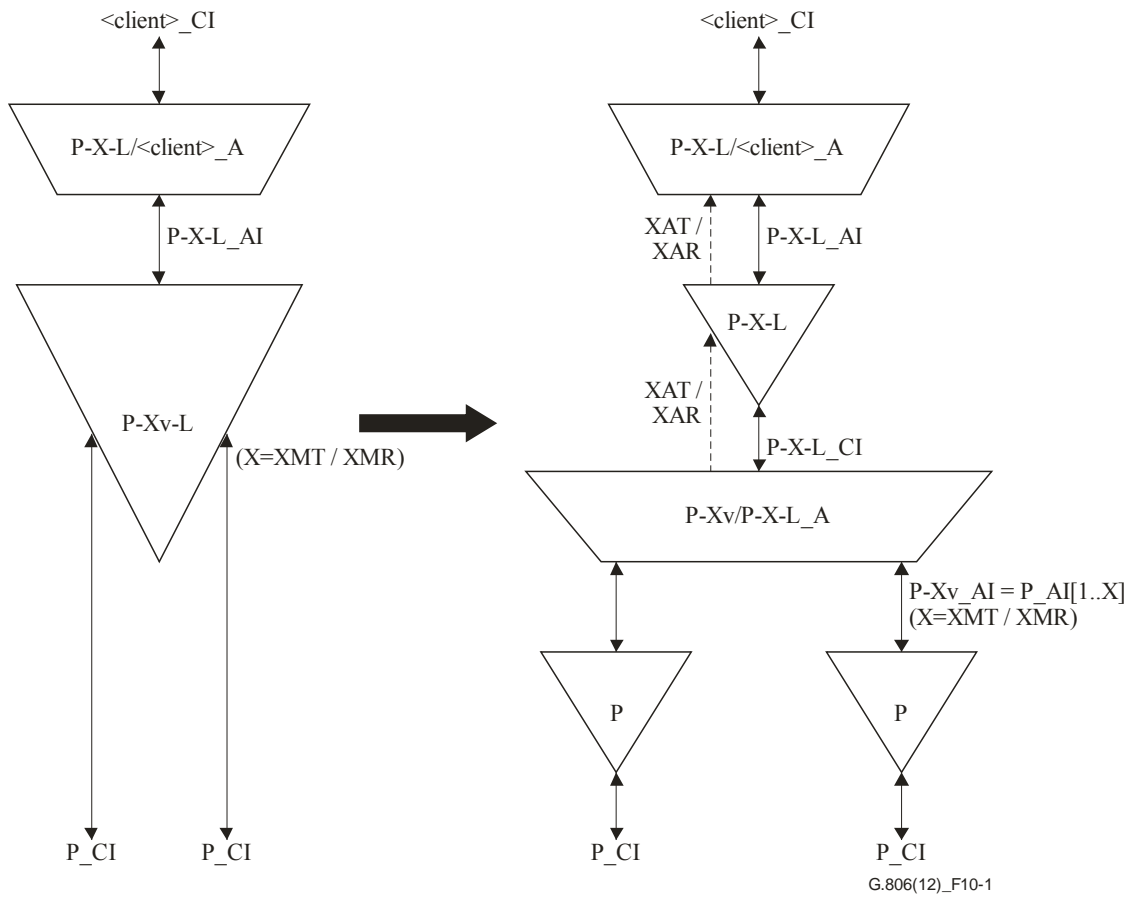


Figure 10-1 – Decomposition of P-Xv-L_TT function

The P_TT functions are the normal path layer trail termination functions as defined in the technology-specific Recommendations.

**10.1.1.1 LCAS-capable virtual concatenated path adaptation source function
P-Xv/P-X-L_A_So**

Symbol:

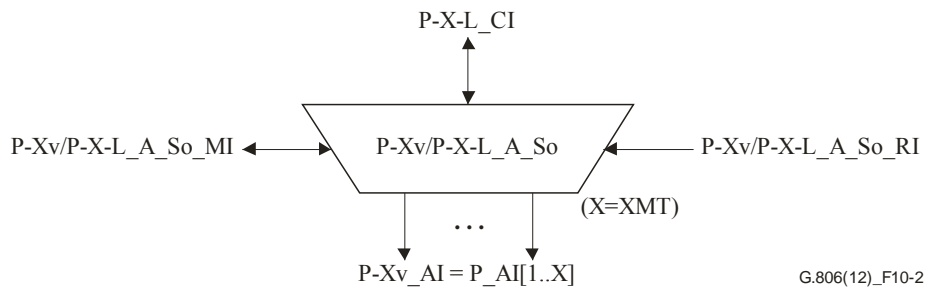


Figure 10-2 – P-Xv/P-X-L_A_So symbol

Interfaces:

Table 10-1 – P-Xv/P-X-L_A_So input and output signals

Inputs	Outputs
P-X-L_CP: P-X-L_CI_D P-X-L_CI_CK P-X-L_CI_FS P-Xv/P-X-L_A_So_MP: P-Xv/P-X-L_A_So_MI_LCASEnable P-Xv/P-X-L_A_So_MI_ProvM[1..X _{MT}] P-Xv/P-X-L_A_So_MI_PLCTThr P-Xv/P-X-L_A_So_RP: P-Xv/P-X-L_A_So_RI_RS_Ack_rec P-Xv/P-X-L_A_So_RI_RS_Ack_gen P-Xv/P-X-L_A_So_RI_MST_rec[0..MST_Range] P-Xv/P-X-L_A_So_RI_MST_gen[0..MST_Range]	P-Xv_AP: P-Xv_AI_D = P_AI[1.. X _{MT}] _D P-Xv_AI_CK = P_AI[1.. X _{MT}] _{CK} P-Xv_AI_FS = P_AI[1.. X _{MT}] _{FS} P-X-L_CP: P-X-L_CI_X _{AT} P-Xv/P-X-L_A_So_MP: P-Xv/P-X-L_A_So_MI_X _{AT} P-Xv/P-X-L_A_So_MI_X _{MT} P-Xv/P-X-L_A_So_MI_TxSQ[1..X _{MT}] P-Xv/P-X-L_A_So_MI_cPLCT P-Xv/P-X-L_A_So_MI_cTLCT P-Xv/P-X-L_A_So_MI_cFOPT

Processes:

This function shall perform the distribution of the incoming P-X-L_CI over the outgoing path layer signals that form the P-Xv_AI (= P_AI[1..X_{MT}]). This function can operate in two modes: with the LCAS functionality enabled or disabled.

If the LCAS functionality is enabled, the number of actual members of the VCG (X_{AT}) will be controlled by the LCAS protocol and will be made available at the P-X-L_CP. Any values of 0 ≤ X_{AT} ≤ X_{PT} are possible. If the LCAS functionality is disabled, the function behaves equivalently to a P-X/P-Xv_A_So function (see below for details), with X = X_{PT}.

The MI_LCASEnable input controls whether the LCAS functionality is enabled for the source function (MI_LCASEnable = true) or disabled (MI_LCASEnable = false).

The MI_ProvM[1..X_{MT}] input controls whether a P[i]_AP at the P-Xv_AP is provisioned to be a member of the VCG (MI_ProvM[i] = 1) or not (MI_ProvM[i] = 0). Note that X_{PT}, as defined above, is given by:

$$X_{PT} = \sum_{i=1}^{X_{MT}} \text{ProvM}[i]$$

Any values of X_{PT} and X_{MT} satisfying 0 ≤ X_{PT} ≤ X_{MT} are possible.

A process diagram of this function is included in Figures 10-3 and 10-4.

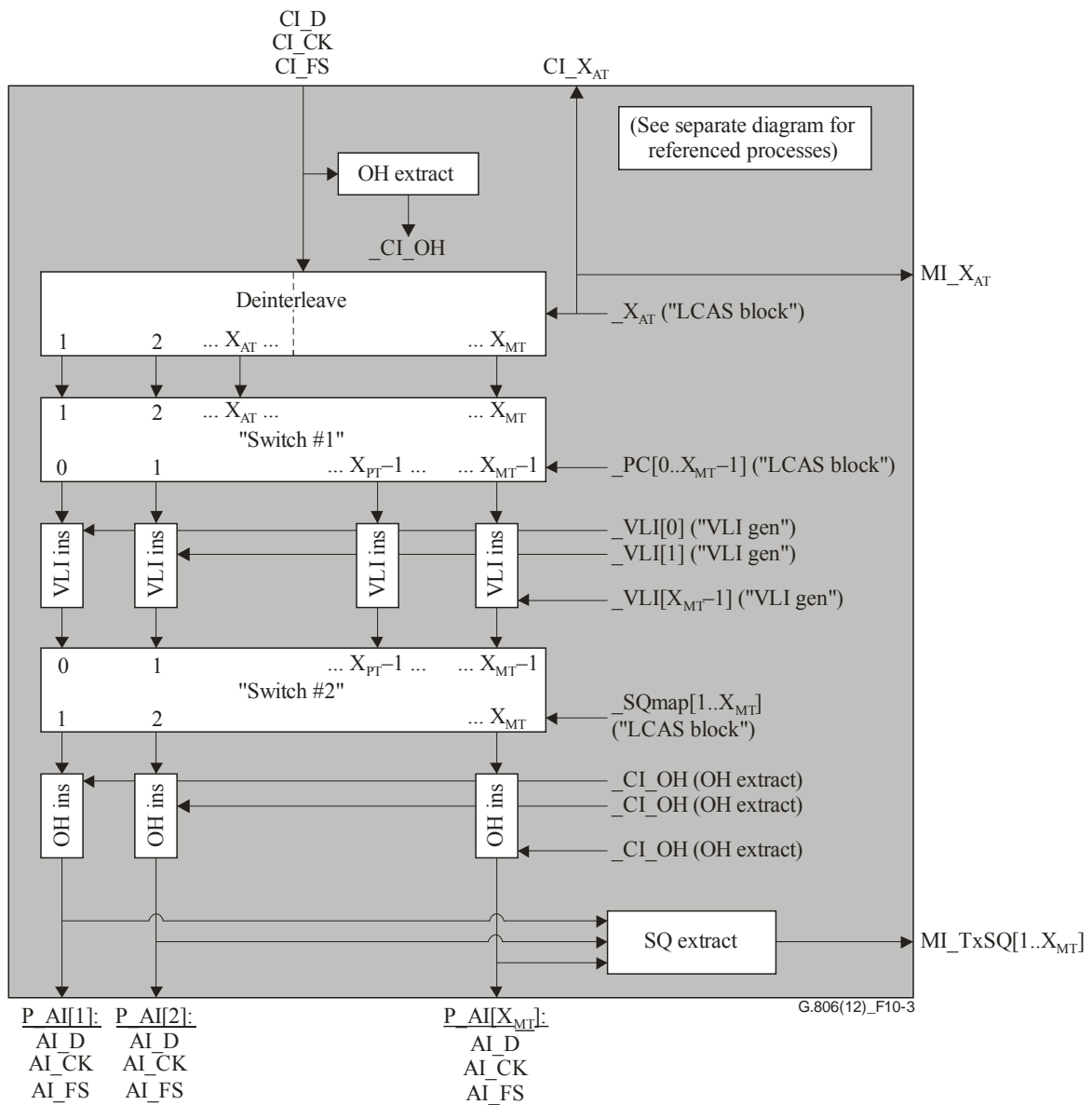
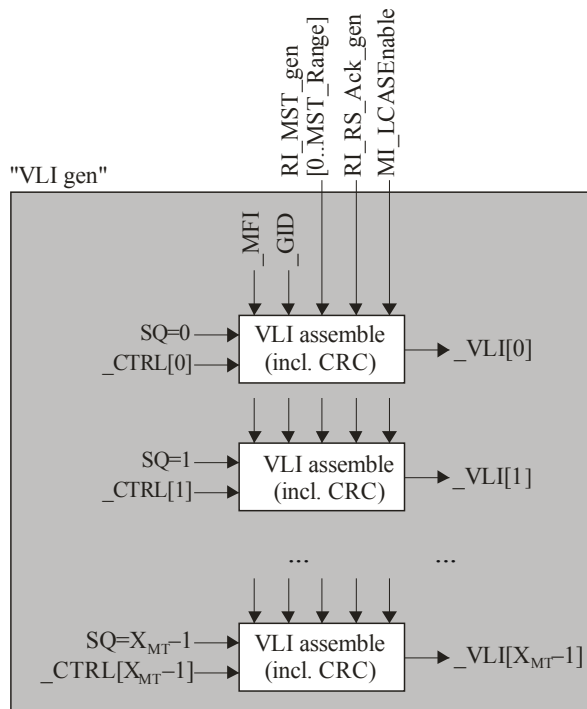
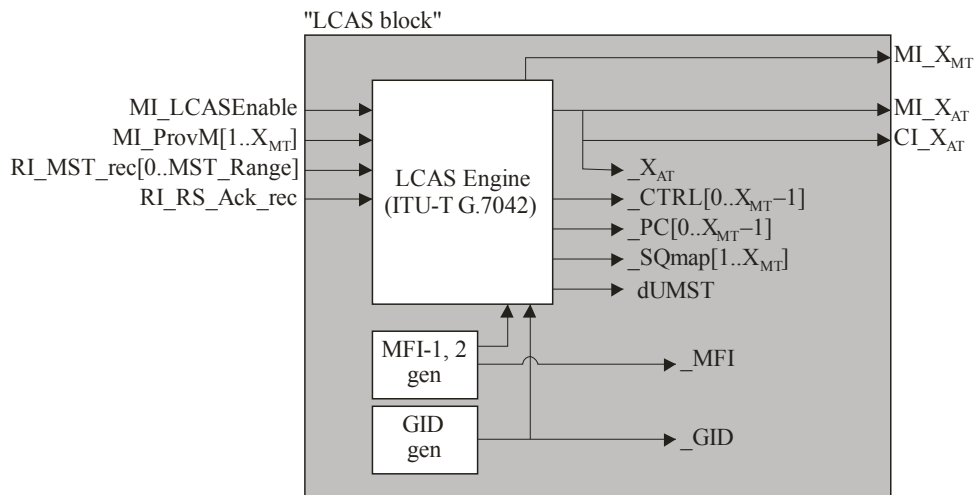
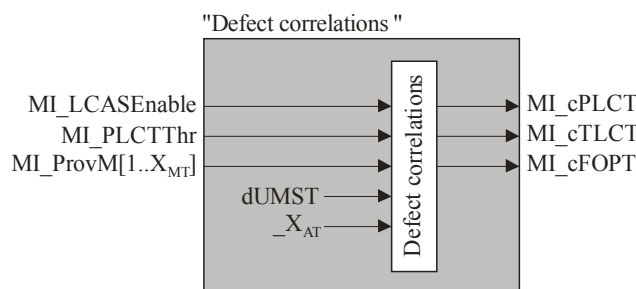


Figure 10-3 – P-Xv/P-X-L_A_So processes (main diagram)



NOTE 1 – The internal inputs `_GID`, `_MFI`, `_CTRL[k]` are sourced at the "LCAS block".



G.806(12)_F10-4

NOTE 2 – The internal inputs `dUMST` and `_X_AT` are sourced at the "LCAS block".

Figure 10-4 – P-Xv/P-X-L_A_So processes (sub-diagrams)

OH extract (Figure 10-3):

This process extracts a set of technology-specific path-layer overhead bytes ($_CI_OH$) from the CI_D signal and makes them available for reinsertion at the OH insert processes (see OH insert below).

Deinterleave (distribution process) (Figure 10-3):

This process distributes the $P-X-L_CI_D$ signal to the X_{AT} path-level signals numbered $1..X_{AT}$ at its output. The value of X_{AT} is given by the input $_X_{AT}$, which is itself controlled by the LCAS protocol (see below).

Each of the X_{MT} outputs of this process is a full-rate P_AI signal and includes a P_AI_D , P_AI_CK and a P_AI_FS .

The distribution procedure of $P-X-L_CI_D$ onto X_{AT} path-level signals is technology-specific. Also, the content of the signal that is sourced in this process for outputs, or the outputs X_{AT+1} , X_{AT+2} , ..., X_{MT} , is technology-specific.

"Switch 1" (assignment of sequence numbers) (Figure 10-3):

This process is a switch that connects the X_{AT} path-level signals at its inputs 1 to X_{AT} of its outputs in the range 0 to $X_{PT}-1$ ($X_{AT} \leq X_{PT}$). The output "s" of this switch is associated with the LCAS sequence number s.

The connections are controlled by the $_PC[s]$ signal, which indicates whether a particular output is carrying payload at a certain time ($_PC[s] = 1$) or not ($_PC[s] = 0$), see LCAS engine process below for details).

The connections from input to output are as follows:

- each payload-carrying output is exactly connected to one input. Payload-carrying outputs are those for which $_PC[s] = 1$;
- all payload-carrying outputs are connected to inputs "1" through " X_{AT} ";
- among the payload-carrying outputs, the output with the lowest index (s) is connected to input 1, the output with the second-lowest index is connected to input 2 and so on until the payload-carrying output with the highest index, which is connected to input X_{AT} ;
- the mapping is fixed in time (i.e., does not change over time as long as $_PC$ does not change).

The content of the signal that is sourced in this process for all non-payload-carrying outputs ($_PC[s] = 0$) is technology-specific³.

Sample scenarios are available in Appendix VII.

VLI insertion (Figure 10-3):

This process inserts the VCAT/LCAS information for each path (as given by $_VLI[s]$) into the corresponding overhead positions in each of the paths. The value of $_VLI[s]$ is calculated by the VLI assemble process (see below).

NOTE 1 – The insertion process (including the used overhead locations and coding) is technology-dependent.

³ Note that, under normal conditions, the LCAS engine will set $_PC[s] = 1$ for $s = 0..X_{AT}-1$ and $_PC[s] = 0$ otherwise. In this case, the switch will just connect its input $s+1$ to its output s. During a signal fail condition (as signalled by the sink function using MST information) of an active P trail with sequence number s, the LCAS engine will set $_PC[s] = 0$ and reduce X_{AT} , temporarily blocking traffic over the failed output (until the sink signals a healthy path again).

VLI assemble and CRC (Figure 10-4):

This process constructs the VCAT/LCAS information sequences to be transmitted in each path-level signal. For each path s , the VCAT/LCAS information $_VLI[s]$ is created, using the following components.

If $MI_LCASEnable$ is active:

- MFI: Multiframe indication, as sourced by the MFI generator process ($_MFI$).
- CTRL: Control channel word, as sourced by the LCAS engine process ($_CTRL[s]$).
- GID: Group identifier, as sourced by the GID generator process ($_GID[s]$).
- SQ: Sequence indicator, as input to the process.
- MST: Generated member status, as received from $RI_MST_gen[0..MST_Range]$.
- RS_Ack: Generated re-sequence acknowledge, as received from $RI_RS_Ack_gen$.
- CRC: Cyclic redundancy check of the LCAS packet information.

If $MI_LCASEnable$ is not active:

- MFI: Multiframe indication, as sourced by the MFI generator process ($_MFI$).
- CTRL: Sourced as all-zeroes.
- GID: Sourced as zero.
- SQ: Sequence indicator, as input to the process.
- MST: Sourced as all-zeroes.
- RS_Ack: Sourced as zero.
- CRC: Sourced as all-zeroes.

The structure of the VLI information, including the particular CRC scheme used and the value of any unused space within that structure, is technology-dependent.

"Switch 2" (mapping to physical resources) (Figure 10-3):

This process is a switch that connects the path-level signals at its inputs in the range 0 to $X_{PT}-1$ to X_{PT} of its outputs in the range 1 to X_{MT} . The output "i" of this switch is associated with the $P_AP[i]$.

This switch changes configuration as provisioning commands activate and deactivate members of the VCG for usage. Sample scenarios are available in Appendix VII.

The connections are controlled by the $_SQmap[i]$ signal, which indicates which input a particular output i is connected to at a certain time (see LCAS engine process below for details).

The connections from input to output are as follows. For each output i :

- If $_SQmap[i] \neq n/a$, output i is connected to input $_SQmap[i]$.
- If $_SQmap[i] = n/a$, this process sources a member signal with the following characteristics:
 - multiframe-synchronous to the other signals at the output of the switch; and
 - with the same multiframe structure and VLI contents as the other members but with an IDLE control word and a sequence number complying with the requirements in [ITU-T G.7042] for non-provisioned members.

LCAS engine (Figure 10-4):

This process executes the LCAS protocol (if MI_LCASEnable is active), and provides appropriate control signals for other processes.

If MI_LCASEnable is active, it produces the following outputs:

- $_PC[0..X_{MT}-1]$: Indication of whether a member with a particular sequence number is an active member (i.e., is carrying payload) at a particular time. For each sequence number s , the process shall calculate $_PC[s]$ as determined by the LCAS protocol in [ITU-T G.7042] (= 1 if the protocol determines it is carrying payload, = 0 otherwise).
- $_X_{AT}$: Current size of the transmitted payload. Note that, with the above definition, $_X_{AT}$ is given by:

$$_X_{AT} = \sum_{s=0}^{X_{MT}-1} _PC[s]$$

- $_CTRL[0..X_{MT}-1]$: LCAS control word for each member, indexed by the sequence number. It is calculated as determined by the LCAS protocol in [ITU-T G.7042].
- $_SQmap[1..X_{MT}]$: Indication of which sequence number ($_SQmap[i]$) is being carried over a particular P_AI (P_AI[i]) signal. For each P_AI signal number i , the process shall calculate $_SQmap[i]$ such that:
 - If MI_ProvM[i] = 1, $_SQmap[i]$ shall be the sequence number for the member to be carried over P_AI[i] as determined by the LCAS protocol in [ITU-T G.7042].
 - If MI_ProvM[i] = 0, $_SQmap[i] = n/a$.

NOTE 2 – This implies that the X_{PT} provisioned outputs (i.e., those with MI_ProvM[i] = 1) will be connected to inputs $0..X_{PT}-1$ by the switch 2 process.

NOTE 3 – $_SQmap$ will vary whenever members are added or removed from an LCAS-enabled link and will be, in general, dependent on the history of additions and removals in the link.

See Appendix VII for examples.

NOTE 4 – This is the only output of the LCAS engine process which has the characteristic of being indexed by the P_AI number, instead of the sequence number.

NOTE 5 – The LCAS protocol, as defined in [ITU-T G.7042], is used in this process to calculate some outputs. The instance of the protocol used here shall have the following characteristics:

- It shall interpret a change in MI_ProvM[i] as a request to add (MI_ProvM[i]: 0→1) or remove (MI_ProvM[i]: 1→0) a member. Multiple simultaneous requests are possible.

If MI_LCASEnable is not active, the outputs are as follows:

- $_PC[0..X_{MT}-1]$: Indication of whether a member with a particular sequence number is an active member (i.e., is carrying payload) at a particular time. For each sequence number s , the process shall calculate $_PC[s]$ as follows:
 - for $0 \leq s \leq X_{PT}-1$, $_PC[s] = 1$
 - for $X_{PT} \leq s \leq X_{MT}-1$, $_PC[s] = 0$
- $_X_{AT}$: Current size of the transmitted payload. Note that, with the above definition, $_X_{AT}$ is given by:

$$_X_{AT} = \sum_{s=0}^{X_{MT}-1} _PC[s]$$

NOTE 6 – This definition makes $_X_{AT} = X_{PT}$ for as long as MI_LCASEnable is not active.

- $_CTRL[0..X_{MT}-1]$: LCAS control word for each member, indexed by the sequence number. It is sourced as an all-zeroes word.

– $_SQmap[1..X_{MT}]$: Indication of which sequence number ($_SQmap[i]$) is being carried over a particular P_AI ($P_AI[i]$) signal. For each sequence number i , the process shall calculate $_SQmap[i]$ such that:

- if $\{i_0, i_1, \dots, i_{X_{PT}-1}\}$ is the set of indexes i_n for which $MI_ProvM[i_n] = 1$, ordered by value ($i_0 < i_1 < \dots < i_{X_{PT}-1}$), then $_SQmap[i_n] = n$;

NOTE 7 – In other words, for the provisioned members (indexes i for which $MI_ProvM[i] = 1$), $_SQmap[i]$ shall be such that $_SQmap[i] = 0$ for the provisioned member with the lowest index, $_SQmap[i] = 1$ for the provisioned member with the second-lowest index and so on until $_SQmap[i] = X_{PT}-1$, which will be the value for the provisioned member with the highest index.

- for all members for which $MI_ProvM[k_i] = 0$, $_SQmap[k_i] = n/a$.

NOTE 8 – This is the only output of the LCAS engine process which has the characteristic of being indexed by the P_AI number, instead of the sequence number.

OH insert (Figure 10-3):

This process inserts a set of technology-specific path-layer overhead bytes ($_CI_OH$), into the appropriate positions at the $AI_D[1..X_{MT}]$ signals.

NOTE 9 – There are X_{MT} of these processes in parallel, one for each P_AI, all of them inserting the same values for $_CI_OH$. The described behaviour enables the sink function to select a source for the overhead bytes among the available members, depending on their status.

Also, note that interworking with non-LCAS-capable sink functions is guaranteed with this overhead handling.

SQ extract (Figure 10-3):

This process extracts the currently-transmitted sequence indicator from each P_AI signal for reporting to the MI.

Defects:

Persistent unexpected MST (dUMST): A persistent detection (longer than a time t_{detect}) of $RI_MST_rec[i] = 0$ (OK), while no RS-ACK is pending, for a member that does not carry the "ADD", "NORM", "EOS" or "DNU" control words shall give rise to a dUMST defect. The defect shall be cleared as soon as $RI_MST_rec[i] = 1$ (FAIL) is detected persistently (longer than t_{clear}) for all members not carrying those control words. The value of the t_{detect} , t_{clear} parameters is ffs.

Consequent actions: None.

Defect correlations:

The next two parameters signal the partial loss (PLCT) or total loss (TLCT) of capacity in an LCAS-enabled VCG source:

$cPLCT \leftarrow MI_LCASEnable \text{ and } (0 < X_{AT}) \text{ and } (X_{AT} < MI_PLCTThr) \text{ and } (X_{PT} > 0)$

$cTLCT \leftarrow MI_LCASEnable \text{ and } (X_{AT} = 0) \text{ and } (X_{PT} > 0)$

$cFOPT \leftarrow MI_LCASEnable \text{ and } dUMST.$

Performance monitoring: None.

10.1.1.2 LCAS-capable virtual concatenated path adaptation sink function P-Xv/P-X-L_A_Sk

Symbol:

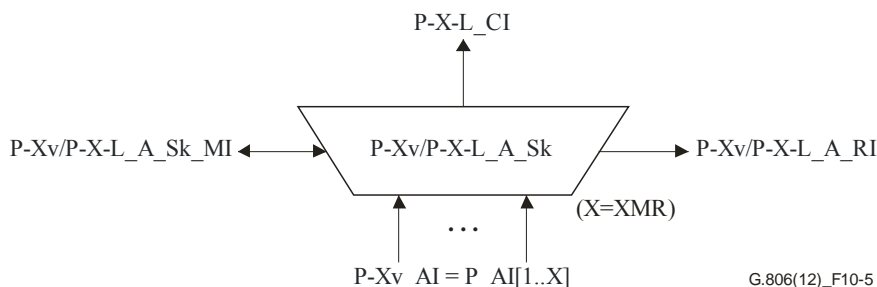


Figure 10-5 – P-Xv/P-X-L_A_Sk symbol

Interfaces:

Table 10-2 – P-Xv/P-X-L_A_Sk input and output signals

Inputs	Outputs
P-Xv_AP: P-Xv_AI_D = P_AI[1..X _{MR}] _D P-Xv_AI_CK = P_AI[1..X _{MR}] _{CK} P-Xv_AI_FS = P_AI[1..X _{MR}] _{FS} P-Xv_AI_TSF = P_AI[1..X _{MR}] _{TSF} P-Xv_AI_TSD = P_AI[1..X _{MR}] _{TSD}	P-X-L_CP: P-X-L_CI_D P-X-L_CI_CK P-X-L_CI_FS P-X-L_CI_SSF P-X-L-CI_X _{AR}
P-Xv/P-X-L_A_Sk_MP: P-Xv/P-X-L_A_Sk_MI_ProvM[1..X _{MR}] P-Xv/P-X-L_A_Sk_MI_LCASEnable P-Xv/P-X-L_A_Sk_MI_PLCRThr P-Xv/P-X-L_A_Sk_MI_TSDEnable P-Xv/P-X-L_A_Sk_MI_HOTime P-Xv/P-X-L_A_Sk_MI_WTRTime P-Xv/P-X-L_A_Sk_MI_RMVTime	P-Xv/P-X-L_A_Sk_MP: P-Xv/P-X-L_A_Sk_MI_X _{MR} P-Xv/P-X-L_A_Sk_MI_X _{AR} P-Xv/P-X-L_A_Sk_MI_DMFI[1..X _{MR}] P-Xv/P-X-L_A_Sk_MI_LCAS_So_Detected P-Xv/P-X-L_A_Sk_MI_cPLCR P-Xv/P-X-L_A_Sk_MI_cTLCR P-Xv/P-X-L_A_Sk_MI_cFOPR P-Xv/P-X-L_A_Sk_MI_cLOM[1..X _{MR}] P-Xv/P-X-L_A_Sk_MI_cSQM[1..X _{MR}] P-Xv/P-X-L_A_Sk_MI_cMND[1..X _{MR}] P-Xv/P-X-L_A_Sk_MI_cLOA P-Xv/P-X-L_A_Sk_MI_AcSQ[1..X _{MR}] P-Xv/P-X-L_A_Sk_RP: P-Xv/P-X-L_A_Sk_RI_RS_Ack_rec P-Xv/P-X-L_A_Sk_RI_RS_Ack_gen P-Xv/P-X-L_A_Sk_RI_MST_rec[0..MST_Range] P-Xv/P-X-L_A_Sk_RI_MST_gen[0..MST_Range]

Processes:

This function shall perform the alignment of the incoming path-layer signals that constitute the P-Xv_AI (= P_AI[1..X_{MR}]) to form the outgoing P-X-L_CI. This function can operate in two modes: with the LCAS functionality active or inactive.

If the LCAS functionality is active, the number of actual members of the VCG (X_{AR}) will be controlled by the LCAS protocol and will be made available at the P-X-L_CP. Any values of $0 \leq X_{AR} \leq X_{PR}$ are possible. If the LCAS functionality is inactive, the function behaves equivalently to a P-X/P-XV_A_Sk function (see below for details), with $X = X_{PR}$.

The MI_LCASEnable input controls whether the LCAS functionality is enabled for the sink function (MI_LCASEnable = true) or disabled (MI_LCASEnable = false). If LCAS is enabled, the function auto-detects which type of source it is interfacing to. The output MI_LCAS_So_Detected reports whether the present sink function detected an LCAS-enabled source function (MI_LCAS_So_Detected = true) or a non-LCAS-enabled source (MI_LCAS_So_Detected = false, see below for details). Only if both MI_LCASEnable and MI_LCAS_So_Detected are true is the LCAS functionality active in the function.

NOTE 1 – The transitional behaviour between LCAS-active and non-LCAS-active mode is for further study.

The MI_ProvM[1..X_{MR}] input controls whether a particular one of the available physical resources at the P-XV_AP is provisioned to be a member of the VCG (MI_ProvM[i] = 1) or not (MI_ProvM[i] = 0). Note that X_{PR}, as defined above, is given by:

$$X_{PR} = \sum_{i=1}^{X_{MR}} \text{ProvM}[i]$$

Any values of X_{PR} and X_{MR} satisfying $0 \leq X_{PR} \leq X_{MR}$ are possible.

The MI_TSDenable input controls whether the sink function uses AI_TSD[i] indications as contributors for signalling defective members back to the LCAS source function (MI_TSDenable = true) or whether it ignores AI_TSD[i] indications altogether (MI_TSDenable = false).

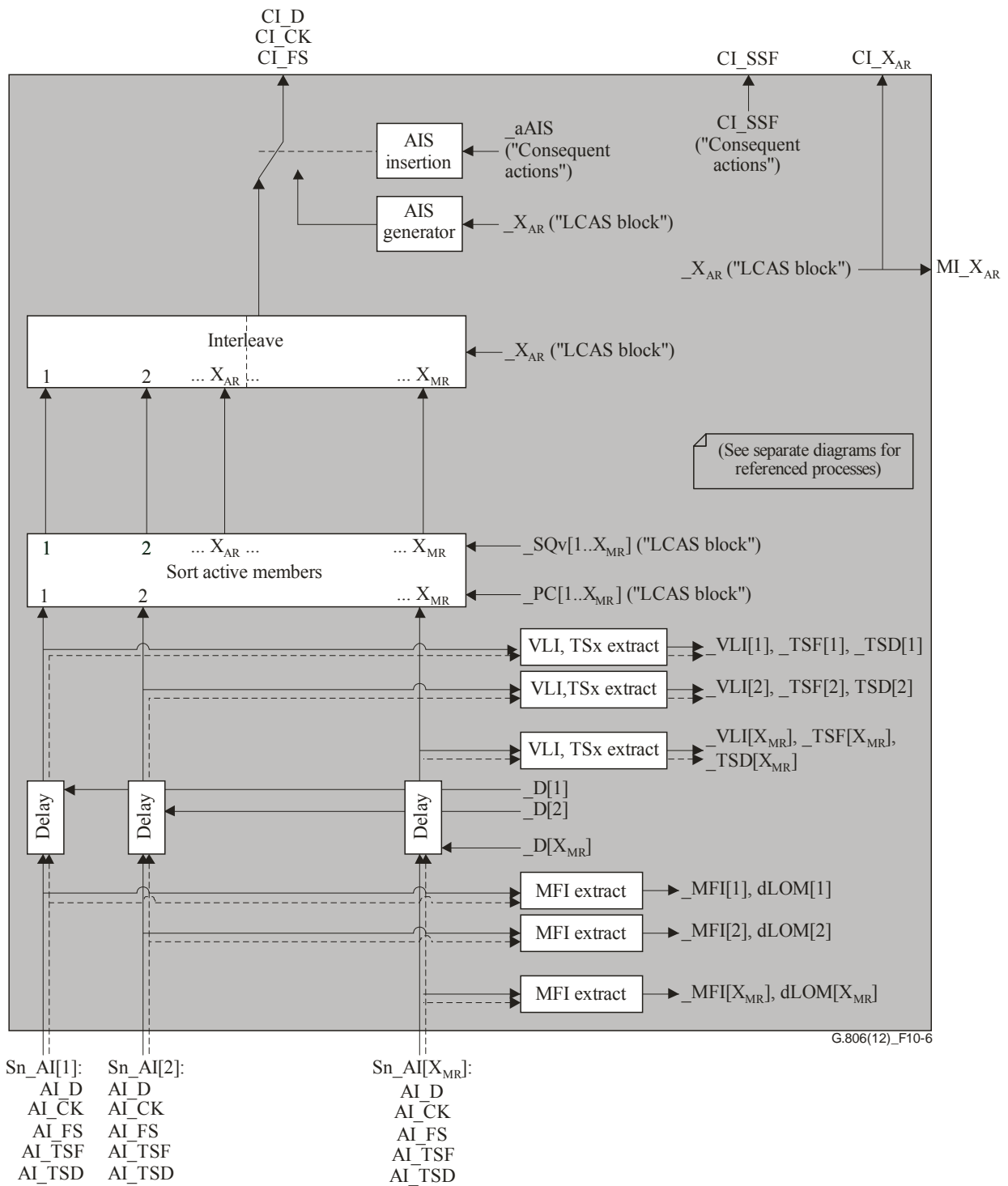
The MI_HOTime input controls whether the hold-off (HO) timer is enabled or disabled for the sink function and, if enabled, what the value of the timer is. If MI_HOTime = 0, the HO timer shall be disabled, if MI_HOTime ≠ 0, it shall be enabled.

The MI_WTRTime input controls whether the wait-to-restore (WTR) timer is enabled or disabled for the sink function and, if enabled, what the value of the timer is. If MI_WTRTime = 0, the WTR timer shall be disabled, if MI_WTRTime ≠ 0, it shall be enabled.

The range of values for the HO/WTR timers is as defined in [ITU-T G.808.1].

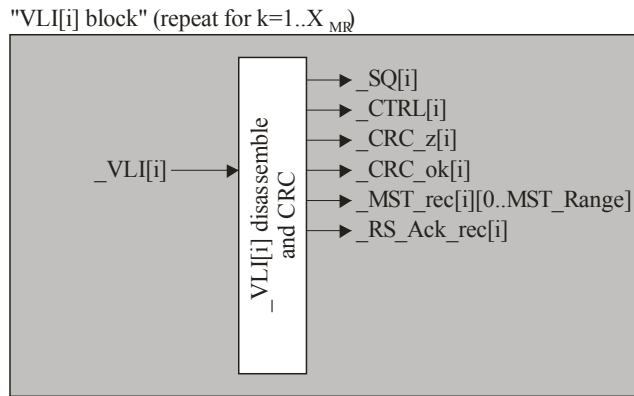
The MI_RMVTime input controls whether the remove (RMV) timer is enabled or disabled for the sink function and, if enabled, what the value of the timer is. If MI_RMVTime = 0, the remove timer shall be disabled, if MI_RMVTime ≠ 0, it shall be enabled.

A process diagram of this function is included in Figures 10-6, 10-7 and 10-8.

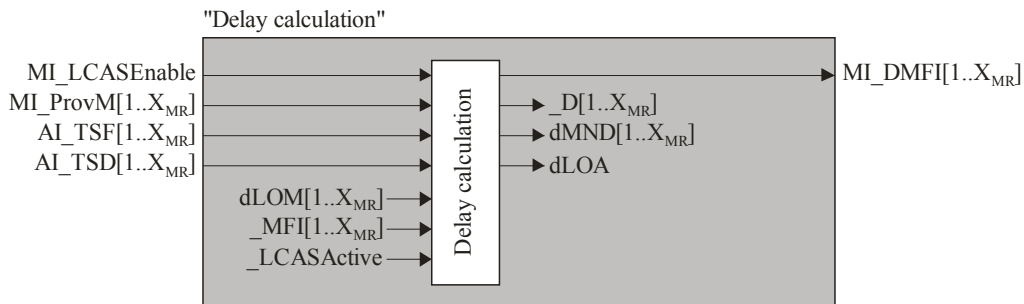


NOTE – The dotted lines represent the TSF and TSD indications.

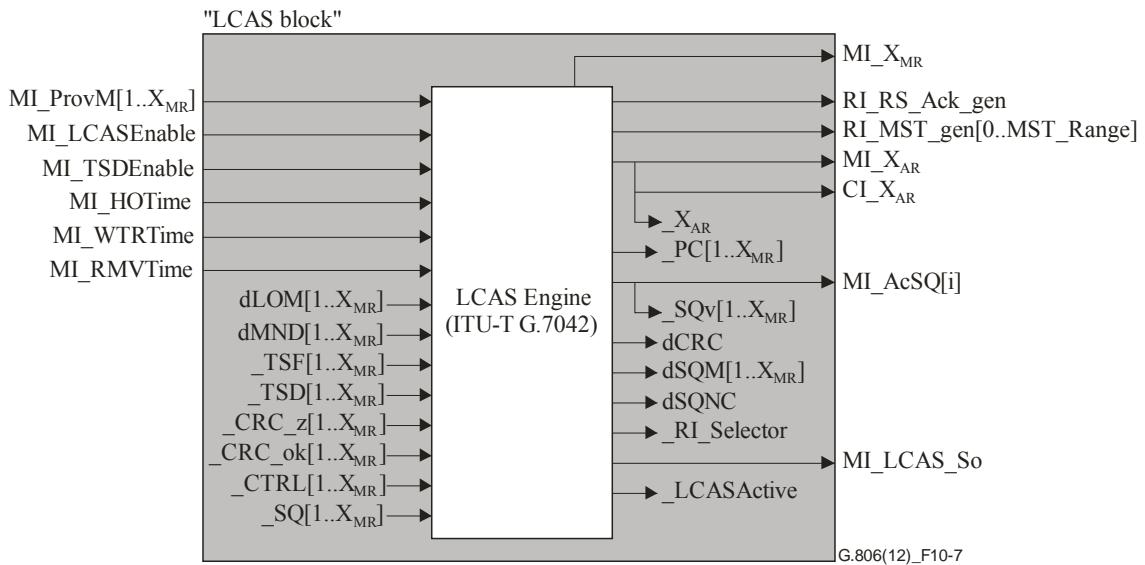
Figure 10-6 – P-Xv/P-X-L_A_Sk processes (main diagram)



NOTE 1 – The internal input `_VLI[i]` is sourced at the "VLI, TSx extract" process.

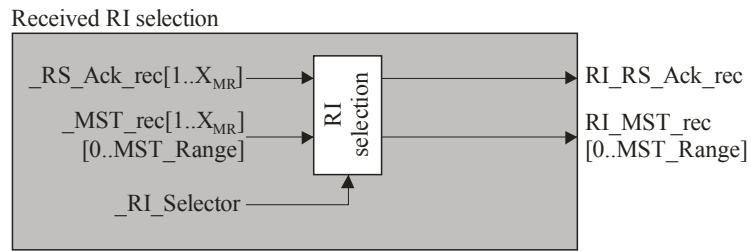


NOTE 2 – The internal inputs `dLOM[i]`, `_MFI[i]` are sourced at the "MFI extract" process.

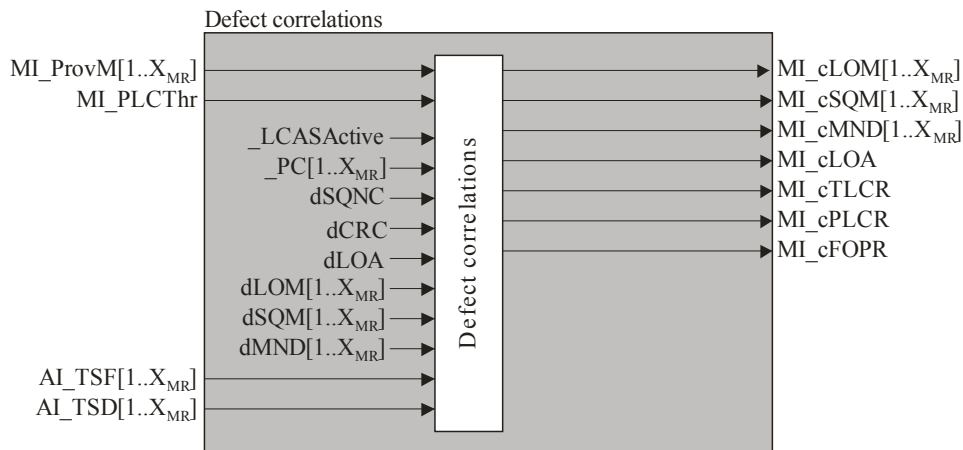


NOTE 3 – The internal inputs `dLOM[i]`, `_MFI[i]` are sourced at the "MFI extract" process, `dMND[i]` at the "delay calculation" process, `_TSF`, `_TSD` at the "VLI, TSx extract" process and `_CRC_z[i]`, `_CRC_ok[i]`, `_CTRL[i]`, `_SQ[i]` at the "VLI disassemble" process.

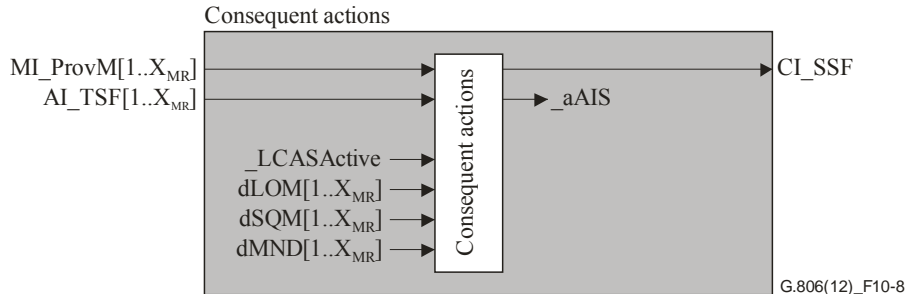
Figure 10-7 – P-Xv/P-X-L_A_Sk processes (subdiagrams I)



NOTE 1 – The internal inputs $_RS_Ack_rec[k]$, $_MST_rec[k]$ are sourced at the "VLI disassemble" process, " $_RI_Selector$ " at the "LCAS block".



NOTE 2 – The internal inputs $_LCASActive$, $_PC$, $dSQNC$, $dCRC$, $dSQM[k]$ are sourced at the "LCAS block", $dLOM[k]$ at the "MFI extract" process, $dMND[k]$, $dLOA$ at the "delay calculation" process.



NOTE 3 – The internal inputs $_LCASActive$, $dSQM[k]$ are sourced at the "LCAS block", $dLOM[k]$ at the "MFI extract" process, $dMND[k]$ at the "delay calculation" process.

Figure 10-8 – P-Xv/P-X-L_A_Sk processes (sub-diagrams II)

MFI extract (Figure 10-6):

This process performs the multiframe alignment, recovers the multiframe information from each of the VCG members and makes it available to the delay calculation process for further processing. The multiframe-alignment process (including the $dLOM[i]$ detection) and the format of the multiframe information are technology-specific.

If $AI_TSF[i] = false$, then the $_MFI[i]$ output of this process is the MFI as contained in the VCAT/LCAS Information (VLI) in $AI_D[i]$.

If $AI_TSF[i] = true$, then the $_MFI[i]$ output of this process shall be an error indication with a technology-specific content.

Delay calculation (Figure 10-7):

This process performs the calculation of the delay for each provisioned member ($_D[i]$), which is necessary in order to multiframe-align all provisioned members. This process also calculates the relative delay ($MI_DMFI[i]$), in MFI units, between each provisioned member and the earliest-arriving member among those considered for the calculation.

The calculation (of both $_D[i]$ and $MI_DMFI[i]$) will be done considering only the multiframe information of the members for which:

$$(MI_ProvM[i] = 1) \text{ AND } (\text{not } AI_TSF[i]) \text{ AND } (\text{not } dLOM[i])$$

is true (i.e., the provisioned members with accepted multiframe information). The output for the members for which the above condition is false shall be: $MI_DMFI[i] = \text{n/a}$ (not available), $_D[i] = \text{ffs}$ (note that it will be important in general that $_D[i]$ for the latter group of members be such that the TSF indication experiences little or no delay across the delay process).

If the delay necessary to multiframe-align all considered members cannot be accommodated by the delay process, this process will:

- If $_LCASActive$ is true, the process will try to determine a subset of them that can be multiframe-aligned. For members in this subset, $_D[i]$ and $MI_DMFI[i]$ will be calculated as described above, and $dMND[i]$ will be clear. For considered members not in this subset, a "member not deskewable" defect ($dMND[i]$, see defects below) shall be declared and $_D[i] = \text{ffs}$, $MI_DMFI[i] = \text{n/a}$. The selection of the subset of members that will be realigned is implementation specific.

NOTE 2 – It is recommended that members carrying a control word of "IDLE" be considered first for exclusion from the subset of members that will be multiframe-aligned.

- If $_LCASActive$ is false, the process will raise the $dLOA$ defect (see defects below).

NOTE 3 – The intention of the above mechanism is to provide, for the $_LCASActive$ case, a certain amount of survivability for the virtual concatenation group in situations where the differential delays among the provisioned members exceed the multiframe-realignment capabilities of an implementation.

NOTE 4 – For the special case in which $X_{PR} = 1$ and $MI_LCASEnable$ is not active, this process shall source $MI_DMFI[i] = 0$, $_D[i] = \text{ffs}^4$ for the member for which $MI_ProvM[i] = 1$, irrespective of the multiframe information and $dLOM[i]$ condition for that member. This is intended to allow interworking with single-member, non-virtual-concatenation source adaptation functions, which generally do not provide virtual concatenation multiframe information.

The delay calculation process shall cover at least a differential delay of 125 μs .

Delay (Figure 10-6):

This process delays each signal by an amount of time indicated by $_D[i]$. The value of $_D[i]$ is calculated by the delay calculation process and is such that all provisioned member signals with a valid multiframe indication are multiframe-aligned at the output of the delay processes.

The delay process shall cover at least a differential delay of 125 μs .

Note that downstream from the delay processes (i.e., from them upward in the diagram), all active members are multiframe-synchronous to each other. This facilitates the hitless addition and removal of members when $MI_LCASEnable$ is active.

⁴ $_D[k]$ will, in general, be implementation-dependent.

VLI, TSx extract (Figure 10-6):

This process extracts the (delayed) VCAT/LCAS information (VLI) from each of the VCG members and makes it available to the VLI disassemble process. It also extracts the TSF, TSD indications from each of the VCG members and makes them available to the LCAS block.

The format of the VLI information is technology-specific.

The outputs `_TSF[i]` and `_TSD[i]` shall be the TSF and TSD indications at the input of the process, respectively.

If `_TSF[i]` is false and `dMND[i]` is false, then the `_VLI[i]` output of this process is the value of the VLI information at the input of this process.

If `_TSF[i]` is true or `dMND[i]` is true, then the `_VLI[i]` output of this process shall be an error indication with a technology-specific content.

VLI disassemble and CRC (Figure 10-7):

This process processes the `_VLI[i]` information. There are X_{MR} instances of this process. Each instance performs the following processing on the corresponding `_VLI[i]`:

- Extract the following information from the `_VLI[i]` from the corresponding multiframe and bit positions:
 - `_CTRL[i]`: Control channel word.
 - `_SQ[i]`: Sequence number.
 - `_MST_rec[i][0..MST_Range]`: Received member status.
 - `_RS_Ack_rec[i]`: Received re-sequence acknowledge.
- Perform the corresponding CRC processing:
 - `_CRC_z[i]`: True if the received CRC word is zero, false otherwise.
 - `_CRC_ok[i]`: Calculate the CRC over the received `_VLI[i]` control packet. Then:
`_CRC_ok[i] = true` if the received CRC word matches the calculated CRC word,
`_CRC_ok[i] = false` otherwise.

NOTE 5 – All processing (including the CRC processing) is performed regardless of the value of `MI_LCASEnable`.

NOTE 6 – The group identifier (GID) bit is not used in the present definition of the sink function.

The structure of the VLI information, including the particular CRC scheme used and the value of any unused space within that structure, is technology-dependent.

"Sort active members" (Figure 10-6):

This process is a switch that connects the active members of the VCG onto fixed signal positions for further processing. Active members are those provisioned members which are carrying payload at a certain point in time, as described above.

The connections are controlled by the `_PC[i]` and the `_SQv[i]` signals ("payload carrying" and "validated sequence number", see LCAS engine process below for definitions).

The connections from input to output are as follows:

- Each active member is exactly connected to one output. Active members are those for which `_PC[i] = 1`.
- All active members are connected to outputs "1" through " X_{AR} ".

- Among the active members, the member with the lowest validated sequence number ($_SQv[i]$) is connected to output "1", the member with the second-lowest validated sequence number is connected to output "2" and so on until the member with the highest validated sequence number, which is connected to output " X_{AR} ".
- The mapping is fixed in time (i.e., does not change over time as long as $_PC$ or $_SQv$ do not change).

For the outputs X_{AR+1} , X_{AR+2} , ..., X_{MR} , this process inserts an all-zeroes signal with the rate and format of a P_AI_D signal.

NOTE 7 – In particular, if only one member has $_PC[i] = 1$, then that input will be connected to output "1", irrespective of the $_SQv[i]$ value. This is relevant in situations where the present sink function is interworking with a single-member, non-virtual-concatenation source function, which does not provide sequence number information.

"Interleave process" (Figure 10-6):

This process recovers the $P-X-L_CI_D$ signal from the X_{AR} path-level signals numbered 1.. X_{AR} at its input. The value of X_{AR} is given by the input $_X_{AR}$, which is itself controlled by the LCAS protocol (see below).

The output to this process is a P_CI signal and includes a P_CI_D , P_CI_CK and a P_CI_FS .

The recovery procedure of $P-X-L_CI_D$ from the X_{AR} path-level signals is technology-specific.

LCAS engine (Figure 10-7):

This process implements the LCAS protocol and related functionality, and provides appropriate control signals for other processes.

The process first calculates the $MI_LCAS_So_Detected$ output as follows:

- 1) $MI_LCAS_So_Detected = \text{false}$ if the following is true:

$$\prod_{MI_ProvM[i]=1} [(_CTRL[i] = \text{FIXED}) \text{ and } _CRC_z[i] \text{ and not } (\text{AI_TSF}[i] \text{ or } dLOM[i] \text{ or } dMND[i])]$$

- 2) $MI_LCAS_So_Detected = \text{true}$ if the following is true:

$$\prod_{MI_ProvM[k]=1} [(_CTRL[i] \neq \text{FIXED}) \text{ and } _CRC_ok[i] \text{ and not } (\text{AI_TSF}[i] \text{ or } dLOM[i] \text{ or } dMND[i])]$$

- 3) Otherwise, $MI_LCAS_So_Detected$ preserves its previous value. The initial value of $MI_LCAS_So_Detected$ is the provisioned value for MI_LCAS_Enable .

NOTE 8 – In other words, the function assumes it is interfacing in a non-LCAS source ($MI_LCAS_So_Detected = \text{false}$) if all provisioned and non-failed members are detected to have a FIXED control word and a zero CRC. It assumes an LCAS source ($MI_LCAS_So_Detected = \text{true}$) if all provisioned and non-failed members have a non-FIXED control word and a correct CRC. Note that there is a hysteresis in the criteria for $MI_LCAS_So_Detected$. This is intended to guard against changing the source assumption without a clear indication.

The process then calculates the internal signal:

$$_LCASActive = MI_LCAS_Enable \text{ and } MI_LCAS_So_Detected$$

If $_LCASActive$ is true, it then produces the following outputs:

- $_PC[1..X_{MR}]$: Indication of whether a particular member is an active member (i.e., is carrying payload) at a particular time. For each index i , the process shall calculate $_PC[i]$ as follows:
 - $MI_ProvM[i] = 0 \rightarrow _PC[i] = 0$

- MI_ProvM[i] = 1 → _PC[i] as determined by the LCAS protocol in [ITU-T G.7042] (= 1 if the protocol determines it is carrying payload, = 0 otherwise)

– _X_{AR}: Current size of the received payload. Note that, with the above definition, _X_{AR} is given by:

$$_X_{AR} = \sum_{i=1}^{X_{MR}} _PC[i]$$

- RI_RS_Ack_gen: (Generated) re-sequence acknowledge bit as defined in [ITU-T G.7042].
- RI_MST_gen[0..MST_Range]: (Generated) member status bits as defined in [ITU-T G.7042]. For each sequence number s, the process shall calculate MI_MST_gen[s] as shown in Annex B.

Note that, according to the definition in [ITU-T G.7042], the index for this output is given by the sequence number. That is MI_MST_gen[0] represents the member status of the member with sequence number 0 (the member i for which _SQv[i] = 0), MI_MST_gen[1] the member status of the member with sequence number 1, and so on.

This is the only output of the LCAS engine process which has this characteristic (to be indexed by the sequence number instead of the P_AI number).

– _SQv[1..X_{MR}]: Validated sequence number. For each index i, the process shall calculate _SQv[i] as follows:

- (MI_ProvM[i] = 0) or ((_TSF[i]) or (dLOM[i]) or (dMND[i])) and (HOTimer not running)) or (_CTRL[i] = IDLE) → _SQv[i] = n/a
- else:
 - _CRC_ok[i] = 1 → _SQv[i] = _SQ[i]
 - _CRC_ok[i] = 0 → _SQv[i] retains its previous value

– _RI_Selector: Member index used to select the remote information set sent to the source (see RI selection process below). _RI_Selector shall be chosen from among those indexes i satisfying:

- (MI_ProvM[i] = 1) and not ((_TSF[i]) or (dLOM[i]) or (dMND[i])) and (_CRC_ok[i] = 1)

If this is an empty set, then _RI_Selector shall be sourced as "n/a".

NOTE 9 – This value is an error indication towards the RI selection process.

Otherwise, the specific choice of _RI_Selector is implementation-specific since the LCAS source sends the same MST values simultaneously in the control packets of all members of a VCG.

NOTE 10 – The LCAS protocol as defined in [ITU-T G.7042] is used in this process to calculate some outputs. The instance of the protocol used here shall have the following characteristics:

- If _CRC_ok[i] = 0, it shall discard all control information for a member i and assume that _CTRL[i] and _SQ[i] are the same as for the previous control packet.
- It shall use the inputs MI_HOTime and MI_WTRTime as the parameters for the hold-off and wait-to-restore timers in the state machines.

If _LCASActive is false, the outputs are as follows:

– _PC[1..X_{MR}]: Indication of whether a particular member is an active member (i.e., is carrying payload) at a particular time. For each index i, the process shall calculate _PC[i] as follows:

- _PC[i] = MI_ProvM[i]

- $_X_{AR}$: Shall be calculated as:

$$_X_{AR} = \sum_{i=1}^{X_{MR}} MI_ProvM[i]$$

NOTE 11 – This definition makes $_X_{AR} = X_{PR}$, for as long as $_LCASActive$ is false.

- $RI_RS_Ack_gen$: (Generated) re-sequence acknowledge bit shall be sourced as a fixed zero.
- $RI_MST_gen[0..MST_Range]$: (Generated) member status bits shall be sourced as all fixed zeroes.
- $_SQv[1..X_{MR}]$: Validated sequence number. $_SQv[i]$ is calculated from $_SQ[i]$ by applying a persistency check. A new received $_SQ[i]$ is forwarded as $_SQv[i]$ if the received sequence has the same value in m consecutive multiframe -1 s, with $3 \leq m \leq 10$.
- $_RI_Selector$: Member index used to select the remote information set sent to the source (see RI selection process below). $_RI_Selector$ shall be calculated as the index i with the lowest $_SQv[i]$ among those satisfying ($_PC[i] = 1$). If this is an empty set, then $_RI_Selector$ shall be sourced as "n/a".

NOTE 12 – This value is an error indication towards the RI selection process.

Received RI selection (Figure 10-8):

This process selects, among the received, the set of remote information that is to be forwarded to the source function. This process also contains a buffer that records the last set of information sent to the source and uses this set in case no valid selector is available.

If ($_RI_Selector \neq n/a$), the outputs are determined by:

- $RI_MST_rec[0..MST_Range] = _MST_rec[_RI_Selector][0..MST_Range]$
- $RI_RS_Ack_rec = _RS_Ack_rec[_RI_Selector]$.

Otherwise, the outputs are sourced from the buffer (i.e., contain the last set of information sent to the source). The initial status of the buffer is an all-zero $_MST_rec[0..MST_Range]$ word and a zero $_RS_Ack_rec$ bit.

NOTE 13 – The purpose of the buffer is to provide the last-known received MST/RS_Ack to the source in the absence of any better information (e.g., in case all the members in the received direction have failed). This mechanism allows the source to continue to transmit without interruption during transient failures affecting the complete set of received path signal(s).

NOTE 14 – The buffer is set to the initial status at the startup of the adaptation function operation.

Defects:

Loss of sequence defect (dSQM[i]):

If $_LCASActive$ is true, then $dSQM[i]$ shall be always cleared.

If $_LCASActive$ is false, $dSQM[i]$ shall be detected if the accepted sequence number (AcSQ) does not match the expected sequence number (ExSQ). $dSQM$ shall be cleared if AcSQ matches ExSQ. The ExSQ of the member at $Sn_AI[i]$ is $i - 1$.

Member not deskewable (dMND[i]):

If $_LCASActive$ is true, then $dMND[i]$ shall be declared for members that have been excluded from multiframe-realignment by the delay calculation process due to the differential delay in the full set of considered members being larger than the realignment capability of the particular instance of the function (see delay calculation process description).

If $_LCASActive$ is false, then $dMND[i]$ shall be always cleared.

Loss of alignment (dLOA):

If `_LCASActive` is true, then dLOA shall be always cleared.

If `_LCASActive` is false, then dLOA shall be detected if the delay calculation process cannot perform the alignment of the considered members to a common multiframe start for any reason (e.g., dLOA is detected if the differential delay exceeds the size of the alignment buffer). The details are ffs.

Persistent CRC errors (dCRC): If k_1 or more CRC errors are detected in n_1 consecutive control packets of any provisioned member, the dCRC defect is declared, provided the CRC and the CTRL word are both unequal to all-zeroes and no `mMSU_L` condition exists for that member.

If k_2 or fewer CRC errors are detected in n_2 consecutive control packets of all provisioned members, dCRC is cleared. The parameters k_1, k_2, n_1, n_2 are ffs.

Inconsistent SQ numbers (dSQNC): Members that carry the "NORM" or "EOS" message are expected to have unique sequence numbers. If this condition is violated, the dSQNC (SQ numbers not consistent) defect is declared. The dSQNC defect is cleared as soon as this condition is satisfied again.

Consequent actions:

`mMSU[i]` ← `MI_ProvM[i]` and (`AI_TSF[i]` or `dLOM[i]` or `dLOA` or `dSQM[i]`)

`mMSU_L[i]` ← `MI_ProvM[i]` and (`AI_TSF[i]` or `dLOM[i]` or `DMND[i]`)

The consequent actions can be defined as:

$$\begin{aligned}
 \text{aAIS} \leftarrow & \left(\text{not } \sum_{i=1}^{X_{MR}} \text{MI_ProvM}[i] \right) \text{or} \\
 & \left((\text{not } _LCASActive) \text{ and } (X_{PR} = 1) \text{ and } \sum_{\text{MI_ProvM}[i]=1} \text{AI_TSF}[i] \right) \text{or} \\
 & \left((\text{not } _LCASActive) \text{ and } (X_{PR} > 1) \text{ and } \sum_{\text{MI_ProvM}[i]=1} \text{mMSU}[i] \right) \text{or} \\
 & \left(_LCASActive \text{ and } \prod_{\text{MI_ProvM}[i]=1} \text{mMSU_L}[i] \right)
 \end{aligned}$$

$$\begin{aligned}
 \text{aSSF} \leftarrow & \left((\text{not } _LCASActive) \text{ and } (X_{PR} = 1) \text{ and } \sum_{\text{MI_ProvM}[i]=1} \text{AI_TSF}[i] \right) \text{or} \\
 & \left((\text{not } _LCASActive) \text{ and } (X_{PR} > 1) \text{ and } \sum_{\text{MI_ProvM}[i]=1} \text{mMSU}[i] \right) \text{or} \\
 & \left(_LCASActive \text{ and } \prod_{\text{MI_ProvM}[i]=1} \text{mMSU_L}[i] \right)
 \end{aligned}$$

NOTE 15 – The summation symbol is used here to denote a logical OR operation over its argument, the product symbol is used here to denote a logical AND operation over its argument⁵.

Note that, for $X_{PR} = 1$ and $_LCASActive$ false, $dLOM[i]$ is not taken into consideration for generating aAIS/aSSF. This allows interworking of this sink function with single path signals sourced by non-virtual-concatenation adaptation functions. See Appendix VII for an example.

The maintenance signal to be output on declaration of aAIS and the time constraints for it are technology-specific. The only general requirement is that the bit rate of this maintenance signal shall be consistent with the value of $_X_{AR}$ as calculated by the processes involved.

Note that, if $_LCASActive$ is true, not all payload discontinuities are signalled with SSF at the output of the P-Xv/P-X-L_A_Sk function. In particular, transient payload discontinuities caused by individual member failures will be signalled back to the source as described in [ITU-T G.7042] but will not trigger aSSF.

Defect correlations:

$cLOM[i] \leftarrow MI_ProvM[i]$ and $dLOM[i]$ and (not $AI_TSF[i]$) and not $((X_{PR} = 1)$ and not $_LCASActive)$

$cMND[i] \leftarrow MI_ProvM[i]$ and $dMND[i]$ and (not $dLOM[i]$) and (not $AI_TSF[i]$)

$cLOA \leftarrow dLOA$ and not $\left(\sum_{MI_ProvM[i]=1} (dLOM[i] \text{ or } AI_TSF[i]) \right)$

$cSQM[i] \leftarrow MI_ProvM[i]$ and $dSQM[i]$ and (not $dLOM[i]$) and (not $dLOA$) and (not $AI_TSF[i]$)

The next two parameters signal the partial loss (PLCR) or total loss (TLCR) of capacity in an LCAS-enabled VCG sink:

$cPLCR \leftarrow _LCASActive$ and $(0 < X_{AR})$ and $(X_{AR} < MI_PLCRThr)$ and $(X_{PR} > 0)$

$cTLCR \leftarrow _LCASActive$ and $(X_{AR} = 0)$ and $(X_{PR} > 0)$

$cFOPR \leftarrow _LCASActive$ and $(dCRC \text{ or } dSQNC)$

Performance monitoring: None.

⁵ Motivation behind the formulae:

- 1) Only the provisioned members are taken into account for aAIS/aSSF calculation.
- 2) aAIS is active if:
 - a) no members are provisioned; or
 - b) (LCAS is inactive and $X_{PR} = 1$ and (that member has AI_TSF)); or
 - c) (LCAS is inactive and $X_{PR} > 1$ and (at least one provisioned member is unavailable)); or
 - d) (LCAS is active and (all provisioned members are unavailable)).
- 3) aSSF is active in the same cases as aAIS, except for the case a, which is considered normal operation for this function (regardless of $MI_LCASEnable$ value).

**10.1.1.3 LCAS-capable virtual concatenated path trail termination source function
P-X-L_TT_So**

Symbol:

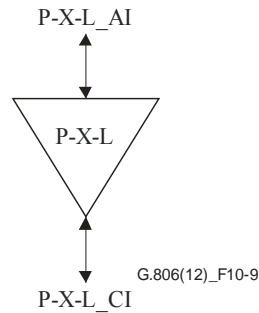


Figure 10-9 – P-X-L_TT_So symbol

Interfaces:

Table 10-3 – P-X-L_TT_So input and output signals

Inputs	Outputs
P-X-L_AP: P-X-L_AI_D P-X-L_AI_CK P-X-L_AI_FS	P-X-L_TCP: P-X-L_CI_D P-X-L_CI_CK P-X-L_CI_FS
P-X-L_TCP: P-X-L_CI_XAT	P-X-L_AP: P-X-L_AI_XAT

Processes: None.

Defects: None.

Consequent actions: None.

Defect correlations: None.

Performance monitoring: None.

**10.1.1.4 LCAS-capable virtual concatenated path trail termination sink function
P-X-L_TT_Sk**

Symbol:

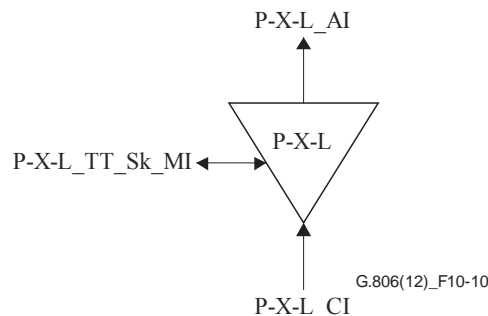


Figure 10-10 – P-X-L_TT_Sk symbol

Interfaces:

Table 10-4 – P-X-L_TT_Sk input and output signals

Inputs	Outputs
P-X-L_TCP: P-X-L_CI_D P-X-L_CI_CK P-X-L_CI_FS P-X-L_CI_SSF P-X-L_CI_XAR	P-X-L_AP: P-X-L_AI_D P-X-L_AI_CK P-X-L_AI_FS P-X-L_AI_TSF P-X-L_AI_XAR
P-X-L_TT_Sk_MP: P-X-L_TT_Sk_MI_SSF_Reported	P-X-L_TT_Sk_MP: P-X-L_TT_Sk_MI_cSSF

Processes: None.

Defects: None.

Consequent actions:

aTSF ← CI_SSF

Defect correlations:

cSSF ← CI_SSF and SSF_Reported

Performance monitoring: None.

Annex A

Assignment and use of signal label, payload type and user payload identifier codes

(This annex forms an integral part of this Recommendation.)

The SDH VC-n signal label (SL), OTN ODUk payload type (PT) and GFP user payload identifier (UPI) codes are used to identify the type and/or composition of the payload data within the VC-n, ODUk and GFP signals respectively.

[ITU-T G.707], [ITU-T G.709] and [ITU-T G.7041] define codes for standardized mappings. To accommodate new mapping development and proprietary mapping schemes, extra codes are reserved for these purposes.

A.1 Experimental code

The experimental code is used to progress development of new payload mappings.

During the development of the mapping, where a standardized code is required but not yet assigned, the experimental mapping code should be used to progress development. As the development matures, a new standardized code shall be requested from ITU-T.

Once the new standard code is assigned to this new mapping and thus replaces the experimental value, the experimental code shall no longer be used for that mapping.

In the case where ITU-T does not accept the new payload mapping for standardization, the vendor/operator intending to deploy the new payload mapping can assign a proprietary code to this new payload mapping.

A.2 Proprietary codes

The proprietary codes allow non-standard payload mappings to be supported. These proprietary codes are not subject to international standardization and can be used by any vendor and/or operator as needed. The selection of a code from any of the reserved proprietary codes is left for the user.

In the case where an initial proprietary payload mapping is accepted, at a later date, by ITU-T as an additional standard payload mapping, a new standardized code will be assigned, thus replacing the original proprietary code. Additionally, if interworking is intended, the proprietary code shall no longer be used for that mapping.

A.3 Request for standardized codes

Standardized codes are administered by ITU-T. A request for new standardized codes shall be made to the study group responsible for the Recommendations which define the codes. The codes are assigned out of the set of codes that are reserved for future standardization when the new mappings are accepted.

The request for new codes shall be supported by a description of the network applications of the new payload mappings and the required functional processing that need to be added to relevant Recommendations.

Annex B

P-Xv/P-X-L_A_Sk: Calculation of RI_MST_gen for _LCASActive = true

(This annex forms an integral part of this Recommendation.)

In the function P-Xv/P-X-L_A_Sk, for _LCASActive = true, the "LCAS engine" process shall calculate RI_MST_gen[0..MST_range] as follows:

- 1) For all sequence numbers s in the set of validated sequence numbers { _SQv[i] }:
→ RI_MST_gen[s] is sourced as determined by the ITU-T G.7042 LCAS protocol.
- 2) For all other sequence numbers:
→ RI_MST_gen[s] = 1 (indicating fail)

NOTE – This definition is motivated as follows:

- a) it covers the members for which the sink can obtain a validated sequence number;
- b) it covers the members for which the sink cannot obtain a validated sequence number.

Appendix I

Connection matrix examples

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

The connection function as defined in clause 5.6.1 is highly flexible, providing full flexibility between its inputs and outputs (see clause I.1). However, the connectivity might be limited due to implementation constraints. Examples are:

- no support of point-to-multipoint connections (broadcast);
- support of bidirectional connections only;
- blocking in a multistage connection matrix;
- no connections within a group of ports (e.g., between add and drop ports of an add/drop matrix) (see clauses I.2, I.3, I.4, I.5, I.6).

If multiplexing is used for the transport of several client signals in a server layer, the client signals have to be assigned to certain address slots (e.g., time slots, frequency/wavelength slots). The address slot assignment is part of the adaptation function to the server layer. An implementation might not support the interchange of address slots of client signals between all or a set of server signals. This is modelled by a connection matrix that allows only connections between ports with identical address slots in the server layer (clauses I.4, I.6).

NOTE – The model assumes that the address slot is only assigned to the client signal along the server layer trail (between adaptation source and adaptation sink) and no address slot is assigned to the client signal outside of this trail. Some signals, however, have this address slot assigned even outside the server layer trail (e.g., wavelength of an optical signal). If the original assignment is performed in the network element itself, possible connections might be modelled, as shown above. If the assignment is, however, done in another network element, possible connections can only be identified from the network view and not locally in the network element itself.

One possibility to represent limited connectivity is to group ports together and define the connectivity between these ports as shown below.

I.1 Connection matrix example for full connectivity

The set of input and output ports is not divided into groups, as shown in Figure I.1. This CM allows full connectivity as given in Table I.1.

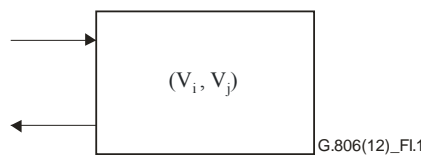


Figure I.1 – Connection matrix example for full connectivity

Table I.1 – Connection matrix example for full connectivity

	V_j
V_i	X
X Indicates V_i - V_j connection possible for any i and j.	

I.2 Connection matrix example for 2-port groups

The set of input and output ports is divided into two groups, each containing both input and output ports, line (L) and tributary (T), as shown in Figure I.2. This CM allows only connectivity between L and T, but not within the L and T group (except for loopbacks) as given in Table I.2.

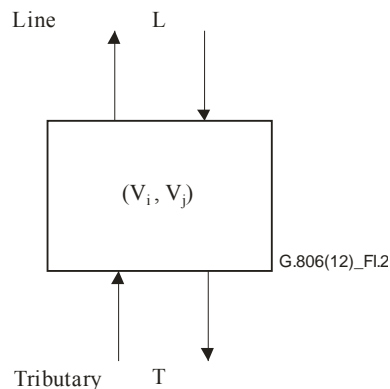


Figure I.2 – Connection matrix example for 2-port groups

Table I.2 – Connection matrix example for 2-port groups

		V_i	
		L	T
V_j	L	$i = j$	X
	T	X	$i = j$

X Indicates V_i - V_j connection possible for any i and j .
 $i = j$ Indicates V_i - V_j connections possible only in the case that $i = j$ (e.g., loopback).

I.3 Connection matrix example for 3-port groups type I

The set of input and output ports is divided into three groups, each containing both input and output ports, west (W), east (E), add/drop (A/D) as shown in Figure I.3. This CM allows connectivity between the groups, but not within the groups as given in Table I.3.

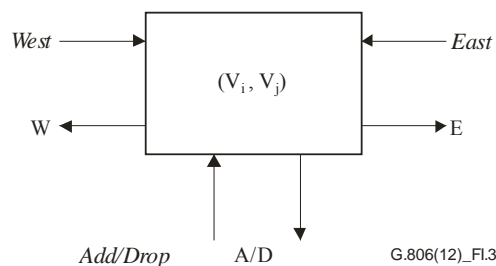


Figure I.3 – Connection matrix example for 3-port groups

Table I.3 – Connection matrix example for 3-port groups type I

		V_i		
		W	E	A/D
V_j	W	–	X	X
	E	X	–	X
	A/D	X	X	–

X Indicates V_i - V_j connection possible for any i and j .
 – Indicates no connection possible.

I.4 Connection matrix example for 3-port groups type II

The set of input and output ports is divided into three groups, each containing both input and output ports, west (W), east (E), add/drop (A/D) as shown in Figure I.3. In addition to the limitations of type I above, connections from W to E and E to W are limited to the same address slot (indicated by identical indices) as shown in Table I.4.

Table I.4 – Connection matrix example for 3-port groups type II

		V_i		
		W	E	A/D
V_j	W	–	$i = j$	X
	E	$i = j$	–	X
	A/D	X	X	–

X Indicates V_i - V_j connection possible for any i and j .
 $i = j$ Indicates V_i - V_j connections possible only in the case that $i = j$ (e.g., no address slot interchange).
 – Indicates no connection possible.

I.5 Connection matrix example for 4-port groups type I

The set of input and output ports is divided into four groups, each containing both input and output ports, west (W), east (E), add/drop east (A/DE) and add/drop west (A/DW) as shown in Figure I.4. This CM allows connectivity between W and E, W and DW and E and EW as given in Table I.5.

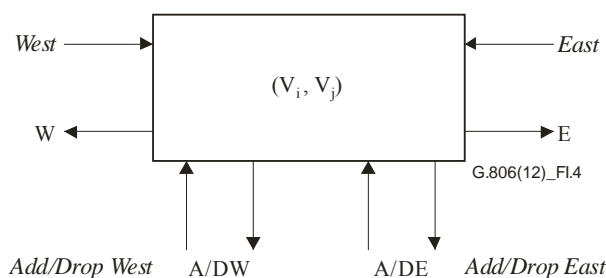


Figure I.4 – Connection matrix example for 4-port groups

Table I.5 – Connection matrix example for 4-port groups type I

		V_i			
		W	E	A/DW	A/DE
V_j	W	–	X	X	–
	E	X	–	–	X
	A/DW	X	–	–	–
	A/DE	–	X	–	–

X Indicates V_i - V_j connection possible for any i and j.
 – Indicates no connection possible.

I.6 Connection matrix example for 4-port groups type II

The set of input and output ports is divided into four subsets, each containing both input and output ports, west (W), east (E), add/drop east (A/DE) and add/drop west (A/DW) as shown in Figure I.4. In addition to the limitations of type I above, connections from W to E and E to W are limited to the same address slot (indicated by identical indices) as given in Table I.6.

Table I.6 – Connection matrix example for 4-port groups type II

		V_i			
		W	E	A/DW	A/DE
V_j	W	–	$i = j$	X	–
	E	$i = j$	–	–	X
	A/DW	X	–	–	–
	A/DE	–	X	–	–

X Indicates V_i - V_j connection possible for any i and j.
 $i = j$ Indicates V_i - V_j connections possible only in the case that $i = j$ (i.e., loopback, no reconfiguration).
 – Indicates no connection possible.

I.7 Example of a provisioned connection matrix

Table 1.7 shows an example of a provisioned connection matrix with unconnected connection points and unprotected, 1+1 SNC/I protected, 1+1 SNC/N protected, unidirectional and bidirectional matrix connections.

Table I.7 – Example of a provisioned connection matrix

Connection inputs ID	Connection outputs ID	Traffic direction	Protection
id #01	–	–	–
id #25	–	–	–
id #65	id #52	Unidirectional	Unprotected
id #91	id #22	Bidirectional	Unprotected
id #69	(N: id #88, P: id #35)	Unidirectional	1+1 SNC/N
(N: id #88, P: id #35)	id #69	Unidirectional	1+1 SNC/N
id #03	(N: id #11, P: id #13)	Bidirectional	1+1 SNC/N

Table I.7 – Example of a provisioned connection matrix

Connection inputs ID	Connection outputs ID	Traffic direction	Protection
id #77	(N: id #88, P: id #35)	Unidirectional	1+1 SNC/I
(N: id #09, P: id #51)	id #42	Unidirectional	1+1 SNC/I
id #10	(N: id #56, P: id #15)	Bidirectional	1+1 SNC/I
...			

NOTE 1 – In order to simplify the content of this table, the connection inputs and outputs are simply identified by an identifier number (id #). Refer to [ITU-T G.7710] and the corresponding technology-specific EMF Recommendations for the correct identification.

NOTE 2 – The notation (N: xxx, P: yyy) identifies the normal and protection trails in case of SNC protection.

I.8 Connection matrix example for 3-port groups (address slot group switching)

The set of input and output ports is divided into three groups, each containing both input and output ports, west (W), east (E), add/drop (A/D) as shown in Figure I.3. In addition to the limitations of type I above, connections from W to E and E to W are limited to the same address slot groups as shown in Table I.8.

Table I.8 – Connection matrix example for 3-port groups (address slot group switching)

		V_i		
		W	E	A/D
V_j	W	–	Y	X
	E	Y	–	X
	A/D	X	X	–

X Indicates V_i - V_j connection possible for any $(i_1, i_2, i_3 \dots i_N)$ and $(j_1, j_2, j_3 \dots j_N)$.

Y Indicates V_i - V_j connections possible only in the case that $(i_1, i_2, i_3 \dots i_N) = (j_1, j_2, j_3 \dots j_N)$ (e.g., waveband switching).

– Indicates no connection possible.

I.9 Connection matrix example for 4-port groups (address slot group switching)

The set of input and output ports is divided into four subsets, each containing both input and output ports, west (W), east (E), add/drop east (A/DE) and add/drop west (A/DW) as shown in Figure I.4. In addition to the limitations of type I above, connections from W to E and E to W are limited to the same address slot groups as shown in Table I.9.

Table I.9 – Connection matrix example for 4-port groups (address slot group switching)

		V_i			
		W	E	A/DW	A/DE
V_j	W	–	Y	X	–
	E	Y	–	–	X
	A/DW	X	–	–	–
	A/DE	–	X	–	–
X	Indicates V_i - V_j connection possible for any $(i_1, i_2, i_3 \dots i_N)$ and $(j_1, j_2, j_3 \dots j_N)$.				
Y	Indicates V_i - V_j connections possible only in the case that $(i_1, i_2, i_3 \dots i_N) = (j_1, j_2, j_3 \dots j_N)$ (e.g., waveband switching).				
–	Indicates no connection possible.				

Appendix II

Example of remote indication operation

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

In order to support single-ended operation, the defect status and the number of detected error code violations of the characteristic information monitored at the trail termination sink shall be conveyed back to the far-end trail termination source (via RDI and REI signals). Hence, in the case where the terminations lie in the domains of different operators, the operations systems (OSs) in both networks will have access to performance information from both trail ends, without the need for OS-to-OS information exchange.

II.1 Remote defect indication (RDI)

RDI signals convey the defect status of the trail signal at the trail destination (i.e., at the trail termination sink function) back to the trail origin (i.e., trail termination source function). This mechanism allows alignment of the near-end and far-end performance monitoring processes.

Examples of RDI signals are the RDI bits in SDH signals, the A-bit in ITU-T G.704 structured 2 Mbit/s signals and the alarm indication bit in other PDH multiplex signals.

Figure II.1 illustrates the RDI insertion and detection/processing for a multiplex section. Figure II.2 illustrates the process for a VC-4 Path:

- At node A, the near-end information represents the performance of the unidirectional section/path from B to A, while the far-end information represents the performance of the unidirectional section/path from A to B.
- At node B, the near-end information represents the performance of the unidirectional section/path from A to B, while the far-end information represents the performance of the unidirectional section/path from B to A.

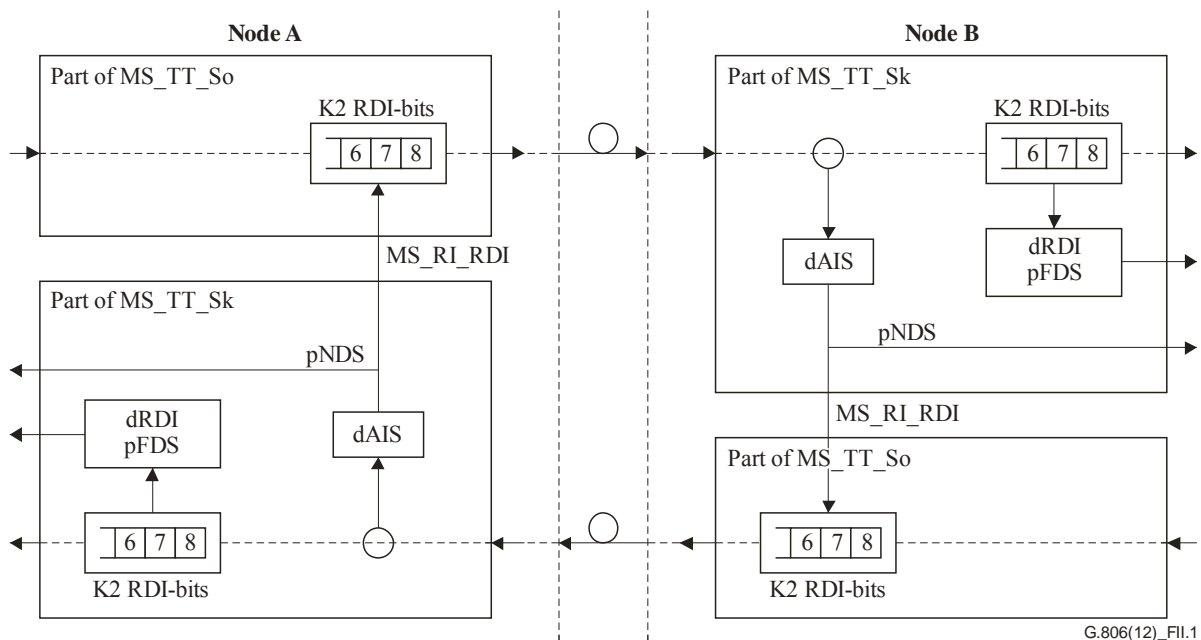


Figure II.1 – RDI insertion control example (multiplex section)

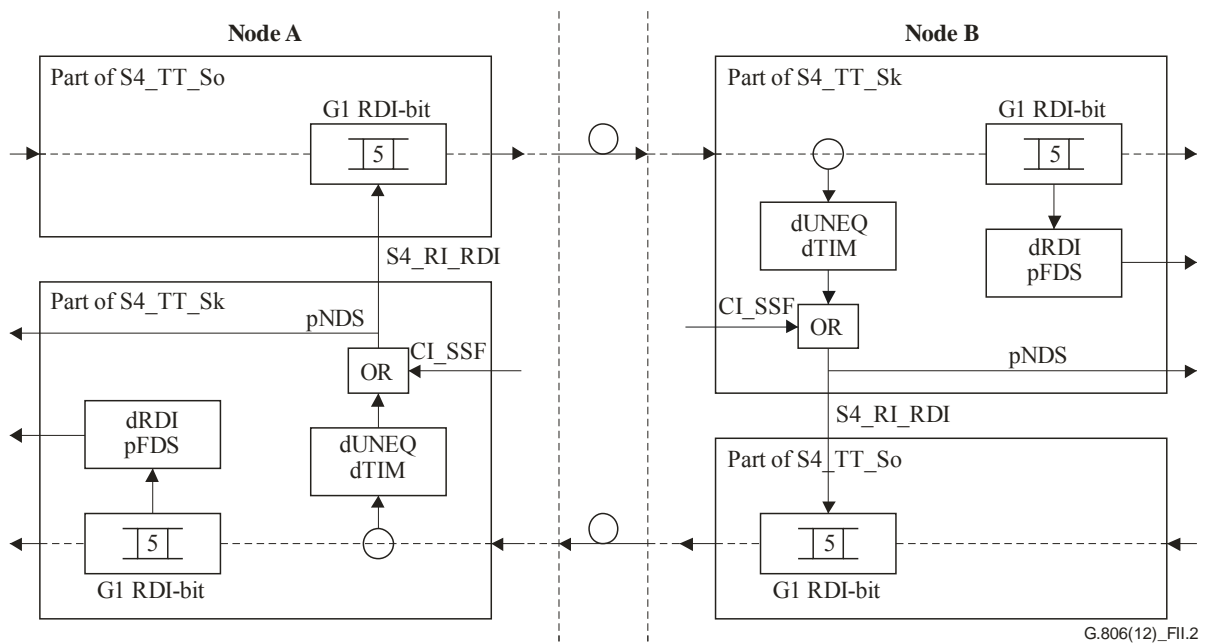


Figure II.2 – RDI insertion control example (VC-4 path)

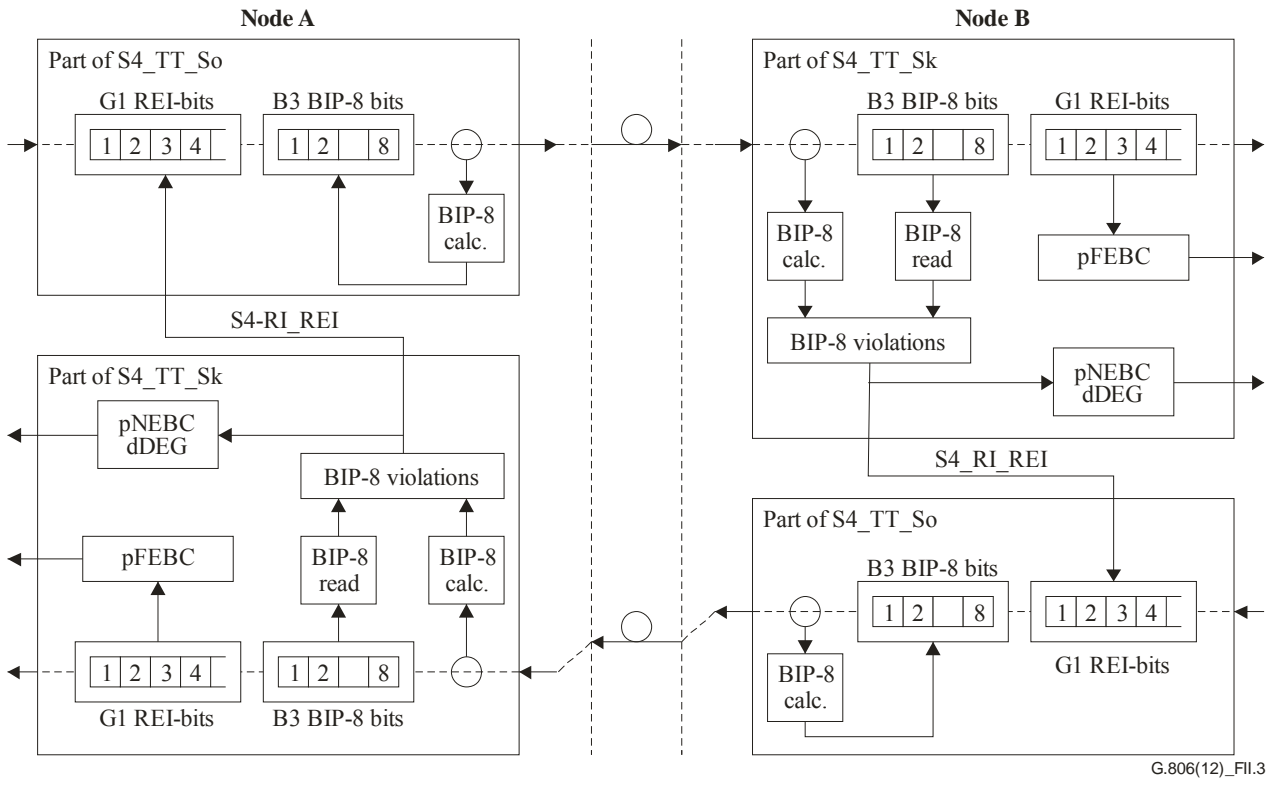
II.2 Remote error indication (REI)

REI signals contain either the exact or truncated⁶ number of error detection code violations detected in the trail signal at the trail termination sink. This information is conveyed to the trail termination source. This mechanism allows alignment of the near-end and far-end performance monitoring processes. Examples of REI signals are the REI bits in SDH signals and the E-bit in ITU-T G.704 structured 2 Mbit/s signals.

Figure II.3 illustrates the REI insertion and extraction/processing for a VC-4 bidirectional path:

- At node A, the near-end information represents the performance of the unidirectional path from B to A, while the far-end information represents the performance of the unidirectional path from A to B.
- At node B, the near-end information represents the performance of the unidirectional path from A to B, while the far-end information represents the performance of the unidirectional path from B to A.

⁶ Refer to the specific atomic functions to determine between exact or truncated number of EDCV transport in the REI.



G.806(12)_Fil.3

Figure II.3 – REI insertion control example (VC-4 path)

Appendix III

Alarm indication signal (AIS)

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

The AIS is an all-ones characteristic or adapted information signal. It is generated to replace the normal traffic signal when it contains a defect condition in order to prevent consequential downstream failures being declared and alarms being raised.

All-ones (AIS) insertion in the sink direction is controlled as follows: every atomic function inserts all-ones on locally detected defects only, with one of the defects being incoming AIS from upstream atomic functions.

Figure III.1 illustrates this process. Due to a LOF defect (STM1dLOF), the OS1/RS1_A_Sk inserts the all-ones signal. This signal is propagated through the RS1 layer. The MS1/TT_Sk detects this all-ones signal by monitoring bits 6-8 of K2. The MS1/S4_A_Sk detects the all-ones signal by monitoring the pointer bytes H1, H2. As a consequence, both functions insert all-ones at their outputs (i.e., they "refresh" the all-ones signal). This behaviour is continued in the other client layers.

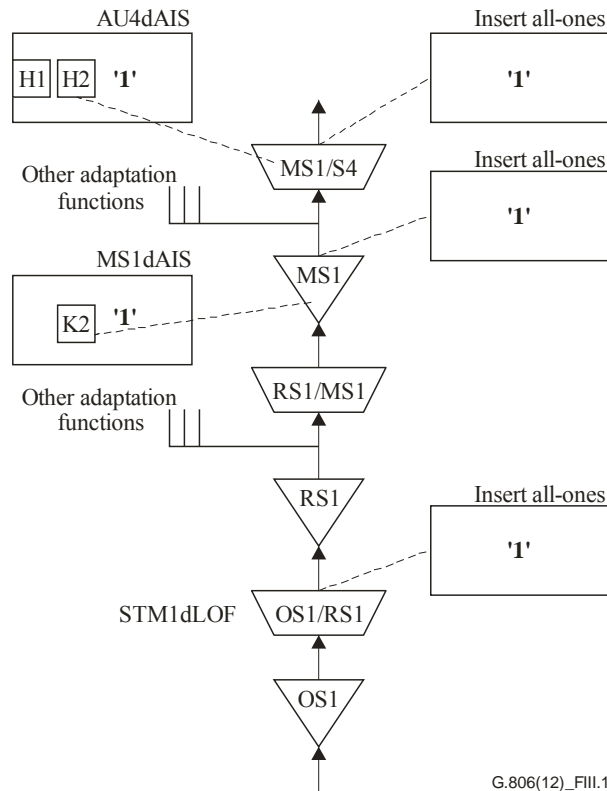


Figure III.1 – All-ones (AIS) insertion and propagation in the sink direction in case of STM1dLOF

As soon as the direction through the layered structure reverts from the sink direction into the source direction, the all-ones (AIS) signal becomes one of the defined AIS patterns:

- MS_n-AIS ($n = 1,4,16$) in case the RS_n/MS_n_A_Sk is connected to the RS_n/MS_n_A_So. This is the case in an STM- n regenerator.
- AU-4-AIS in case the MS_n/S4_A_Sk is connected to the MS_n/S4_A_So. This is the case in a VC-4 add-drop multiplexer and a VC-4 digital cross-connect (Figure III.2).

- TUm-AIS (m = 12,2,3) in case the S4/Sm_A_Sk is connected to the S4/Sm_A_So. This is the case in a VC-m ADM and a VC-m DXC.
- PDH AIS: Ex-AIS, a complete all-ones signal, in the ITU-T G.703 type signal.

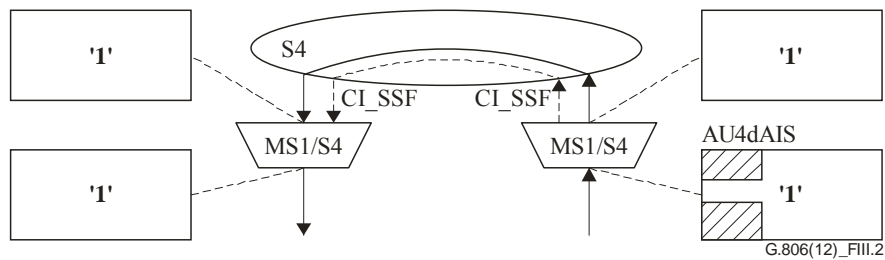


Figure III.2 – All-ones propagation from sink to source direction

The all-ones and CI_SSF signal applied at the input of the MS1/S4_A_So (Figure III.3) results in the generation of an all-ones signal at the output. The MS1_TT_So and the other MS1 adaptation functions (e.g., MS1/OW_A_So) add the MSOH to the all-ones signal. The RS1_TT_So and the RS1 adaptation functions add the RSOH. The result is the so-called AU-4 AIS signal. This signal is transmitted to the far-end. The STM-1 signal passes through the functions up to the MS1_TT_Sk. Then the MS1/S4_A_Sk function detects AU-4 AIS. It declares the AU4dAIS defect and inserts all-ones at its output.

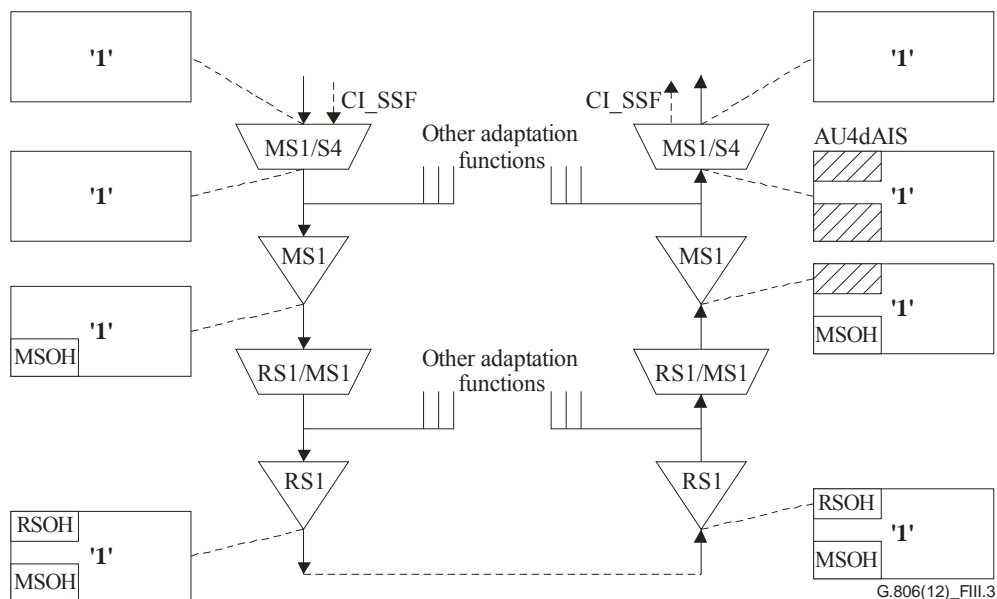


Figure III.3 – All-ones (AIS) generation in the source and detection in the sink direction

Similarly, the reception of an all-ones signal at the S4/S12_A_So results in the generation of an all-ones (TU) signal at the output of the function. This signal is multiplexed with the other TUs, after which the VC-4 overhead, AU-4 pointer, MSOH and RSOH are added. The result is an STM-N signal with a TU carrying TU-AIS.

Appendix IV

Signal fail (SF) and signal degrade (SD)

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

IV.1 Server signal fail (SSF) signal

The CI_SSF signal (generated by the adaptation sink function under control of aSSF) informs the next downstream function of the "signal fail" condition of the associated data signal (which contains, due to that "signal fail" condition, the all-ones (AIS) pattern).

The CI_SSF signal, when connected to a connection function with protection functionality, represents the signal fail (SF) conditions.

IV.2 Server signal degrade (SSD) signal

The CI_SSD signal informs the next downstream function of the "signal degrade" condition of the associated data signal.

The CI_SSD signal is defined only in the adaptation sink function in protection sublayers. The signal relays the AI_TSD signal generated by the trail termination sink function towards the protection connection function in the protection sublayer.

IV.3 Trail signal fail (TSF) signal

The AI_TSF signal (generated by a trail termination sink function under control of aTSF) informs the next downstream function(s) of the "signal fail" condition of the associated data signal (which contains, due to that "signal fail" condition, the all-ones (AIS) pattern).

The AI_TSF signal, when connected to a connection function with protection functionality, represents a signal fail (SF) condition.

IV.4 Trail signal degrade (TSD) signal

The AI_TSD signal (generated by a trail termination sink function under control of aTSD) informs the next function(s) of the "signal degrade" condition of the associated data signal.

The AI_TSD signal is only connected to a connection function with protection functionality, and represents the signal degrade (SD) conditions.

Appendix V

Description of error detection code (EDC) terminology $N \times \text{BIP-m}$

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

The notation BIP- X is defined in [ITU-T G.707]. It refers only to the EDC, i.e., the number of BIP bits, and not to the EDC usage (i.e., what quantities are counted). This appendix describes the term EDC usage and discusses the distinction between the notation $N \times \text{BIP-m}$ and BIP- X . It will be seen that if the EDC usage for BIP- X is $N \times \text{BIP-m}$, then $X = mN$.

To define the EDC usage and the notation $N \times \text{BIP-m}$, divide the X BIP bits into N groups of m bits each, as shown in Figure V.1. This figure illustrates the general case of BIP- X , with $X = mN$. Each BIP parity bit is a parity check over a set of A/X bits of the frame for which the error supervision is performed, where A is the number of bits in the frame. Each such set of bits is referred to as a thread, and there are a total of $X = mN$ threads. Divide this set of X threads into N subsets, where each subset contains m threads as shown in the left-hand portion of Figure V.1. Furthermore, let the BIP bits corresponding to the threads in each subset be consecutive, as shown in the right-hand portion of Figure V.1. Each subset containing m threads is referred to as a block. The number of bits in a block is A/N , and there are N blocks in the payload portion of one frame. In Figure V.1, the k th block consists of all the bits of all the group k 's.

With the above terminology, we define a $N \times \text{BIP-m}$ EDC usage as a set of N counters, with each counter corresponding to one of the subsets of m threads. Each counter is incremented by 1 when one or more of the m threads in the subset has a parity violation. This is illustrated in the right-hand portion of Figure V.1.

To see that the above partitioning may be done for BIP- X with any integers m and N and $mN = X$, consider Figure V.2. This figure shows the successive X -bit sequences, with each X bit sequence consisting of N Groups of m bits each and $mN = X$. In each sequence, bit j of Group k is shown. This bit is the $[(k-1)m + j]$ th bit in the sequence. The BIP bit corresponding to this is the $[(k-1)m + j]$ th bit in the BIP- X , which is also bit j of group k in the BIP- X . This BIP bit is calculated so that the modulo 2 sum of it and bit j of group k of all the other X -bit sequences is zero. It is seen that, regardless of how we do the partitioning, the $[(k-1)m + j]$ th bit of an X -bit sequence is always bit j of group k of the sequence; the same holds for the BIP- X . Therefore, the BIP- X bits can be calculated at the transmitter independently of the values of m and N , as long as $mN = X$.

As an example, consider the case where we have a single BIP byte ($X = 8$) and count individual code violations. This example is shown in Figure V.3. Here, there are 8 counters, and the EDC is referred to as $8 \times \text{BIP-1}$. In this case, each thread forms a block. As another example, consider the case where we have a single BIP byte ($X = 8$), but treat the entire frame area as a block and have a single counter (which is incremented when one or more of the 8 threads have a parity violation). This EDC is referred to as $1 \times \text{BIP-8}$ and is shown in Figure V.4. As additional examples, note that the EDC for VC-4- X c paths, which uses a single B3 byte, is $1 \times \text{BIP-8}$; and the EDC for STM- N MS, which uses $3N$ B2 bytes, is $24N \times \text{BIP-1}$. The $N \times \text{BIP-m}$ notation is consistent with notation used in [b-ITU-T G.828], [b-ITU-T G.829] and [b-ITU-T G.8201].

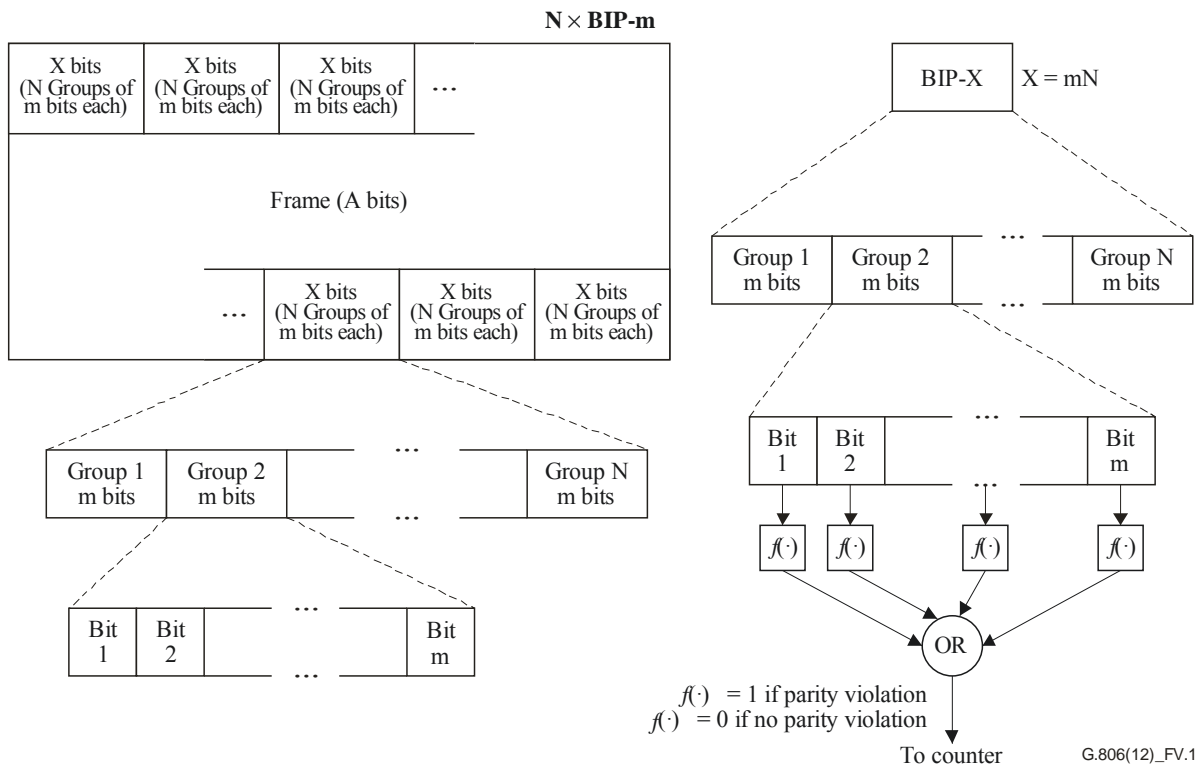


Figure V.1 – Illustration of $N \times \text{BIP-m}$ EDC usage

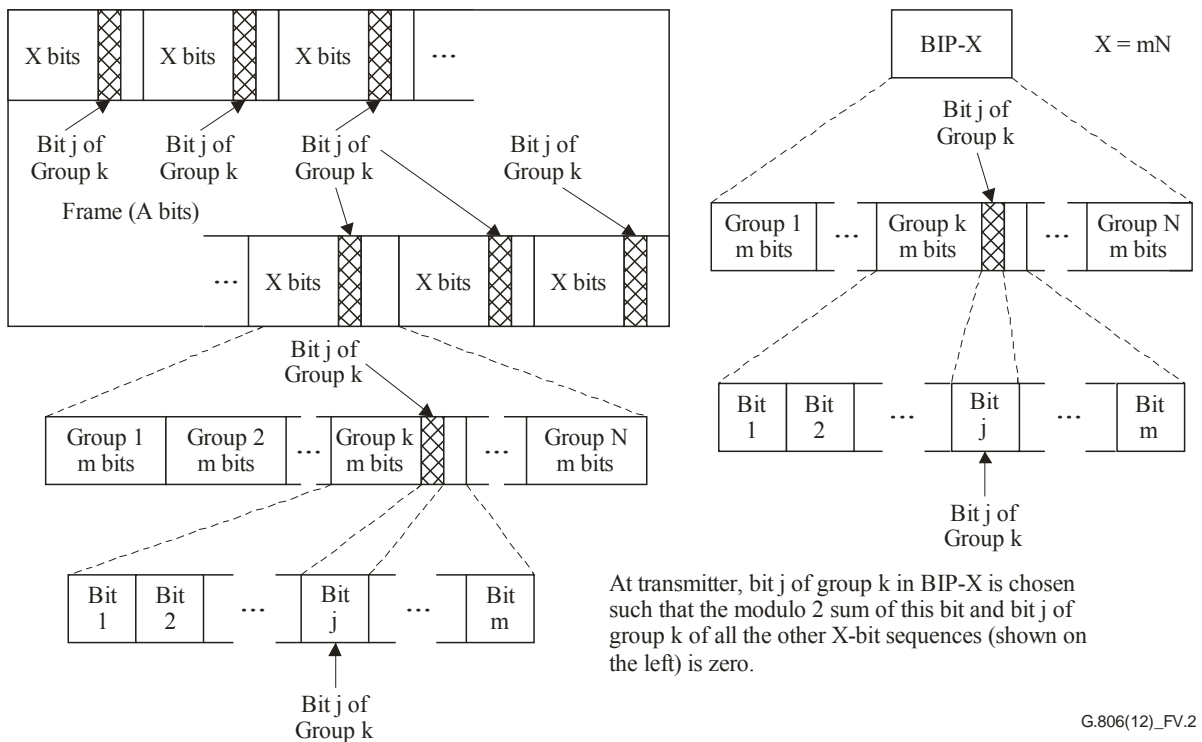


Figure V.2 – Illustration of BIP- X calculation and $N \times \text{BIP-m}$ EDC usage, showing that BIP- X calculation at transmitter is independent of m and N

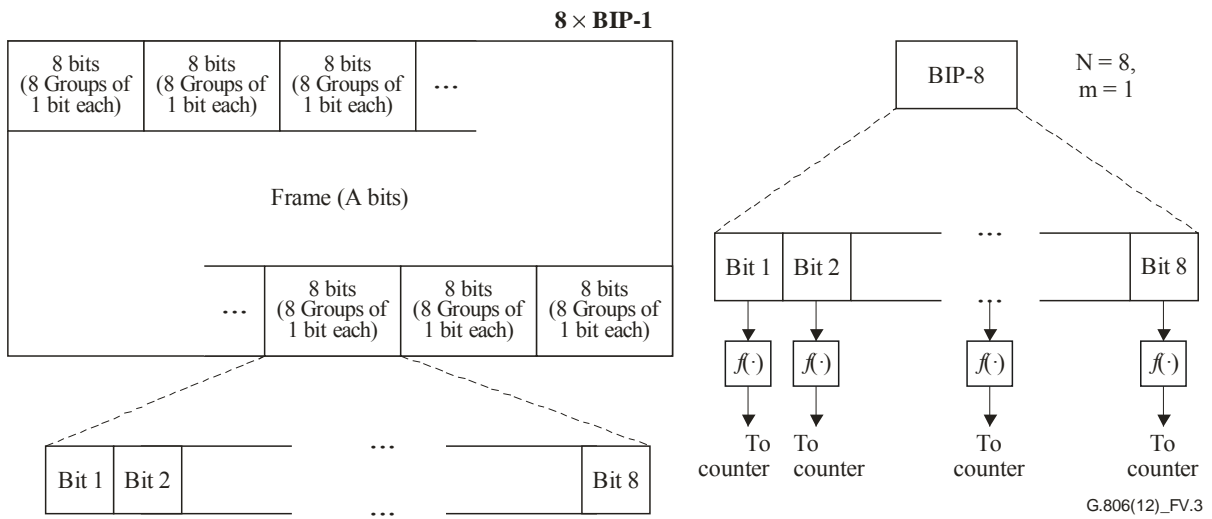


Figure V.3 – Illustration of 8 × BIP-1 EDC usage

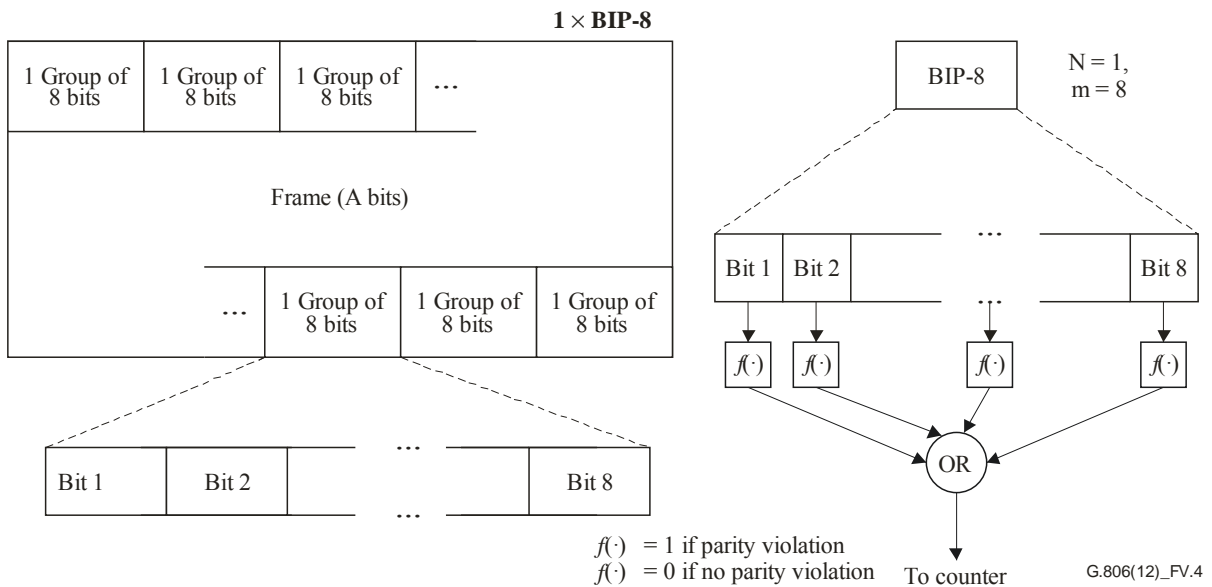


Figure V.4 – Illustration of 1 × BIP-8 EDC usage

Appendix VI

Calculations leading to BIP saturation results in Tables 6-4 and 6-5

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

VI.1 Introduction

Tables 6-4 and 6-5 indicate the average BER values where error detection has saturated for various VC-n and VC-4-Xc. This appendix provides the detailed calculations leading to these results.

VI.2 Calculations and results

Table VI.1 shows, for each VC-n and VC-4-Xc, the EDC usage and block size in bits. The block sizes are taken from Table B.1 of [b-ITU-T G.828]. Table VI.1 also shows the thread size in bits. For EDC usage of $N \times \text{BIP-m}$, the number of threads is equal to Nm (see Appendix V for details on this and also for the definition of the term *thread*) and, therefore, the thread size is equal to the block size divided by Nm .

Let p be the average bit error ratio (BER), and assume the errors are random, i.e., Poisson. Then p is the probability that any bit will be errored. Let the number of bits in a thread be n . Then the probability that the thread will be detected as being errored, $P_{th,det}$ is equal to the probability that the number of bit errors in the thread is odd, i.e.,

$$P_{th,det} = \sum_{k=0}^{2k+1 \leq n} \binom{n}{2k+1} p^{2k+1} (1-p)^{n-2k-1} \quad (\text{VI-1})$$

A closed form expression for this sum is given in reference [b-Cornaglia] as

$$P_{th,det} = \frac{1 - (1 - 2p)^n}{2} \quad (\text{VI-2})$$

NOTE – This result may be derived by:

- 1) writing the binominal series for the expression $(x + y)^n - (x - y)^n$ and noting that the terms involving even powers cancel;
- 2) substituting $x = 1 - p$ and $y = p$; and
- 3) noting that the first sum is 1 and the second sum is $(1 - 2p)^n$.

Equation VI-2 is plotted in Figure VI.1 as a function of average BER, p , for each of the thread sizes n given in Table VI.1. The VC-n and VC-4-Xc paths shown in the legend correspond to the successive curves starting with the curve of lowest errored thread probability values. The saturation BER is defined as the BER where an extension of the sloped portion of the respective curve meets the asymptote at 0.5. This BER may be obtained by expanding equation VI-2 to first order in p , setting $P_{th,det}$ equal to 0.5, and solving for p . The result is

$$p_{sat} = \frac{1}{2n} \quad (\text{VI-3})$$

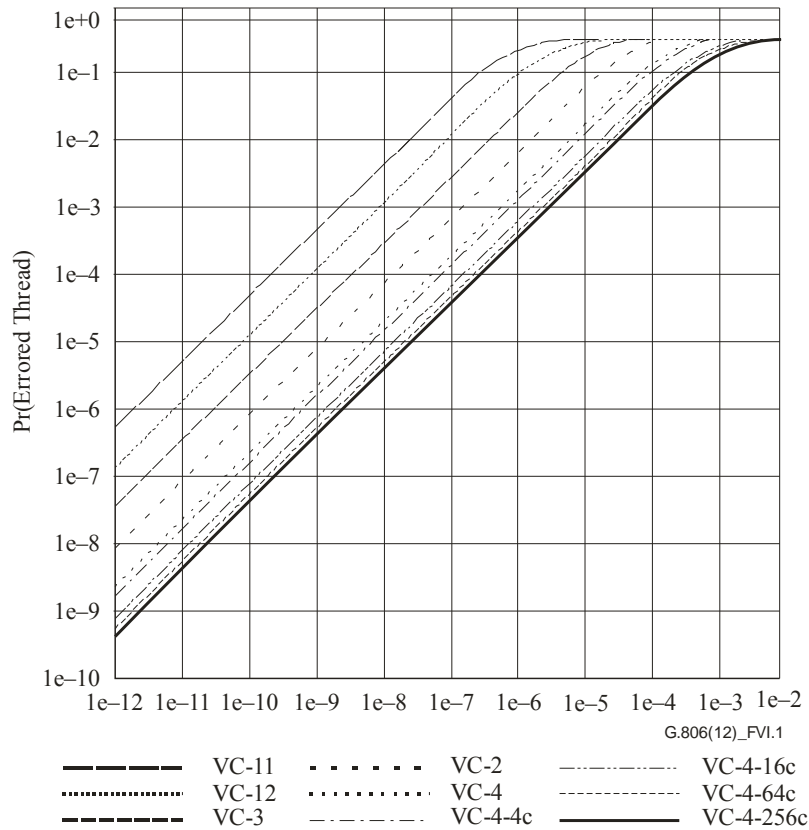
where p_{sat} denotes the saturation BER. Using this definition, the saturation BER for each VC-n and VC-4-Xc is obtained and given in Table VI.1.

Saturation of the BIP means that almost every thread contains bit errors. When saturation occurs, approximately half the errored threads will be detected (i.e., the number of BIP violations will be one half the total number of BIP bits during the measurement interval) because it is approximately equally likely that a thread will contain an even or odd number of errors (an even number of errors does not produce a BIP violation).

Table VI.1 – EDC usage, block size, and thread size for VC-n and VC-4-Xc

Path	EDC usage	Block size (bits) (Note)	Thread size (bits)	Saturation BER
VC-11	1 × BIP-2	832	416	1.2×10^{-3}
VC-12	1 × BIP-2	1120	560	8.9×10^{-4}
VC-2	1 × BIP-2	3424	1712	2.9×10^{-4}
VC-3	1 × BIP-8	6120	765	6.5×10^{-4}
VC-4	1 × BIP-8	18792	2349	2.1×10^{-4}
VC-4-4c	1 × BIP-8	75168	9396	5.3×10^{-5}
VC-4-16c	1 × BIP-8	300672	37584	1.3×10^{-5}
VC-4-64c	1 × BIP-8	1202688	150336	3.3×10^{-6}
VC-4-256c	1 × BIP-8	4810752	601344	8.3×10^{-7}

NOTE – See Table B.1 of [b-ITU-T G.828].



NOTE – The VC-n and VC-4-Xc paths indicated in the legend correspond to the successive curves starting with the curve of lowest errored thread probability values.

Figure VI.1 – Errored thread probability as a function of average BER, assuming random (Poisson) errors

Appendix VII

Examples for the operation of the processes within LCAS-capable adaptation functions

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

This appendix shows some examples for the operation of the processes within LCAS-capable adaptation functions. These examples are intended to illustrate the dynamic interactions among the processes, and between the processes and the LCAS protocol.

VII.1 Basic configuration

The configuration in Figure VII.1 is used as the basis for the examples in this appendix.

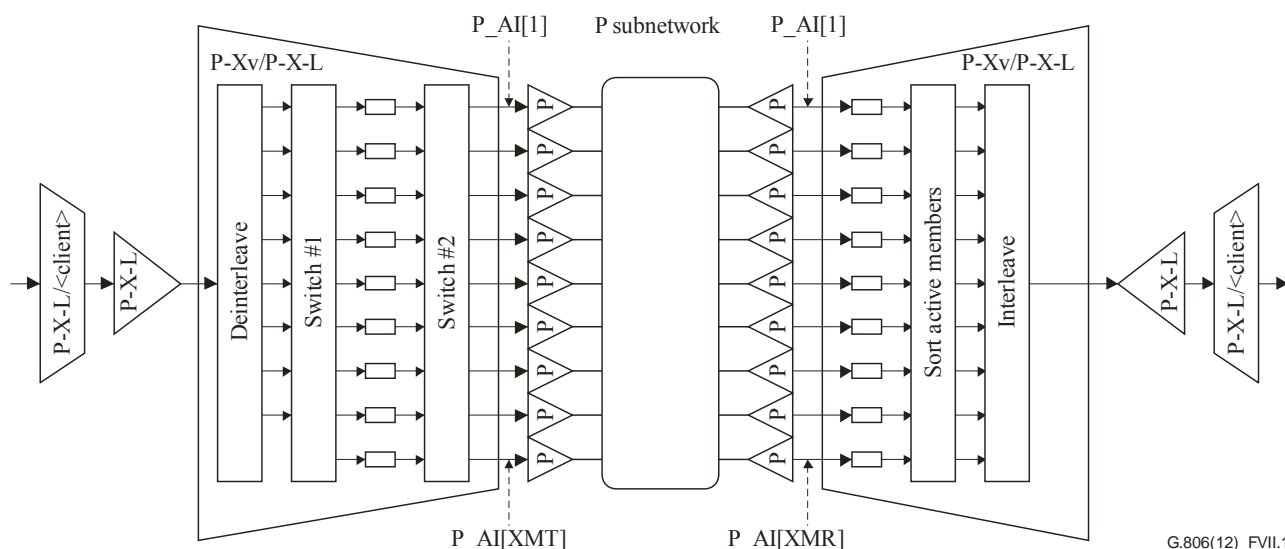


Figure VII.1 – Basic configuration

In this configuration, one direction of transmission is illustrated encompassing an LCAS-capable source and an LCAS-capable sink function. For the LCAS-capable functions, some internal process details are shown. Also in the figure, a P subnetwork is shown, representing the path-layer connectivity between the P_TT functions.

If no connectivity (arrow) is shown in the P subnetwork towards a P_TT sink function, the function will be assumed to be receiving an unequipped signal.

To particularize the examples, the following parameters will be assumed: $X_{MT} = 9$, $X_{MR} = 9$.

VII.2 LCAS-enabled So and Sk functions

Some basic scenarios are shown in this clause for a link in which MI_LCASEnable is active in both source and sink adaptation functions.

VII.2.1 Scenario 1: Establish a 3-member VCG

This scenario considers establishing a 3-member VCG from scratch. There are three things that need to be done to achieve this goal: provisioning the path-layer connectivity, provisioning the source to use the three members and provisioning the sink to use the three members.

These three steps can be taken in any order and the operation will be hitless for the client signal, irrespective of the order. This scenario will illustrate the order as mentioned above.

VII.2.1.1 Step 1: Connectivity established

So, assuming the intended path-layer connectivity has been established and both source and sink are still provisioned for a size of zero ($So_MI_ProvM[1..X_{MT}] = 0$, $Sk_MI_ProvM[1..X_{MR}] = 0$), the status of the adaptation functions will be the following (Figure VII.2):

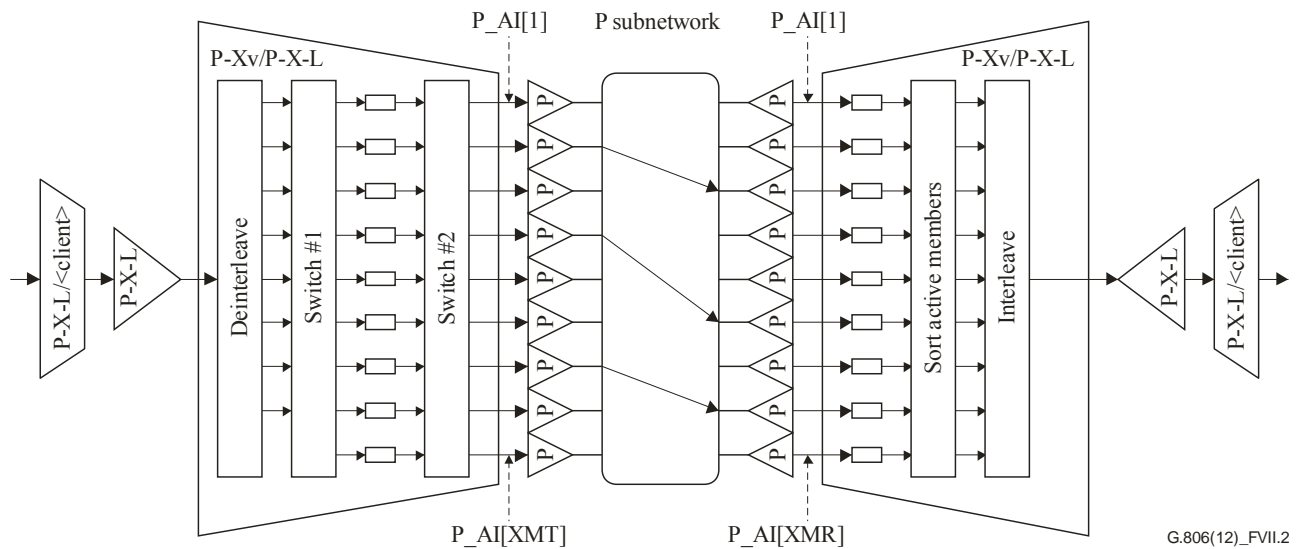


Figure VII.2 – Initial configuration

- For the source:
 - *LCAS engine*
 - Will calculate $_X_{AT} = 0$, $_CTRL[0..X_{MT}-1] = IDLE$, $_PC[0..X_{MT}-1] = 0$
 - Will also initialize $_SQmap[1..X_{MT}]$ to some value, e.g.,⁷:

K	1	2	3	4	5	6	7	8	9
$_SQmap[i]$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a

- *Deinterleave*
Since $_X_{AT} = 0$, this process will insert all-zero path-layer signals at all its outputs.
- *"Switch 1"*
Since $_PC[0..X_{MT}-1] = 0$, this process will insert all-zero path-layer signals at all its outputs.
- *"Switch 2"*
Since $_SQmap[1..X_{MT}] = n/a$, this process will insert path-layer signals at all its outputs with an IDLE control word and an SQ number according to [ITU-T G.7042] for non-provisioned members.

As a result, the source will produce at every $P_AI[i]$ a path-layer signal with a valid VLI overhead structure, a sequence number according to [ITU-T G.7042] for non-provisioned members, an IDLE control word and $_CI_OH$ bytes equal to those at the $P-X-L_CI$.

⁷ The initialization of $_SQmap$, for the case where there are no provisioned members, is not covered by the normative text and will be, in general, implementation-dependent. Here, a simple example is used.

– For the sink:

- *MFI extract*

For the members with a connection ($k = 3, 6, 8$), this process will recover the multiframe information. For the rest of the members, $AI_TSF[i]$ is true and, hence, $_MFI[i]$ will be the error indication (furthermore, $dLOM[i]$ will be declared for these members).

- *Delay calculation*

Since $MI_ProvM[i] = 0$ for all i , this process will output $MI_DMFI[i] = n/a$, $_D[i] = ffs$ for all members, $dMND[i] = false$.

- *Delay*

Will delay all members by $_D[i] = ffs$.

- *LCAS engine*

The inputs (inputs for which $P_CP[i]$ has a connection are shaded) to this process will be:

k	1	2	3	4	5	6	7	8	9
MI_ProvM[i]	0	0	0	0	0	0	0	0	0
dLOM[i]	T	T	F	T	T	F	T	F	T
_TSF[i]	T	T	F	T	T	F	T	F	T
_CRC_z[i] ⁸	X	X	F	X	X	F	X	F	X
_CRC_ok[i]	X	X	T	X	X	T	X	T	X
_CTRL[i]	X	X	Idle	X	X	Idle	X	Idle	X
_SQ[i]	X	X	1	X	X	3	X	6	X

This process will, hence, calculate:

- $MI_LCAS_So_Detected = true$, $_LCASActive = true$
- $_X_{AR} = 0$, $_PC[1..X_{MR}] = 0$, $dSQM[1..X_{MR}] = false$
- For $_SQV[1..X_{MR}]$:

K	1	2	3	4	5	6	7	8	9
_SQV[i]	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a

- And $_RI_Selector = n/a$.

- *Sort active members*

Since $_X_{AR} = 0$, this process will insert all-zero path-layer signals at all its outputs.

⁸ In this and the following tables, "X" is used to denote an indeterminate entry. As discussed by the VLI extract process description, the $_VLI[i]$ information for members with $_TSF[i] = true$ is a technology-specific error indication. The normative text does not specify which values for $_CRC_z[i]$, $_CRC_ok[i]$, $_CTRL[i]$ and $_SQ[i]$ the VLI disassemble process will produce out of the $_VLI[i]$ error indication in this case. As is seen in the course of the example, the particular value of "X" in an implementation is irrelevant for the further operation of the function.

- *Interleave + AIS generator + AIS insertion*
An AIS signal with a rate of ffs (currently not defined for $X_{AR} = 0$) will be inserted towards the P-X-L_CI.
- Also, the following will be signalled towards the P-X-L_CI: CI_SSF = false, CI_X_{AR} = 0.
- Also, the following will be signalled towards the P-X-L_MI: MI_X_{MR} = 9, MI_X_{AR} = 0, MI_DMFI[1..X_{MR}] = n/a, MI_cLOM[1..X_{MR}] = false, MI_cSQM[1..X_{MR}] = false, MI_cLOA = false, MI_cPLCR = false, MI_cTLCR = false.
- MI_Ac_SQ[1..X_{MR}] = _SQv[1..X_{MR}] (see above).

VII.2.1.2 Step 2: Source provisioned

Now, if the source is configured to use the outputs 2, 4, 7 (So_MI_ProvM[2, 4, 7] = 1), the status of the system will become:

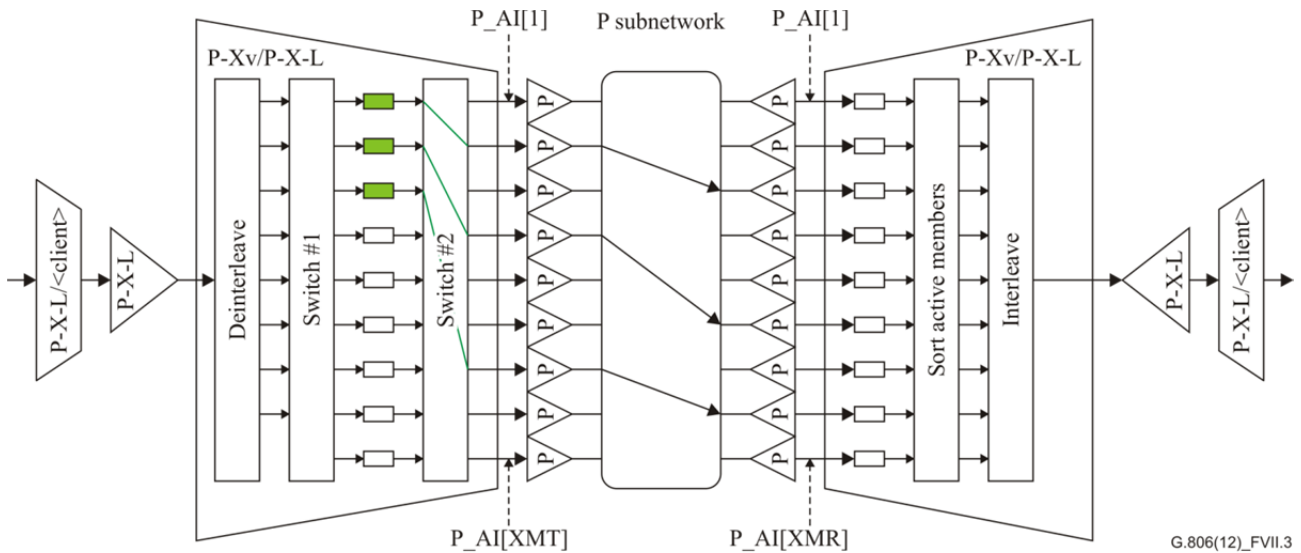
- For the source:
 - *LCAS engine*
 - _CTRL[0..2] = ADD (will not change yet to NORM, since the sink is providing RI_MST[0..2] = 1);
 - _CTRL[3..8] = IDLE;
 - _PC[0..8] = 0;
 - Will calculate _X_{AT} = 0;
 - Will also calculate _SQmap[1..X_{MT}] such that inputs 0, 1, 2 are connected to outputs 2, 4, 7, e.g.,⁹:

K	1	2	3	4	5	6	7	8	9
_SQmap[i]	n/a	0	n/a	1	n/a	n/a	2	n/a	n/a

- *Deinterleave*
Since _X_{AT} = 0, this process will insert all-zero path-layer signals at all its outputs.
- *"Switch 1"*
Since _PC[0..8] = 0, this process will insert all-zero path-layer signals at all its outputs.
- *"Switch 2"*
Will just connect output i to input _SQmap[i] for the provisioned members, and insert path-layer signals at all its non-provisioned outputs with an IDLE control word and a SQ number according to [ITU-T G.7042] for non-provisioned members.

As a result, the source will produce path-layer signals at P_AI[1..9]. All P-AI[i] signals will have a valid VLI overhead structure and a sequence number according to _SQmap[i] (k = 2, 4, 7) or [ITU-T G.7042] for non-provisioned members. The control words will indicate ADD for P_AI[2, 4, 7] and IDLE for the rest of the outputs. This situation is shown in Figure VII.3.

⁹ There are several possible output values of _SQmap after this operation which would comply with the requirements stated in the normative text. In particular, the individual SQ for the provisioned members could have been assigned to the members in any order. Here, an example is used.



G.806(12)_FVII.3

NOTE – The light-grey-coloured items indicate the provisioned members in the source which do not (yet) carry payload.

Figure VII.3 – Status after provisioning source

- For the sink:
 - For the sink, nothing changes from the situation as shown in the previous step, except that the control words extracted from the $_VLI[i]$ information for the members with a connection ($k = 3, 6, 8$) become "add" (instead of "idle"). All outputs of the function remain the same.

VII.2.1.3 Step 3: Sink provisioned

Now, if the sink is configured to use the inputs 3, 6, 8 ($Sk_MI_ProvM[3, 6, 8] = 1$), the status of the system will become:

- For the source:
 - *LCAS engine*
 - $_CTRL[0..2] = ADD$ transitionally, and after the sink signals $RI_MST[0..2] = 0$ (see below), then finally $_CTRL[0..1] = NORM, _CTRL[2] = EOS$;
 - $_CTRL[3..8] = IDLE$ will not change;
 - $_PC[0..2] = 1$;
 - $_PC[3..8] = 0$ will not change;
 - Will calculate $_X_{AT} = 3$;
 - $_SQmap[1..X_{MT}]$ will not change:

K	1	2	3	4	5	6	7	8	9
$_SQmap[i]$	n/a	0	n/a	1	n/a	n/a	2	n/a	n/a

- *Deinterleave*
Since $_X_{AT} = 3$, this process will spread the CI_D information over its exits 1..3 and insert all-zero path-layer signals at all its other outputs.
- *"Switch 1"*
Since $_PC[0..2] = 1$ and $_PC[3..8] = 0$, this process will connect inputs 1..3 to outputs 0..2 and insert all-zero path-layer signals at all its other outputs.

- *"Switch 2"*

Will just connect output *i* to input `_SQmap[i]` for the provisioned members, and insert path-layer signals at all its non-provisioned outputs with an IDLE control word and a SQ number according to [ITU-T G.7042] for non-provisioned members.

As a result, the source will produce three path-layer signals at `P_AI[2, 4, 7]` containing the deinterleaved payload from `CI_D` and non-payload-carrying path-layer signals at the rest of the `P_AI[i]`. All `P_AI[i]` signals will have a valid VLI overhead structure, a sequence number according to `_SQmap[i]` ($k = 2, 4, 7$) or [ITU-T G.7042] for non-provisioned members, a NORM, EOS or IDLE control word and `_CI_OH` bytes equal to those at the `P-X-L_AI`.

– For the sink:

- *MFI extract*

For the members with a connection ($k = 3, 6, 8$), this process will recover the multiframe information. For the rest of the members, `AI_TSF[i]` is true and, hence, `_MFI[i]` will be the error indication (furthermore, `dLOM[i]` will be declared for these members).

- *Delay calculation*

Now, since `MI_ProvM[i] = 1` for $i = 3, 6, 8$, this process will calculate `_D[i]` as necessary to compensate the differential delay. This will be done considering only these inputs, as for the rest `MI_ProvM[i] = 0` and, therefore, `MI_DMFI[i] = n/a`, `_D[i] = ffs`.

Assuming the relative delays among the considered members are supported, `dMND[i] = false`.

- *Delay*

Will multiframe-align `P_AI[3, 6, 8]` and delay all other members by `_D[i] = ffs`.

- *LCAS engine*

The inputs (inputs for which `P_CP[i]` has a connection are shaded) to this process will be:

k	1	2	3	4	5	6	7	8	9
<code>MI_ProvM[i]</code>	0	0	1	0	0	1	0	1	0
<code>dLOM[i]</code>	T	T	F	T	T	F	T	F	T
<code>_TSF[i]</code>	T	T	F	T	T	F	T	F	T
<code>_CRC_z[i]</code>	X	X	F	X	X	F	X	F	X
<code>_CRC_ok[i]</code>	X	X	T	X	X	T	X	T	X
<code>_CTRL[i]</code>	X	X	Add, then Norm	X	X	Add, then Norm	X	Add, then EOS	X
<code>_SQ[i]</code>	X	X	0	X	X	1	X	2	X

This process will, hence, accept the three members as active and calculate:

- `_XAR = 3`, `_PC[3, 6, 8] = 1`, `_PC[1, 2, 4, 5, 7, 9] = 0`, `dSQM[1..XMR] = false`;
- (unchanged: `MI_LCAS_So_Detected = true`, `_LCASActive = true`);

- For $_SQv[1..X_{MR}]$:

K	1	2	3	4	5	6	7	8	9
$_SQv[i]$	n/a	n/a	0	n/a	n/a	1	n/a	2	n/a

- And $_RI_Selector = 3$, $RI_MST_gen[0..2] = 0$, $RI_MST_gen[3..255] = 1$.

- *Sort active members*

This process will connect inputs 3, 6, 8 to outputs 1, 2, 3, respectively. For the other outputs, this process will insert all-zero path-layer signals.

- *Interleave*

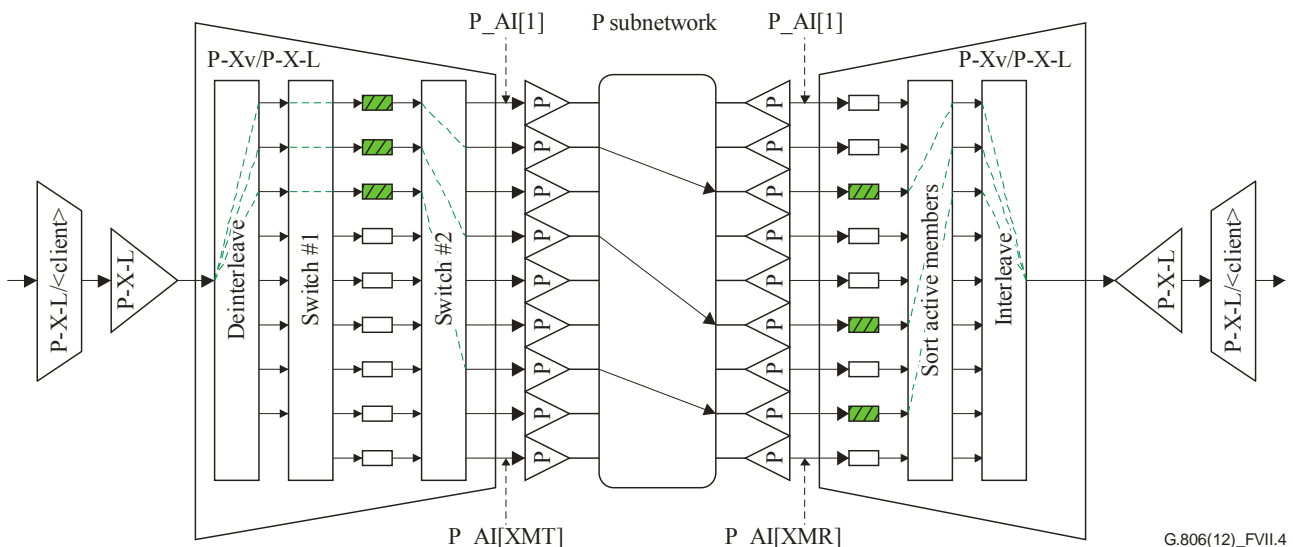
Since $_X_{AR} = 3$, this process will recover a path-layer-3c signal by interleaving the 3 path-layer signals at its inputs 1..3.

- *AIS generator + AIS insertion*

Since $aAIS = false$, no AIS signal will be inserted towards the P-X-L_CI.

- The following will also be signalled towards the P-X-L_CI: $CI_SSF = false$, $CI_X_{AR} = 3$.
- The following will also be signalled towards the P-X-L_MI: $MI_X_{MR} = 9$, $MI_X_{AR} = 3$, $MI_DMFI[3, 6, 8] = xxx$, $MI_DMFI[1, 2, 4, 5, 7, 9] = n/a$, $MI_cLOM[1..X_{MR}] = false$, $MI_cSQM[1..X_{MR}] = false$, $MI_cLOA = false$, $MI_cPLCR = false$, $MI_cTLCR = false$.
- $MI_Ac_SQ[1..X_{MR}] = _SQv[1..X_{MR}]$ (see above).

As a result, the three-member VCG will be established. This situation is shown in Figure VII.4.



NOTE – Hatched (green-coloured) items indicate the path taken by the CI_D payload.

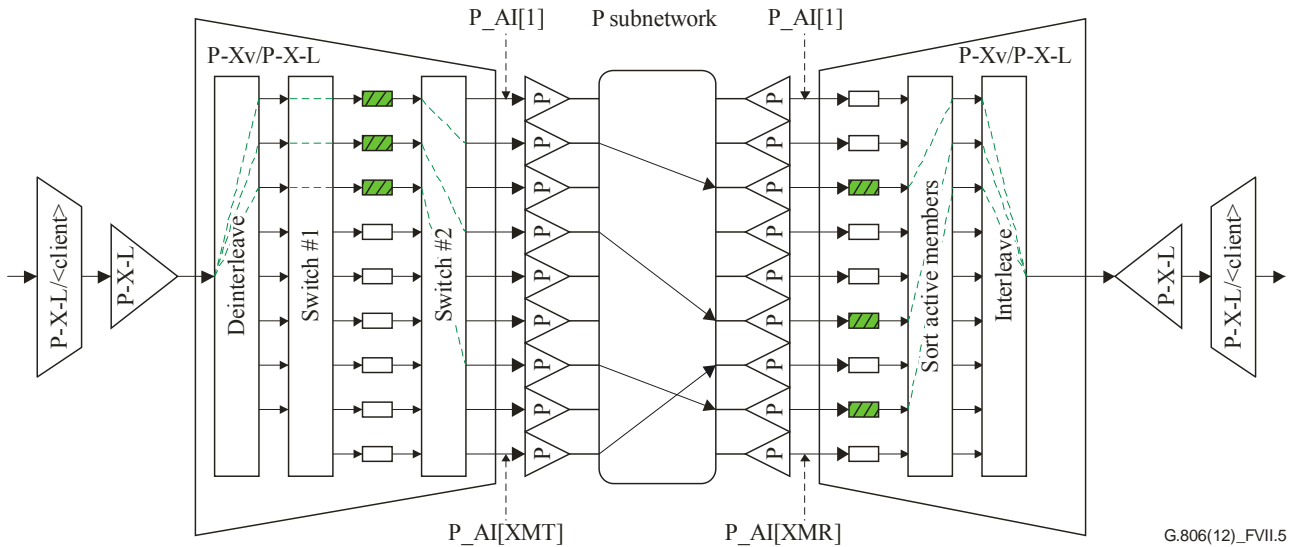
Figure VII.4 – Status after provisioning the sink

VII.2.2 Scenario 2: Addition of a member

In order to add a member, three items need to be provisioned: the source and the sink MI_ProvM and the path connectivity in-between. These three operations can be done in any order and the operation will be hitless for the client signal, irrespective of the order.

VII.2.2.1 Step 1: Connectivity established

For this example, it will be assumed that the connectivity is done first, e.g., connecting P_CP[9] in the source to P_CP[7] in the sink (see Figure VII.5).



NOTE – Hatched (green-coloured) items indicate the path taken by the CI_D payload.

Figure VII.5 – Status after provisioning the connectivity

VII.2.2.2 Step 2: Source provisioned

Assuming the source is updated next (with MI_ProvM[9] = 1), the status of the system will be:

- For the source:
 - *LCAS engine*
 - Will set $_CTRL[3] = ADD$. Since this process is also receiving $_MST_rec[3] = 1$ (FAIL), this $_CTRL[3]$ value will be sent continuously.
 - Will continue to have $_X_{AT} = 3$, $_CTRL[0..1] = NORM$, $_CTRL[2] = EOS$, $_CTRL[4..8] = IDLE$, $_PC[0..2] = 1$, $_PC[3..8] = 0$.
 - Will update $_SQmap[i]$ such that sequence number 3 be put out onto P_AI[9].

k	1	2	3	4	5	6	7	8	9
$_SQmap[i]$	n/a	0	n/a	1	n/a	n/a	2	n/a	3

- *Deinterleave*

Since $_X_{AT} = 3$, this process will spread the CI_D information¹⁰ over its outputs 1..3 and insert all-zero path-layer signals at all its other outputs.
- *"Switch 1"*

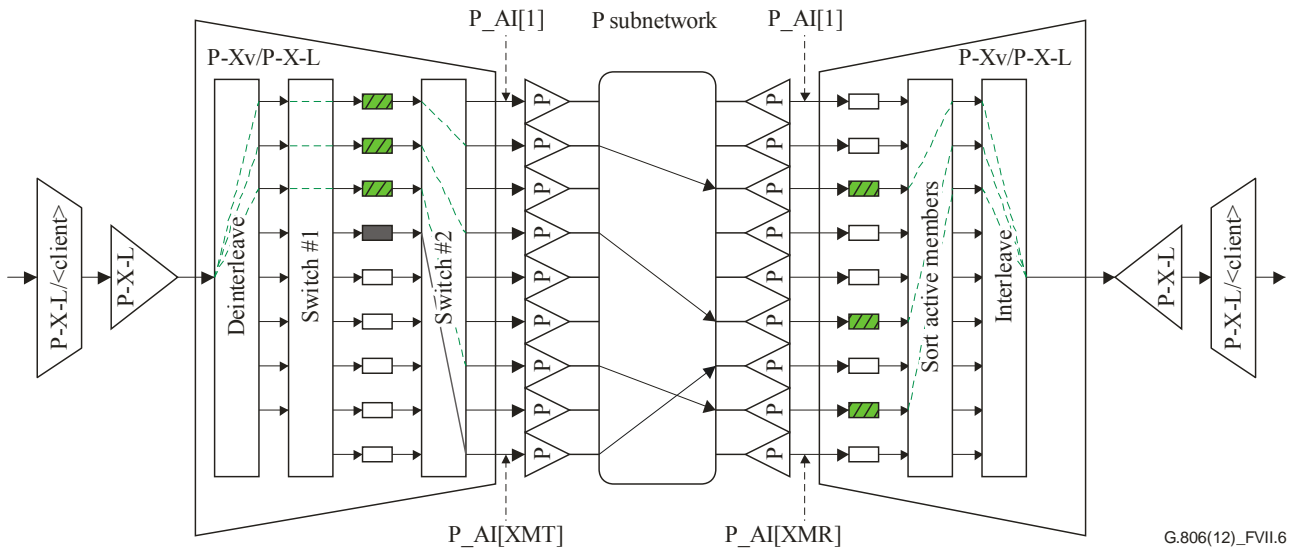
Since $_PC[0..2] = 1$ and $_PC[3..8] = 0$, this process will connect inputs 1..3 to outputs 0..2 and insert all-zero path-layer signals at all its other outputs.

¹⁰ Which is expected to be a path-layer-3c, since $CI_{X_{AT}} = 3$ is being communicated to the upper-layer functions.

- "Switch 2"

Will just connect output *i* to input *_SQmap[i]* for the provisioned members, and insert path-layer signals at all its non-provisioned outputs with an IDLE control word and a SQ number according to [ITU-T G.7042] for non-provisioned members.

As a result, the source will produce the same payload mapping as before having been provisioned and, in addition, will send a *_CTRL[3] = ADD* request over the physical *P_CP[9]*. This situation is shown in Figure VII.6.



G.806(12)_FVII.6

NOTE – Hatched (green-coloured) items indicate the path taken by the CI_D payload.
The light-grey-coloured items indicate the provisioned member which does not (yet) carry payload.

Figure VII.6 – Status after provisioning the source

– For the sink:

- *Delay calculation, delay*

No changes from last section, since the provisioned membership is the same.

- *LCAS engine*

Among the inputs to this process, the only ones that will change will be the ones related to the new signal that is incoming (in bold below) (inputs for which *P_CP[i]* has a connection are shaded):

k	1	2	3	4	5	6	7	8	9
MI_ProvM[i]	0	0	1	0	0	1	0	1	0
dLOM[i]	T	T	F	T	T	F	F	F	T
_TSF[i]	T	T	F	T	T	F	F	F	T
_CRC_z[i]	X	X	F	X	X	F	F	F	X
_CRC_ok[i]	X	X	T	X	X	T	T	T	X
_CTRL[i]	X	X	Norm	X	X	Norm	ADD	EOS	X
_SQ[i]	X	X	0	X	X	1	3	2	X

Since *MI_ProvM[i]* has not changed, all the outputs of this process will stay the same:

- *_XAR = 3, _PC[3, 6, 8] = 1, _PC[1, 2, 4, 5, 7, 9] = 0, dSQM[1..XMR] = false;*
- *MI_LCAS_So_Detected = true, _LCASActive = true;*

- For $_SQv[1..X_{MR}]$:

k	1	2	3	4	5	6	7	8	9
$_SQv[i]$	n/a	n/a	0	n/a	n/a	1	n/a	2	n/a

- And $_RI_Selector = 3$.
- *Sort active members, interleave, AIS generator + insertion*
Since there have been no changes to $MI_ProvM[i]$, these processes do not change their outputs.

As a result, the sink will see an ADD request in one of its members, but will not react to it because this member is provisioned for non-operation ($MI_ProvM[i] = 0$).

VII.2.2.3 Step 3: Sink provisioned

Assuming the sink is updated, then to use this member (by setting $Sk_MI_ProvM[7] = 1$), the status of the system will become:

- For the source:
 - *LCAS engine*
 - The only change is that, as explained below (sink description), the source LCAS process will receive $_MST_rec[3] = 0$ now. As a consequence, it will set $_X_{AT} = 4$ and $_CTRL[0..2] = NORM$, $_CTRL[3] = EOS$, $_CTRL[4..8] = IDLE$, $_PC[0..3] = 1$, $_PC[4..8] = 0$.
 - $_SQmap[i]$ does not change, since the sequence numbers to be transported over each $P_AI[i]$ signal do not change.
 - *Deinterleave*
Since $_X_{AT} = 4$, this process will now spread the CI_D information over its outputs 1..4 and insert all-zero path-layer signals at all its other outputs.
 - *"Switch 1"*
Since $_PC[0..3] = 1$ and $_PC[4..8] = 0$, this process will connect inputs 1..4 to outputs 0..3 and insert all-zero path-layer signals at all its other outputs.
 - *"Switch 2"*
Will just connect output i to input $_SQmap[i]$.

As a result, the source will produce four path-layer signals at $P_AI[2, 4, 7, 9]$ containing the deinterleaved payload from CI_D and all-zero-payload path-layer signals at the rest of the $P_AI[i]$. All $P_AI[i]$ signals will have a valid VLI overhead structure, a sequence number according to $_SQmap[i]$, a NORM, EOS or IDLE control word and $_CI_OH$ bytes equal to those at the $P-X-L_AI$.

- For the sink:
 - *MFI extract*
For the members with a connection ($k = 3, 6, 7, 8$), this process will recover the multiframe information. For the rest of the members, $AI_TSF[i]$ is true and, hence, $_MFI[i]$ will be the error indication (furthermore, $dLOM[i]$ will be declared for these members).
 - *Delay calculation*
For the now four P_APs for which $MI_ProvM[i] = 1$ ($k = 3, 6, 7, 8$), this process will calculate $_D[i]$ as necessary to compensate the differential delay. This will be done considering only these inputs, as for the rest $MI_ProvM[i] = 0$ and, therefore, $MI_DMFI[i] = n/a$, $_D[i] = ffs$.

Assuming the relative delays among the considered members are supported, dMND[i] = false.

- *Delay*

Will multiframe-align P_AI[3, 6, 7, 8] and delay the rest by _D[i] = ffs.

- *LCAS engine*

Among the inputs to this process, the only ones that will change will be MI_ProvM[7] and the control words (in bold below):

k	1	2	3	4	5	6	7	8	9
MI_ProvM[i]	0	0	1	0	0	1	1	1	0
dLOM[i]	T	T	F	T	T	F	F	F	T
_TSF[i]	T	T	F	T	T	F	F	F	T
_CRC_z[i]	X	X	F	X	X	F	F	F	X
_CRC_ok[i]	X	X	T	X	X	T	T	T	X
_CTRL[i]	X	X	Norm	X	X	Norm	Add, then EOS	EOS, then Norm	X
_SQ[i]	X	X	0	X	X	1	3	2	X

This process will, hence, accept the new member as active and calculate:

- $_X_{AR} = 4$, $_PC[3, 6, 7, 8] = 1$, $_PC[1, 2, 4, 5, 9] = 0$, $dSQM[1..X_{MR}] = \text{false}$;
- RI_XXX will be as defined by [ITU-T G.7042] ($RI_MST_gen[0..3] = 0$, $RI_MST_gen[4..8] = 1$);
- For $_SQV[1..X_{MR}]$ the value for $i = 7$ will change:

k	1	2	3	4	5	6	7	8	9
$_SQV[i]$	n/a	n/a	0	n/a	n/a	1	3	2	n/a

- and still $_RI_Selector = 3$.

- *Sort active members*

This process will connect inputs 3, 6, 8, 7 to outputs 1, 2, 3, 4, respectively. For the other outputs, this process will insert all-zero path-layer signals.

- *Interleave*

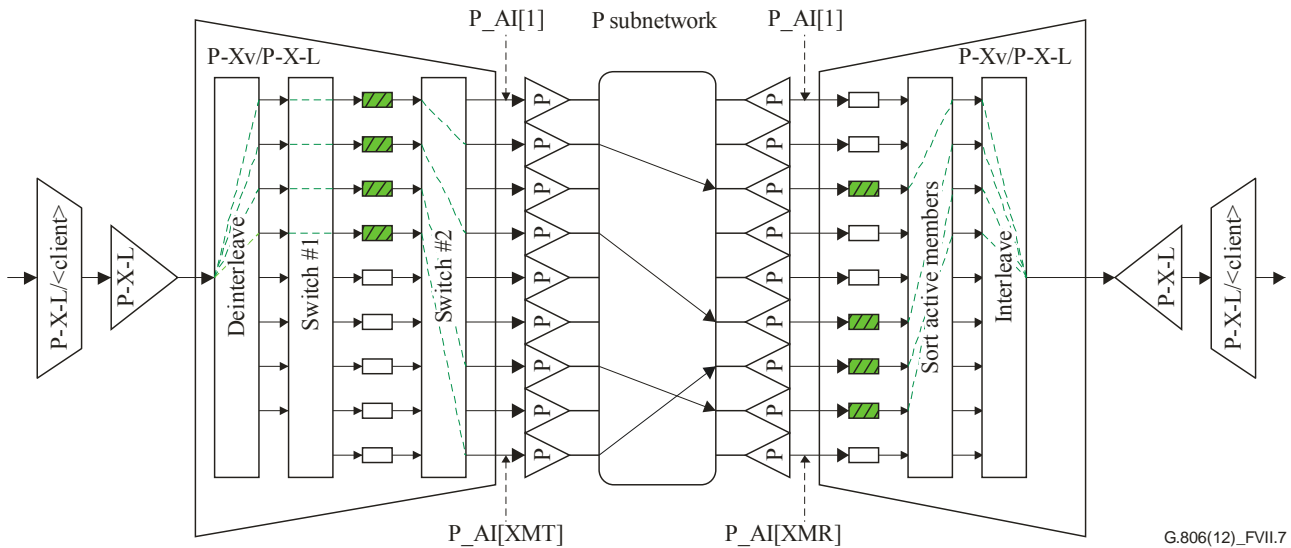
Since $_X_{AR} = 4$, this process will recover a path-layer-4c signal by interleaving the 4 path-layer signals at its inputs 1..4.

- *AIS generator + AIS insertion*

Since aAIS = false, no AIS signal will be inserted towards the P-X-L_CI.

- The following will also be signalled towards the P-X-L_CI: $CI_SSF = \text{false}$, $CI_X_{AR} = 4$.
- The following will also be signalled towards the P-X-L_MI: $MI_X_{MR} = 9$, $MI_X_{AR} = 4$, $MI_DMFI[3, 6, 7, 8] = \text{xxx}$, $MI_DMFI[1, 2, 4, 5, 9] = \text{n/a}$, $MI_cLOM[1..X_{MR}] = \text{false}$, $MI_cSQM[1..X_{MR}] = \text{false}$, $MI_cLOA = \text{false}$, $MI_cPLCR = \text{false}$, $MI_cTLCR = \text{false}$.
- $MI_Ac_SQ[1..X_{MR}] = _SQV[1..X_{MR}]$ (see above).

As a result, the sink will accept the new member as an active member and start using its payload. This situation is shown in Figure VII.7.



G.806(12)_FVII.7

NOTE – Hatched (green-coloured) items indicate the path taken by the CI_D payload.

Figure VII.7 – Status after provisioning the sink

VII.2.3 Scenario 3: Removal of a member

In order to delete a member, three items need to be provisioned: the source and the sink MI_ProvM and the path connectivity in-between. These three operations can be done in any order, but only the removal of the member at the source end first will be hitless to the client signal. Both the taking down of the connection for the member, and the removal of the member at the sink end first will result in a temporary disruption of the client signal until the resulting $MST = FAIL$ status for the member reaches the source end, and the member is removed from the active multiplex there (and $CTRL = DNU$ is inserted). The subsequent removal of the member at the source will have no effect on the client signal.

For the sake of this example, the member is first deleted at source, then at the sink and then the connectivity will be removed. The example assumes a starting condition with a four-active-member group established, as shown in Figure VII.7.

VII.2.3.1 Step 1: Source provisioned

For this example, it will be assumed that the source provisioning is done first. If the member to be removed is, e.g., the one with $i = 7$ (i.e., $MI_ProvM[7] = 0$), the status of the system will be:

- For the source:
 - *LCAS engine*
 - Will set $_CTRL[3] = IDLE$, as per [ITU-T G.7042]. Accordingly, $_X_{AT} = 3$, $_CTRL[0..1] = NORM$, $_CTRL[2] = EOS$, $_CTRL[4..8] = IDLE$, $_PC[0..2] = 1$, $_PC[3..8] = 0$.
 - Will update $_SQmap[i]$ such that the remaining provisioned members ($k = 2, 4, 9$) carry the remaining members ($SQ = 0..2$).

k	1	2	3	4	5	6	7	8	9
$_SQmap[i]$	n/a	0	n/a	1	n/a	n/a	n/a	n/a	2

- *Deinterleave*
Since $_X_{AT} = 3$, this process will spread the CI_D information¹¹ over its outputs 1..3 and insert all-zero path-layer signals at all its other outputs.
- *"Switch 1"*
Since $_PC[0..2] = 1$ and $_PC[3..8] = 0$, this process will connect inputs 1..3 to outputs 0..2 and insert all-zero path-layer signals at all its other outputs.
- *"Switch 2"*
Will just connect output i to input $_SQmap[i]$ for the provisioned members, and insert path-layer signals at all its non-provisioned outputs with an IDLE control word and a SQ number, according to [ITU-T G.7042], for non-provisioned members.

As a result, the source will stop mapping payload onto $P_AI[7]$ and will reduce the available bandwidth to the client layers to $CI_X_{AR} = 3$. This client bandwidth will be mapped onto the three remaining provisioned members. For $P_AI[7]$, a signal with an IDLE control word and a sequence number, as per the requirements in [ITU-T G.7042] for non-provisioned members, will be sourced, indicating to the sink that this member no longer carries payload.

– For the sink:

- *Delay calculation, delay*
No changes from last section, since the provisioned membership is still the same.
- *LCAS engine*
Among the inputs to this process, the only ones that will change will be the ones related to the new incoming control words due to the member that was removed at the source (changes in bold below) (inputs for which $P_CP[i]$ has a connection are shaded):

k	1	2	3	4	5	6	7	8	9
$MI_ProvM[i]$	0	0	1	0	0	1	1	1	0
$dLOM[i]$	T	T	F	T	T	F	F	F	T
$_TSF[i]$	T	T	F	T	T	F	F	F	T
$_CRC_z[i]$	X	X	F	X	X	F	F	F	X
$_CRC_ok[i]$	X	X	T	X	X	T	T	T	X
$_CTRL[i]$	X	X	Norm	X	X	Norm	EOS	Idle	X
$_SQ[i]$	X	X	0	X	X	1	2	X' ¹²	X

This process will hence stop accepting payload from $P_AI[8]$ and calculate:

- (unchanged: $MI_LCAS_So_Detected = true$, $_LCASActive = true$);
- $_X_{AR} = 3$, $_PC[3, 6, 7] = 1$, $_PC[1, 2, 4, 5, 8, 9] = 0$, $dSQM[1..X_{MR}] = false$;
- RI_XXX will be as defined by [ITU-T G.7042] ($RI_MST_gen[0..2] = 0$, $RI_MST_gen[3..8] = 1$);

¹¹ Which is expected to be a path-layer-3c, since $CI_X_{AT} = 3$ is being communicated to the upper-layer functions.

¹² X' represents the sequence number the source is using for this non-provisioned member. It is assumed to comply with the requirements in [ITU-T G.7042] to be greater than any of the SQ used for "NORM", "EOS" and "DNU" members.

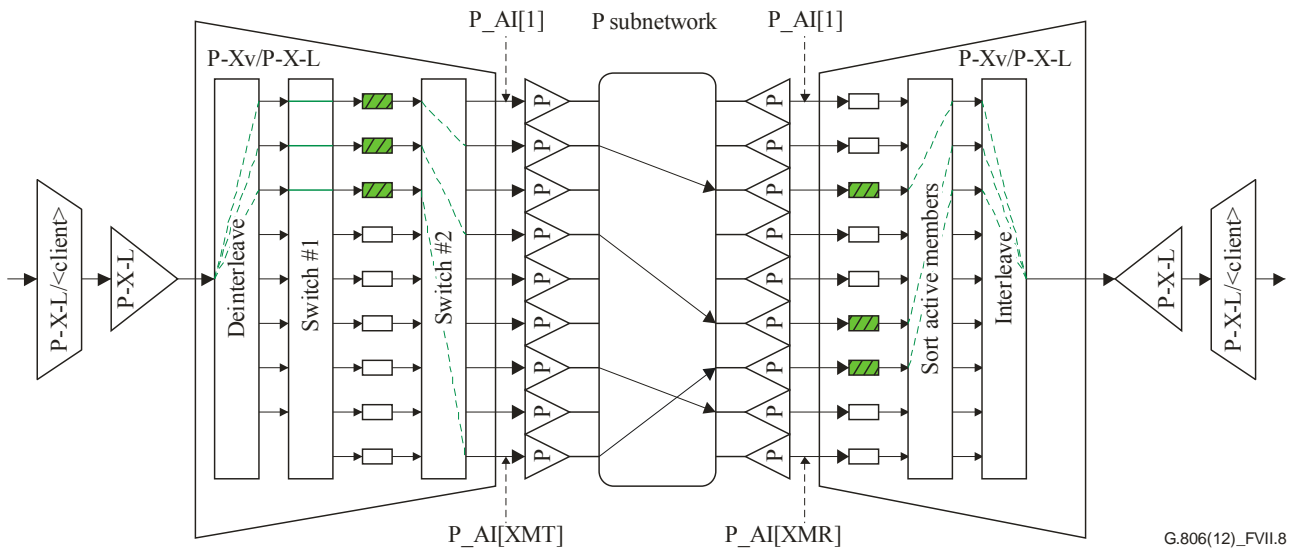
- For $_SQV[1..X_{MR}]$ the values for $i = 7$ and $i = 8$ will change:

k	1	2	3	4	5	6	7	8	9
$_SQV[i]$	n/a	n/a	0	n/a	n/a	1	2	X'	n/a

- And still $_RI_Selector = 3$.
- *Sort active members*
This process will connect inputs 3, 6, 7 to outputs 1, 2, 3, respectively. For the other outputs, this process will insert all-zero path-layer signals, since $_PC[i] = 0$ for these members.
- *Interleave*
Since $_X_{AR} = 3$, this process will recover a path-layer-3c signal by interleaving the 3 path-layer signals at its inputs 1..3.
- *AIS generator + AIS insertion*
Since $aAIS = false$, no AIS signal will be inserted towards the P-X-L_CI.
- The following will also be signalled towards the P-X-L_CI: $CI_SSF = false$, $CI_X_{AR} = 3$.
- The following will also be signalled towards the P-X-L_MI: $MI_X_{MR} = 9$, $MI_X_{AR} = 3$, $MI_DMFI[3, 6, 7, 8] = xxx$, $MI_DMFI[1, 2, 4, 5, 9] = n/a$, $MI_cLOM[1..X_{MR}] = false$, $MI_cSQM[1..X_{MR}] = false$, $MI_cLOA = false$, $MI_cPLCR = false$ ¹³, $MI_cTLCR = false$.
- $MI_Ac_SQ[1..X_{MR}] = _SQV[1..X_{MR}]$ (see above).

As a result, the sink will stop accepting payload from the member indicating "idle" in the control word and will reduce the bandwidth forwarded towards the client functions to $CI_X_{AR} = 3$. Since the member is still provisioned, however, it would still be taken into consideration for realignment and its VLI would still be analysed for LCAS purposes. This situation is shown in Figure VII.8.

¹³ Assuming that $MI_PLCRThr \leq 3$, otherwise $MI_cPLCR = true$.



G.806(12)_FVII.8

NOTE – Hatched (green-coloured) items indicate the path taken by the CI_D payload.

Figure VII.8 – Status after provisioning the source

VII.2.3.2 Step 2: Sink provisioned

Assuming the sink is updated, then by setting $Sk_MI_ProvM[8] = 0$ and, consequently, by not using this member, the status of the system will become:

- For the source:
 - Nothing will change for the source inputs or outputs.
- For the sink:
 - *MFI extract*
No change.
 - *Delay calculation*
Now for the three P_APs for which $MI_ProvM[i] = 1$ ($k = 3, 6, 7$), this process will continue to calculate $_D[i]$ as it did before. For $i = 8$, as well as for the other members, $MI_ProvM[i] = 0$ and, therefore, $MI_DMFI[i] = n/a, _D[i] = ffs$.
In other words, P_AI[8] will no longer be considered for multiframe alignment.
 - *Delay*
Will multiframe-align P_AI[3, 6, 7] and delay the rest by $_D[i] = ffs$.
 - *LCAS engine*
Among the inputs to this process, the only one that will change will be $MI_ProvM[8]$ (in bold below):

k	1	2	3	4	5	6	7	8	9
MI_ProvM[i]	0	0	1	0	0	1	1	0	0
dLOM[i]	T	T	F	T	T	F	F	F	T
_TSF[i]	T	T	F	T	T	F	F	F	T
_CRC_z[i]	X	X	F	X	X	F	F	F	X
_CRC_ok[i]	X	X	T	X	X	T	T	T	X
_CTRL[i]	X	X	Norm	X	X	Norm	EOS	Idle	X
_SQ[i]	X	X	0	X	X	1	2	X'	X

This process, hence, will keep:

- MI_LCAS_So_Detected = true, _LCASActive = true;
- _X_{AR} = 3, _PC[3, 6, 7] = 1, _PC[1, 2, 4, 5, 8, 9] = 0, dSQM[1..X_{MR}] = false;
- RI_{XXX} will be as defined by [ITU-T G.7042] (RI_MST_gen[0..2] = 0, RI_MST_gen[3..8] = 1);
- For _SQv[1..X_{MR}] the values for i = 8 will change:

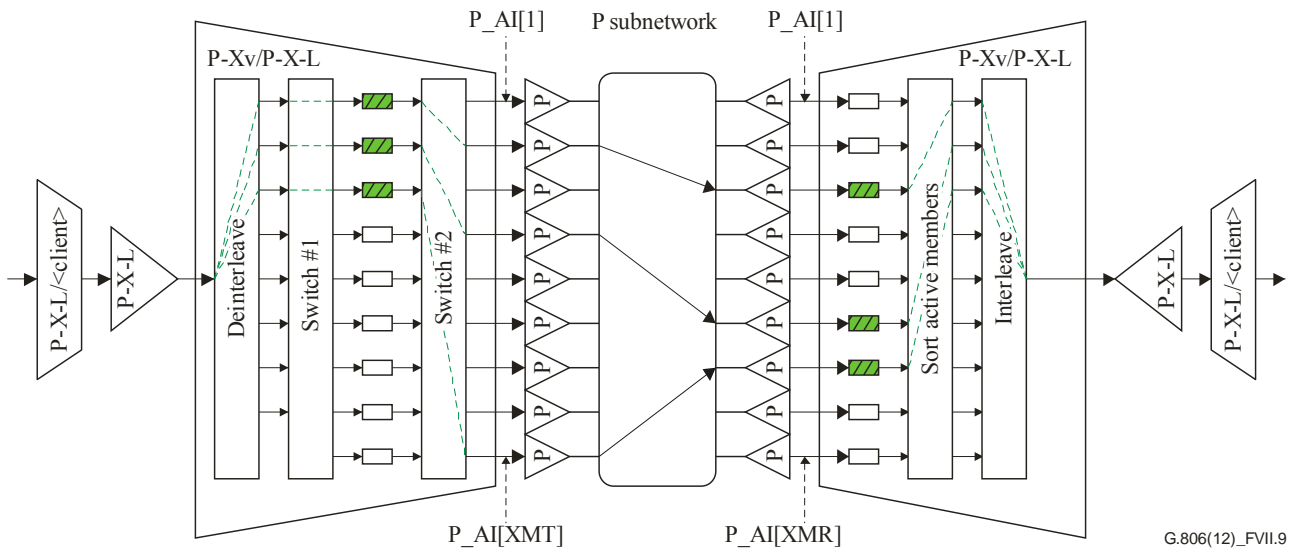
k	1	2	3	4	5	6	7	8	9
_SQv[i]	n/a	n/a	0	n/a	n/a	1	2	n/a	n/a

- and still _RI_Selector = 3.
- *Sort active members, interleave*
No change, since _PC[i] does not change and _SQv[3, 6, 7] also does not change.
- *AIS generator + AIS insertion*
Since aAIS = false, no AIS signal will be inserted towards the P-X-L_CI.
- The following will also be signalled towards the P-X-L_CI: CI_SSF = false, CI_X_{AR} = 3 (no change).
- The following will also be signalled towards the P-X-L_MI: MI_X_{MR} = 9, MI_X_{AR} = 3, MI_DMFI[3, 6, 7] = xxx, MI_DMFI[1, 2, 4, 5, 8, 9] = n/a, MI_cLOM[1..X_{MR}] = false, MI_cSQM[1..X_{MR}] = false, MI_cLOA = false, MI_cPLCR = false, MI_cTLCR = false (the only change is that MI_DMFI[8] = n/a, instead of the value it had before).
- MI_Ac_SQ[1..X_{MR}] = _SQv[1..X_{MR}] (see above).

As a result, the sink will just stop considering P_AI[8] for any purposes.

VII.2.3.3 Step 3: Connectivity removed

If, at this point, the connectivity for the removed member is deleted (see Figure VII.9), the state of the system only changes slightly:



G.806(12)_FVII.9

NOTE – Hatched (green-coloured) items indicate the path taken by the CI_D payload.

Figure VII.9 – Status after deleting the connectivity

- For the source:
 - No changes.
- For the sink:
 - *MFI extract*
The only change is that now AI_TSF[8] is present and, hence, _MFI[8] will be the error indication (furthermore, dLOM[8] will be declared for this member).
 - *Delay calculation, delay*
No changes from last section, since the provisioned membership is the same.
 - *LCAS engine*
Among the inputs to this process, the only ones that will change will be those related to the incoming _TSF[8] (in bold below):

k	1	2	3	4	5	6	7	8	9
MI_ProvM[i]	0	0	1	0	0	1	1	0	0
dLOM[i]	T	T	F	T	T	F	F	T	T
_TSF[i]	T	T	F	T	T	F	F	T	T
_CRC_z[i]	X	X	F	X	X	F	F	X	X
_CRC_ok[i]	X	X	T	X	X	T	T	X	X
_CTRL[i]	X	X	Norm	X	X	Norm	EOS	X	X
_SQ[i]	X	X	0	X	X	1	2	X	X

This process, hence, will change none of its outputs.

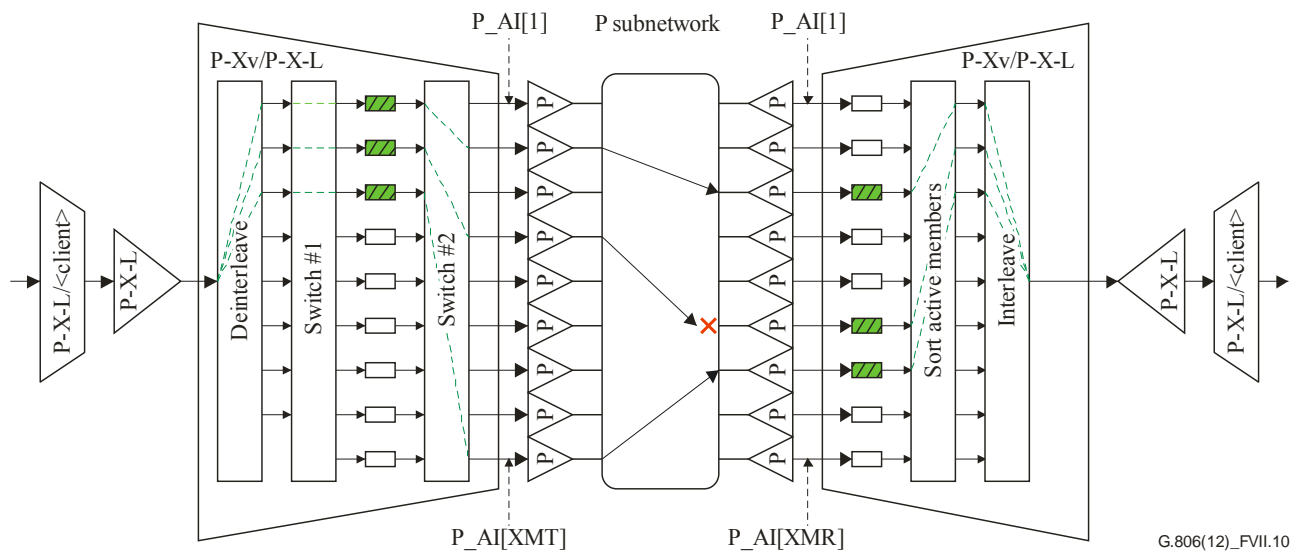
- The same applies to the rest of the processes in the function (no changes in their outputs) and, hence, the function changes none of its outputs.

As a result, the sink will just start receiving AI_TSF[8], but since this member was not provisioned for service anyway, the visible behaviour of the function will not change.

VII.2.4 Scenario 4: Member failure

When an active member fails, the LCAS protocol removes that member from service and continues operation with a reduced set of active members. In this scenario, the failure of one of the members is exercised.

Assuming the member arriving at the sink via P_AI[6] fails (Figure VII.10), the following will happen:



NOTE – Hatched (green-coloured) items indicate the path taken by the CI_D payload.
"X" represents the path failure.

Figure VII.10 – Status after the failure of a member

– For the sink:

- *MFI extract*

The only change is that now AI_TSF[6] is present and, hence, _MFI[6] will be the error indication (furthermore, dLOM[6] will be declared for this member).

- *Delay calculation*

Now for the two P_APs for which MI_ProvM[i] = 1 and AI_TSF[i] is not active (k = 3, 7), this process will continue to calculate _D[i] as it did before. For i = 6, as well as for the other members, AI_TSF[i] is active and, therefore, MI_DMFI[i] = n/a, _D[i] = ffs.

In other words, P_AI[6] will no longer be considered for multiframe alignment.

- *Delay*

Will multiframe-align P_AI[3, 7] and delay the rest by _D[i] = ffs.

- *LCAS engine*

Among the inputs to this process, the ones that will change will be those related to the incoming _TSF[6] (in bold below):

k	1	2	3	4	5	6	7	8	9
MI_ProvM[i]	0	0	1	0	0	1	1	0	0
dLOM[i]	T	T	F	T	T	T	F	T	T
_TSF[i]	T	T	F	T	T	T	F	T	T
_CRC_z[i]	X	X	F	X	X	X	F	X	X
_CRC_ok[i]	X	X	T	X	X	X	T	X	X
_CTRL[i]	X	X	Norm	X	X	X	EOS	X	X
_SQ[i]	X	X	0	X	X	X	2	X	X

This process will, hence, change its outputs to:

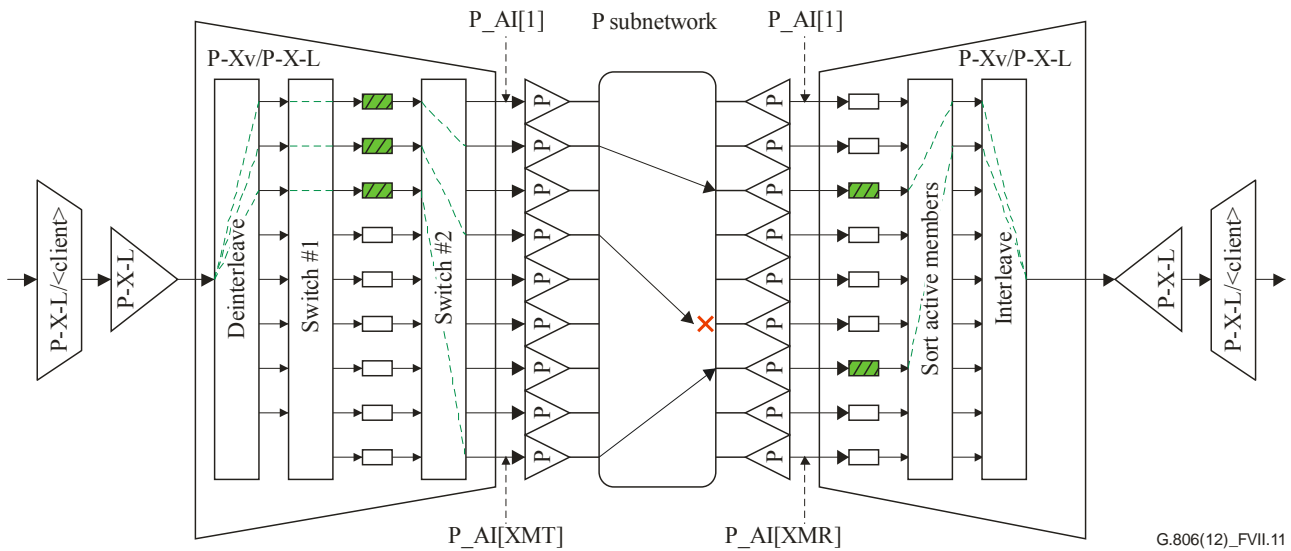
- (Unchanged: MI_LCAS_So_Detected = true, _LCASActive = true).
- $_X_{AR} = 2$, $_PC[3, 7] = 1$, $_PC[1, 2, 4, 5, 6, 8, 9] = 0$, $dSQM[1..X_{MR}] = \text{false}$.
- RI_XXX will be as defined by [ITU-T G.7042] ($RI_MST_gen[0, 2] = 0$, $RI_MST_gen[1, 3..8] = 1$). That is, the sink will start signalling to the source that a failure was detected for sequence number 1.
- For $_SQv[1..X_{MR}]$ the value for $i = 6$ will change:

k	1	2	3	4	5	6	7	8	9
$_SQv[i]$	n/a	n/a	0	n/a	n/a	n/a	2	n/a	n/a

- And still $_RI_Selector = 3$.
- *Sort active members*
This process will connect inputs 3, 7 to outputs 1, 2 respectively. The failed input 6 is no longer taken into consideration, as $_PC[6] = 0$. For the other outputs, this process will insert all-zero path-layer signals.
- *Interleave*
Since $_X_{AR} = 2$, this process will recover a path-layer-2c signal by interleaving the 2 path-layer signals at its inputs 1..2.
- *AIS generator + AIS insertion*
Since $aAIS = \text{false}$, no AIS signal will be inserted towards the P-X-L_CI.
- The following will also be signalled towards the P-X-L_CI: $CI_SSF = \text{false}$, $CI_X_{AR} = 2$ (CI_X_{AR} changes).
- The following will also be signalled towards the P-X-L_MI: $MI_X_{MR} = 9$, $MI_X_{AR} = 2$, $MI_DMFI[3, 7] = \text{xxx}$, $MI_DMFI[1, 2, 4, 5, 6, 8, 9] = \text{n/a}$, $MI_cLOM[1..X_{MR}] = \text{false}$, $MI_cSQM[1..X_{MR}] = \text{false}$, $MI_cLOA = \text{false}$, $MI_cPLCR = \text{false}$ ¹⁴, $MI_cTLCR = \text{false}$ (MI_X_{AR} and $MI_DMFI[6]$ change).
- $MI_Ac_SQ[1..X_{MR}] = _SQv[1..X_{MR}]$ (see above).

As a result, the sink will stop accepting payload from the failed member P_AI[6] and will reduce the bandwidth forwarded towards the client functions to $CI_X_{AR} = 2$. At the same time, it will start signalling towards the source that a failure was detected for sequence number 1. This transient situation is shown in Figure VII.11.

¹⁴ Assuming that $MI_PLCRThr \leq 2$, otherwise $MI_cPLCR = \text{true}$.



G.806(12)_FVII.11

NOTE – Hatched (green-coloured) items indicate the path taken by the CI_D payload.
 "X" represents the failure path

Figure VII.11 – Transient status after the failure of a member and reaction by the sink (the source function has not yet reacted to the sink failure indication in this diagram)

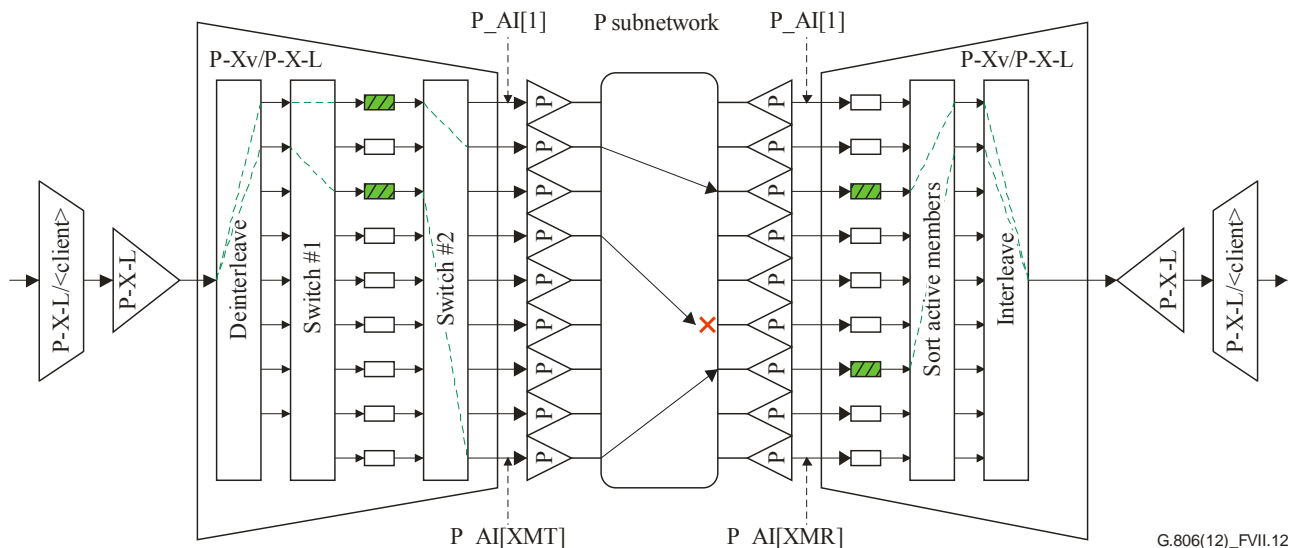
- For the source, the following reaction will happen as soon as it receives the RI_MST[1] = 1 (fail), which is being reported by the sink:
 - *LCAS engine*
 - Will set $_CTRL[1] = DNU$, as per [ITU-T G.7042]. Accordingly, $_X_{AT} = 2$, $_CTRL[0] = NORM$, $_CTRL[2] = EOS$, $_CTRL[3..8] = IDLE$, $_PC[0, 2] = 1$, $_PC[1, 3..8] = 0$.
 - $_SQmap[i]$ does not change, since each sequence number continues to be carried over the same $P_AI[i]$ signals. It will remain:

k	1	2	3	4	5	6	7	8	9
$_SQmap[i]$	n/a	0	n/a	1	n/a	n/a	n/a	n/a	2

- *Deinterleave*
 Since $_X_{AT} = 2$ this process will spread the CI_D information¹⁵ over its outputs 1..2 and insert all-zero path-layer signals at all its other outputs.
- *"Switch 1"*
 Since $_PC[0, 2] = 1$ and $_PC[1, 3..8] = 0$, this process will connect inputs 1, 2 to outputs 0, 2 and insert all-zero path-layer signals at all its other outputs.
- *"Switch 2"*
 Will just connect output i to input $_SQmap[i]$ for the provisioned members, and insert path-layer signals at all its non-provisioned outputs with an IDLE control word and a SQ number, according to [ITU-T G.7042], for non-provisioned members.

¹⁵ Which is expected to be a path-layer-2c, since $CI_X_{AT} = 2$ is being communicated to the upper-layer functions.

As a result, the source will stop mapping payload onto P_AI[4] and will reduce the available bandwidth to the client layers to $CI_{X_{AR}} = 2$. This client bandwidth will be mapped onto the two remaining provisioned, non-failed members. For P_AI[4], a signal with a DNU control word, a sequence number as per $_{SQmap}[4]$ and all-zero payload will be sourced, indicating to the sink that this member no longer carries payload. This situation is shown in Figure VII.12.



G.806(12)_FVII.12

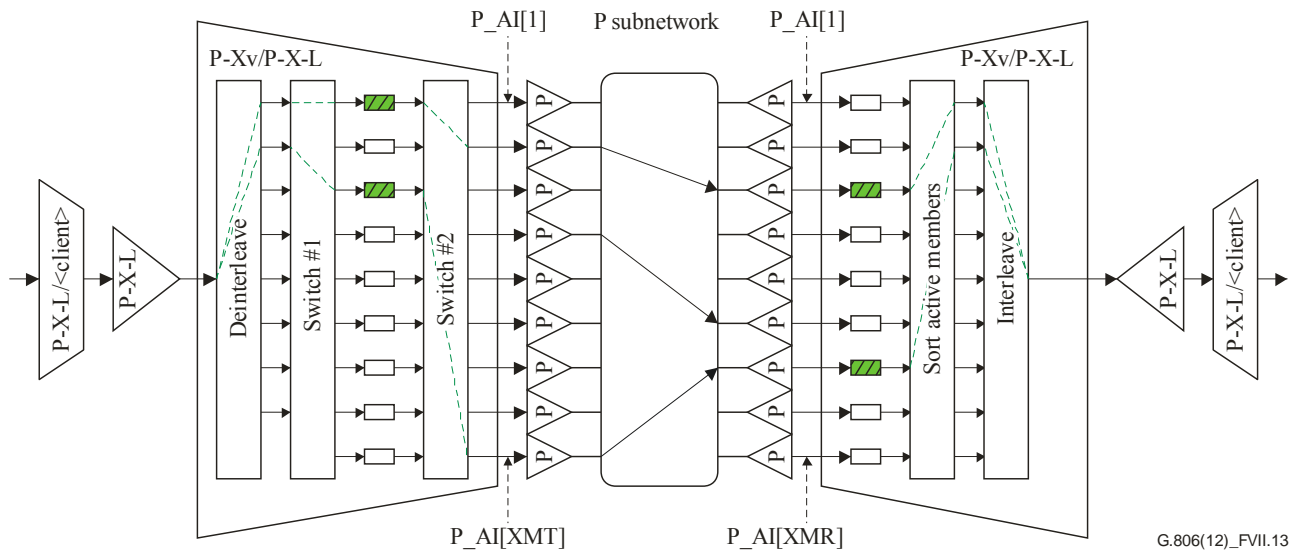
NOTE – Hatched (green-coloured) items indicate the path taken by the CI_D payload.
"X" represents the failure path

Figure VII.12 – Status after the failure of a member and reaction by the sink and source

VII.2.5 Scenario 5: Member recovery

When a failed member recovers, the LCAS protocol reinstates that member into service, continuing operation with the expanded set of active members. In this scenario, the recovery of one of the members is exercised.

Assuming the situation as described in the previous scenario, and the event that the member arriving at the sink via P_AI[6] recovers (Figure VII.13), the following will happen:



G.806(12)_FVII.13

NOTE – Hatched (green-coloured) items indicate the path taken by the CI_D payload.

Figure VII.13 – Status immediately after the recovery of the member arriving at P_AI[6] at the sink (neither the sink nor the source have yet reacted to the recovery)

- For the sink:
 - *MFI extract*
The only change is that AI_TSF[6] is not present any more and, hence, _MFI[6] will be recovered normally (and dLOM[6] cleared).
 - *Delay calculation*
Now for the three P_APs for which MI_ProvM[i] = 1 and AI_TSF[i] is not active (k = 3, 6, 7), this process will calculate _D[i] and MI_DMFI[i] as required. For the other members, AI_TSF[i] is active and therefore MI_DMFI[i] = n/a, _D[i] = ffs.
In other words, P_AI[6] will be considered again for multiframe alignment.
 - *Delay*
Will multiframe-align P_AI[3, 6, 7] and delay the rest by _D[i] = ffs.
 - *LCAS engine*
Among the inputs to this process, the ones that will change will be those related to the inactive _TSF[6] (in bold below):

k	1	2	3	4	5	6	7	8	9
MI_ProvM[i]	0	0	1	0	0	1	1	0	0
dLOM[i]	T	T	F	T	T	F	F	T	T
_TSF[i]	T	T	F	T	T	F	F	T	T
_CRC_z[i]	X	X	F	X	X	F	F	X	X
_CRC_ok[i]	X	X	T	X	X	T	T	X	X
_CTRL[i]	X	X	Norm	X	X	DNU	EOS	X	X
_SQ[i]	X	X	0	X	X	1	2	X	X

This process, hence, will output:

- (Unchanged: MI_LCAS_So_Detected = true, _LCASActive = true).
- $_X_{AR} = 2$, $_PC[3, 7] = 1$, $_PC[1, 2, 4, 5, 6, 8, 9] = 0$, $dSQM[1..X_{MR}] = \text{false}$ (no changes).
- RI_XXX will be as defined by [ITU-T G.7042] ($RI_MST_gen[0..2] = 0$, $RI_MST_gen[3..8] = 1$). That is, the sink will start signalling to the source that the failure has cleared for sequence number 1.
- For $_SQv[1..X_{MR}]$ the value for $i = 6$ will change:

k	1	2	3	4	5	6	7	8	9
$_SQv[i]$	n/a	n/a	0	n/a	n/a	1	2	n/a	n/a

- And still $_RI_Selector = 3$.
- *Sort active members, interleave*
Since there are no changes in $_PC[i]$ or $_X_{AR}$, these processes continue recovering a path-layer-2c signal by interleaving the two path-layer signals at the inputs with $_PC[i] = 1$.
- *AIS generator + AIS insertion*
Since $aAIS = \text{false}$, no AIS signal will be inserted towards the P-X-L_CI.
- The following will also be signalled towards the P-X-L_CI: $CI_SSF = \text{false}$, $CI_X_{AR} = 2$ (no changes).
- The following will also be signalled towards the P-X-L_MI: $MI_X_{MR} = 9$, $MI_X_{AR} = 2$, $MI_DMFI[3, 6, 7] = \text{xxx}$, $MI_DMFI[1, 2, 4, 5, 8, 9] = \text{n/a}$, $MI_cLOM[1..X_{MR}] = \text{false}$, $MI_cSQM[1..X_{MR}] = \text{false}$, $MI_cLOA = \text{false}$, $MI_cPLCR = \text{false}$, $MI_cTLCR = \text{false}$ ($MI_DMFI[6]$ change).
- $MI_Ac_SQ[1..X_{MR}] = _SQv[1..X_{MR}]$ (see above).

As a result, the sink will start considering the recovered member P_AI[6] for realignment and LCAS purposes. At the same time, it will start signalling towards the source that the failure condition was removed for sequence number 1.

- For the source, the following reaction will happen as soon as it receives the $RI_MST[1] = 0$ (OK), which is being reported by the sink:
 - *LCAS engine*
 - Will set $_CTRL[1] = \text{NORM}$, as per [ITU-T G.7042]. Accordingly, $_X_{AT} = 3$, $_CTRL[0] = \text{NORM}$, $_CTRL[2] = \text{EOS}$, $_CTRL[3..8] = \text{IDLE}$, $_PC[0..2] = 1$, $_PC[3..8] = 0$.
 - $_SQmap[i]$ does not change, since each sequence number continues to be carried over the same P_AI[i] signals. It will remain:

k	1	2	3	4	5	6	7	8	9
$_SQmap[i]$	n/a	0	n/a	1	n/a	n/a	n/a	n/a	2

- *Deinterleave*
Since $_X_{AT} = 3$, this process will spread the CI_D information¹⁶ over its outputs 1..2 and insert all-zero path-layer signals at all its other outputs.
- *"Switch 1"*
Since $_PC[0..2] = 1$ and $_PC[3..8] = 0$, this process will connect inputs 1, 2, 3 to outputs 0..2 and insert all-zero path-layer signals at all its other outputs.
- *"Switch 2"*
Will just connect output i to input $_SQmap[i]$.

As a result, the source will start mapping payload onto $P_AI[4]$ and will enlarge the available bandwidth to the client layers to $CI_X_{AR} = 3$. This client bandwidth will be mapped onto the three provisioned members. For $P_AI[4]$, a signal with a NORM control word, a sequence number as per $_SQmap[4]$ will be sourced, indicating to the sink that this member again carries payload.

– As soon as these signalling changes from the source arrive at the sink, the following will change at that function:

- *MFI extract, delay calculation, delay*: No changes.
- *LCAS engine*

Among the inputs to this process, the one that will change will be the control word for $i = 6$ (in bold below):

k	1	2	3	4	5	6	7	8	9
MI_ProvM[i]	0	0	1	0	0	1	1	0	0
dLOM[i]	T	T	F	T	T	F	F	T	T
_TSF[i]	T	T	F	T	T	F	F	T	T
_CRC_z[i]	X	X	F	X	X	F	F	X	X
_CRC_ok[i]	X	X	T	X	X	T	T	X	X
_CTRL[i]	X	X	Norm	X	X	Norm	EOS	X	X
_SQ[i]	X	X	0	X	X	1	2	X	X

This process will hence start accepting payload from sequence number 1 and output:

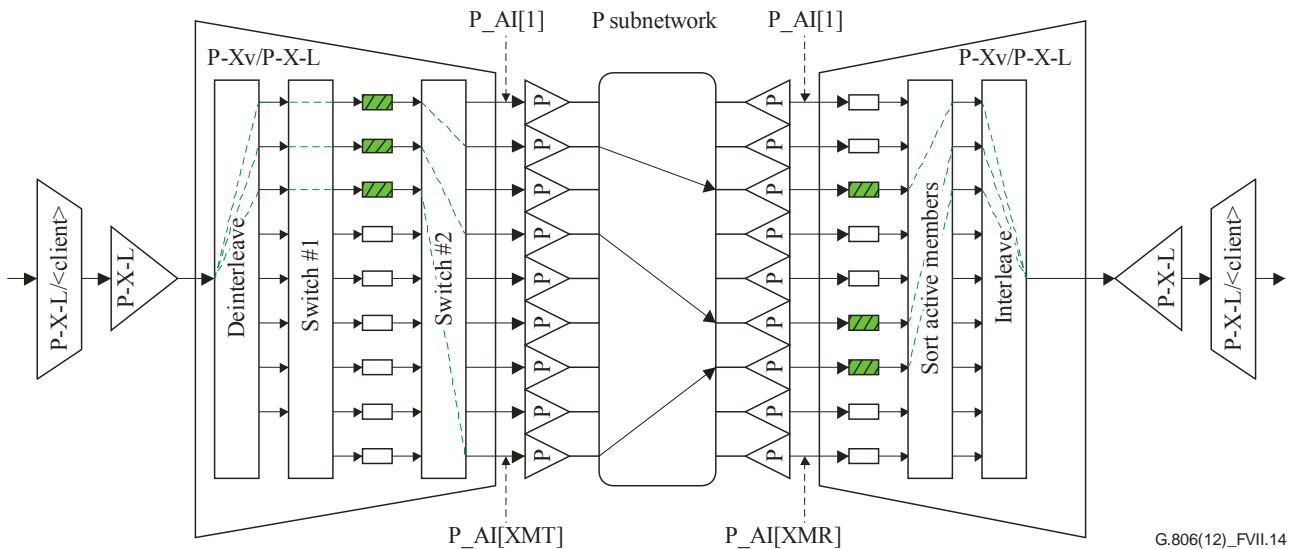
- (Unchanged: $MI_LCAS_So_Detected = true$, $_LCASActive = true$).
- $_X_{AR} = 3$, $_PC[3, 6, 7] = 1$, $_PC[1, 2, 4, 5, 8, 9] = 0$, $dSQM[1..X_{MR}] = false$ ($_PC[6]$ changes).
- RI_xxx will be as defined by [ITU-T G.7042] ($RI_MST_gen[0..2] = 0$, $RI_MST_gen[3..8] = 1$). That is, no changes.
- $_SQv[1..X_{MR}]$ will not change:

k	1	2	3	4	5	6	7	8	9
_SQv[i]	n/a	n/a	0	n/a	n/a	1	2	n/a	n/a

¹⁶ Which is expected to be a path-layer-3c, since $CI_X_{AT} = 3$ is being communicated to the upper-layer functions.

- *Sort active members*
This process will connect inputs 3, 6, 7 to outputs 1, 2, 3, respectively. For the other outputs, this process will insert all-zero path-layer signals.
- *Interleave*
Since $_X_{AR} = 3$, this process will recover a path-layer-3c signal by interleaving the 3 path-layer signals at its inputs 1..3.
- The following will also be signalled towards the P-X-L_CI: $CI_SSF = \text{false}$, $CI_X_{AR} = 3$ (CI_X_{AR} changes).
- The following will also be signalled towards the P-X-L_MI: $MI_X_{MR} = 9$, $MI_X_{AR} = 3$, $MI_DMFI[3, 6, 7] = \text{xxx}$, $MI_DMFI[1, 2, 4, 5, 8, 9] = \text{n/a}$, $MI_cLOM[1..X_{MR}] = \text{false}$, $MI_cSQM[1..X_{MR}] = \text{false}$, $MI_cLOA = \text{false}$, $MI_cPLCR = \text{false}$, $MI_cTLCR = \text{false}$ (MI_X_{AR} changes).
- $MI_Ac_SQ[1..X_{MR}] = _SQv[1..X_{MR}]$ (see above).

As a result, the sink will start accepting payload from the restored member $P_AI[6]$ and will increase the bandwidth forwarded towards the client functions to $CI_X_{AR} = 3$. This situation is shown in Figure VII.14.



NOTE – Hatched (green-coloured) items indicate the path taken by the CI_D payload.

Figure VII.14 – Steady state after the recovery of the member arriving at $P_AI[6]$ at the sink

VII.3 LCAS-enabled So and LCAS-disabled Sk functions

For further study.

VII.4 LCAS-disabled So and LCAS-enabled Sk functions

For further study.

VII.5 Non-virtual-concatenation So and LCAS-capable virtual-concatenation Sk functions

For further study.

Appendix VIII

Consequent actions for signals with no AIS/FDI defined

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

In transport networks in the case of server layer defects, normally an AIS/FDI is generated for the downstream client signal at the server/client adaptation sink function. The AIS/FDI signal is an indication to the downstream network elements that the client signal is missing due to a server layer defect. AIS/FDI suppresses downstream alarms and initiates protection/restoration actions at the client layer, if applicable.

Some client signals may not have an AIS/FDI signal defined due to various reasons (e.g., not needed in the original application of the client signal, the signal was originally at the bottom of the layer stack and it was not expected to be transported over a server layer).

In the case where no client signal AIS/FDI is available, but a downstream indication of the defect condition is needed (e.g., to trigger protection switching) the following actions can be considered at the output port for the client signal:

- Turn off the output transmitting device.
- Insert error code (e.g., /V/, 10B_ERR for 1 GbE).

These actions are only possible in the case where a single instance of a client signal is transported over the output port, as all the client signal instances at the output port are affected by such actions. Note that these actions will result in server layer defects at the downstream input port which may lead to the wrong assumption that the server layer trail has a problem, which is not the case (see Figure VIII.1).

Other options are:

- Insert signal fail message (if defined for that particular client signal and a client management channel exists between the network elements, e.g., CSF in GFP).
- No action.

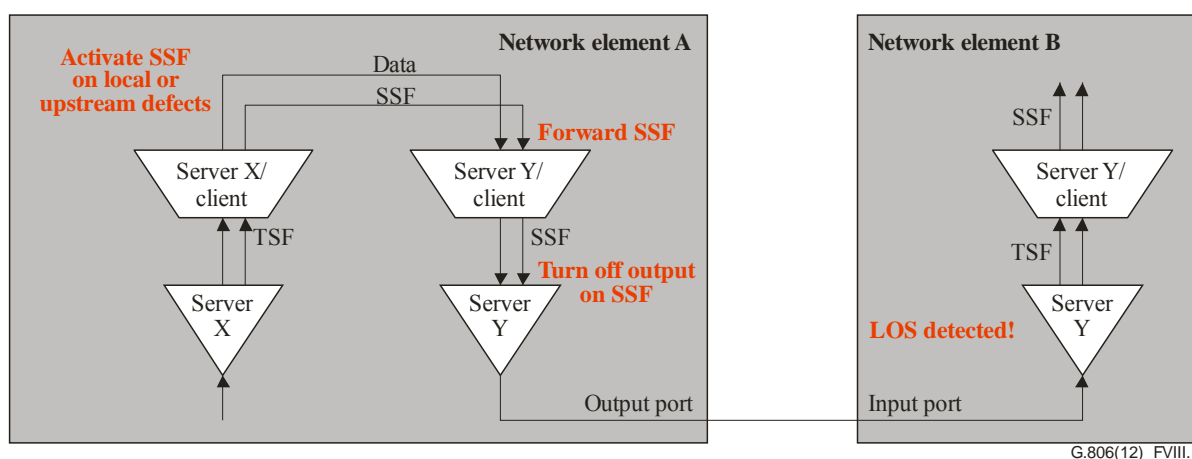


Figure VIII.1 – Example of "turn off output" action

Appendix IX

Types of flow forwarding processes

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

There are two main classes of flow forwarding processes; the flow unaware forwarding processes and the flow aware forwarding processes.

A flow unaware forwarding process forwards information applied at an input port to all the connected output port or ports or to a subset of those output ports. The forwarding is performed independent of information within the characteristic information.

- Connectivity of a Type Ia and Type IIa flow forwarding process cannot be changed. Information applied at the input port is forwarded to all output ports.
- Connectivity of a Type Ib flow forwarding process can be changed under the control of MI, in which case a loopback can be established or removed; refer to [ITU-T M.125] for loopback types and their specific connectivity.
- Connectivity of a Type III, IVa and IVb flow forwarding process can be modified under the control of signal fail/degrade states of the incoming signals themselves or external protection switching commands (i.e., protection switching).

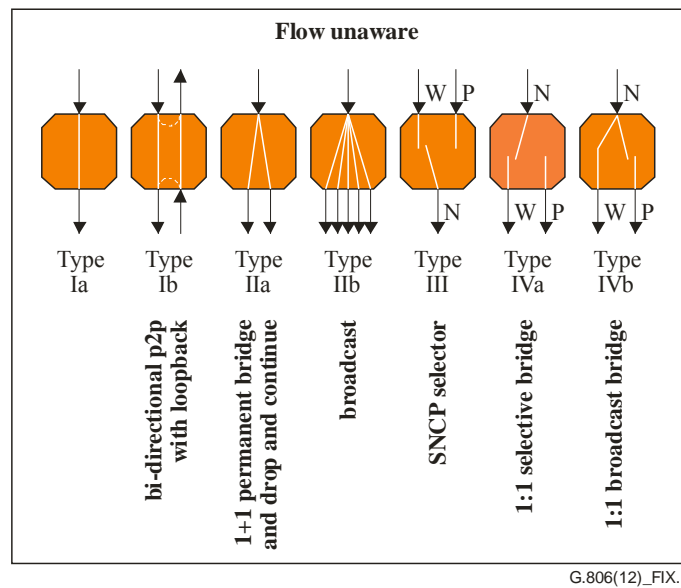
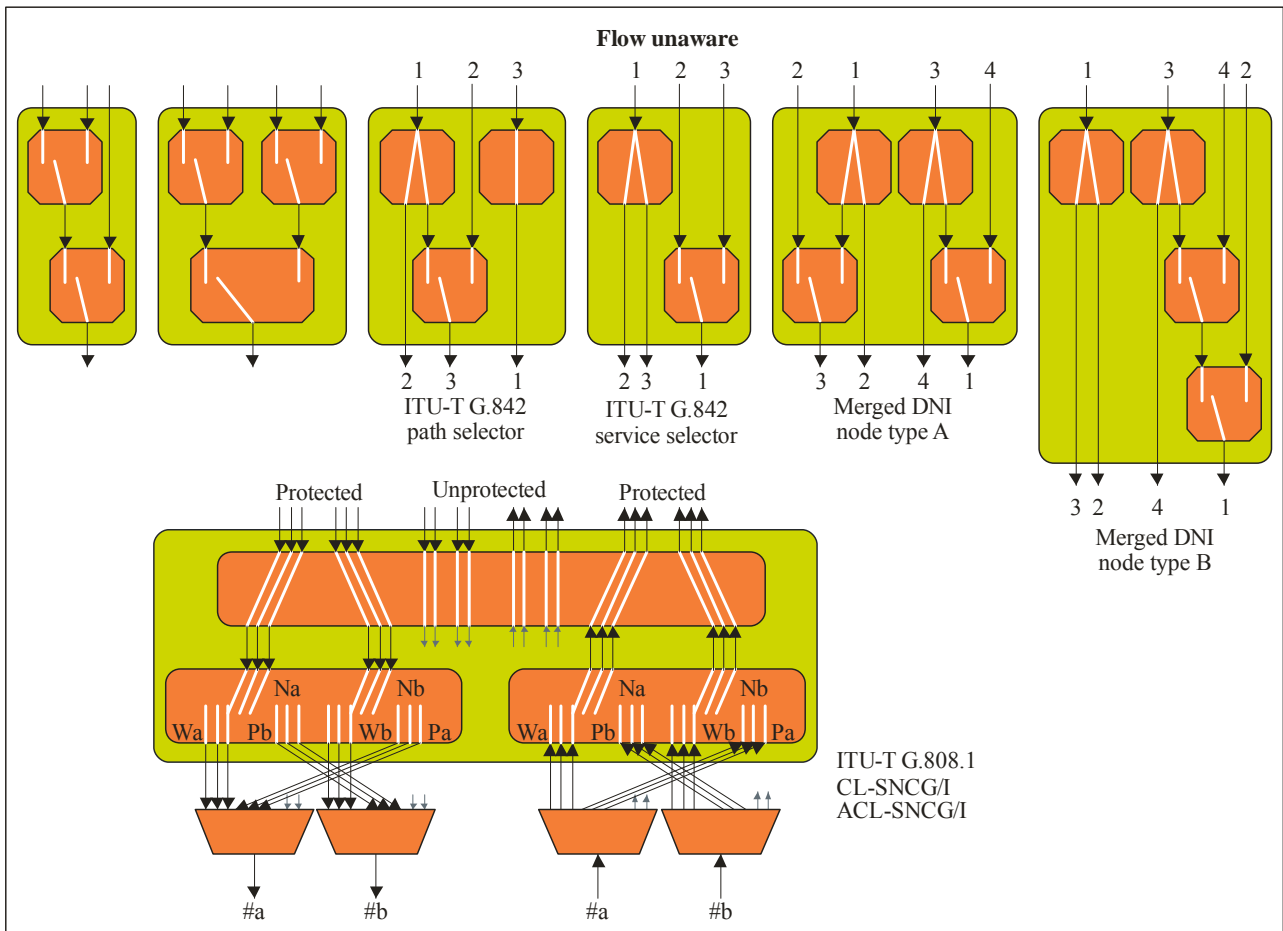


Figure IX.1 – Basic flow unaware forwarding process types

Flow unaware forwarding processes may be compounded to create more complex flow forwarding processes. Figure IX.1 illustrates a number of such compound flow unaware forwarding processes which support specific protection switching capabilities.

- The two top left compound flow forwarding processes support two stage SNC protection switching selectors with three or four input signals.
- The two top middle compound flow forwarding processes support the ITU-T G.842 path selector and service selector.

- The two top left compound flow forwarding processes support two types of dual node interconnect (DNI) protection for the case where two subnetworks are interconnected via two nodes which have NNI ports in both subnetworks. It provides for a variation of dual node interconnection scenarios described in [ITU-T G.842], in which the interfaces between the two subnetworks are virtualized; i.e., supported within the switch fabric of one node.
- The bottom compound flow forwarding process supports the [ITU-T G.808.1] (adaptive) compound link SNC group protection with load sharing.



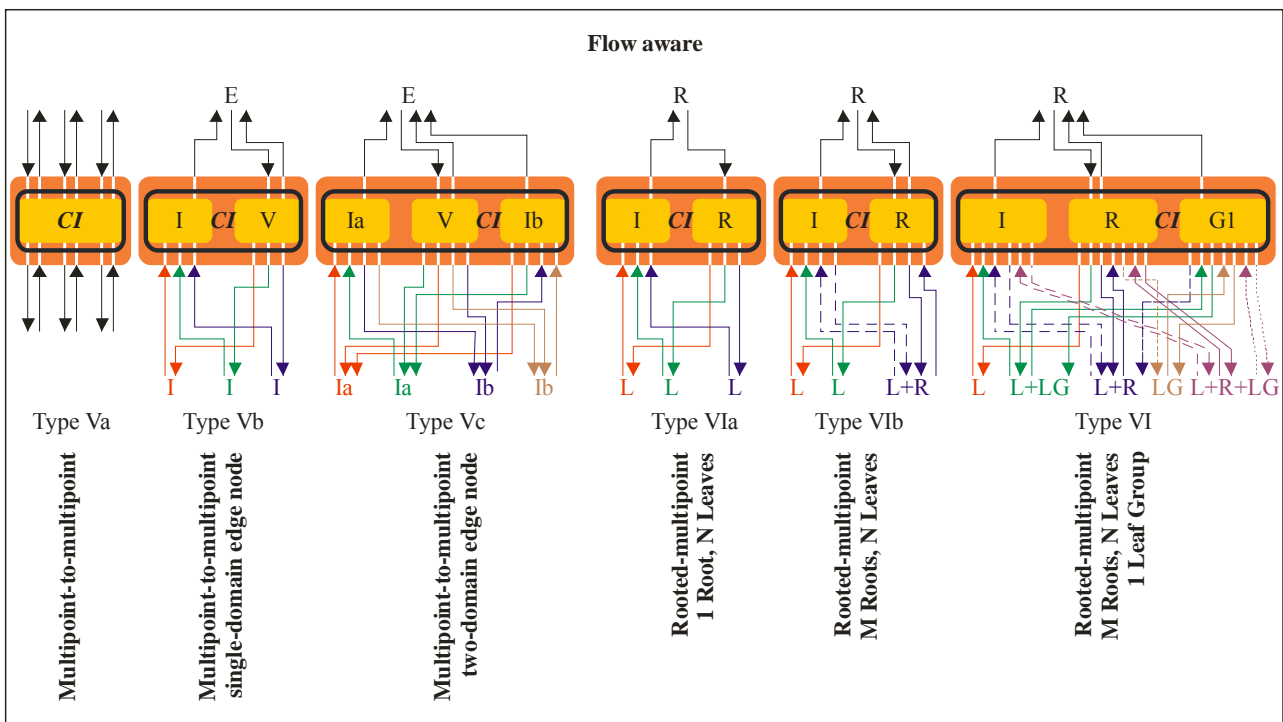
G.806(12)_FIX.2

Figure IX.2 – compound flow unaware forwarding process types

A flow aware forwarding process forwards information applied at an input port either to one of the output ports, or to a subset of the output ports, or to all output ports with the exception of the output port associated with the input port. The forwarding is performed on the basis of information elements within the characteristic information and configured relationships (under control of local learning, network management and/or control plane) associating specific values of these information elements with one or more or all of the output ports.

- Connectivity of a Type Va flow forwarding process is any input to any output with the exception of the output port associated with the input port.
- Connectivity of a Type Vb flow forwarding process is:
 - from an external input port E to any internal output port I and any external output port E with the exception of the output port E associated with the input port E;
 - from an internal input port I to any external output port E.

- Connectivity of a Type Vc flow forwarding process is:
 - from an external input port E to any internal output port I and any external output port E with the exception of the output port E associated with the input port E;
 - from an internal input port Ia or Ib to any external output port E;
 - from an internal input port Ia to any internal output port Ib;
 - from an internal input port Ib to any internal output port Ia.
- Connectivity of a Type VIa flow forwarding process is:
 - from the root input port R to any leaf output port;
 - from a leaf input port I to the root output port R.
- Connectivity of a Type VIb flow forwarding process is:
 - from a root input port R to any leaf output port L and any root output port R with the exception of the root output port R associated with the input port R;
 - from a leaf input port L to any root output port R.
- Connectivity of a Type VIc flow forwarding process is:
 - from a root input port R to any leaf output port L and any root output port R with the exception of the output port R associated with the input port R;
 - from a leaf input port L to any root output port R;
 - from a leaf group input port Gi to any root output port R and any leaf group output port Gi with the exception of the output port Gi associated with the input port Gi.



G.806(12)_FIX.3

Figure IX.3 – Basic flow aware forwarding process types

Bibliography

- [b-ITU-T G.828] Recommendation ITU-T G.828 (2000), *Error performance parameters and objectives for international, constant bit-rate synchronous digital paths*.
- [b-ITU-T G.829] Recommendation ITU-T G.829 (2002), *Error performance events for SDH multiplex and regenerator sections*.
- [b-ITU-T G.957] Recommendation ITU-T G.957 (2006), *Optical interfaces for equipments and systems relating to the synchronous digital hierarchy*.
- [b-ITU-T G.8201] Recommendation ITU-T G.8201 (2003), *Error performance parameters and objectives for multi-operator international paths within the Optical Transport Network (OTN)*.
- [b-ITU-T X.731] Recommendation ITU-T X.731 (1992), *Information technology – Open Systems Interconnection – Systems management: State management function*.
- [b-ITU-T Handbook] ITU-T Handbook (2004), *Quality of Service and Network Performance*, ITU, Geneva.
- [b-IEEE 802.3] IEEE 802.3 (in force), *IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) access method and Physical Layer specifications – Section Five*.
<http://ieeexplore.ieee.org/xpl/standards.jsp>
- [b-Cornaglia] Cornaglia, B., Pane, P., and Spini, M. (1995), *Errored block detection with bit interleaved parity failures in SDH network*, IEEE Transactions on Communications, Vol. 43, No. 12, December; pp. 2904-2906.

SERIES OF ITU-T RECOMMENDATIONS

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