International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

G.8021/Y.1341

Amendment 1

(06/2006)

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

Ethernet over Transport aspects – General aspects

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

Internet protocol aspects – Transport

Characteristics of Ethernet transport network equipment functional blocks

Amendment 1

ITU-T Recommendation G.8021/Y.1341 (2004) – Amendment 1

ITU-T

ITU-T G-SERIES RECOMMENDATIONS

TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS

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

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

ITU-T Recommendation G.8021/Y.1341

Characteristics of Ethernet transport network equipment functional blocks

Amendment 1

Summary

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

This amendment contains additional material to be incorporated into ITU-T Recommendation G.8021/Y.1341. It presents additional functions required to support link aggregation and Ethernet OAM.

Source

Amendment 1 to ITU-T Recommendation G.8021/Y.1341 (2004) was approved on 6 June 2006 by ITU-T Study Group 15 (2005-2008) under the ITU-T Recommendation A.8 procedure.

FOREWORD

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

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

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

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

NOTE

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

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

INTELLECTUAL PROPERTY RIGHTS

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

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

CONTENTS

ITU-T Recommendation G.8021/Y.1341

Characteristics of Ethernet transport network equipment functional blocks

Amendment 1

1 Scope

This amendment provides updated material describing the functions required to support link aggregation and a subset of the Ethernet OAM, i.e., clauses 8.1 (OAM related processes), 9 (Ethernet layer functions), 9.1 (ETH_FD function), 9.2 (ETHx/ETH_A adaptation function), 9.3 (ETHG/ETH_A adaptation function), 9.5 (<server>/ETH_A adaptation function) and 9.7 (ETH link aggregation function).

The intent of ITU-T Rec. G.8021/Y.1341 is to be aligned with the ongoing work in IEEE 802.1ag. It is expected that IEEE 802.1ag will be approved before the approval of the next version of this Recommendation. Inconsistencies between G.8021/Y.1341 (2004) + Amendment 1 and IEEE 802.1ag will be taken into account in the development of the next version of this Recommendation.

2 References

– ITU-T Recommendation Y.1731 (2006), OAM *functions and mechanisms for Ethernet based networks*.

– IEEE Std. 802.1ad-2005, *IEEE Standards for Local and Metropolitan Area Networks Virtual Bridged Local Area Networks – Revision – Amendment 4: Provider Bridges*.

– IEEE Std. 802.1X-2004, *IEEE Standards for Local and Metropolitan Area Networks: Port-Based Network Access Control*.

– IEEE Std. 802.1D-2004, *IEEE standard for local and metropolitan area networks: Media Access Control (MAC) Bridges*.

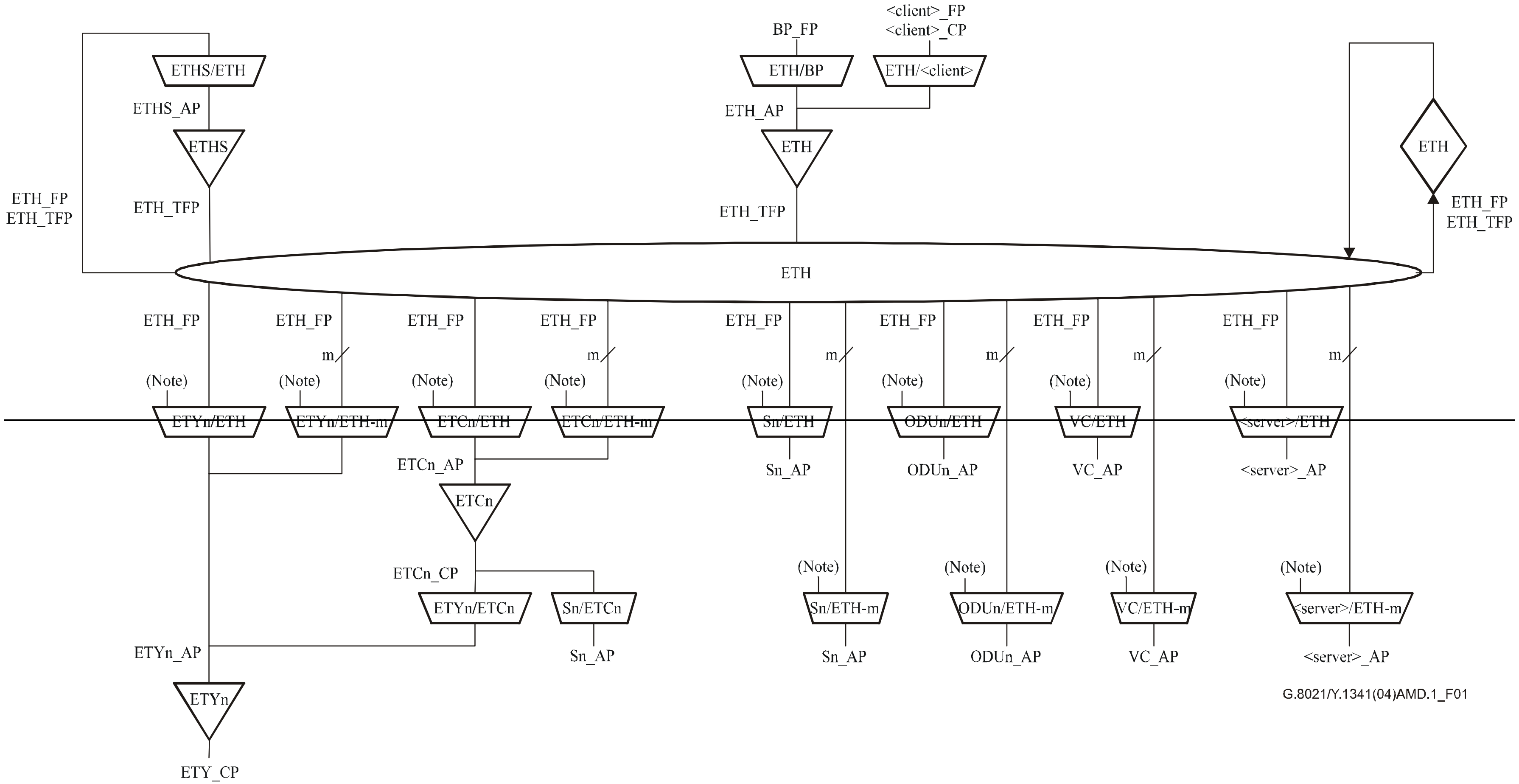
– IEEE Std. 802.1Q-2005, *IEEE Standard for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks*.

3 Conventions and Methodology

Characteristic and adapted information signals consist of D <Traffic Unit Data>, DE <Drop Eligible> and P <Priority> signals.

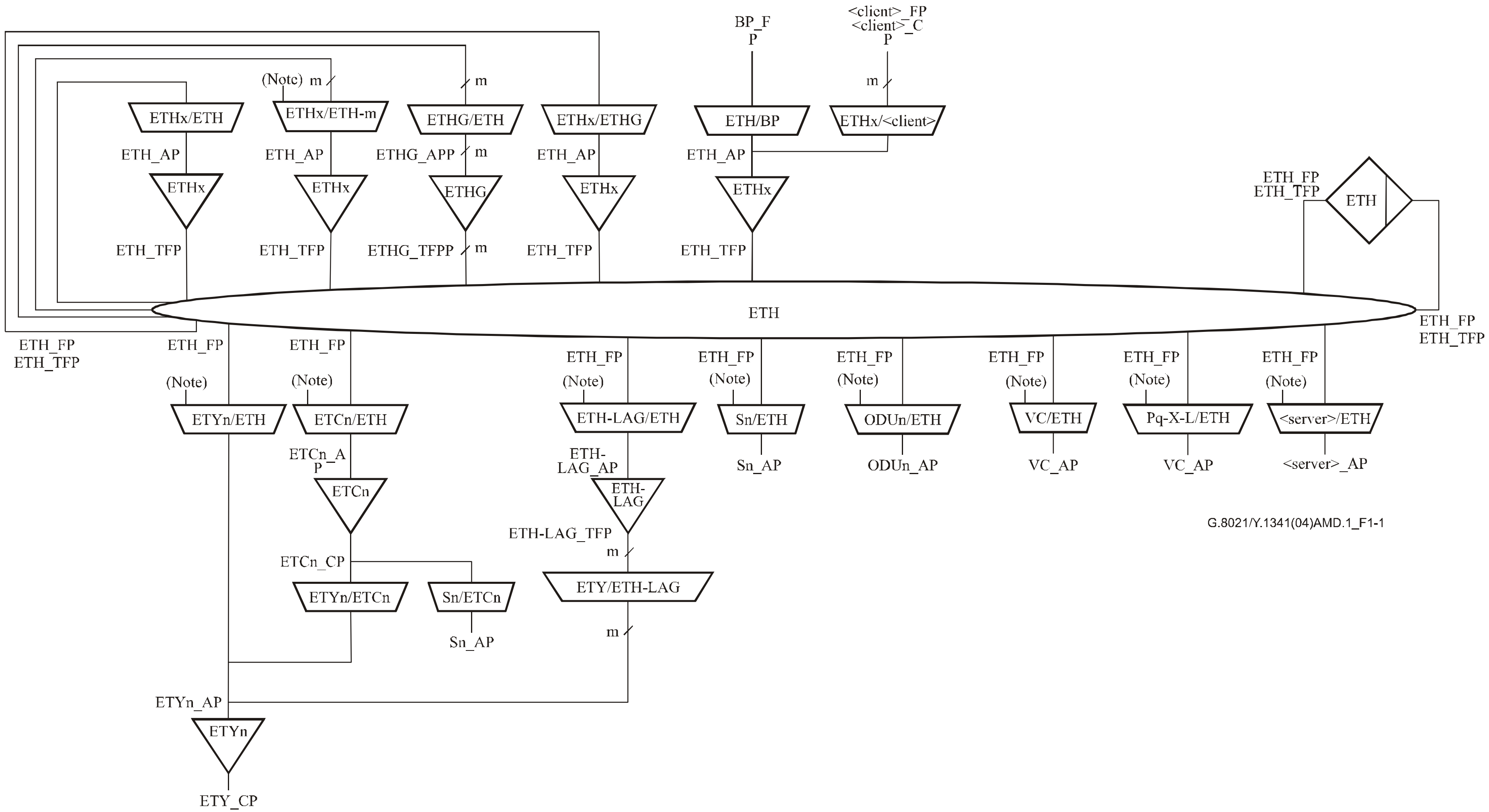
4 Changes to clause 1

Replace Figure 1 with Figure 1-1.



NOTE – ETH_TFP interface of adaptation functions towards the ETH_FT functions for logical link control. See G.8010 and function definition for details.

Figure 1 – Overview of G.8021/Y.1341 Atomic Model Functions



NOTE – ETH_TFP interface of adaptation functions towards the ETH_FT functions connects to logical link control. See ITU-T Rec. G.8010/Y.1306 and function definition for details.

Figure 1-1 – Overview of G.8021/Y.1341 atomic model functions

5 Changes to clause 8 Generic processes

5.1 Changes to clause 8

Replace text between header clause 8 and clause 8.1 by:

Generic processes are defined in clause 8/G.806. This clause defines processes specific to equipment supporting the Ethernet transport network.

5.2 Changes to clause 8.1

Replace the text with the following:

8.1 Mux/Demux process

For further study.

With the following text:

8.1 OAM-related processes

8.1.1 OAM MEG level filter process

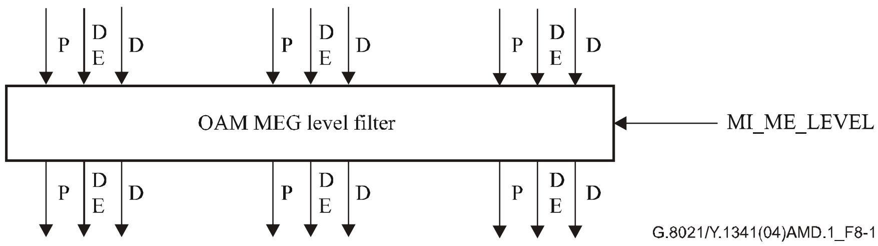


Figure 8-1 – OAM MEG level filter process

The OAM MEG level filter process filters incoming ETH OAM traffic units based on the MEG level they carry. All traffic units with an ME level equal to or lower than the MEG level provided by the MI_ME_Level signal are discarded.

The criteria for filtering depend on the values of the fields in the M_SDU field of the ETH_CI_D signal:

- Length/type field = OAM Ethertype (for further study); and
- ME level field ≤ MI_ME_Level.

Note that in the next version of this Recommendation the OAM Ethertype will be assigned the value as will be defined in IEEE 802.1ag.

8.1.2 LCK generate process

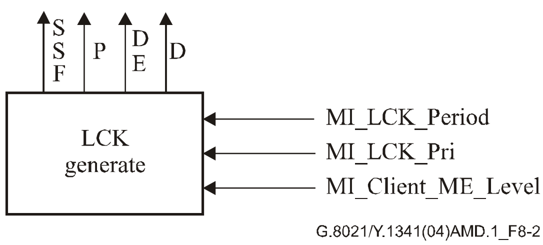


Figure 8-2 – LCK generate process

The LCK generate process generates ETH_CI traffic units where the ETH_CI_D signal contains the LCK signal.

The ETH_CI_D signal contains source and destination address fields and an M_SDU field. The format of the M_SDU field is defined in 9.1 and 9.8/Y.1731. The ME level in the M_SDU field is determined by the MI_Client_ME_Level input parameter.

The values of the source and destination address fields in the M_SDU field are determined by the Local MAC address (SA) and the multicast class 1 DA as described in ITU-T Rec. Y.1731 (DA). Note that in the next version of this Recommendation the class multicast 1 DA will be assigned the value as will be defined in IEEE 802.1ag.

The value of the ETH_CI_P signal associated with the generated LCK traffic unit is defined by the MI_LCK_Pri input parameter; valid values are in the range 0-7.

The value of the ETH_CI_DE signal associated with the generated LCK traffic units is always set to drop ineligible.

The period between two consecutive traffic units is determined by the MI_LCK_Period input signal. Allowed values are once per second and once per minute.

Figure 8-2a below shows the M_SDU signal format where the LCK-specific values are shown.

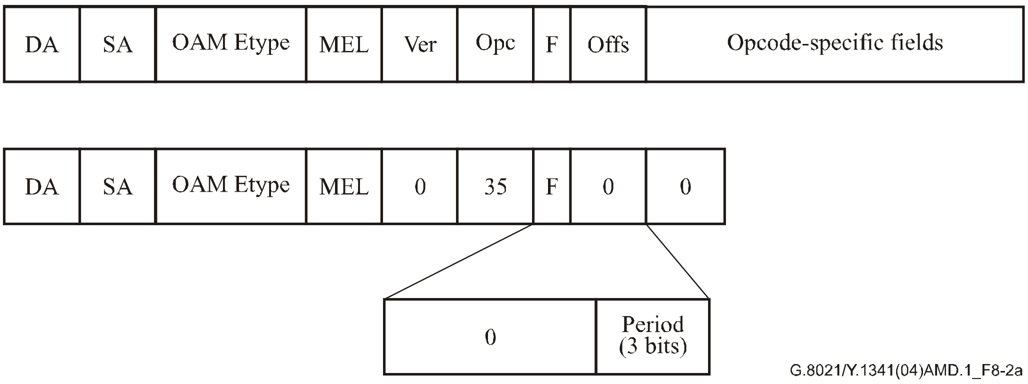


Figure 8-2a – M_SDU signal format for the LCK generate process

8.1.3 Selector process

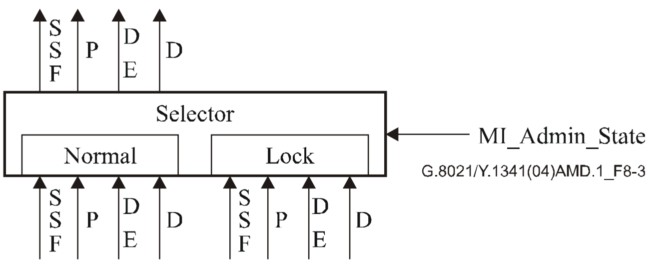


Figure 8-3 – Selector process

The selector process may replace the normal ETH_CI signal by the lock ETH_CI signal (as generated by the LCK generate process). The normal signal is replaced if the MI_Admin_State is LOCKED.

8.1.4 AIS insert process

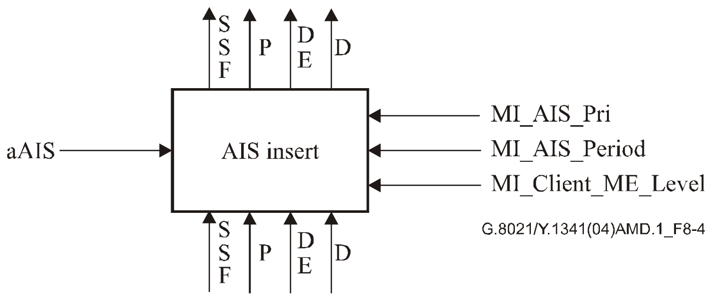


Figure 8-4 – AIS insert process

If the aAIS signal is present, the AIS insert process continuously generates ETH_CI traffic units where the ETH_CI_D signal contains the AIS signal. The generated AIS traffic units are inserted in the incoming stream, i.e., the output stream contains the incoming traffic units and the generated AIS traffic units.

The effect of the aAIS signal on the SSF signal is *for further study*.

The ETH_CI_D signal contains source and destination address fields and an M_SDU field. The format of the M_SDU field for AIS traffic units is defined in 9.1 and 9.7/Y.1731. The ME level in the M_SDU field is determined by the MI_Client_ME_Level input parameter.

The values of the source and destination address fields in the M_SDU field are determined by the local MAC address (SA) and the multicast class 1 DA as described in ITU-T Rec. Y.1731 (DA). Note that in the next version of this Recommendation the multicast class 1 DA will be assigned the value as will be defined in IEEE 802.1ag.

The value of the ETH_CI_P signal associated with the generated AIS traffic units is defined by the MI_AIS_Pri input parameter.

The value of the ETH_CI_DE signal associated with the generated AIS traffic units is always set to drop ineligible.

The period between the generation of consecutive AIS traffic units is determined by the MI_AIS_period parameter. Allowed values are once per second and once per minute.

Figure 8-4a below shows the M_SDU signal format, where the AIS-specific values are shown.

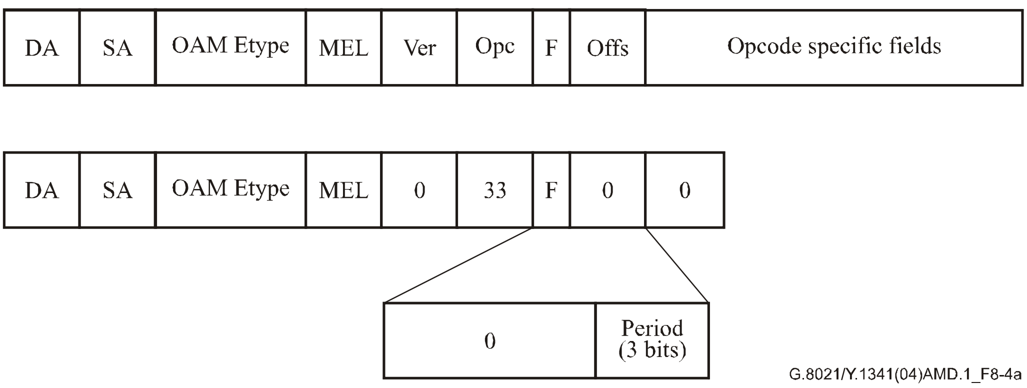


Figure 8-4a – M_SDU signal format for the AIS insert process

8.1.5 APS insert process

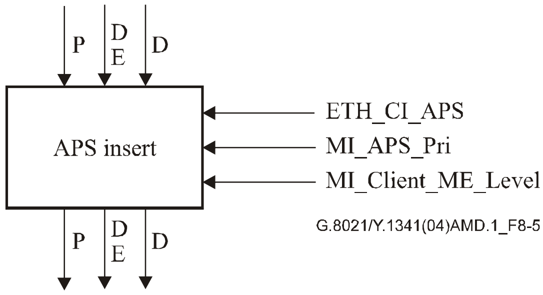


Figure 8-5 – APS insert process

The APS insert process encodes the ETH_CI_APS signal into the ETH_CI_D signal of an ETH_CI traffic unit; the resulting APS traffic unit is inserted into the stream of incoming traffic units, i.e., the outgoing stream consists of the incoming traffic units and the inserted APS traffic units.

The ETH_CI_D signal contains source and destination address fields and an M_SDU field. The format of the M_SDU field for APS traffic units is defined in 9.1 and 9.10/Y.1731. The ME level in the M_SDU field is determined by the MI_Client_ME_Level input parameter.

The values of the source and destination address fields in the M_SDU field are determined by the local MAC address (SA) and the multicast class 1 DA or unicast address as described in ITU-T Rec. Y.1731 (DA). Note that in the next version of this Recommendation the class Multicast 1 DA will be assigned the value as will be defined in IEEE 802.1ag.

The value of the ETH_CI_P signal associated with the generated APS traffic units is determined by the MI_APS_Pri input parameter.

The value of the ETH_CI_DE signal associated with the generated APS traffic units is set to drop ineligible.

8.1.6 APS extract process

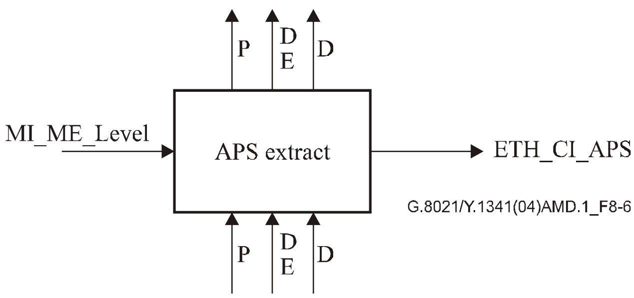


Figure 8-6 – APS extract process

The APS extract process extracts ETH_CI_APS signals from the incoming stream of ETH_CI traffic units. ETH_CI_APS signals are only extracted if they belong to the ME level as defined by the MI_ME_Level input parameter.

If an incoming traffic unit is an APS traffic unit belonging to the ME level defined by MI_ME_Level, the ETH_CI_APS signal will be extracted from this traffic unit and the traffic unit will be filtered. All other traffic units will be transparently forwarded. The encoding of the ETH_CI_D signal for APS frames is defined in 9.10/Y.1731.

The criteria for filtering are based on the values of the fields within the M_SDU field of the ETH_CI_D signal:

- length/type field equals the OAM Ethertype (*for further study*);
- ME level field equals MI_ME_Level; and
- OAM type equals APS (39), as defined in 9.1/Y.1731.

Note that in the next version of this Recommendation the OAM Ethertype will be assigned the value as will be defined in IEEE_802.1ag.

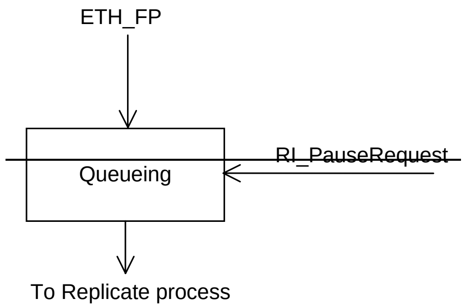
5.3 Changes to clause 8.2

As indicated by changebars:

8.2 Queuing Process

~~The queuing process buffers received ETH frames for output (see Figure 3). The queuing process is also responsible for dropping frames if their rate at the ETH_CI is higher than the <Srv>_AI_D can accommodate, as well as maintaining PM counters for dropped frames. Additional performance monitor counters per IEEE 802.3-2002 clause 30 are *for further study*.~~

~~In response to RI_PauseRequest asserted, the Queuing process halts the flow of frames to the Replicate process. Note that RI_PauseRequest is not connected in transport network equipment.~~



~~**Figure 3/G.8021/Y.1341 – Queuing process**~~

The queuing process buffers the received ETH_CI_D for output (see Figure 8-7). The queuing process is also responsible for discarding frames if their rate at the ETH_CI_D is higher than the <server>_AI_D can accommodate, as well as maintaining PM counters for discarded frames. Additional performance monitor counters (MI_PM_count) per IEEE 802.1Q are *for further study*.

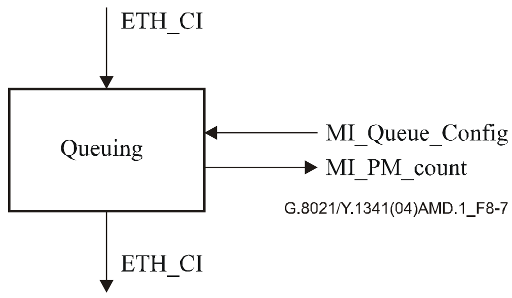


Figure 8-7 – Queuing process

The queuing process is configured using the MI_Queue_Config input parameter. This parameter specifies the mapping of ETH_CI_D into the available queues based on the value of the ETH_CI_P signal.

Furthermore it specifies whether the value of the ETH_CI_DE signal should be taken into account when discarding frames. If this needs to be taken into account, ETH_CI with ETH_CI_DE set to drop eligible should have a higher probability of being discarded than ETH_CI with ETH_CI_DE set to drop ineligible.

5.4 Remove clauses 8.2.1 and 8.2.2

~~8.2.1. IEEE 802.1D Queuing process~~

~~The IEEE 802.1D Queuing process is applicable to <Srv>/ETH_A_So functions. This process is defined in IEEE 802.1D (clauses 7.7.3 and 7.7.4).~~

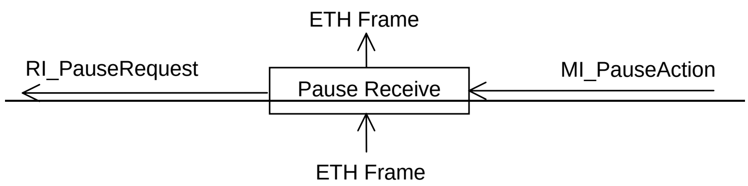
~~This process never takes into account the value of the ETH_CI_DE signal.~~

~~8.2.2. IEEE 802.1Q Queuing Process~~

~~The IEEE 802.1Q Queuing process is applicable to <Srv>/ETH-m_A_So functions. This process is defined in IEEE 802.1Q and IEEE802.1ad (clauses 8.6.6, 8.6.7 and 8.6.8).~~

5.5 Changes to clause 8.5.1.1.2

8.5.1.1.2 Pause receive process



~~**Figure 8/G.8021/Y.1341 – Receive Pause Process**~~

~~On receipt of a pause frame, the corresponding action shall be performed according to the MI_PauseAction configuration. Valid actions are "block" and "process."~~

- ~~Process: A received Pause frame results in a RI_PauseRequest, conveying the received pause_time value, to the paired <Srv>/ETH_A_So.~~
- ~~Block: Discard the received Pause frame~~

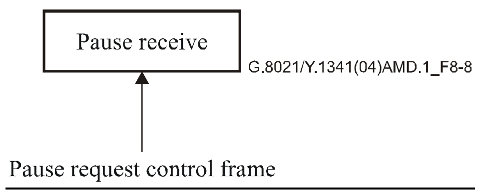


Figure 8-8 – Receive pause process

On receipt of a pause request control frame no action shall be performed (i.e., the pause request control frame shall be silently discarded).

5.6 Replace clauses 8.6 and 8.7

8.6 MAC length check

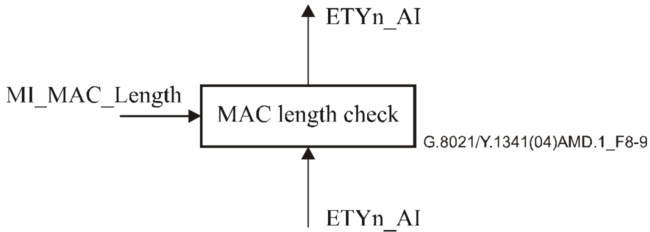


Figure 8-9 – MAC length check function

This process checks whether the length of the MAC frame is allowed. Frames shorter than 64 bytes are discarded and the corresponding counter is incremented. Frames longer than MI_MAC_Length are passed and the corresponding counter is incremented.

Table 8-1 shows the values corresponding to the IEEE-defined frame lengths.

Table 8-1 – IEEE 802.3 MI_MAC_Length values

Frame type	MI_MAC_Length
Basic	1518
Q-tagged	1522
Envelope	2000

8.7 MAC frame counter

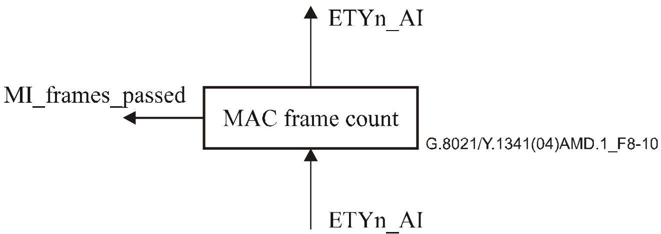


Figure 8-10 – MAC frame count function

This process passes MAC frames and counts the number of frames that are passed.

5.7 Replace clause 8.9

8.9 "Server-Specific" common processes

For some server signals MAC FCS generation is not supported. This will be defined in the server-specific adaptation functions.

8.9.1 MAC FCS generation

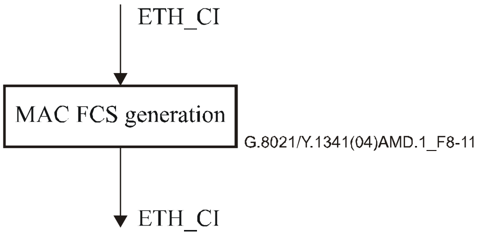


Figure 8-11 – MAC FCS generation process

The MAC FCS is calculated over the ETH_CI traffic unit and inserted into the MAC FCS fields of the frame as defined in 4.2.3.1.2/IEEE 802.3.

8.9.2 MAC frame check

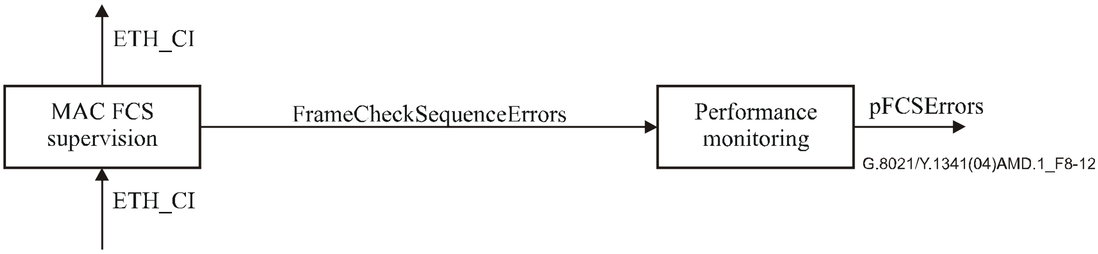


Figure 8-12 – MAC frame check process

The MAC FCS is calculated over the ETH_CI traffic unit and checked as specified in 4.2.4.1.2/IEEE 802.3. If errors are detected, the frame is discarded. Errored frames are indicated by FrameCheckSequenceErrors.

8.9.3 802.1AB/X protocols processes

802.1AB/X protocols processes include source and sink handling of 802.1AB and 802.1X protocols, as shown in Figures 8-13a and 8-13b. These processes are used in ETYn/ETH_A functions.

The following clauses specify processes for each of the illustrated process blocks.

8.9.3.1 802.1X protocol

The 802.1X protocol block implements the port-based network access control as per IEEE 802.1X-2004.

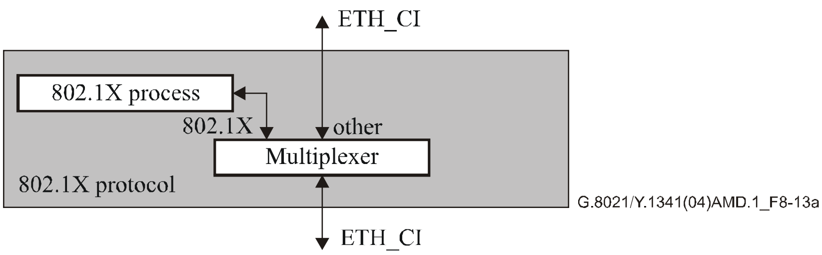


Figure 8-13a – 802.1X protocol processes

In the sink direction, the multiplexer separates the 802.1X PDUs from the rest of the frames based on MAC address 01-80-C2-00-00-03. The former are delivered to the 802.1X process, the latter are passed on in the sink direction. In the source direction, 802.1X PDUs are multiplexed with the rest of the frames.

In the function descriptions where it appears, the 802.1X process is optional.

8.9.3.2 802.1AB protocol

The 802.1AB protocol block implements the link layer discovery protocol as per IEEE 802.1AB-2004.

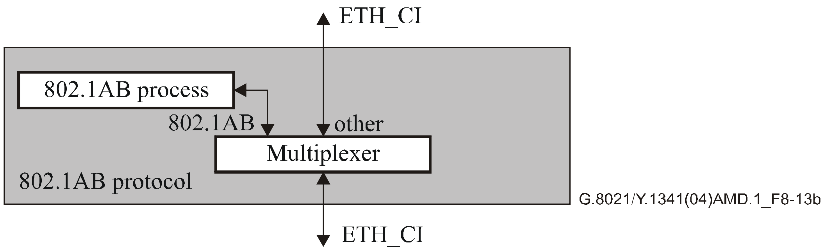


Figure 8-13b – 802.1AB protocol processes

In the sink direction, the multiplexer separates the 802.1AB PDUs from the rest of the frames. The former are delivered to the 802.1AB process, the latter are passed on in the sink direction. In the source direction, 802.1AB PDUs are multiplexed with the rest of the frames. Frames are defined by MAC address 01-80-C2-00-00-0E, Ethertype 88-CC.

In the function description where it appears, the 802.1AB process is optional.

8.9.4 Link quality supervision

Counts of transmitted and received octets and frames are maintained in <Srv>/ETH_A functions per the requirements of clause 30/IEEE 802.3. Discarded jabber frames are counted in ETYn/ETH_A_So functions.

Additional link quality performance monitors per clause 30/IEEE 802.3-2002 are *for further study*.

8.9.5 FDI/BDI generation and detection

For further study.

6 Changes to clause 9

Replace clause 9 as follow:

9 Ethernet layer functions

Figure 1-1 illustrates all the ETH layer network, server and client adaptation functions. The information crossing the ETH flow point (ETH_FP) is referred to as the ETH characteristic information (ETH_CI). The information crossing the ETH access point (ETH_AP) is referred to as ETH adapted information (ETH_AI).

ETH sublayers can be created by expanding an ETH_FP as illustrated in Figure 9-1.

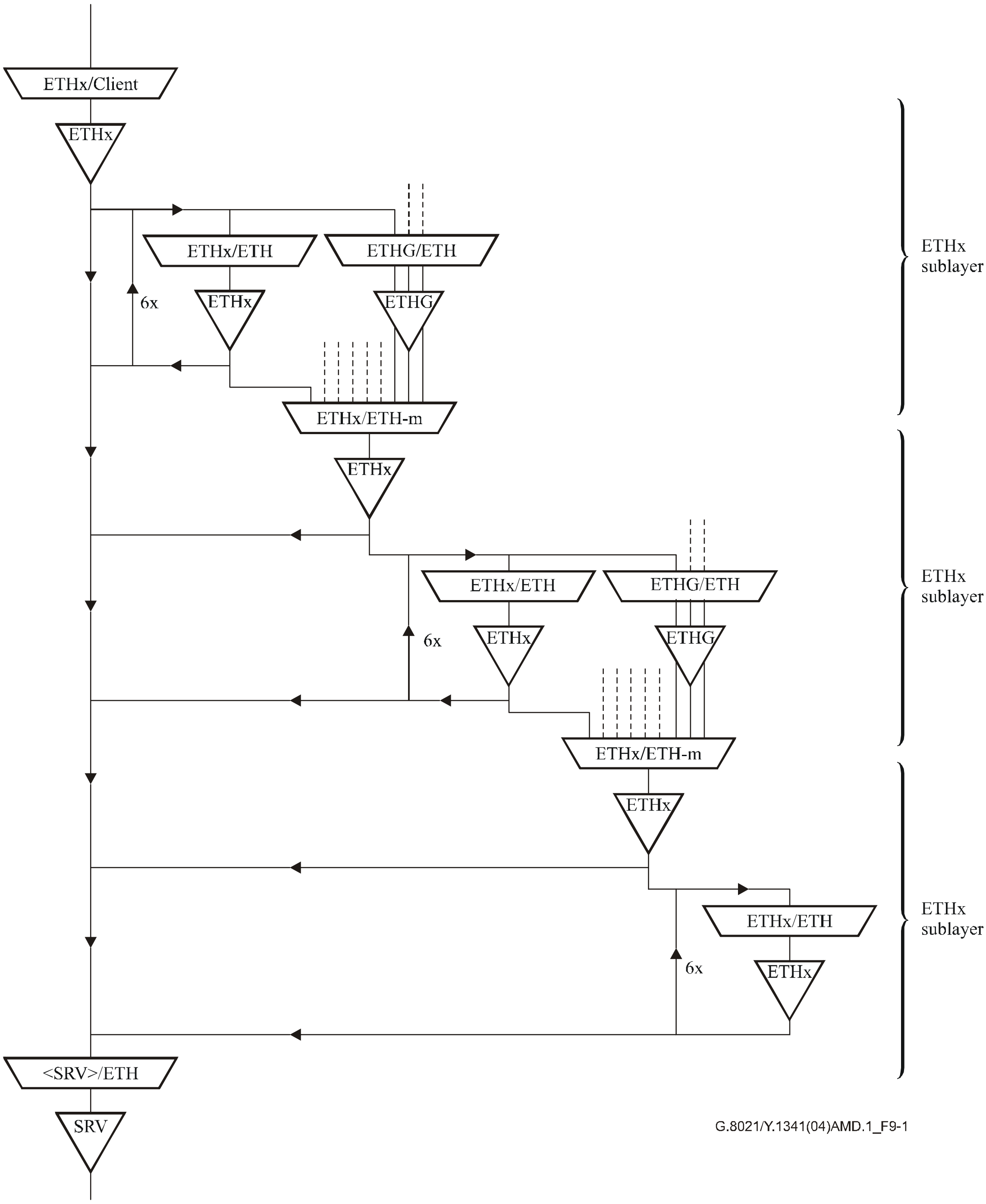


Figure 9-1 – ETH sublayering

Figure 9-1 illustrates the basic flow termination and adaptation functions involved and the possible orderings of these functions. The ETHx/ETH-m functions multiplex ETH_CI streams. The ETHx and ETHG flow termination functions insert and extract the proactive Y.1731 OAM information (e.g., CCM). The ETHDy flow termination functions insert and extract the on-demand Y.1731 OAM information (e.g., LBM, LTM). The ETHx/ETH adaptation functions insert and extract the administrative and management Y.1731 OAM information (e.g., LCK, APS).

Any combination that can be constructed by following the directions in Figure 9-1 is allowed. Some recursion is allowed as indicated by the arrows upwards; the number next to the arrow defines the number of recursions allowed.

Note that the ETHx sublayers in Figure 9-1 correspond to the ETH0 (top), ETH1 (middle) and ETH2 (bottom) in Figure 7-5/G.8010/Y.1306.

ETH characteristic information

The ETH_CI is a stream of ETH_CI traffic units complemented with ETH_CI_P, ETH_CI_DE and ETH_CI_SSF signals. An ETH_CI traffic unit defines the ETH_CI_D signal as illustrated in Figure 9-2. Each ETH_CI traffic unit contains a source address (SA) field, a destination address (DA) field and an M_SDU field. This can be further decomposed into a length/type field and a payload field; the payload field may be padded.

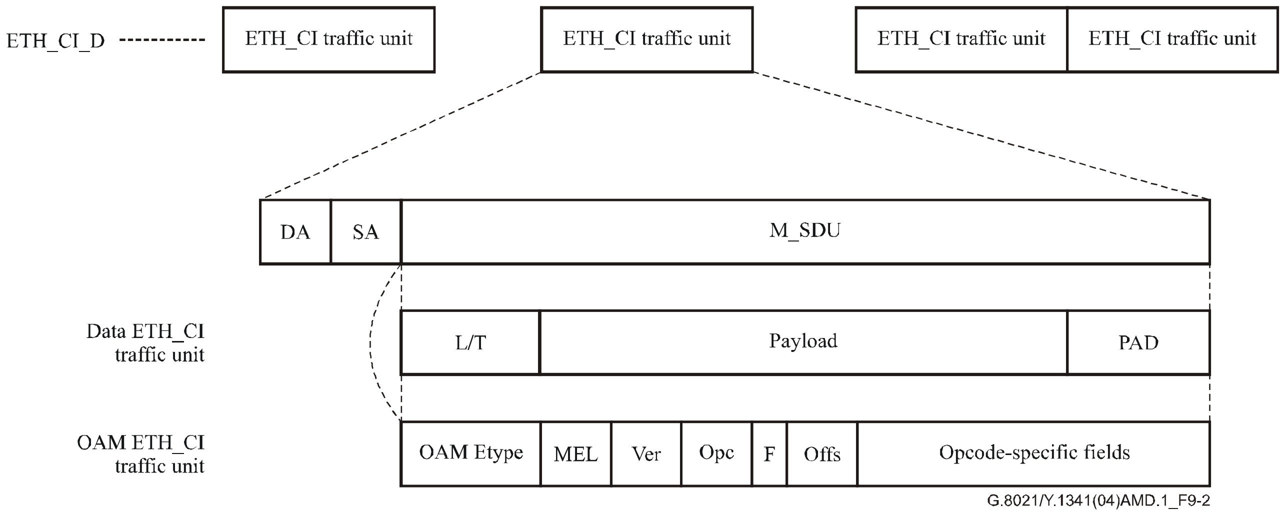


Figure 9-2 – ETH characteristic information

The SA and DA fields contain 48-byte MAC addresses as defined in IEEE 802.3.

There are two types of ETH_CI traffic units: data traffic units and OAM traffic units. If the L/T field equals the OAM Etype value (*for further study*, see Note 2), the ETH_CI traffic unit is an ETH_CI OAM traffic unit, otherwise it is an ETH_CI data traffic unit.

The payload field of an ETH_CI OAM traffic unit can be decomposed into the maintenance entity group level field (MEL), the version field (Ver), the opcode field (Opc), the flags field (F), the TLV offset field (Offs) and opcode-specific fields. This structure of ETH_CI OAM traffic units is defined in clause 9/Y.1731.

NOTE 1 – The ETH_CI contains no VID field as the ETH_CI is defined per VLAN.

ETH adapted information

The ETH_AI is a stream of ETH_AI traffic units complemented with the following signals: ETH_AI_P, ETH_AI_DE and ETH_AI_TSF. The ETH_AI traffic units define the ETH_AI_D signal. The ETH_AI traffic unit structure is shown in Figure 9-3.

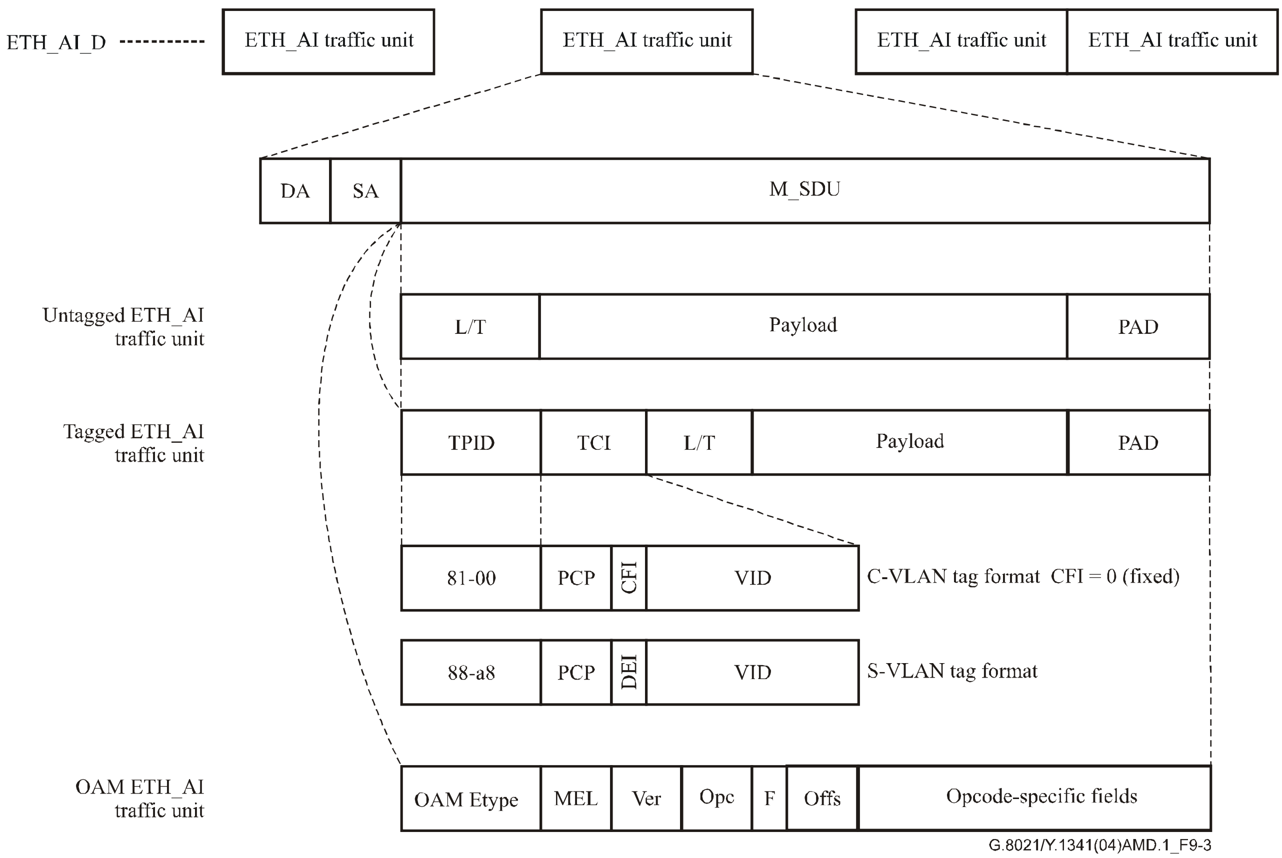


Figure 9-3 – ETH adapted information

The ETH_AI traffic unit contains the M_SDU and the DA and SA fields. The M_SDU field can be further decomposed into L/T, payload and PAD fields. These fields are the same as in ETH_CI traffic units.

There are three types of ETH_AI traffic units: tagged, untagged and OAM traffic units. The tagged and untagged types are defined in IEEE 802.1Q and IEEE 802.1ad. The OAM traffic units are defined in ITU-T Rec. Y.1731.

The L/T field determines the type of the ETH_AI traffic unit:

- if the L/T field contains the OAM Ethertype value *(for further study*, see Note 2*)*, the traffic unit is an OAM traffic unit, otherwise;
- if the L/T field contains one of the tag protocol identifier values indicated in Figure 9-3, the traffic unit is a tagged traffic unit, otherwise;
- the traffic unit is an untagged traffic unit.

NOTE 2 – In the next version of this Recommendation the OAM Ethertype will be assigned the value as will be defined in IEEE 802.1ag.

The payload field of an ETH_CI OAM traffic unit can be decomposed into the maintenance entity group level field (MEL), the version field (Ver), the opcode field (Opc), the flags field (F), TLV offset field (Offs) and opcode-specific fields, as for the ETH_CI OAM traffic units. This structure of ETH_CI OAM traffic units is defined in clause 9/Y.1731.

There are two types of tagged traffic units: C-VLAN tagged and S-VLAN tagged. Both of these types has its own TPI value: 81-00 for C-VLAN tagged; and 88-a8 for S-VLAN tagged, as defined in IEEE 802.1Q and IEEE 802.1ad respectively.

In a tagged frame (C-VLAN and S-VLAN tagged) a tag control information (TCI) field follows the TPI field. This field consists of a priority code point (PCP), VLAN ID (VID) and canonical format identifier (CFI) for C-VLAN tagged, or a drop eligible indicator (DEI) field for S-VLAN tagged traffic units.

The PCP field may be used to carry the ETH_CI_P and ETH_CI_DE signal values from an ETH_FP. The DEI field may be used to carry the ETH_CI_DE signal from an ETH_FP.

All ETH_AI traffic units may come from one ETH_FP or from different ETH_FP (in the case of multiplexing in ETHx/ETH-m_A function). In the latter case the VID field value is used to identify the ETH_FP the traffic unit is associated with.

Note that because of the stacking of ETH sublayers, the ETH_CI of a client ETH sublayer is encapsulated in ETH_AI to be transferred via a server ETH sublayer. Figure 9-4 shows an ETH_CI OAM traffic unit encapsulated in an ETH_AI data traffic unit. The grey fields constitute the original ETH_CI OAM traffic unit. The encapsulating traffic unit is no longer an OAM traffic unit, but a tagged traffic unit. Adding a VLAN tag hides the OAM information, and transforms an OAM ETH_CI traffic unit into an ETH_AI tagged traffic unit.

DA	SA	TPI	TCI	OAM Etype	MEL	Ver	Opc	Opcode-specific OAM information	Pad

Figure 9-4 – Tagged ETH_AI carrying ETH_CI OAM

This ETH_AI tagged traffic unit will be transformed into a ETH_CI data traffic unit by the ETHx_FT source function, resulting in an ETH_CI data traffic unit carrying a client layer ETH_CI OAM traffic unit.

9.1 Connection functions

9.1.1 ETH flow forwarding function (ETH_FF)

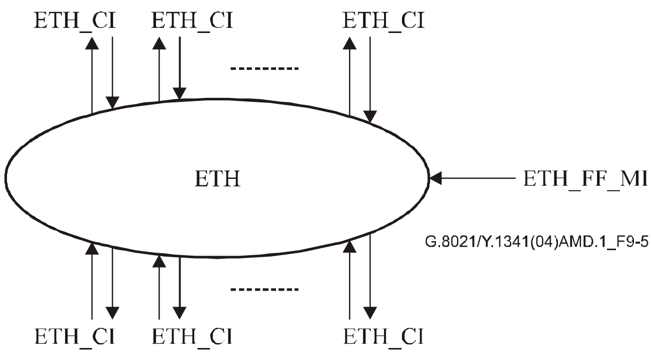


Figure 9-5 – ETH flow forwarding function

The ETH flow forwarding function, as shown in Figure 9-5, forwards ETH_CI signals at its input ports to its output ports. The forwarding may take into account the value of the SA field of the ETH_CI traffic unit.

Processes

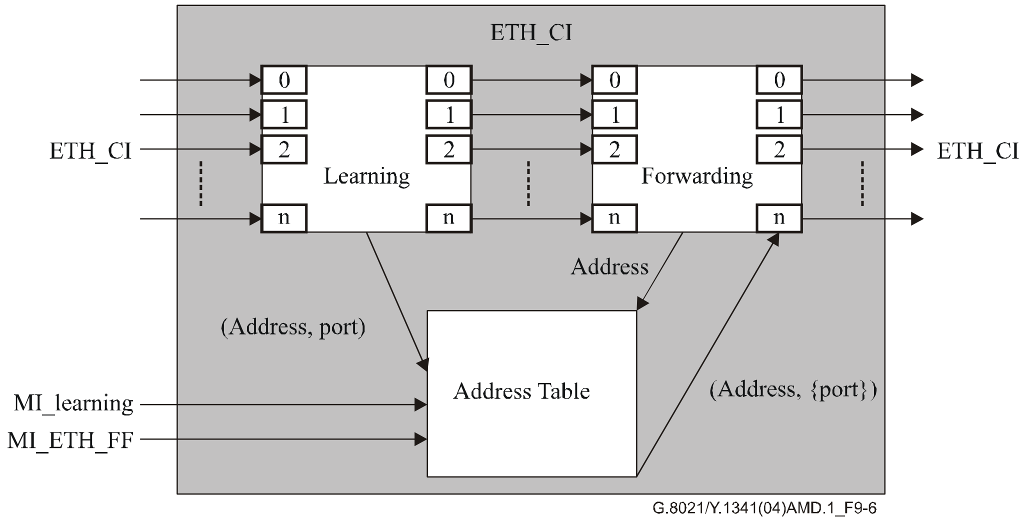


Figure 9-6 – ETH flow forwarding process

Address table process:

The AddressTable process maintains a list of tuples (address, {ports}). This list may be configured using ETH_FF_MI input signal and by the learning process. The AddressTable process processes address requests from the forwarding process, and responds with the tuple (address, {port}) for the specified address. If the tuple does not exist, the port set ({port}) is empty.

Learning process:

If the value of MI_Learning is enable, the learning process reads the SA field of the incoming ETH_CI traffic unit and forwards a tuple (address, {port}) to the address table process. The address contains the value of the SA field of the ETH_CI traffic unit, and the port is the port on which the traffic unit was received.

The ETH_CI itself is forwarded unchanged to the output of the learning process.

Forwarding process:

The forwarding process reads the DA field of the incoming ETH_CI traffic unit and sends this to the AddressTable process, the AddressTable will send a tuple (address, {port}) back in response. It will forward the ETH_CI on all ports listed in the port set field of the tuple. If the port set is empty, the ETH_CI will be forwarded on all ports (flooding). However, in all cases the ETH_CI is never forwarded from the same port as it was received on.

Defects *None*.

Consequent actions *None*.

Defect correlations *None*.

Performance monitoring *None*.

9.1.2 ETH split horizon flow forwarding function (ETH_SH_FF)

For further study.

9.1.3 Subnetwork connection protection process

For further study.

9.2 Termination functions

For further study.

9.2.1 ~~ETH Flow~~ETHx flow termination functions (~~ETH_FT~~ETHx_FT)

For further study.

9.2.1.1 ~~ETH Flow~~ETHx flow termination source function (~~ETH_FT~~ETHx_FT_So)

For further study.

9.2.1.2 ~~ETH Flow~~ETHx flow termination sink function (~~ETH_FT~~ETHx_FT_Sk)

For further study.

9.3 Adaptation functions

9.3.1 ETH to client adaptation functions (ETH/<client>_A)

For further study.

9.3.2 ETH to ETH adaptation function (ETHx/ETH_A)

9.3.2.1 ETH to ETH adaptation source function (ETHx/ETH_A_So)

This function maps client ETH_CI traffic units into server ETH_AI traffic units.

Symbol

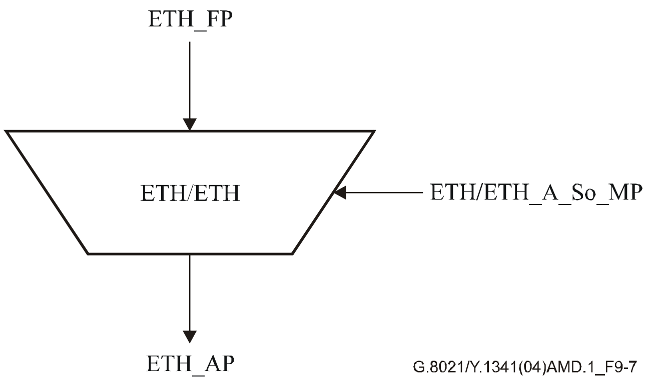


Figure 9-7 – ETHx/ETH_A_So function

Interfaces

Table 9-1 – ETHx/ETH_A_So input and outputs

Inputs	Outputs
ETH_FP: ETH_CI_D ETH_CI_P ETH_CI_DE ETH_CI_APS **ETHx/ETH_A_So_MP**: ETHx/ETH_A_So_MI_ME_Level ETHx/ETH_A_So_MI_LCK_Period ETHx/ETH_A_So_MI_LCK_Pri ETHx/ETH_A_So_MI_Client_ME_Level ETHx/ETH_A_So_MI_Admin_State ETHx/ETH_A_So_APS_Pri	**ETH_AP**: ETH_AI_D ETH_AI_P ETH_AI_DE

Processes

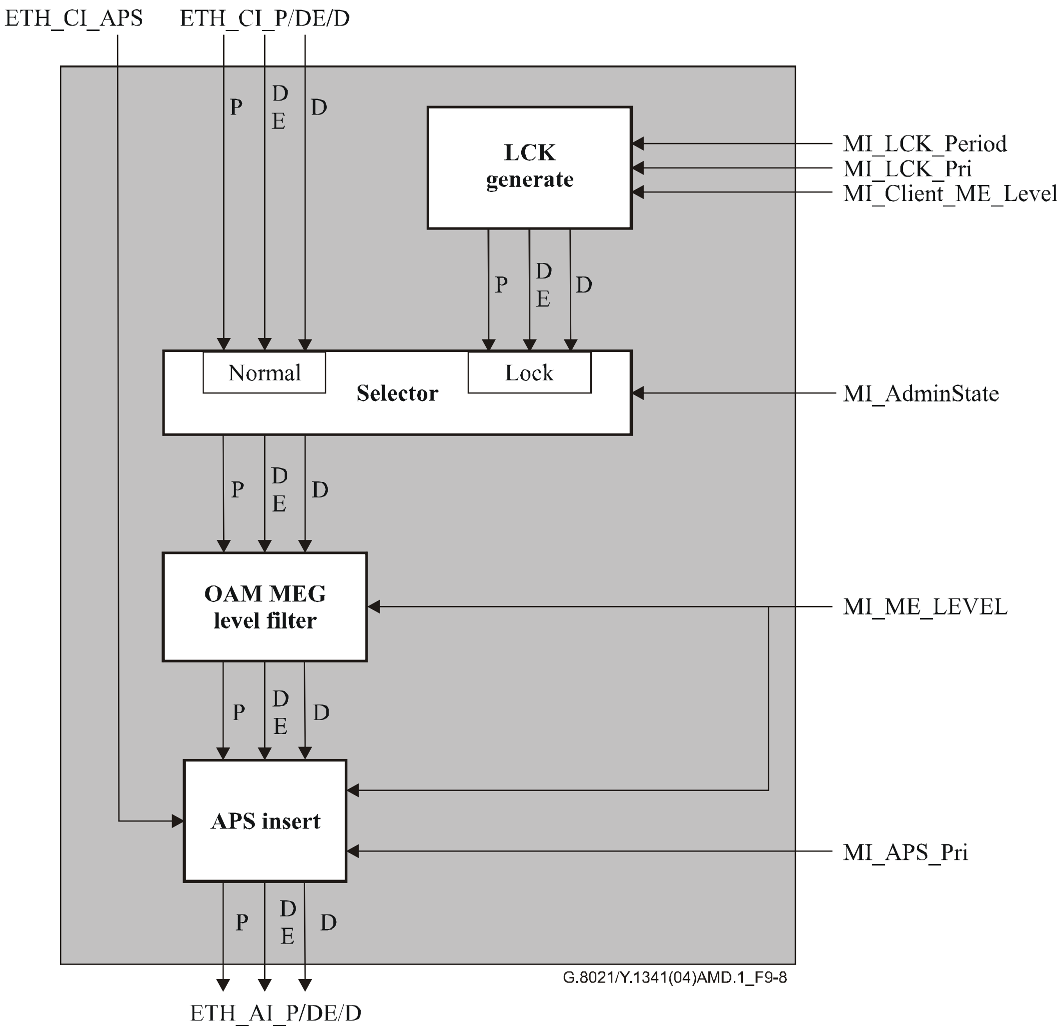


Figure 9-8 – Source direction ETHx/ETHA_So process

LCK generate process:

As defined in 8.1.2.

Selector process:

As defined in 8.1.3.

OAM MEG Level filter process:

As defined in 8.1.1.

APS insert process:

As defined in 8.1.5.

When activated LCK admin state shall be unlocked (see 7.5.2.2/G.8010/Y.1306).

Defects *None*.

Consequent actions *None*.

Defect correlations *None*.

Performance monitoring *None*.

9.3.2.2 ETH to ETH adaptation sink function (ETHx/ETH_A_Sk)

This function retrieves client ETH_CI traffic units from server ETH_AI traffic units.

Symbol

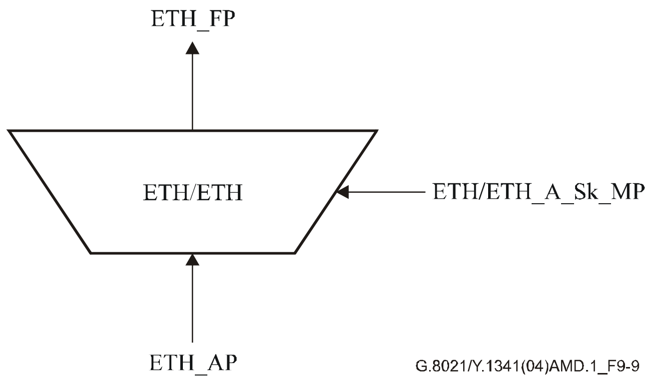


Figure 9-9 – ETHx/ETH_A_Sk function

Interfaces

Table 9-2 – ETHx/ETH_A_Sk input and outputs

Inputs	Outputs
ETH_AP: ETH_AI_D ETH_AI_P ETH_AI_DE ETH_AI_TSF ETH_AI_AIS **ETHx/ETH_A_Sk_MP**: ETHx/ETH_A_Sk_MI_Admin_State ETHx/ETH_A_Sk_MI_LCK_Period ETHx/ETH_A_Sk_MI_LCK_Pri ETHx/ETH_A_Sk_MI_Client_ME_Level ETHx/ETH_A_Sk_MI_AIS_Pri ETHx/ETH_A_Sk_MI_AIS_Period ETHx/ETH_A_Sk_MI_ME_Level	**ETH_FP**: ETH_CI_D ETH_CI_P ETH_CI_DE ETH_CI_APS ETH_CI_SSF

Processes

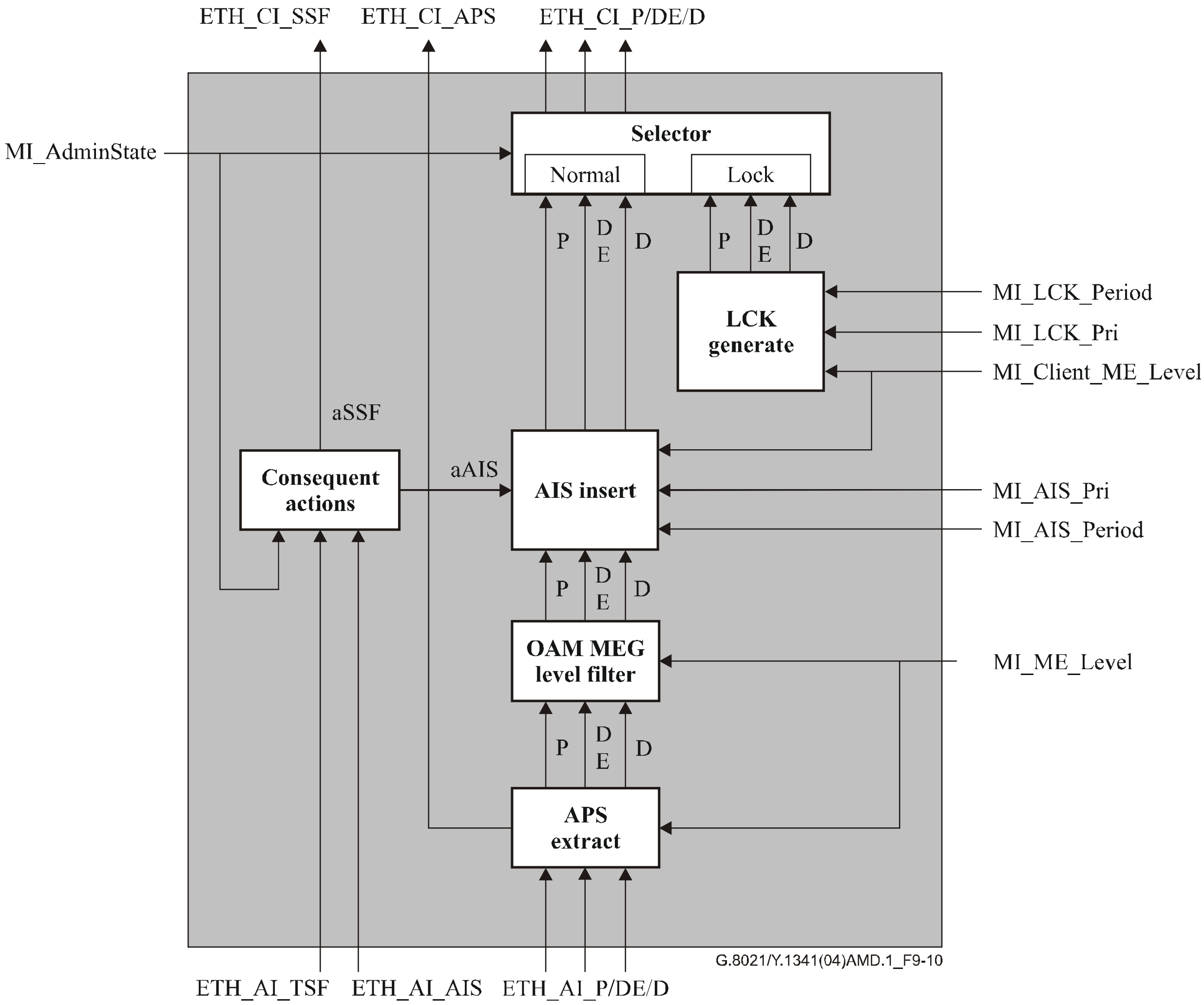


Figure 9-10 – Sink direction ETHx/ETH_A_Sk process

APS extract process:

As defined in 8.1.6.

OAM MEG level filter process:

As defined in 8.1.1.

AIS insert process:

As defined in 8.1.4.

LCK generate process:

As defined in 8.1.2.

Selector process:

As defined in 8.1.3.

Defects *None*.

Consequent actions

aSSF ← AI_TSF and (not MI_Admin_State = LOCKED)

aAIS ← AI_TSF and (not MI_Admin_State = LOCKED)

Defect correlations *None*.

Performance monitoring *None*.

9.3.3 ETH to ETH multiplexing adaptation function (ETHx/ETH-m)

This adaptation function multiplexes different ETH_CI streams into a single ETH_AI stream in the source direction and demultiplexes the ETH_AI stream into individual ETH_CI streams.

Symbol

The ETHx/ETH-m_A (Figure 9-11) function is further decomposed into separate source and sink adaptation functions that are interconnected as shown in Figure 9-12.

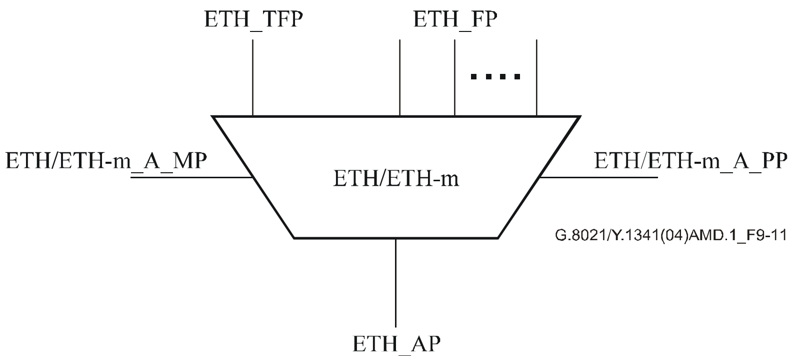


Figure 9-11 – ETHx/ETH-m_A function

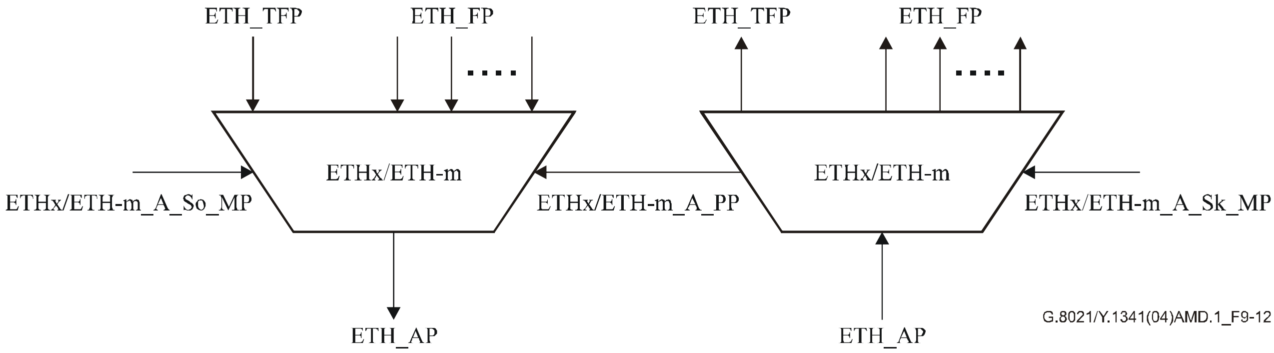


Figure 9-12 – ETHx/ETH-m_A source and sink functions

9.3.3.1 ETH to ETH multiplexing adaptation source function (ETHx/ETH-m_A_So)

This function multiplexes individuals ETH_CI streams into a single ETH_AI stream.

Symbol

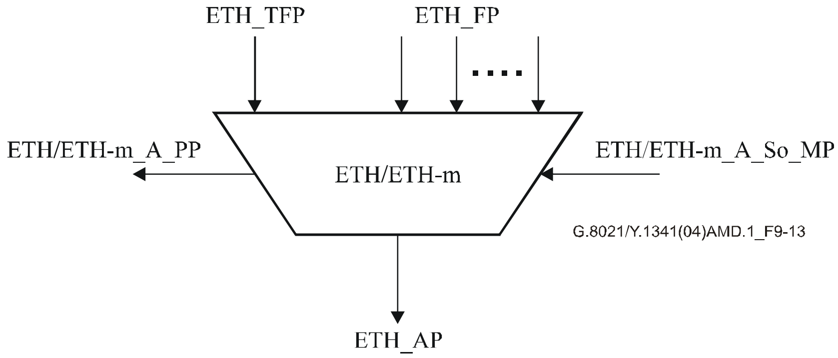


Figure 9-13 – ETHx/ETH-m_A_So function

Interfaces

Table 9-3 – ETHx/ETH-m_A_So interfaces

Inputs	Outputs
ETH_FP: ETH_CI_D[1...M] ETH_CI_P[1...M] ETH_CI_DE[1...M] **ETH_TFP**: ETH_CI_D ETH_CI_P ETH_CI_DE **ETHx/ETH**-m**_A_So_MP**: ETHx/ETH-m_A_So_MI_ME_Level[1…M] ETHx/ETH-m_A_So_MI_LCK_Period[1…M] ETHx/ETH-m_A_So_MI_LCK_Pri[1…M] ETHx/ETH-m_A_So_MI_Client_ME_Level[…M] ETHx/ETH-m_A_So_MI_Admin_State ETHx/ETH-m_A_So_MI_Vlan_Config ETHx/ETH-m_A_So_MI_Etype ETHx/ETH-m_A_So_PCP_Config ETHx/ETH-m_A_So_Queue_Config	**ETH_AP**: ETH_AI_D ETH_AI_P ETH_AI_DE **ETHx/ETH-m_A_PP**: ETH_PI_P ETH_PI_DE ETH_PI_D

Processes

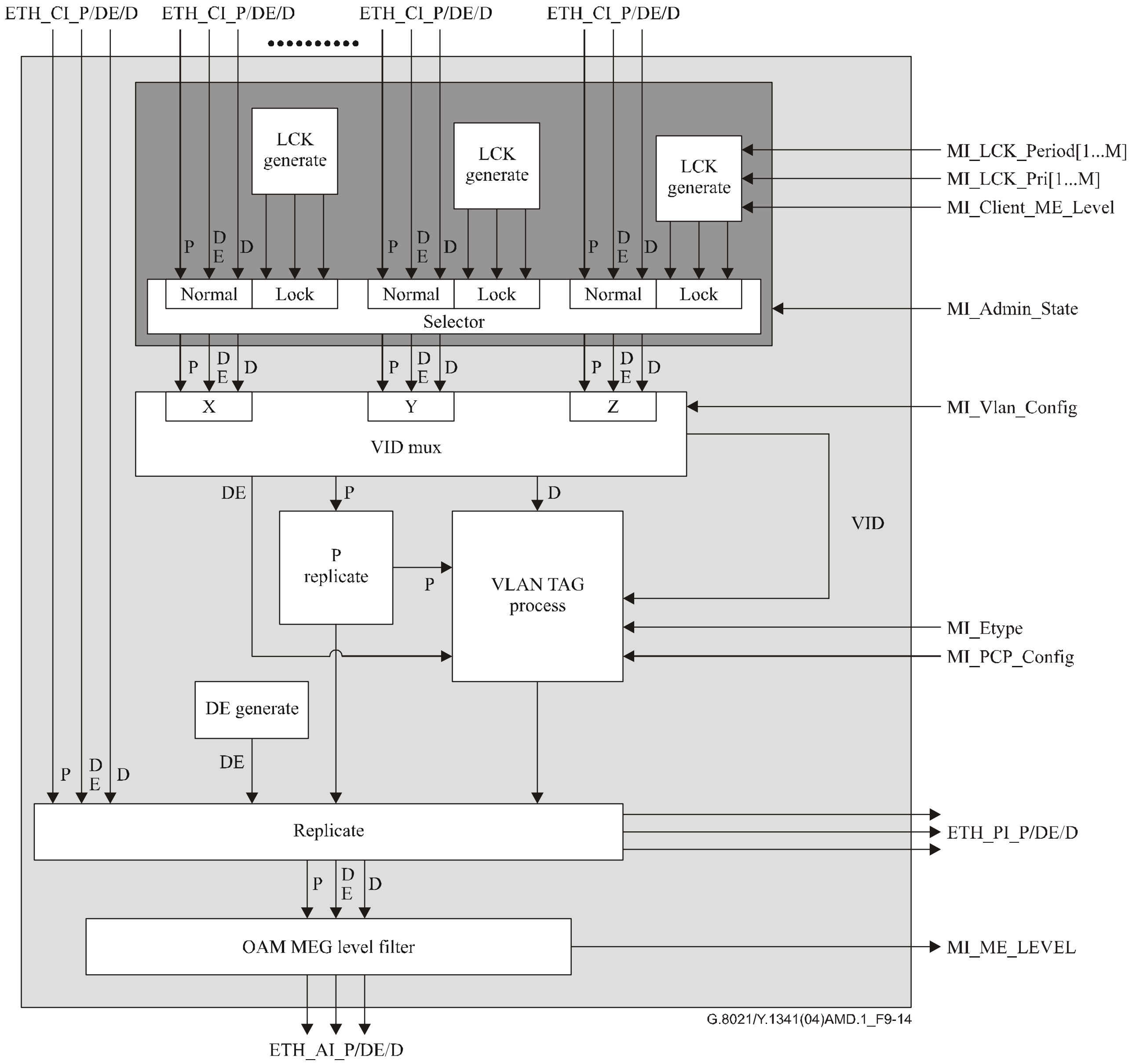


Figure 9-14 – ETHx/ETH-m_A_So process

LCK generate process:

As defined in 8.1.2. Each FP has its LCK generate process.

Selector process:

As defined in 8.1.3. Replaces the normal CI by the individual lock CI if Admin_State = LOCKED.

VID mux process:

The VID mux process interleaves the signal sets (P, D and DE) from the input ports (X, Y and Z). For each incoming signal set on forwarding the signal set, a VID signal is generated. The value of the VID signal is based on the port on which the signal set is received and the configuration from the MI_VLAN_Config input parameter.

For every input port, the MI_VLAN_Config input parameter determines the associated VID value. The allowed values for the VID signal are untagged, priority tagged and 1-4094. The following restriction applies to the allowed MI_VLAN_Config values:

- Every VID value is only used once.

Note that IEEE 802.1 standards do not allow IEEE bridges to generate priority tagged frames. Priority tagged frames are only generated by end stations. However a C-VLAN bridge may create S-VLAN priority tagged frames.

VLAN tag process:

This process inserts a VLAN tag into the M_SDU field of the incoming D signal. The Ethertype to be used is determined by the value of the MI_Etype input parameter. The MI_PCP_Config signal determines the encoding of the P and DE signals in the VLAN tag. This parameter defines a mapping from P value to PCP value in the case of C-VLAN tags, and from P value to PCP and DEI values in the case of S-VLAN tags.

The VID signal determines the VID value in the VLAN tag. If the VID signal equals priority tagged, the VID value to be used is zero. If the VID signal equals untagged, no VLAN tag is inserted in the M_SDU field.

P replicate process:

The P replicate process replicates the incoming P signal to both output ports without changing the value of the signal.

DE generate process:

The DE generate process generates a DE signal with the value drop ineligible.

Replicate process:

As defined in 8.4.

OAM MEG level filter process:

As defined in 8.1.1.

Defects *None*.

Consequent actions *None*.

Defect correlations *None*.

Performance monitoring *None*.

9.3.3.2 ETH to ETH multiplexing adaptation sink function (ETHx/ETH-m_A_Sk)

Symbol

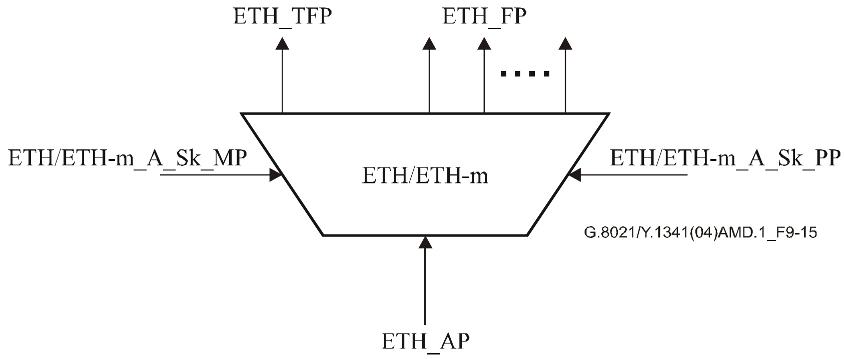


Figure 9-15 – ETHx/ETH-m_A_Sk function

Interfaces

Table 9-4 – ETHx/ETH-m_A_Sk interfaces

Inputs	Outputs
ETH_AP*:* ETH_AI_D ETH_AI_P ETH_AI_DE ETH_AI_TSF ETH_AI_AIS **ETHx/ETH-m_A_Sk_PP***:* ETH_PI_D ETH_PI_P ETH_PI_DE **ETHx/ETH-m_A_Sk_MP***:* ETHx/ETH-m_A_Sk_MI_Admin_State ETHx/ETH-m _A_Sk_MI_LCK/AIS_Period[1…M] ETHx/ETH-m _A_Sk_MI_LCK/AIS_Pri[1…M] ETHx/ETH-m _A_Sk_MI_Client_ME_Level[1…M] ETHx/ETH-m_A_Sk_MI_VLAN_Config ETHx/ETH-m_A_Sk_MI_P_Regenerate ETHx/ETH-m_A_Sk_MI_PVID ETHx/ETH-m_A_Sk_MI_PCP_Config ETHx/ETH-m_A_Sk_MI_Etype ETHx/ETH-m_A_Sk_MI_ME_Level ETHx/ETH-m_A_Sk_MI_Frametype_Config ETHx/ETH-m_A_Sk_MI_Filter_Config	**ETH_FP***:* ETH_CI_P[1...M] ETH_CI_DE[1...M] ETH_CI_D[1...M] ETH_CI_SSF[1...M] **ETH_TFP***:* ETH_CI_D ETH_CI_P ETH_CI_DE

Processes

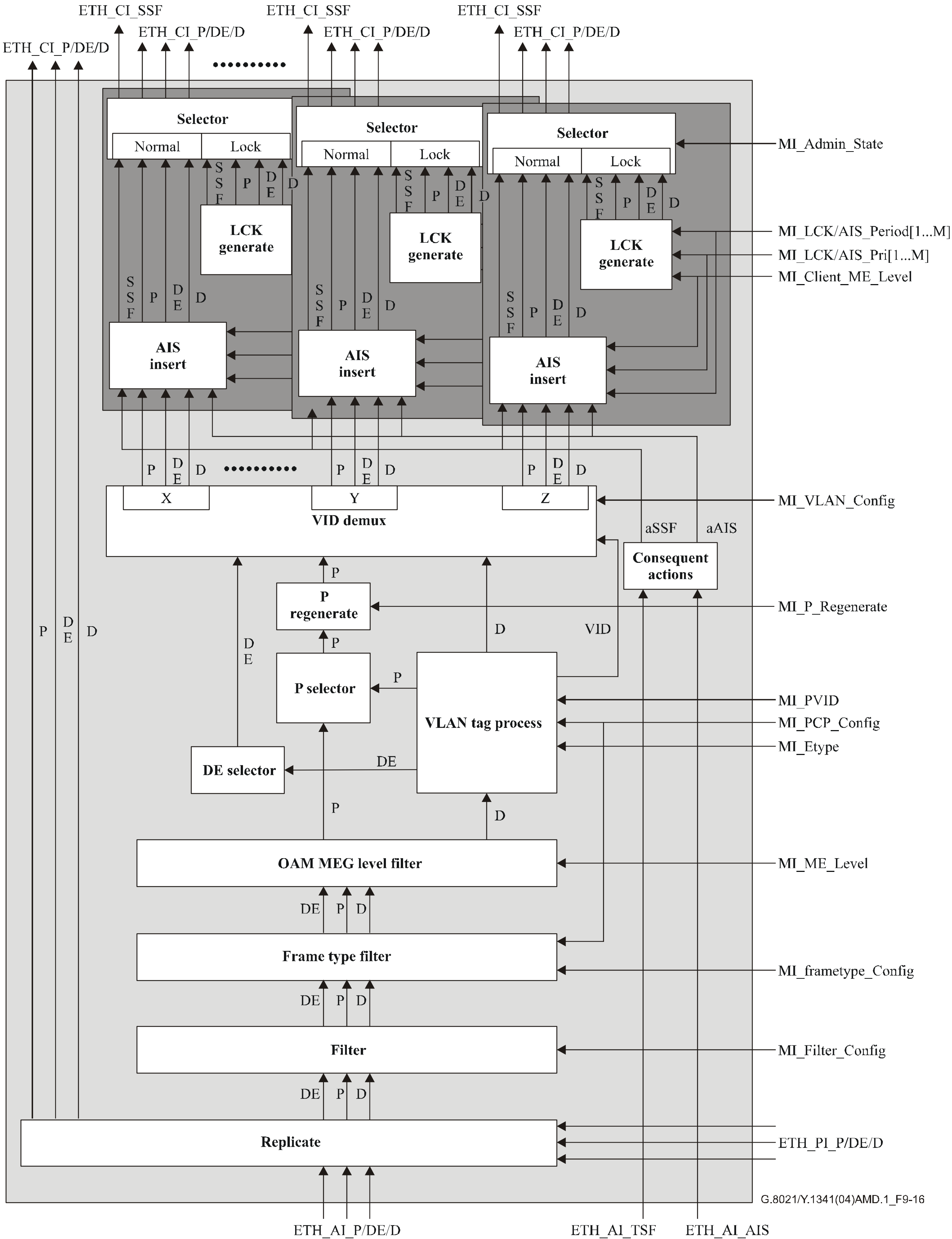


Figure 9-16 – ETHx/ETH-m_A_Sk process

Replicate process:

As defined in 8.4.

Filter process:

As defined in 8.3.

Frame type filter process:

The frame type filter process filters the ETH_CI depending on the value of the MI_frametype_Config input parameter. There are three possible values for this parameter:

- all frames;
- only VLAN tagged;
- only untagged and priority tagged.

If the value of MI_frametype_Config is set to all frames, all ETH_CI is passed through. For the other two values the process inspects the M_SDU field of the ETH_CI_D signal. It inspects the length/type field and, if applicable, the VID field.

If MI_frametype_Config is set to only untagged and priority tagged, all frames with L/T equal to MI_Etype and VID in the range 1…4094 are filtered.

If MI_frametype_Config is set to only VLAN tagged, all frames with L/T not equal to MI_Etype, and all frames with L/T equal to MI_Etype and VID equal to zero, are filtered.

OAM MEG level filter process:

As defined in 8.1.1.

VLAN tag process:

The VLAN tag process inspects the incoming D signal; if the value in the L/T field is equal to the value provisioned by the MI_Etype input parameter, a VLAN tag is present in the D signal.

If there is no VLAN tag present, the VID signal gets the value presented by the MI_PVID input parameter.

If there is a VLAN tag present, the VLAN tag process extracts the P, DE and VID information from this VLAN tag. The VID value is taken from the VID field in the VLAN tag. The P and DE values are decoded from the PCP field of the VLAN tag (C-VLAN), or from the PCP and DEI fields of the VLAN tag (S-VLAN), using the decoding information presented via the MI_PCP_Config input parameter. The P value is presented to the P selector process and the DE value is presented to the DE selector process.

DE selector process:

This process forwards the incoming DE signal. If there is no incoming DE signal present, it generates a DE signal with value drop ineligible.

P selector process:

This process forwards the P signal coming from the VLAN tag process. If this signal is not present, the P signal coming from the OAM ME level process is forwarded.

P regenerate process:

This process regenerates the incoming P signal based on the MI_P_Regenerate input signal. The MI_P_Regenerate signal specifies a mapping table from P value to P value.

VID demux process:

The VID demux process deinterleaves the incoming signal set (DE, P and D) to the different ports (X, Y and Z in Figure 9-16). The VID signal determines the port to be selected based on the MI_VLAN_Config input parameter.

The MI_VLAN_Config parameter specifies the possible VID values for the ports to be used. If there is no port assigned to a specific VID value, and this VID value is used, the VID demux process will filter the incoming signal set.

AIS insert process:

As defined in 8.1.4.

LCK generate process:

As defined in 8.1.2. Each FP has its own LCK generate process.

Selector process:

As defined in 8.1.3. Replaces the normal CI by the individual lock CI if Admin_State = LOCKED.

Defects *None*.

Consequent actions:

aSSF ← AI_TSF and (not MI_Admin_State = LOCKED)

aAIS ← AI_TSF and (not MI_Admin_State = LOCKED)

Defect correlations *None*.

Performance monitoring *None*.

9.3.4 ETH group to ETH adaptation function (ETHG/ETH)

9.3.4.1 ETH group to ETH adaptation source function (ETHG/ETH_A_So)

Symbol

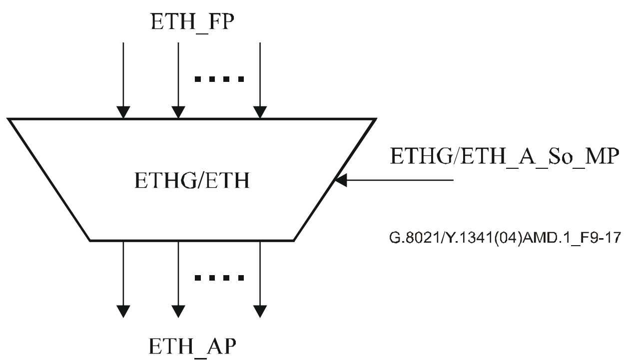


Figure 9-17 – ETHG/ETH_A_So function

Interfaces

Table 9-5 – ETHG/ETH_A_So interfaces

Inputs	Outputs
ETH_FP: ETH_CI_D[1…M] ETH_CI_P[1…M] ETH_CI_DE[1…M] ETH_CI_APS **ETHG/ETH_A_So_MP**: ETHG/ETH_A_So_MI_ME_Level ETHG/ETH_A_So_MI_LCK_Period[1…M] ETHG/ETH_A_So_MI_LCK_Pri[1…M] ETHG/ETH_A_So_MI_Client_ME_Level ETHG/ETH_A_So_MI_Admin_State ETHG/ETH_A_So_MI_APS_Pri	**ETH_AP**: ETH_AI_D[1...M] ETH_AI_P[1...M] ETH_AI_DE[1...M]

Processes

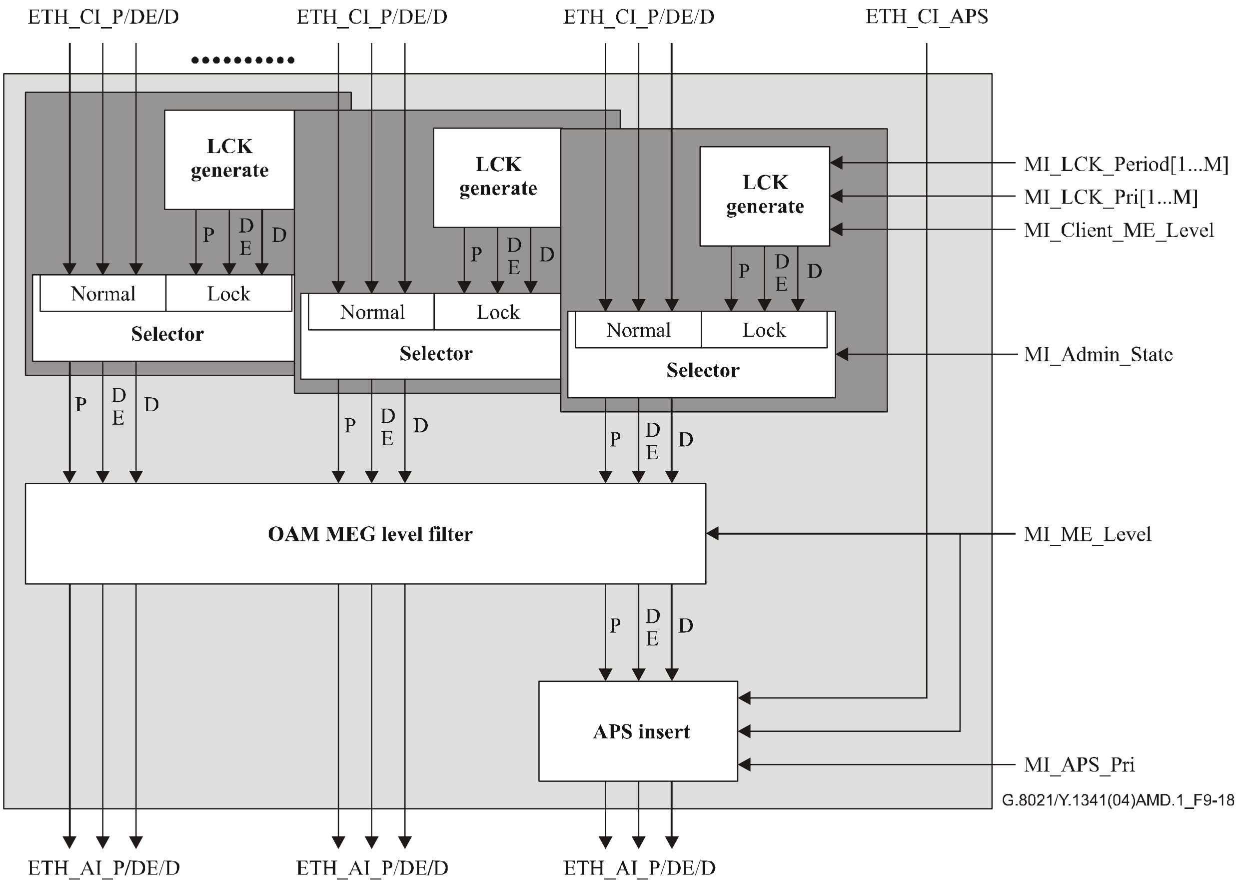


Figure 9-18 – ETHG/ETH_A_So process

LCK generate process:

As defined in 8.1.2. There is a single LCK Generate process for each ETH.

Selector Process:

As defined in 8.1.3. Replaces the normal CI of each input by the Lock CI if Admin_State = LOCKED.

OAM MEG level filter process:

As defined in 8.1.1.

APS insert process:

As defined in 8.1.5.

Defects *None*.

Consequent Actions *None*.

Defect correlations *None*.

Performance Monitoring *None*.

9.3.4.2 ETH group to ETH adaptation sink function (ETHG/ETH_A_Sk)

Symbol

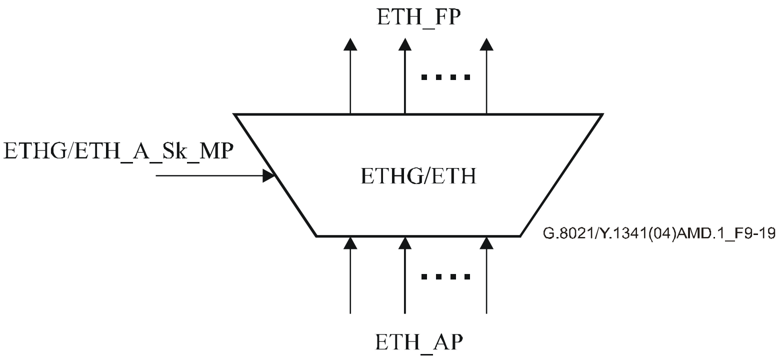


Figure 9-19 – ETHG/ETH_A_Sk function

Interfaces

Table 9-6 – ETHG/ETH_A_Sk interfaces

Inputs	Outputs
ETH_AP: ETH_AI_D[1...M] ETH_AI_P[1...M] ETH_AI_DE[1...M] ETH_AI_TSF ETH_AI_AIS **ETHG/ETH_A_Sk_MP**: ETHG/ETH_A_Sk_MI_Admin_State ETHG/ETH_A_Sk_MI_LCK/AIS_Period[1…M] ETHG/ETH_A_Sk_MI_LCK/AIS_Pri[1…M] ETHG/ETH_A_Sk_MI_Client_ME_Level ETHG/ETH_A_Sk_MI_ME_Level	**ETH_FP**: ETH_CI_D[1…M] ETH_CI_P[1…M] ETH_CI_DE[1…M] ETH_CI_APS ETH_CI_SSF[1…M]

Processes

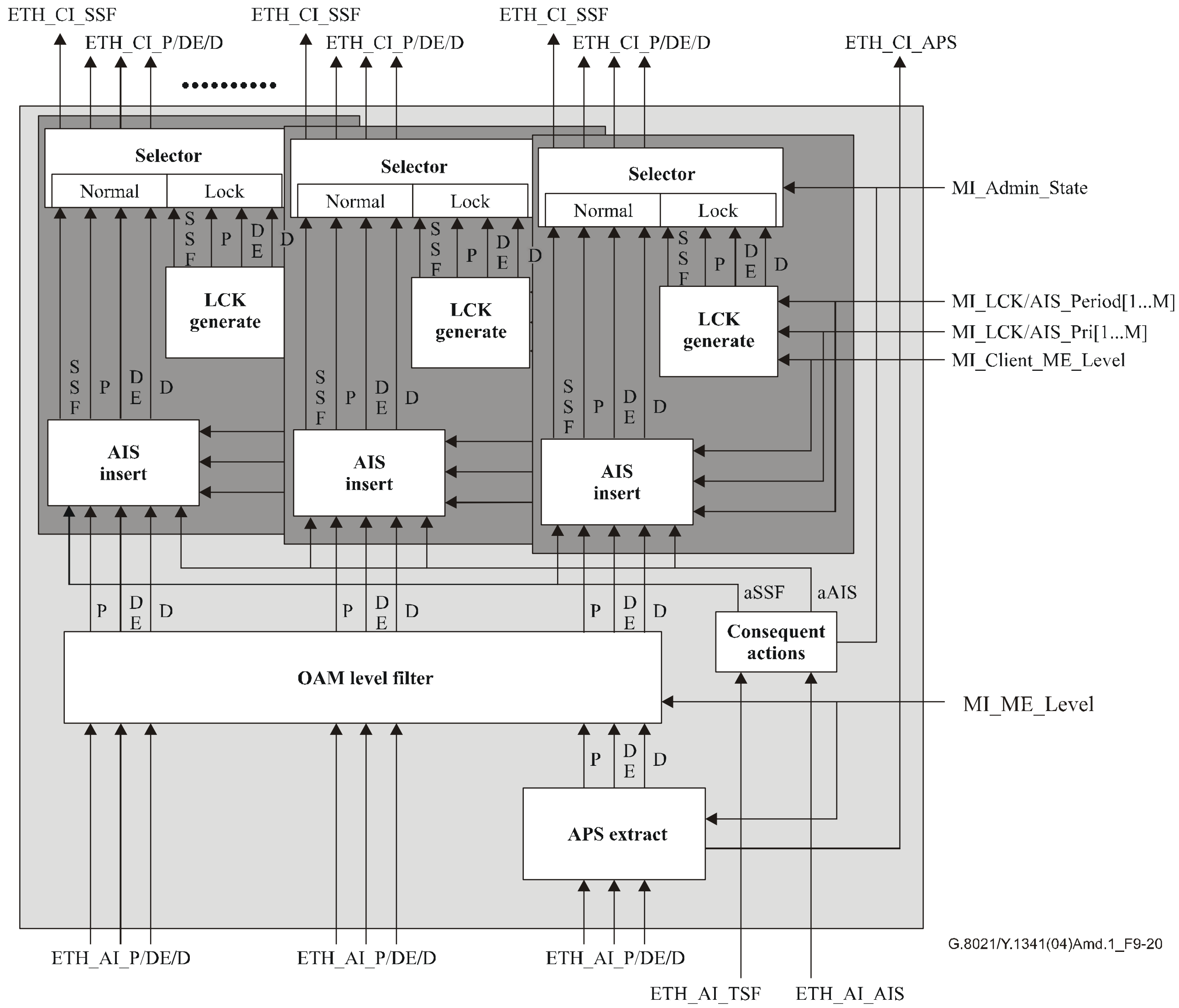


Figure 9-20 – ETHG/ETH_A_Sk process

APS extract process:

As defined in 8.1.6.

OAM MEG level filter process:

As defined in 8.1.1.

AIS insert process:

As defined in 8.1.4. There is a single AIS insert process for each ETH.

LCK generate process:

As defined in 8.1.2. There is a single LCK generate process for each ETH.

Selector process:

As defined in 8.1.3. Replaces the normal CI of each input by the Lock CI if Admin_State = LOCKED.

Defects *None.*

Consequent Actions

aSSF ← AI_TSF and (not MI_Admin_State == Locked)

aAIS ← AI_TSF and (not MI_Admin_State == Locked)

Defect correlations *None*.

Performance Monitoring *None*.

9.3.5 ETHx to ETHG adaptation function

For further study.

9.4 ETH diagnostic functions

9.4.1 ETH diagnostic flow termination function

For further study.

9.4.1.1 ETH diagnostic flow termination source function

For further study.

9.4.1.2 ETH diagnostic flow termination sink function

For further study.

9.4.2 ETHD/ETH adaptation function

For further study.

9.4.2.1 ETHD/ETH source adaptation function

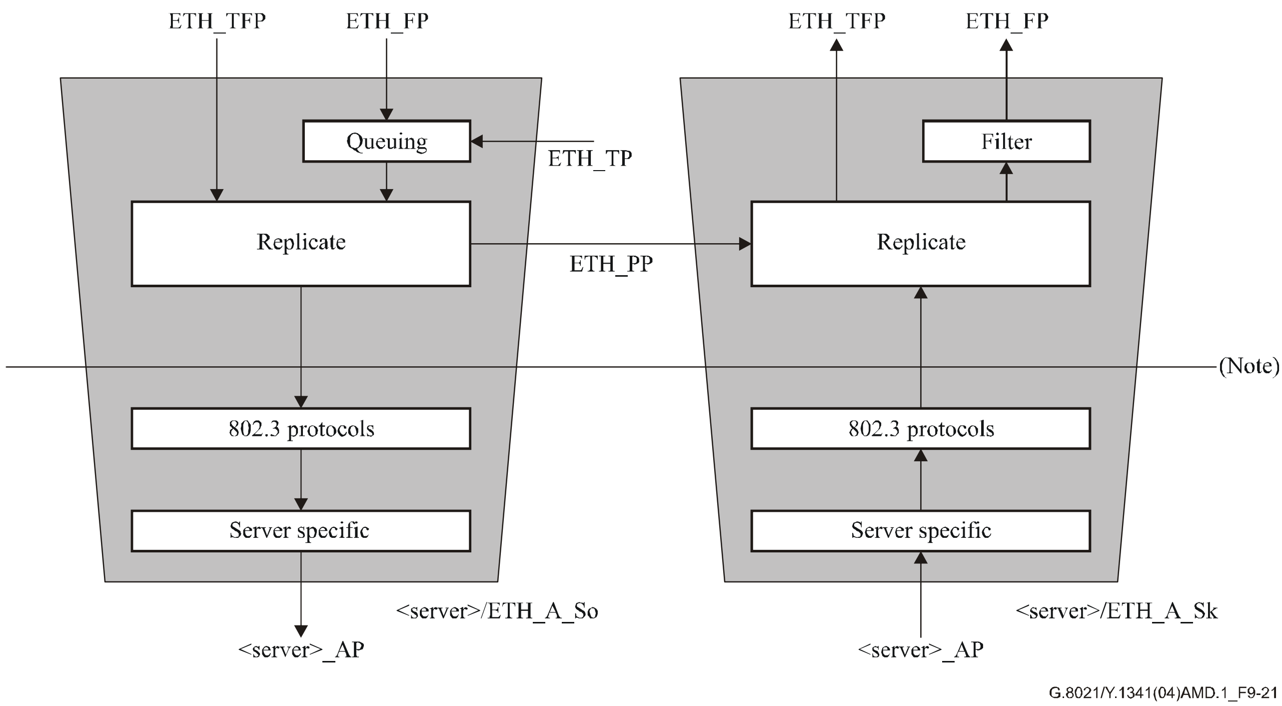
For further study.

9.4.2.2 ETHD/ETH sink adaptation function

For further study.

9.5 Server to ETH adaptation function <server>/ETH_A

Figure 9-21 presents a high level view of the processes that are present in a generic server to ETH adaptation function (<server>/ETH). The information crossing the <server>/ETH termination flow point (ETH_TFP) is referred to as the ETH characteristic information (ETH_CI). The information crossing the server layer access point (<server>_AP) is referred to as the server-specific adapted information (<server>_AI). Note that for some server signals not all processes need to be present, as defined in the server-specific adaptation functions.



NOTE – This interface is shown for reference only. It corresponds to the ISS interface in the IEEE 802 model.

Figure 9-21 – Server to ETH adaptation functions

The following generic processes are specified:

– "filter process" in clause 8.3;

– "queuing process" in clause 8.2;

– "replicate process" in clause 8.4, and

– "802.3 protocols process" in clause 8.5.

Server-specific processes are specified in server-specific clauses.

NOTE 1 – Filtering in the <server>/ETH_A sink adaptation function is not applied to frames forwarded to the ETH_TFP. The processes connected to this ETH_TFP should filter or process ETH_CI.

NOTE 2 – Queuing of frames in the source direction is also not applied to frames from the ETH_TFP. If queuing of frames in the sink direction is required when traffic conditioning is applied, this will be included in the traffic conditioning function.

NOTE 3 – For the G.8011.1 EPL service, ETH_TFP is unconnected. For services supporting ETH_TFP in the source direction, prioritization of frames received across the ETH_FP and ETH_TFP interfaces will be required. Such prioritization is *for further study*.

9.6 ETH traffic conditioning and shaping function (ETH_TCS)

For further study.

9.7 ETH link aggregation functions

The ETH link aggregation functions model the link aggregation functionality as described in clause 43 of IEEE 802.3. Where necessary, the definitions in this clause provide references to the appropriate generic process definitions in clause 8/G.806.

The generic model used is shown in Figures 9-22 and 9-23. Figure 9-22 shows the simplified model for the case of a single aggregator, while Figure 9-23 shows the generic model for the case of several aggregators. Np denotes the number of ETYn_AP interfaces (interfaces to the

IEEE 802.3 MAC layer), while Na is the number of ETH-LAG_FP interfaces (interfaces to the IEEE 802.3 MAC layer).

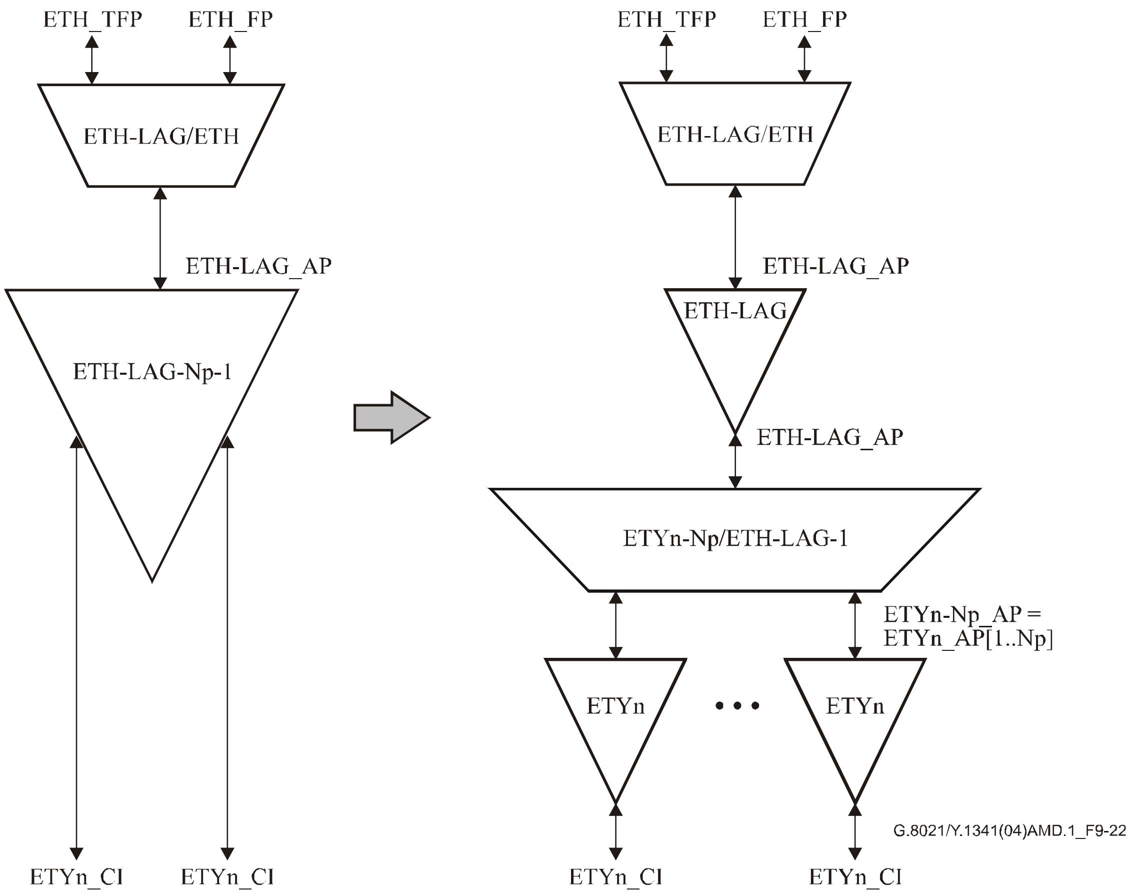


Figure 9-22 – Simplified model of Ethernet link aggregation with decomposition of ETH-LAG-Np-Na_TT function for Na=1

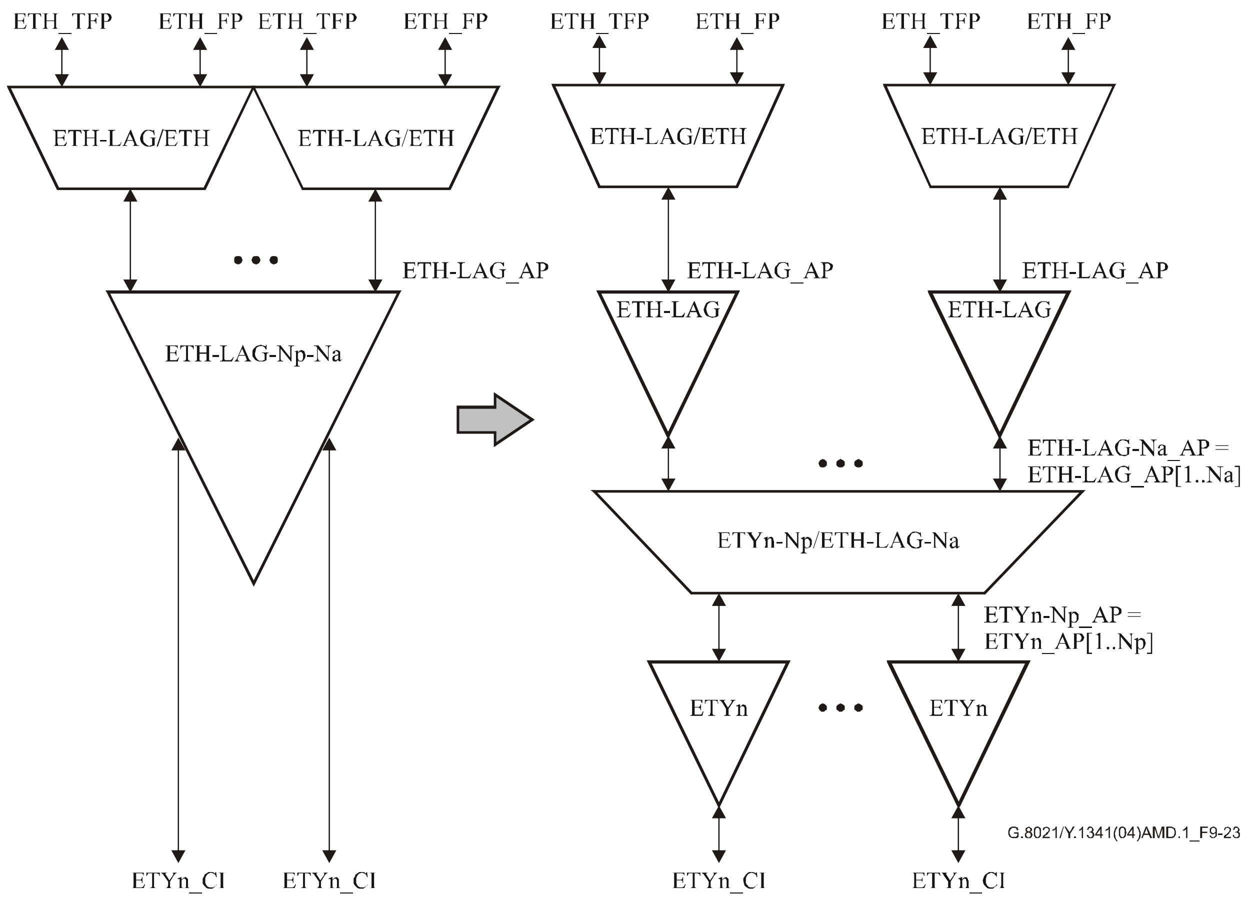


Figure 9-23 – Generic model of Ethernet link aggregation with decomposition of ETH-LAG-Np-Na_TT function

9.7.1 ETH link aggregation layer trail termination function (ETH-LAG-Np-Na_TT)

The ETH-LAG-Np-Na_TT function is decomposed as shown in Figure 9-23.

NOTE – ETH-LAG-Np-Na_TT functions always consist of a pair of identically-sized source and sink functions (i.e. a source function with certain values of Na/Np and a sink function with the same Na/Np values), as per IEEE 802.3.

9.7.1.1 ETH link aggregation adaptation source function (ETY-Np/ETH-LAG-Na_A_So)

Symbol

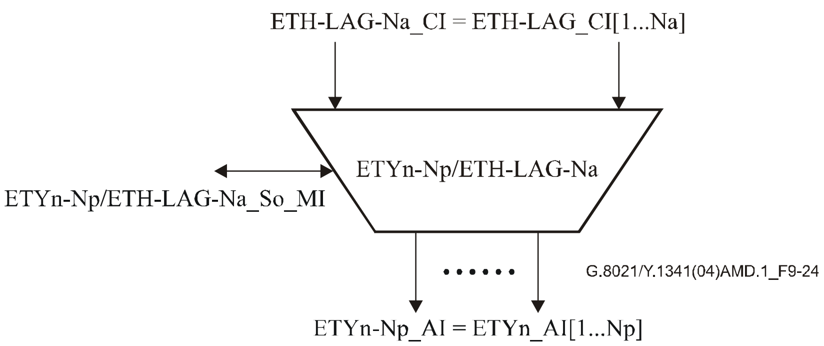


Figure 9-24 – ETY-Np/ETH-LAG-Na_A_So symbol

Interfaces

Table 9-7 – ETY-Np/ETH-LAG-Na_A_So interfaces

Inputs	Outputs
ETH-LAG-Na_FP: ETH-LAG-Na_CI_D = ETH-LAG_CI[1..Na]_D ETH-LAG-Na_CI_P = ETH-LAG_CI[1..Na]_P ETH-LAG-Na_CI_DE = ETH-LAG_CI[1..Na]_DE ETH-LAG-Na_CI_Clock = ETH-LAG_CI[1..Na]_Clock **ETYn-Np/ETH-LAG-Na _A_So_MI**: ETYn-Np/ETH-LAG-Na_A_So_ MI_TxPauseEnable ETYn-Np/ETH-LAG-Na_A_So_ MI_Agg[1..Na]_AP_List ETYn-Np/ETH-LAG-Na_A_So_ MI_AggPort[1..Np]_ Actor~~AdminState~~Admin_State	**ETYn-Np_AP**: ETYn-Np_AI_Data = ETYn_AI[1..Np]_Data ETYn-Np_AI_Clock = ETYn_AI[1..Np]_Clock **ETYn-Np/ETH-LAG-Na _A_So_MI**: ETYn-Np/ETH-LAG-Na_A_So_ MI_Agg[1..Na]_ ActorSystemID ActorSystemPriority ActorOperKey PartnerSystemID PartnerSystemPriority PartnerOperKey DataRate CollectorMaxDelay ETYn-Np/ETH-LAG-Na_A_So_ MI_AggPort[1..Np]_ ActorOperKey PartnerOperSystemPriority PartnerOperSystemID PartnerOperKey ActorPort ActorPortPriority PartnerOperPort PartnerOperPortPriority ActorOperState PartnerOperState ETYn-Np/ETH-LAG-Na_A_So_ MI_pAggOctetsTxOK[1..Na] ETYn-Np/ETH-LAG-Na_A_So_ MI_pAggFramesTxOK[1..Na] ETYn-Np/ETH-LAG-Na_A_So_ MI_pFramesTransmittedOK[1..Np] ETYn-Np/ETH-LAG-Na_A_So_ MI_pOctetsTransmittedOK[1..Np]

NOTE 1 – The signals MI_Agg[1..Na]_… and MI_AggPort[1..Np]_… represent the attributes of the "aggregator" and "aggregator port" objects of the same name in the model in clause 30.7/IEEE 802.3. As an example, the output MI_Agg[k]_PartnerSystemID corresponds to the IEEE read-only attribute aAggPartnerSystemID for aggregator object #k.

NOTE 2 – For the purposes of Ethernet transport equipment, Table 9-7 contains the minimum set of aggregator and aggregator port inputs and outputs to be supported. This set is a subset of the IEEE 802.3 model, of which some attributes have been omitted because they are specific to the IEEE management philosophy or for simplification in transport equipment. All parameters not explicitly settable per Table 9-7 take their default values as per clause 43/IEEE 802.3.

NOTE 3 – This is the minimum set of common requirements that transport equipment must fulfil.

Processes

A process diagram of this function is shown in Figure 9-25.

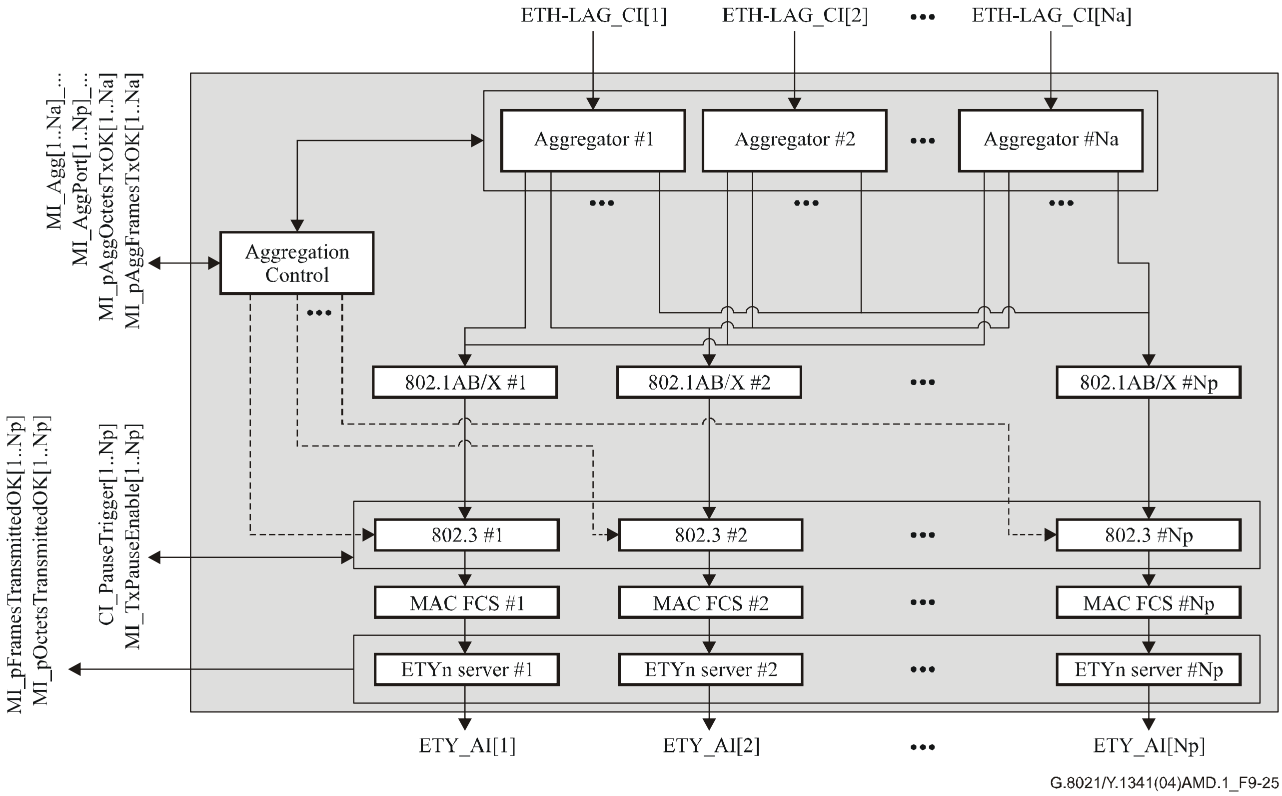


Figure 9-25 – ETY-Np/ETH-LAG-Na_A_So process diagram

The input MI_Agg[1..Na]_AP_List defines, for each aggregator, which ports (access points) are provisioned to be assigned to it. The AP_List attributes for all aggregators are disjunct lists.

The system shall assign a unique value for the parameter aAggActorAdminKey for each aggregator in the system. The system shall also assign the value used for each aggregator to the parameter aAggPortActorAdminKey of all ports in its assigned port list (AP_List).

NOTE 4 – This automated AdminKey assignment is a simplification of the IEEE provisioning model, where the keys are provisioned explicitly for each port and aggregator.

NOTE 5 – Automated assignment of PartnerAdminKey attributes is *for further study*.

ETYn server

This process is identical to the server-specific common process defined in clause 8.9.

MAC FCS, 802.1AB/X, 802.3

These processes are as per the definitions of the MAC FCS generation in clause 8.9.1, 802.1AB/X protocols processes in clause 8.9.3 and 802.3 protocols processes in clause 8.5.

Aggregation control

This process is the source part of the process of the same name in clause 43/IEEE 802.3.

NOTE 6 – The aggregation control process is a single process shared between the source and the sink of a pair of source/sink adaptation functions.

NOTE 7 – As per the IEEE model, and given the automated key assignment, only ports from each aggregator's AP_List will be eligible to be selected by that aggregator.

Aggregator

This process is the source part of the process of the same name in clause 43/IEEE 802.3. A coupled mux state machine model is used.

NOTE 8 – Each aggregator #k process is a single process shared between the source and the sink of a pair of source/sink adaptation functions.

Defects *None*.

Consequent actions *None*.

Defect correlations *None*.

Performance monitoring

For each aggregator:

- MI_pAggOctetsTxOK[1..Na] per clause 30/IEEE 802.3.
- MI_pAggFramesTxOK[1..Na] per clause 30/IEEE 802.3.

For each access point:

- MI_pOctetsTransmittedOK[1..Np] per clause 30/IEEE 802.3.
- MI_pFramesTransmittedOK[1..Np] per clause 30/IEEE 802.3.

9.7.1.2 ETH link aggregation adaptation sink function (ETY-Np/ETH-LAG-Na_A_Sk)

Symbol

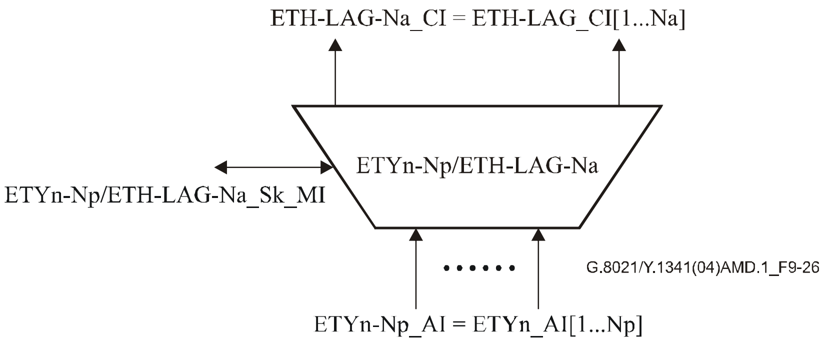


Figure 9-26 – ETY-Np/ETH-LAG-Na_A_Sk symbol

Interfaces

Table 9-8 – ETY-Np/ETH-LAG-Na_A_Sk interfaces

Inputs	Outputs
ETYn-Np_AP: ETYn-Np_AI_D= ETYn_AI[1..Np]_D ETYn-Np_AI_P= ETYn_AI[1..Np]_P ETYn-Np_AI_DE= ETYn_AI[1..Np]_DE ETYn-Np_AI_Clock ETYn_AI[1..Np]_Clock **ETYn-Np/ETH-LAG-Na _A_Sk_MI**: ETYn-Np/ETH-LAG-Na_A_Sk_ MI_PLLThr[1..Na]	**ETH-LAG-Na_FP**: ETH-LAG-Na_CI_D= ETH-LAG_CI[1..Na]_D ETH-LAG-Na_CI_P= ETH-LAG_CI[1..Na]_P ETH-LAG-Na_CI_DE= ETH-LAG_CI[1..Na]_DE ETH-LAG-Na_CI_Clock= ETH-LAG_CI[1..Na]_Clock ETH- LAG-Na_CI_aSSF= ETH-LAG_CI[1..Na]_aSSF **ETYn-Np/ETH-LAG-Na _A_Sk_MI**: ETYn-Np/ETH-LAG-Na_A_Sk_ MI_cPLL[1..Na] ETYn-Np/ETH-LAG-Na_A_Sk_ MI_cTLL[1..Na] ETYn-Np/ETH-LAG-Na_A_Sk_ MI_pAggOctetsRxOK[1..Na] ETYn-Np/ETH-LAG-Na_A_Sk_ MI_pAggFramesRxOK[1..Na] ETYn-Np/ETH-LAG-Na_A_Sk_ MI_pFramesReceivedOK[1..Np] ETYn-Np/ETH-LAG-Na_A_Sk_ MI_pOctetsReceivedOK[1..Np] ETYn-Np/ETH-LAG-Na_A_Sk_ MI_pFCSErrors[1..Np.]

Processes

A process diagram of this function is shown in Figure 9-27.

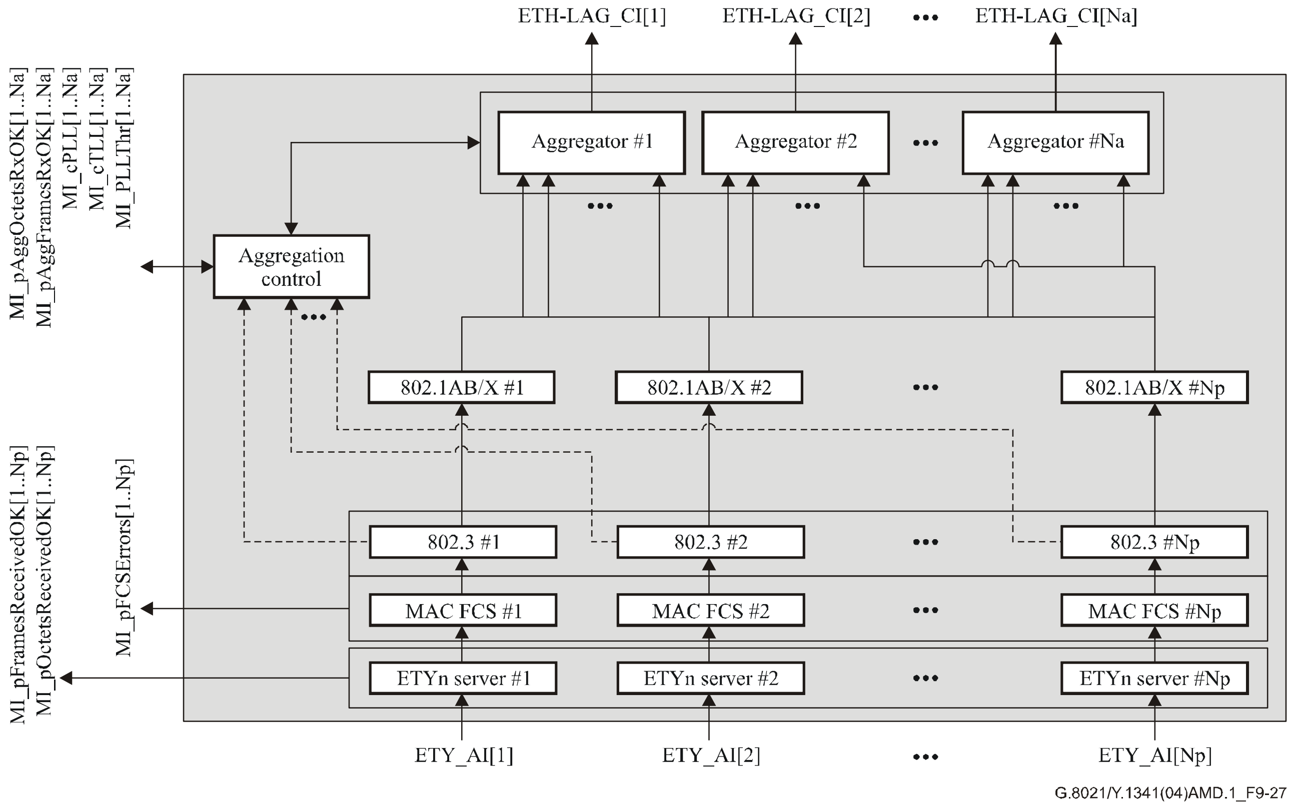


Figure 9-27 – ETY-Np/ETH-LAG-Na_A_Sk process diagram

ETYn Server:

This process is identical to the ETYn server-specific process defined in clause 9.7.1.1.

MAC FCS, 802.1AB/X, 802.3:

These processes are as per the definitions of the MAC frame check in clause 8.9.2, "802.1AB/X protocols processes" in clause 8.9.3 and "802.3 protocols processes" in clause 8.5.

Aggregation Control:

This process is the source part of the process of the same name in clause 43/IEEE 802.3.

NOTE 1 – The aggregation control process is a single process shared between the source and the sink of a pair of source/sink adaptation functions. The parameters used by this bidirectional process are defined in the interface section of the source adaptation function.

Aggregator:

This process is the source part of the process of the same name in clause 43/IEEE 802.3. A coupled mux state machine model is used.

NOTE 2 – Each aggregator #k process is a single process shared between the source and the sink of a pair of source/sink adaptation functions. The parameters used by this bidirectional process are defined in the interface section of the source adaptation function.

Defects

dMNCD[j] (member j not collecting/distributing): The defect shall be raised if an access point (port) in an aggregator's AP_List stays outside of the COLLECTING_DISTRIBUTING state for longer than X_{raise} seconds. The defect shall be cleared if the port enters the COLLECTING_DISTRIBUTING state and stays there for X_{clear} seconds.

$$X_{raise} = X_{clear} = 1 \text{ second}$$

Consequent actions

$$\text{ETH-LAG\_CI[k]\_aSSF} \leftarrow \prod_{j \in \text{MI\_AP\_List}[k]} \text{dMNCD}[j]$$

NOTE 3 – In other words, aSSF will be raised at the output ETH-LAG_CI[k] of an aggregator if all ports in its assigned port list (AP_List[k]) have the dMNCD defect active.

Defect correlations

Defining:

$$\text{mAP\_Active}[k] = \sum_{j \in \text{MI\_AP\_List}[k]} (\text{not dMNCD}[j])$$

i.e., the number of active (no-defect) ports among those in an aggregator's AP_List, then:

$$\text{ETH-LAG\_cTLL}[k] \leftarrow \text{mAP\_Active}[k] = 0$$

$$\text{ETH-LAG\_cPLL}[k] \leftarrow (0 < \text{mAP\_Active}[k]) \text{ and } (\text{mAP\_Active}[k] < \text{MI\_PLLThr}[k])$$

NOTE 4 – In other words, a cTLL (total link loss) fault cause will be raised if no ports are active for an aggregator. A cPLL (partial link loss) fault cause shall be raised if the number of active ports is less than the provisioned threshold.

Performance monitoring

For each aggregator:

- MI_pAggOctetsRxOK[1..Na] per clause 30/IEEE 802.3.
- MI_pAggFramesRxOK[1..Na] per clause 30/IEEE 802.3.

For each access point:

- MI_pFCSErrors[1..Np] per clause 30/IEEE 802.3.
- MI_pOctetsReceivedOK[1..Np] per clause 30/IEEE 802.3.
- MI_pFramesReceivedOK[1..Np] per clause 30/IEEE 802.3.

9.7.1.3 ETH link aggregation flow termination source function (ETH-LAG_FT_So)

Symbol

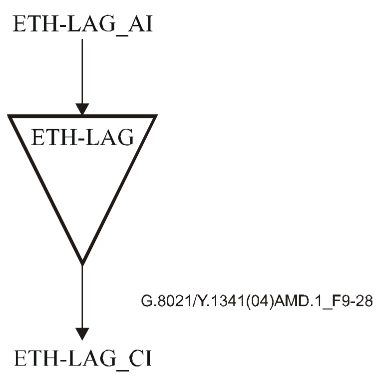


Figure 9-28 – ETH-LAG_FT_So symbol

Interfaces

Table 9-9 – ETH-LAG_FT_So interfaces

Inputs	Outputs
ETH-LAG_AP: ETH-LAG_AI_D ETH-LAG_AI_P ETH-LAG_AI_DE ETH-LAG_AI_Clock	**ETH-LAG_FP**: ETH-LAG_CI_D ETH-LAG_CI_P ETH-LAG_CI_DE ETH-LAG_CI_Clock

Processes

This function only forwards the ETH-LAG_AP information onto the ETH-LAG_FP without manipulation.

Defects: *None*.

Consequent Actions: *None*.

Defect Correlations: *None*.

Performance monitoring: *None*.

9.7.1.4 ETH link aggregation flow termination sink function (ETH-LAG_FT_Sk)

Symbol

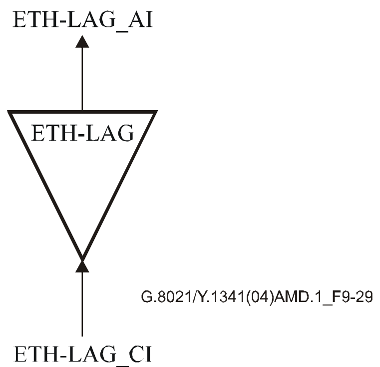


Figure 9-29 – ETH-LAG_FT_Sk symbol

Interfaces

Table 9-10 – ETH-LAG_FT_Sk interfaces

Inputs	Outputs
ETH-LAG_FP: ETH-LAG_CI_D ETH-LAG_CI_P ETH-LAG_CI_DE ETH-LAG_CI_Clock ETH-LAG_CI_SSF **ETH-LAG_MP**: ETH-LAG_TT_Sk_MI_SSF_Reported	**ETH-LAG_AP**: ETH-LAG_AI_D ETH-LAG_AI_P ETH-LAG_AI_DE ETH-LAG_AI_Clock ETH-LAG_AI_TSF ETH-LAG_AI_AIS **ETH-LAG_MP**: ETH-LAG_TT_Sk_MI_cSSF

Processes

This function only forwards the ETH-LAG_FP information onto the ETH-LAG_AP without manipulation.

Defects: *None*.

Consequent actions

aTSF ← CI_SSF

Defect correlations

cSSF ← CI_SSF and SSF_Reported

Performance monitoring: *None*.

9.7.2 ETH-LAG to ETH adaptation function (ETH-LAG/ETH_A)

9.7.2.1 ETH-LAG to ETH link aggregation adaptation source function (ETH-LAG/ETH _A_So)

Symbol

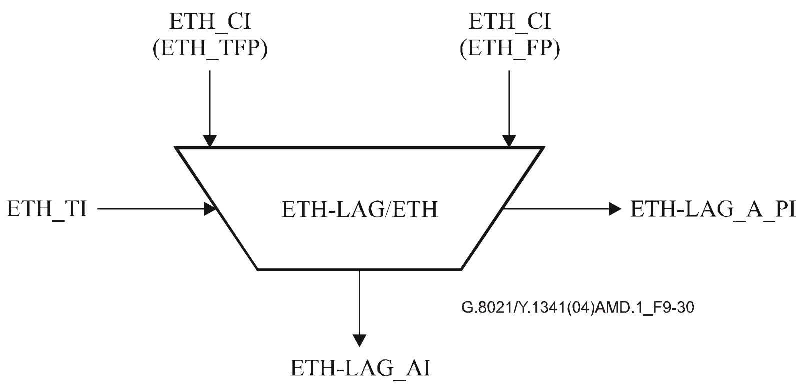


Figure 9-30 – ETH-LAG/ETH_A_So symbol

Interfaces

Table 9-11 – ETH-LAG/ETH_A_So interfaces

Inputs	Outputs
ETH_TFP: ETH_CI_D ETH_CI_P ETH_CI_DE ETH_CI_Clock **ETH_FP**: ETH_CI_D ETH_CI_P ETH_CI_DE ETH_CI_Clock **ETH_TP**: ETH_TI_Clock	**ETH-LAG_AP**: ETH-LAG_AI_D ETH-LAG_AI_P ETH-LAG_AI_DE ETH-LAG_AI_Clock **ETH-LAG_A_PP**: ETH-LAG_A_PI_D (ETHTF_PP) ETH-LAG_A_PI_P (ETHTF_PP) ETH-LAG_A_PI_DE (ETHTF_PP) ETH-LAG_A_PI_D (ETHF_PP) ETH-LAG_A_PI_P (ETHF_PP) ETH-LAG_A_PI_DE (ETHF_PP)

Processes

A process diagram of this function is shown in Figure 9-31.

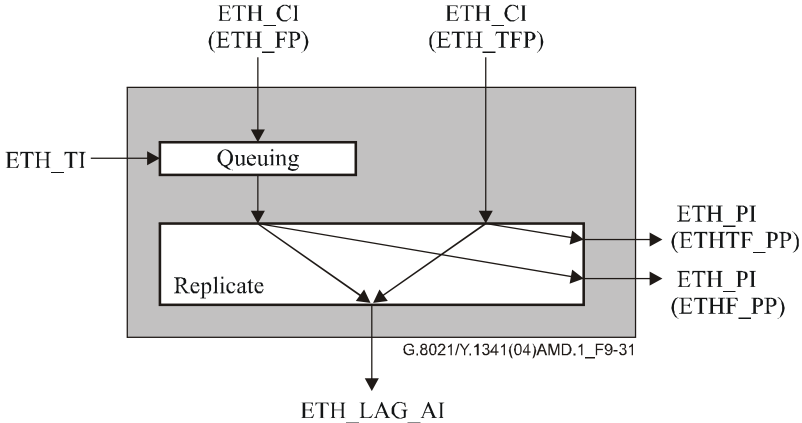


Figure 9-31 – ETH-LAG/ETH_A_So process diagram

These processes are as per the definitions of the queuing process in clause 8.2 and the replicate process in clause 8.4.

Defects *None*.

Consequent actions *None*.

Defect correlations *None*.

Performance monitoring *None*.

9.7.2.2 ETH-LAG to ETH link aggregation adaptation sink function (ETH-LAG/ETH_A_Sk)

Symbol

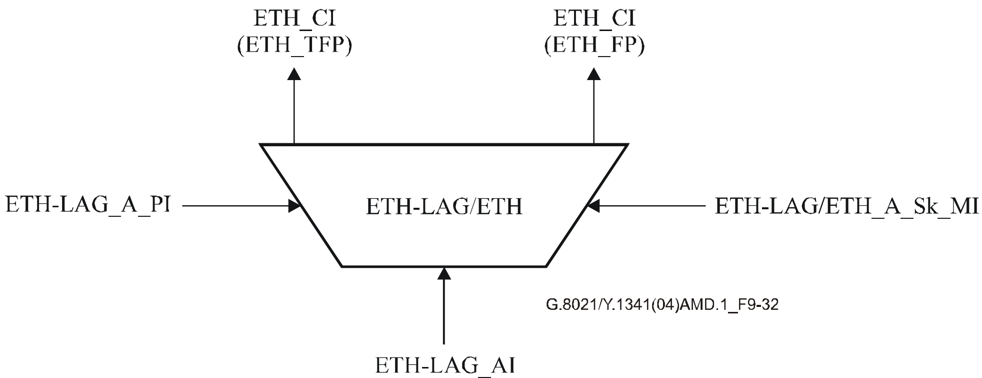


Figure 9-32 – ETH-LAG/ETH_A_Sk symbol

Interfaces

Table 9-12 – ETH-LAG/ETH_A_Sk interfaces

Inputs	Outputs
ETH-LAG_AP: ETH-LAG_AI_D ETH-LAG_AI_P ETH-LAG_AI_DE ETH-LAG_AI_Clock ETH-LAG-AI_TSF ETH-LAG-AI_AIS **ETH-LAG/ETH_A_Sk_MI**: ETH-LAG/ETH_A_Sk_MI_FilterConfig **ETH-LAG_A_PP**: ETH-LAG_A_PI_D (ETHTF_PP) ETH-LAG_A_PI_P (ETHTF_PP) ETH-LAG_A_PI_DE (ETHTF_PP) ETH-LAG_A_PI_D (ETHF_PP) ETH-LAG_A_PI_P (ETHF_PP) ETH-LAG_A_PI_DE (ETHF_PP)	**ETH_TFP**: ETH_CI_D ETH_CI_P ETH_CI_DE ETH_CI_Clock ETH_CI_SSF **ETH_FP**: ETH_CI_D ETH_CI_P ETH_CI_DE ETH_CI_Clock ETH_CI_SSF

Processes

A process diagram of this function is shown in Figure 9-33.

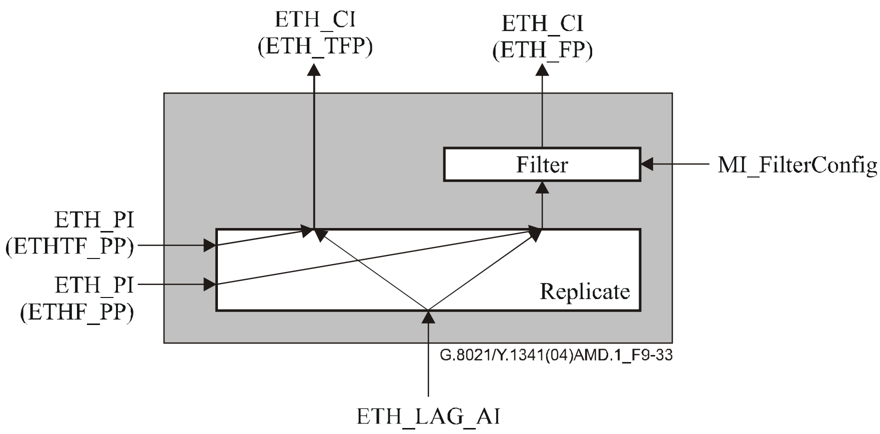


Figure 9-33 – ETH-LAG/ETH_A_Sk process diagram

These processes are as per the definitions of the filter process in clause 8.3 and the replicate process in clause 8.4.

Defects *None*.

Consequent actions *None*.

Defect correlations *None*.

Performance monitoring *None*.

7 Changes to clause 10

Modify clause 10 as follows:

10 Ethernet PHY Layer (ETYn) functions

This Recommendation supports the following full-duplex Ethernet PHYs:

- ETY1: 10BASE-T (twisted pair electrical; full-duplex only);
- ETY2.1: 100BASE-TX (twisted pair electrical; full-duplex only; *for further study*);
- ETY2.2: 100BASE-FX (optical; full-duplex only; *for further study*);
- ETY3.1: 1000BASE-T (copper; for *further study*);
- ETY3.2: 1000BASE-LX/SX (long- and short-haul optical; full duplex only);
- ETY3.3: 1000BASE-CX (short-haul copper; full duplex only; *for further study*);
- ETY4: 10GBASE-S/L/E (optical; *for further study*).

10.1 ETYn connection functions

Not applicable; there are no connection functions defined for this layer.

10.2 Ethernet PHY trail termination functions (ETYn_TT)

In the sink direction, Ethernet PHY trail termination functions (ETYn_TT) terminate received optical or electrical Ethernet signals, delivering a conditioned signal to the ETYn/ETH_Sk_A sink adaptation function. In the source direction, ETYn_TT trail termination accepts an electrical signal from the ETYn/ETH_So_A source adaptation function, and outputs an appropriate electrical or optical signal to the Ethernet electrical or optical delivery medium.

NOTE – The ETYn_TT functions are intended to encapsulate the whole functionality of the physical layer in the IEEE 802.3-2002 model. The models in this Recommendation define this functionality only by reference to the IEEE model and intentionally do not provide additional detail, as this functionality is well understood from the IEEE work.

The types of ETYn functions are as defined in Table 10-1:

Table 10-1 – ETYn types

ETYn type	IEEE 802.3-2002 interface type
ETY1	10BASE-T
ETY2.1	100BASE-TX
ETY2.2	100BASE-FX
ETY3.1	1000BASE-T
ETY3.2	1000BASE-LX/SX
ETY3.3	1000BASE-CX
ETY4	10GBASE-S/L/E

10.2.1 ETYn trail termination source function (ETYn_TT_So)

Symbol

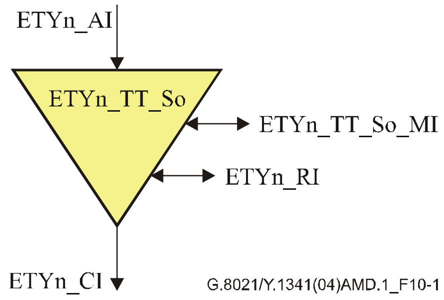


Figure 10-1 – ETYn_TT_So symbol

Interfaces

Table 10-2 – ETYn_TT_So interfaces

Inputs	Outputs
ETYn_AP: ETYn_AI_Data ETYn_AI_Clock ETYn_AI_SSF **ETHYn_RP**: ETYn_RI_RSF **ETYnTT_So_MP**: ETYn_TT_So_MI_FTSEnable	**ETYn_TFP**: ETYn_CI_Data ETYn_CI_Clock **ETYn_RP**: ETYn_RI_FTS **ETYn_TT_So_MP**: ETYn_TT_So_MI_PHYType ETYn_TT_So_MI_PHYTypeList

Processes

This source function together with the corresponding sink function implements all processes in the physical layer in the IEEE 802.3-2002 model.

Fault propagation process:

When the AI_SSF and the FTSEnable (forced transmitter shutdown) are true and the RI_RSF (remote signal fail) is false, this process forces the transmitter shutdown by either turning off the output transmitting device or inserting error codes (e.g., /V/, 10B_ERR for 1 GbE).

As soon as the transmitter shutdown is forced, the RI_FTS is asserted. The RI_FTS is reset [*for further study*] seconds after the enforcement of transmitter shutdown is removed.

NOTE – Further detail is intentionally left out of this Recommendation.

Defects *None.*

Consequent actions *None.*

Defect correlations *None.*

Performance monitoring *None.*

10.2.2 ETYn trail termination sink function (ETYn_TT_Sk)

Symbol

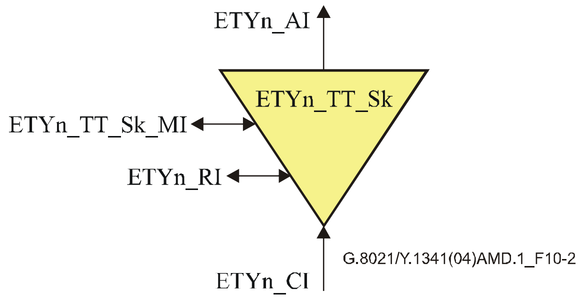


Figure 10-2 – ETYn_TT_Sk symbol

Interfaces

Table 10-3 – ETYn_TT_Sk interfaces

Inputs	Outputs
ETYn_TFP: ETYn_CI_Data **ETYn_RP**: ETYn_RI_FTS	**ETYn_AP**: ETYn_AI_Data ETYn_AI_Clock ETYn_AI_TSF **ETYn_RP**: ETYn_RI_RSF **ETYn_TT_Sk_MP**: ETYn_TT_Sk_MI_cLOS

Processes

This sink function together with the corresponding source function implements all processes in the physical layer in the IEEE 802.3-2002 model.

NOTE – Further detail is intentionally left out of this Recommendation.

Defects

dLOS: The defect is detected as soon as the aMediaAvailable parameter (as defined in IEEE 802.3-2002) gets a value different from "available" and the RI_FTS is false. The defect is cleared as soon as the aMediaAvailable parameter becomes "available".

Consequent actions

aTSF ← dLOS

aRSF ← dLOS

Defect correlations

cLOS ← dLOS

Performance monitoring *None.*

10.3 ETYn to ETH adaptation functions (ETYn/ETH_A)

Figures 10-3 and 10-4 illustrate the Ethernet trail termination to ETH adaptation function (ETYn/ETH_A and ETYn/ETH-m_A). Information crossing the ETH flow point (ETH_FP) and ETH termination flow point (ETH_TFP) is referred to as ETH characteristic information (ETH_CI). Information crossing the ETYn access point (ETY_AP) is referred to as ETYn adapted information (ETYn_AI).

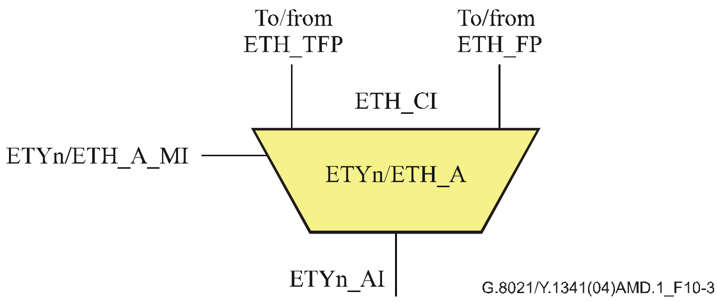


Figure 10-3 – ETYn server to ETH adaptation function

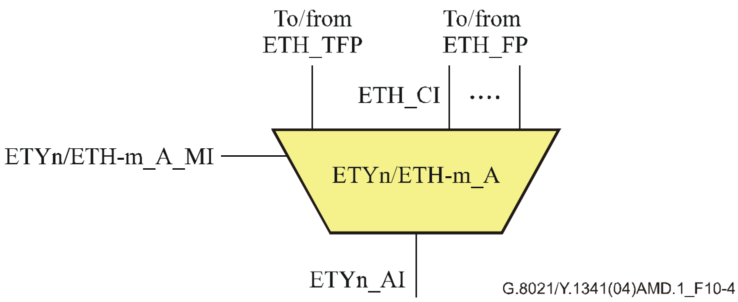


Figure 10-4 – ETYn server to ETH-m adaptation function

The ETYn/ETH_A adaptation function shown in Figure 10-3 can be further decomposed into separate source and sink adaptation functions as shown in Figure 10-5:

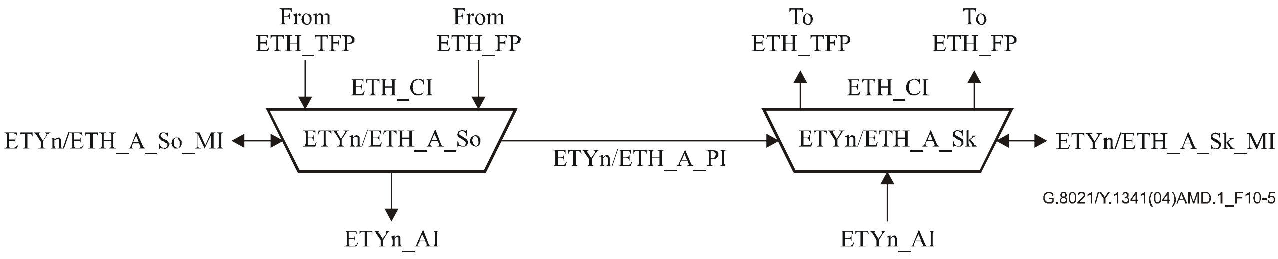


Figure 10-5 – ETYn/ETH_A source and sink adaptation functions

10.3.1 ETYn to ETH_A adaptation source function (ETYn/ETH_A_So)

Symbol

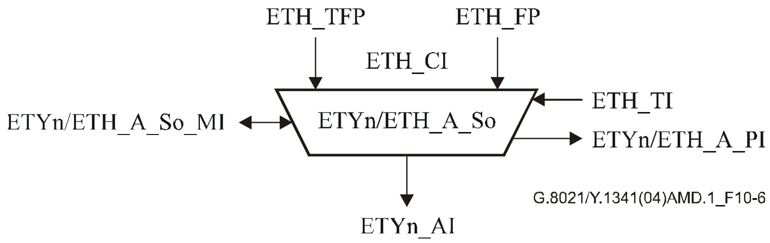


Figure 10-6 – ETYn/ETH_A_So symbol

Interfaces

Table 10-4 – ETYn/ETH_A_So interfaces

Inputs	Outputs
ETH_FP and ETH_TFP: ETH_CI_Data ETH_CI_Clock ETH_A_CI_PauseTrigger ETH_CI_SSF **ETYn/ETH_A_So_MP**: ETYn/ETH_A_So_MI_TxPauseEnable **ETH_TP**: ETH_TI_Clock	**ETYn_AP**: ETYn_AI_Data ETYn_AI_Clock ETYn_AI_SSF **ETH_PP**: ETH_PI_Data **ETYn/ETH_A_So_MP**: ETYn/ETH_A_So_MI_FramesTransmittedOK ETYn/ETH_A_So_MI_OctetsTransmittedOK

Processes

A process diagram of this function is shown in Figure 10-7.

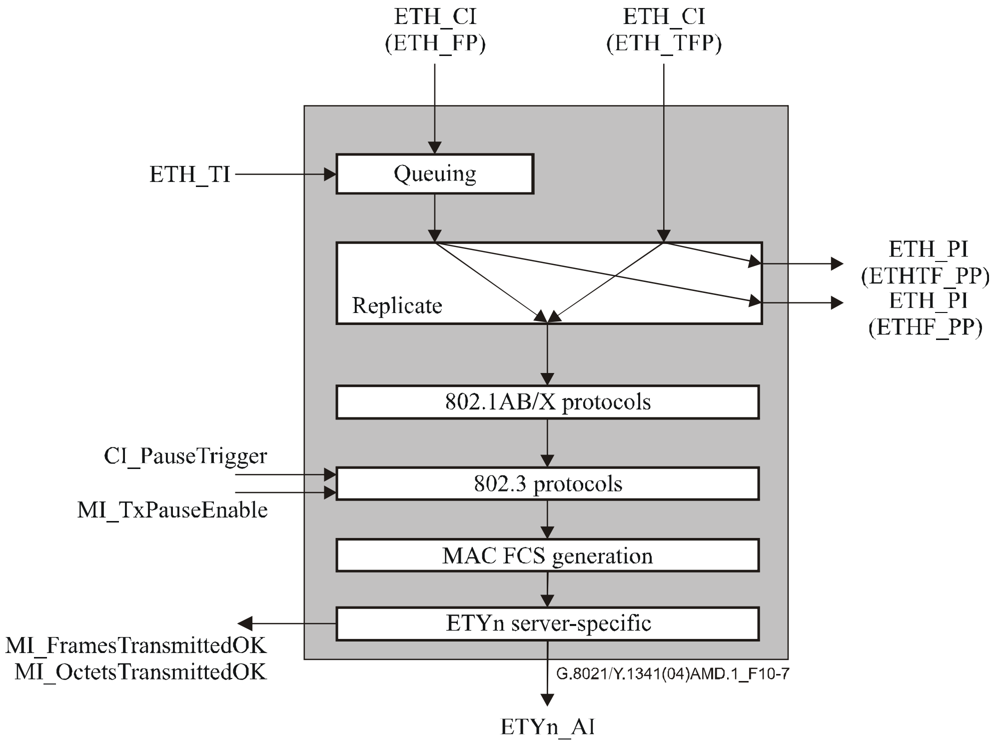


Figure 10-7 – ETYn/ETH_A_So process diagram

Processes

The "queuing", "replicate", "802.3 protocols", "802.1AB/X protocols" and "MAC FCS generation" processes are defined in clause 8 ("generic processes").

The "ETYn server-specific" source process is a null process.

NOTE – All source processes related to the Ethernet physical layer are encapsulated in this Recommendation by the ETYn_TT_So function.

The MAC frame counting process location is *for further study*.

Defects *None*.

Consequent actions *None*.

Defect correlations *None*.

Performance monitoring *For further study*.

10.3.2 ETYn to ETH_A adaptation sink function (ETYn/ETH_A_Sk)

Symbol

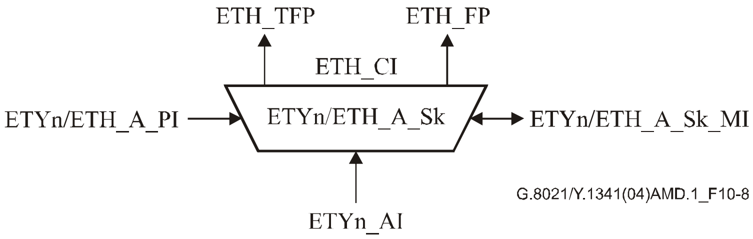


Figure 10-8 – ETYn/ETH_A_Sk symbol

Interfaces

Table 10-5 – ETYn/ETH_A_Sk interfaces

Inputs	Outputs
ETYn_AP: ETYn_AI_Data ETYn_AI_Clock **ETH_PP**: ETH_PI_Data **ETYn/ETH_A_Sk_MP**: ETYn/ETH_A_Sk_MI_FilterConfig ETYn/ETH_A_Sk_MI_MAC_Length	**ETH_FP and ETH_TFP**: ETH_CI_Data ETH_CI_Clock ETH_CI_SSF **ETYn/ETH_A_Sk_MP**: ETYn/ETH_A_Sk_MI_pErrors ETYn/ETH_A_Sk_MI_FramesReceivedOK ETYn/ETH_A_Sk_MI_OctetsReceivedOK

Processes

A process diagram of this function is shown in Figure 10-9.

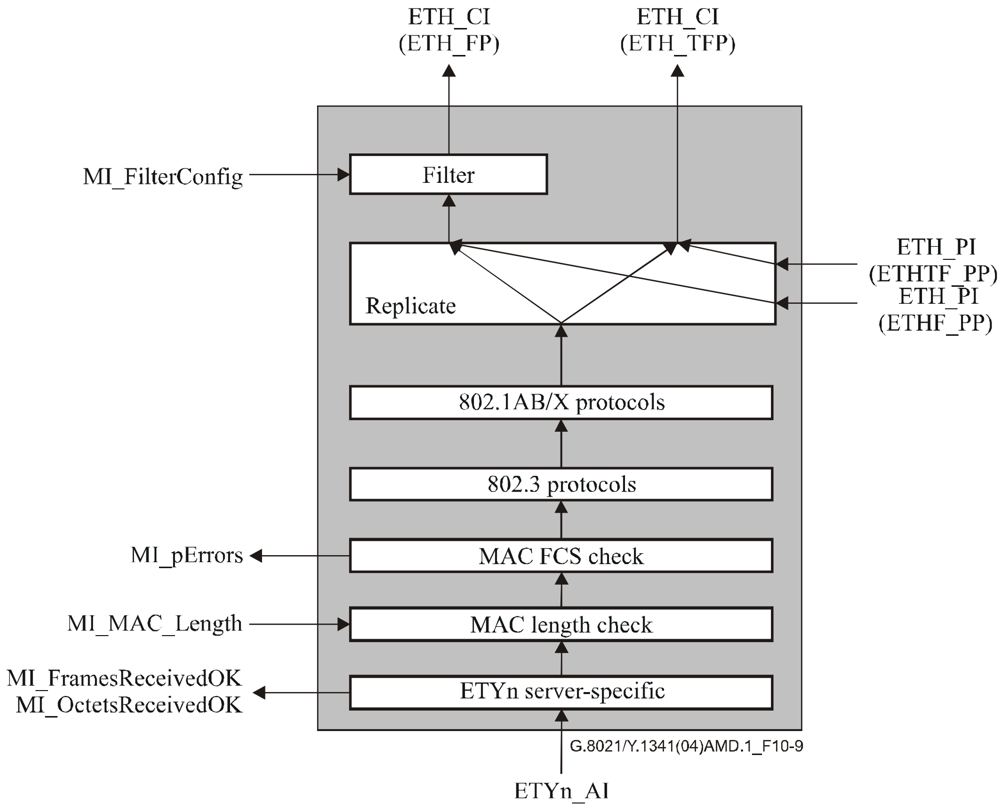


Figure 10-9 – ETYn/ETH_A_Sk process diagram

The "filter", "replicate", "802.3 protocols", "802.1AB/X protocols", "MAC frame check" and "MAC length check" processes are defined in clause 8 ("generic processes").

The "ETYn server-specific" sink process is a null process.

NOTE – All sink processes related to the Ethernet physical layer are encapsulated in this Recommendation by the ETYn_TT_Sk function.

MAC frame counting is *for further study*.

Defects *None*.

Consequent actions

aSSF ← AI_TSF

Defect correlations *None*.

Performance monitoring *For further study.*

10.4 1000BASE-(SX/LX/CX) ETY/coding sub-layer adaptation functions (ETY3/ETC3_A)

This adaptation function adapts 1000BASE-SX, -LX or -CX physical layer signals from/to 8B/10B-encoded codewords. Codewords may be extracted from or mapped into GFP-T frames, per clause 11.2 SDH/ETC adaptation functions (S4-X/ETC3_A).

For further study.

8 Changes to clause 11

8.1 Changes to clauses 11.1.1.1 and 11.1.1.2

Changes are indicated by change bars.

11.1.1.1 VC-n/ETH adaptation source function (Sn/ETH_A_So)

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

Data at the Sn_AP is a VC-n (n = 3, 3-X, 4, 4-X), having a payload as described in ITU-T G.707/Y.1322, but with indeterminate POH bytes: J1, B3, G1.

Symbol

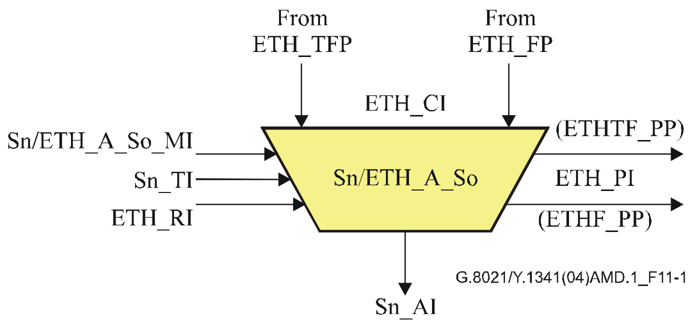


Figure ~~21~~11-1 – Sn/ETH_A_So symbol

Interfaces

Table ~~5~~11-1 – Sn/ETH_A_So interfaces

Inputs	Outputs
ETH_TFP: ETH_CI_Data **ETH_FP**: ETH_CI_Data ETH_CI_SSF **ETH_RP**: ETH_RI_RSF **Sn_TI**: Sn_TI_Clock Sn_TI_FrameStart **Sn/ETH_A_So_MI**: Sn/ETH_A_So_MI_CSFEnable	**Sn_AP**: Sn_AI_Data Sn_AI_Clock Sn_AI_FrameStart **ETHF_PP**: ETH_PI_Data **ETHTF_PP**: ETH_PI_Data

Processes

A process diagram of this function is shown in Figure ~~22~~11-2.

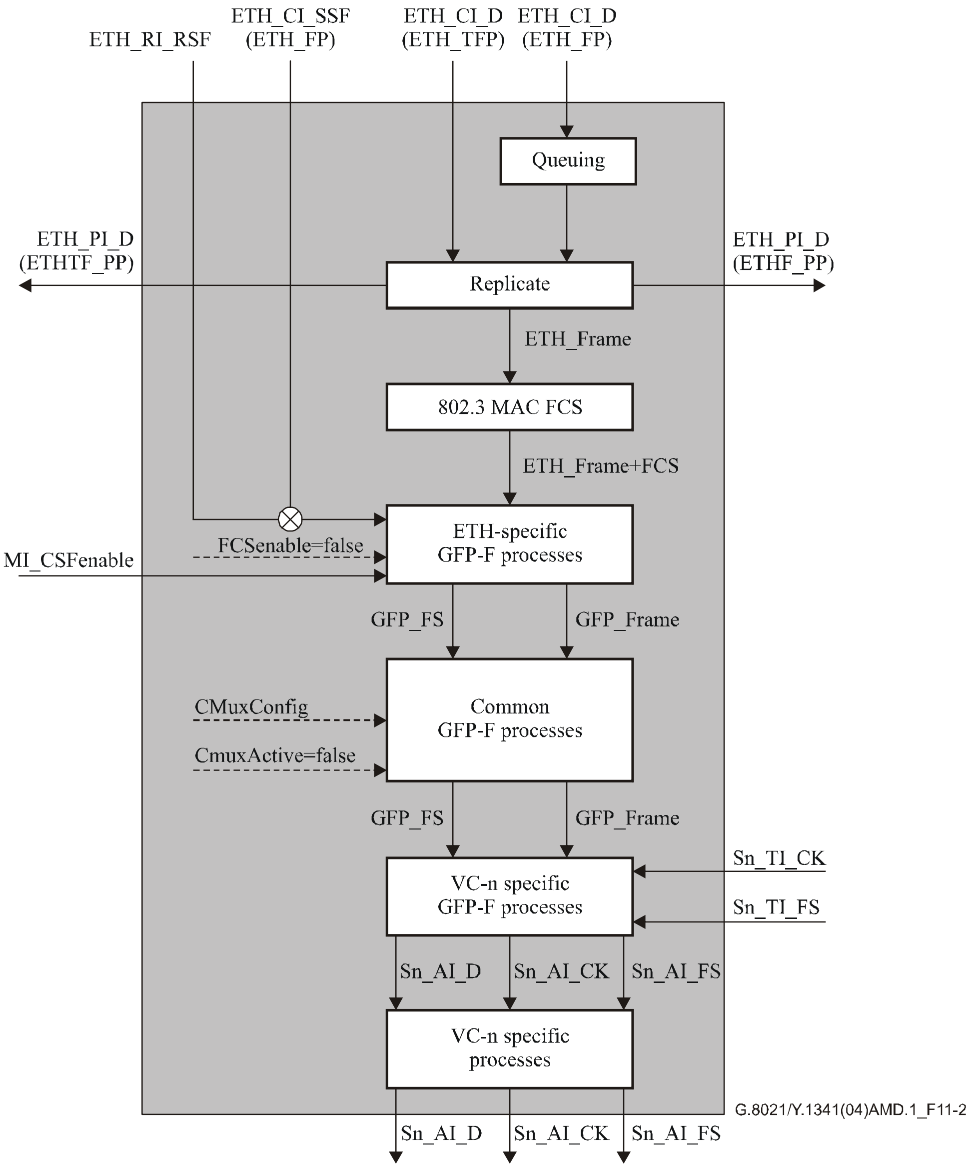


Figure ~~22~~11-2 – Sn/ETH_A_So process diagram

Queuing process:

See 8.2.

Replicate process:

See 8.4.

802.3 MAC FCS generation:

See ~~8.6~~8.9.1.

Ethernet-specific GFP-F source process:

See 8.5.4.1.1/G.806. GFP pFCS generation is disabled (FCSenable=false). The UPI value for frame-mapped Ethernet shall be inserted (see Table 6-3/G.7041/Y.1303). The Ethernet frames are inserted into the client payload information field of the GFP-F frames according to 7.1/G.7041/Y.1303.

Response to ETH_CI_SSF is *for further study*.

Common GFP source process:

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

VC-n specific GFP source process:

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

VC-n specific source process:

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

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

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

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

Counter processes: *For further study.*

Defects *None.*

Consequent actions *None.*

Defect correlations *None.*

Performance monitoring *For further study.*

11.1.1.2 VC-n/ETH adaptation sink function (Sn/ETH_A_Sk)

This function extracts ETH_CI information from the Sn_AI signal (n = 3, 3-X, 4, 4-X), delivering ETH_CI to ETH_TFP and ETH_FP.

Data at the Sn_AP is as described in ITU-T Rec. G.707/Y.1322.

Symbol

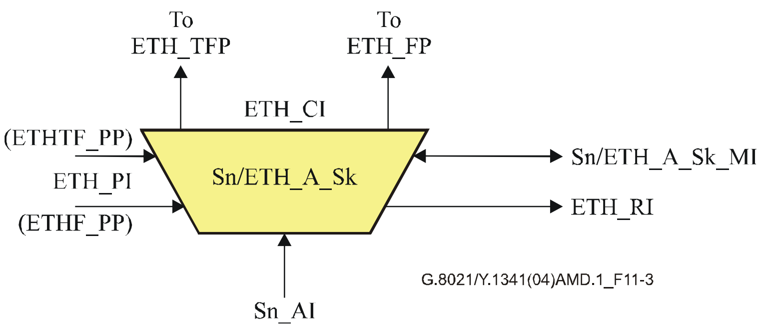


Figure ~~23~~11-3 – Sn/ETH_A_Sk symbol

Interfaces

Table ~~6~~11-2 – Sn/ETH_A_Sk interfaces

Inputs	Outputs
Sn_AP: Sn_AI_Data Sn_AI_ClocK Sn_AI_FrameStart Sn_AI_TSF **ETHF_PP**: ETH_PI_Data **ETHTF_PP**: ETH_PI_Data **Sn/ETH_A_Sk_MI**: Sn/ETH_A_Sk_MI_FilterConfig Sn/ETH_A_Sk_MI_CSF_Reported Sn/ETH_A_Sk_MI_MAC_Length	**ETH_TFP**: ETH_CI_Data ETH_CI_SSF **ETH_FP**: ETH_CI_Data ETH_CI_SSF **ETH_RP**: ETH_RI_RSF **Sn/ETH_A_Sk_MI**: Sn/ETH_A_Sk_MI_AcSL Sn/ETH_A_Sk_MI_AcEXI Sn/ETH_A_Sk_MI_AcUPI Sn/ETH_A_Sk_MI_cPLM Sn/ETH_A_Sk_MI_cLFD Sn/ETH_A_Sk_MI_cUPM Sn/ETH_A_Sk_MI_cEXM Sn/ETH_A_Sk_MI_cCSF Sn/ETH_A_Sk_MI_pErrors

Processes

A process diagram of this function is shown in Figure ~~24~~11-4.

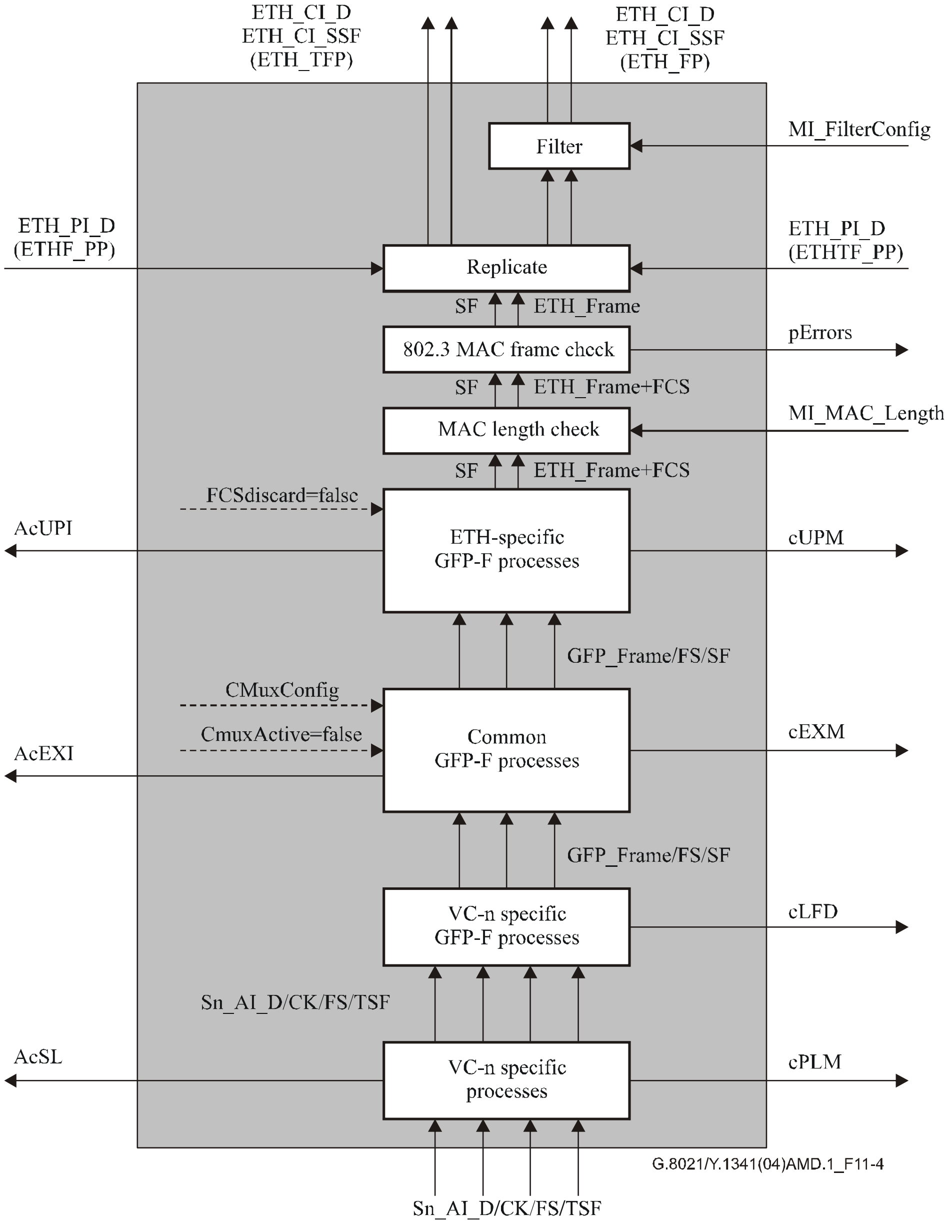


Figure ~~24~~11-4 – Sn/ETH_A_Sk process diagram

Filter process:

See 8.3.

Replicate process:

See 8.4.

802.3 MAC frame check process:

See ~~8.7~~8.9.2.

Ethernet-specific GFP-F sink process:

See 8.5.4.1.2/G.806. GFP pFCS checking, GFP p_FCSError, p_FDis are not supported (FCSdiscard=false). The UPI value for frame-mapped Ethernet shall be expected (see Table 6-3/G.7041/Y.1303). The Ethernet frames are extracted from the client payload information field of the GFP-F frames according to 7.1/G.7041/Y.1303.

Common GFP sink process:

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

VC-n specific GFP sink process:

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

VC-n specific sink process:

C2: The signal label is recovered from the C2 byte as per 6.2.4.2/G.806. The signal label for "GFP mapping" in Table 9-11/G.707/Y.1322 shall be expected. The accepted value of the signal label is also available at the Sn/ETH_A_Sk_MP.

<u>**MAC frame counter**: *For further study.*</u>

Defects

dPLM – See 6.2.4.2/G.806.

dLFD – See 6.2.5.2/G.806.

dUPM – See 6.2.4.3/G.806.

dEXM – See 6.2.4.4/G.806.

Consequent actions

The function shall perform the following consequent actions:

aSSF ← AI_TSF or dPLM or dLFD or dUPM or dEXM or dCSF

<u>aRSF ← AI_TSF or dPLM or dLFD or dUPM or dEXM</u>

Defect correlations

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

cPLM ← dPLM and (not AI_TSF)

cLFD ← dLFD and (not dPLM) and (not AI_TSF)

cUPM ← dUPM and <u>(not dEXM) and </u>(not dPLM) and (not dLFD) and (not AI_TSF)

cEXM ← dEXM and ~~(not dUPM) and~~ (not dPLM) and (not dLFD) and (not AI_TSF)

cCSF ← <u>dCFS and (not dEXM) and (not dUPM) and (not dPLM) and (not dLFD) and (not AI_TSF) and CSF_Reported</u>~~per G.806 section 8.5.4.1.2~~.

Performance monitoring *For further study.*

8.2 Changes to clause 11.2

11.2 SDH/ETC adaptation functions (S4-X/ETC3_A)

This covers GFP-T-based mapping of Gigabit Ethernet codewords into VC-4-Xv.

Replace the text:

For further study.

By:

11.2.1 VC-4-X/ETC3 adaptation source function (S4-X/ETC3_A_So)

This function maps ETC_CI information from an ETC3 onto an S4-X_AI signal. This mapping is currently only defined for X = 7.

Data at the S4-X_AP is a VC-4-Xv, having a payload as described in ITU-T Rec. G.707/Y.1322, but with indeterminate POH bytes: J1, B3 and G1.

Symbol

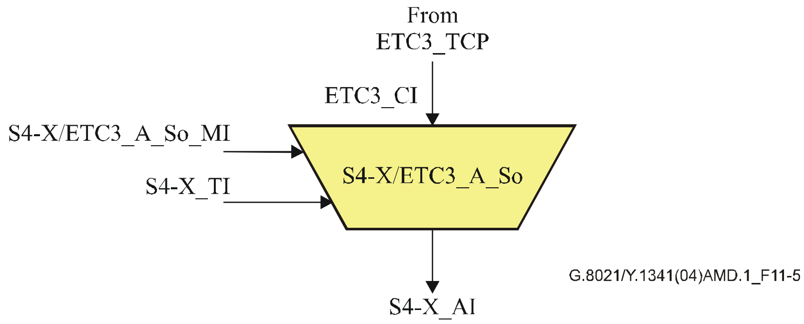


Figure 11-5 – S4-X/ETC3_A_So symbol

Interfaces

Table 11-3 – S4-X/ETC3_A_So interfaces

Inputs	Outputs
ETC3_TCP: ETC3_CI_Data_Control ETC3_CI_Clock ETC3_CI_Control_Ind ETC3_CI_SSF **S4-X_TP**: S4-X_TI_Clock S4-X_TI_FrameStart **S4-X/ETC3_A_So_MP**: S4-X/ETC3_A_So_MI_CSFEnable	**S4-X _AP**: S4-X_AI_Data S4-X_AI_Clock S4-X_AI_FrameStart

Processes

A process diagram of this function is shown in Figure 11-16.

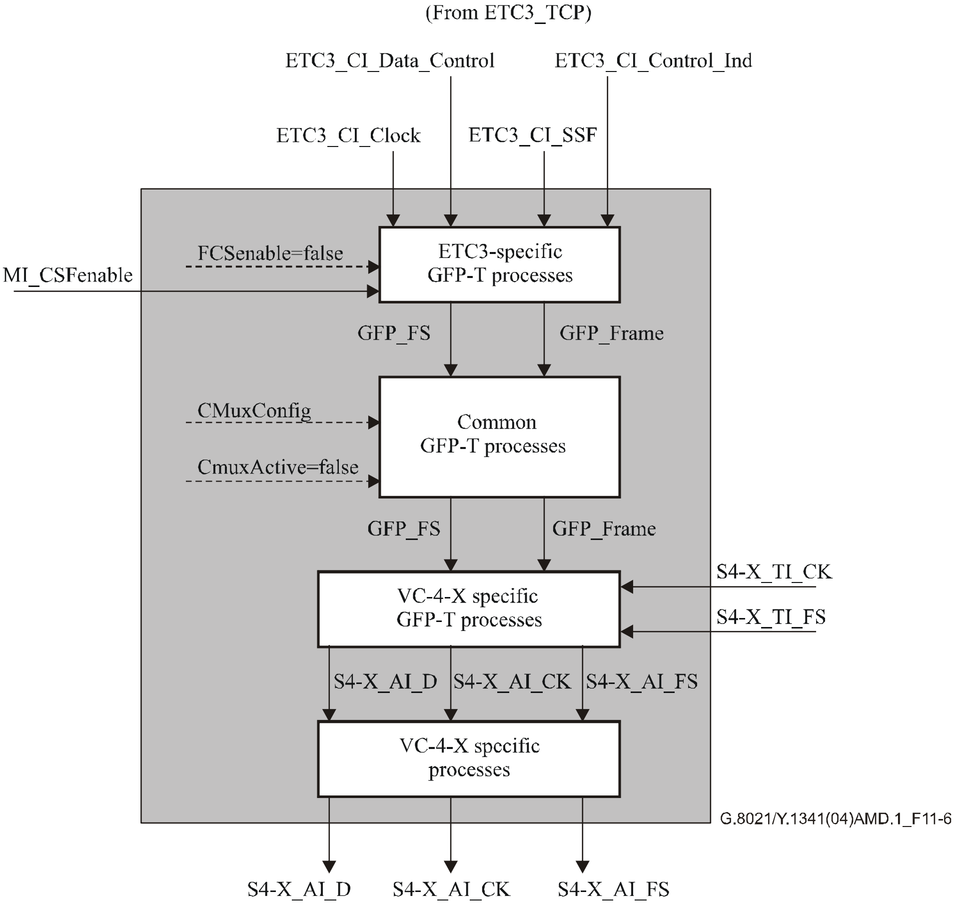


Figure 11-6 – S4-X/ETC3_A_So process diagram

Ethernet-specific GFP-T source process:

See 8.5.4.2.1/G.806. GFP pFCS generation is disabled (FCSenable=false). The UPI value for transparent Gb Ethernet shall be inserted (Table 6-3/G.7041/Y.1303). The Ethernet codeword information is inserted into the client payload information field of the GFP-T frames according to clause 8/G.7041/Y.1303.

Response to ETC3_CI_SSF is according to the principles in 8.3 and 8.3.4/G.7041/Y.1303 and Appendix VIII/G.806. Details are *for further study*.

Common GFP source process:

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

VC-4-X specific GFP source process:

See 8.5.2.1/G.806. The GFP frames are mapped into the VC-4-X payload area according to 10.6/G.707/Y.1322.

VC-4-X specific source process:

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

NOTE – For S4-X/ETC3_A_So, the H4, K3, F2 and F3 bytes are undefined at the S4-X_AP output of this function (as per clause 12/G.783).

Defects *None*.

Consequent actions *None*.

Defect correlations *None*.

Performance monitoring *For further study*.

11.2.2 VC-4-X/ETC3 adaptation sink function (S4-X/ETC3_A_Sk)

This function extracts ETC3_CI information from the S4-X_AI signal, delivering ETC3_CI to the ETC3_TCP.

Data at the S4-X_AP is as described in ITU-T Rec. G.707/Y.1322.

Symbol

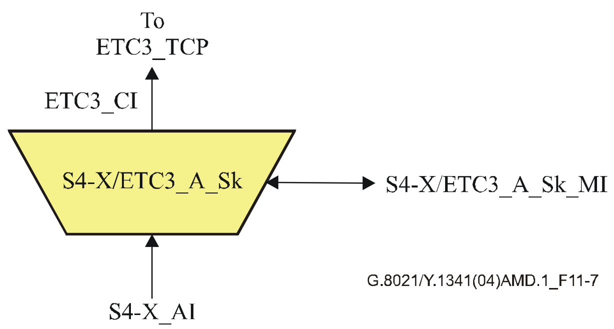


Figure 11-7 – S4-X/ETC3_A_Sk symbol

Interfaces

Table 11-4 – S4-X/ETC3_A_Sk interfaces

Inputs	Outputs
S4-X_AP: S4-X_AI_Data S4-X_AI_ClocK S4-X_AI_FrameStart S4-X_AI_TSF **S4-X/ETC3_A_Sk_MI**: S4-X/ETC3_A_Sk_MI_CSF_Reported	**ETC3_TCP**: ETC3_CI_Data_Control ETC3_CI_Clock ETC3_CI_Control_Ind ETC3_CI_SSF **S4-X/ETC3_A_Sk_MI**: S4-X/ETC3_A_Sk_MI_AcSL S4-X/ETC3_A_Sk_MI_AcEXI S4-X/ETC3_A_Sk_MI_AcPFI S4-X/ETC3_A_Sk_MI_AcUPI S4-X/ETC3_A_Sk_MI_cPLM S4-X/ETC3_A_Sk_MI_cLFD S4-X/ETC3_A_Sk_MI_cUPM S4-X/ETC3_A_Sk_MI_cEXM S4-X/ETC3_A_Sk_MI_cCSF S4-X/ETC3_A_Sk_MI_pCRC16Errors

Processes

A process diagram of this function is shown in Figure 11-8.

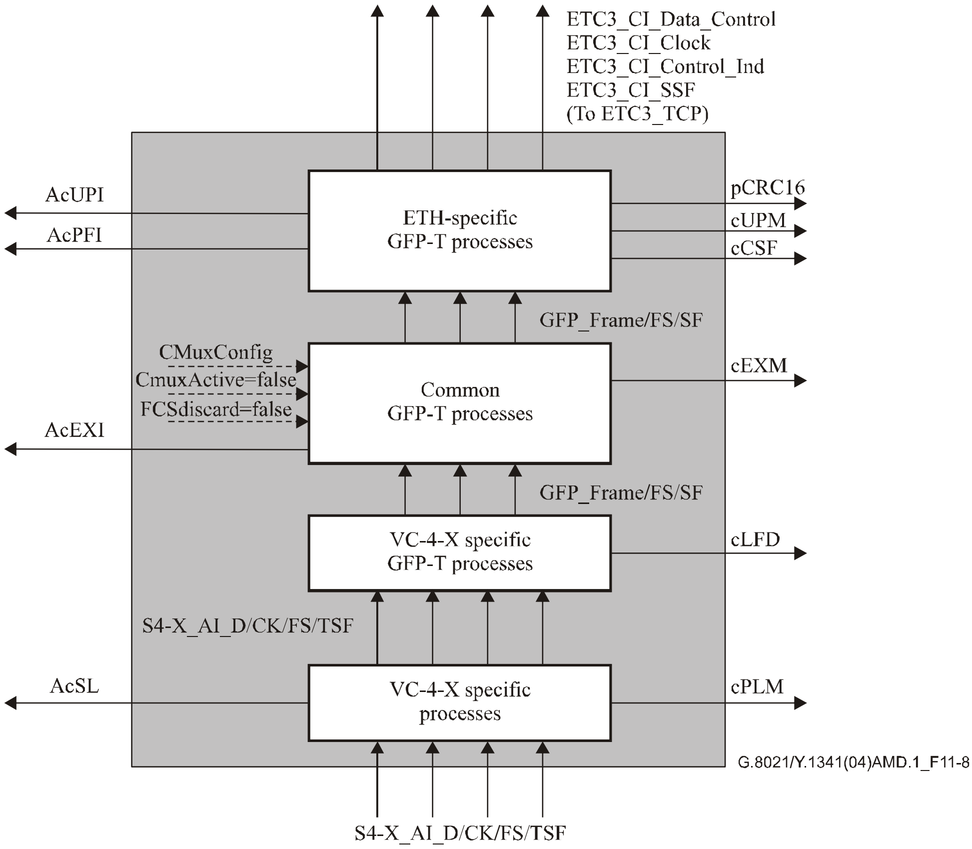


Figure 11-8 – S4-X/ETC3_A_Sk process diagram

Ethernet-specific GFP-T sink process:

See 8.5.4.1.2/G.806. GFP pFCS checking and GFP p_FCSError are not supported (FCSdiscard=false). The UPI value for transparent Gb Ethernet shall be expected (Table 6-3/G.7041/Y.1303). Frames discarded due to incorrect PFI or UPI values shall be counted in _pFDis. Errors detected in a received superblock are reported as a _pCRC16Error. If ECenable=true, then single transmission channel errors in the superblock shall be corrected using the superblock CRC-16. The Ethernet codeword information is extracted from the client payload information field of the GFP-F frames according to 8/G.7041/Y.1303.

Common GFP sink process:

See 8.5.3.1/G.806. GFP channel multiplexing is not supported (CMuxActive=false). Frames discarded due to EXI mismatch or errors detected by the tHEC shall be counted in _pFDis.

VC-4-X specific GFP sink process:

See 8.5.2.1/G.806. The GFP frames are demapped from the VC-4-X payload area according to 10.6/G.707/Y.1322.

VC-4-X specific sink process:

C2: The signal label is recovered from the C2 byte as per 6.2.4.2/G.806. The signal label for "GFP mapping" in Table 9-11/G.707/Y.1322 shall be expected. The accepted value of the signal label is also available at the S4-X/ETC3_A_Sk_MP.

Defects

dPLM – See 6.2.4.2/G.806.

dLFD – See 6.2.5.2/G.806.

dUPM – See 6.2.4.3/G.806.

dEXM – See 6.2.4.4/G.806.

dCSF – See 6.2.6.4/G.806.

Consequent actions

The function shall perform the following consequent actions:

aSSF ← AI_TSF or dPLM or dLFD or dUPM or dEXM or dCSF

Defect correlations

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

cPLM ← dPLM and (not AI_TSF)

cLFD ← dLFD and (not dPLM) and (not AI_TSF)

cUPM ← dUPM and (not dPLM) and (not dLFD) and (not AI_TSF)

cEXM ← dEXM and (not dUPM) and (not dPLM) and (not dLFD) and (not AI_TSF)

cCSF per 8.5.4.2.2/G.806.

Performance monitoring

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

pCRC16Errors: count of superblock CRC-16 errors per second

_pFDis = sum (n_FDis_tHEC + n_FDis_eHEC_EXI + n_FDis_PTI_UPI)

8.3 Change to clause 11.4

Replace the text:

For further study.

By:

11.4 PDH/ETH adaptation functions (p/ETH_A)

11.4.1 Pq/ETH adaptation functions (Pq/ETH_A; q = 11s, 12s, 31s, 32e)

11.4.1.1 Pq/ETH adaptation source function (Pq/ETH_A_So)

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

Data at the Pq_AP is a Pq (q = 11s, 12s, 31s, 32e), having a payload as described in ITU-T Rec. G.7043/Y.1343 with a value of N=1. The VLI byte is reserved and not used for payload data.

Symbol

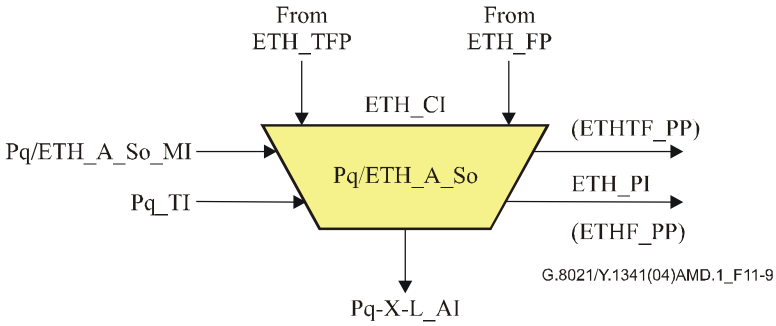


Figure 11-9 – Pq/ETH_A_So symbol

Interfaces

Table 11-5 – Pq/ETH_A_So interfaces

Inputs	Outputs
ETH_TFP: ETH_CI_D ETH_CI_P ETH_CI_DE **ETH_FP**: ETH_CI_Data ETH_CI_SSF **Pq_TP**: Pq_TI_ClocK Pq_TI_FrameStart **Pq/ETH_A_So_MP**: Pq/ETH_A_So_MI_CSFEnable	**Pq_AP**: Pq_AI_Data Pq_AI_ClocK Pq_AI_FrameStart **ETHF_PP**: ETH_PI_D ETH_PI_P ETH_PI_DE **ETHTF_PP**: ETH_PI_D ETH_PI_P ETH_PI_DE

Processes

A process diagram of this function is shown in Figure 11-10.

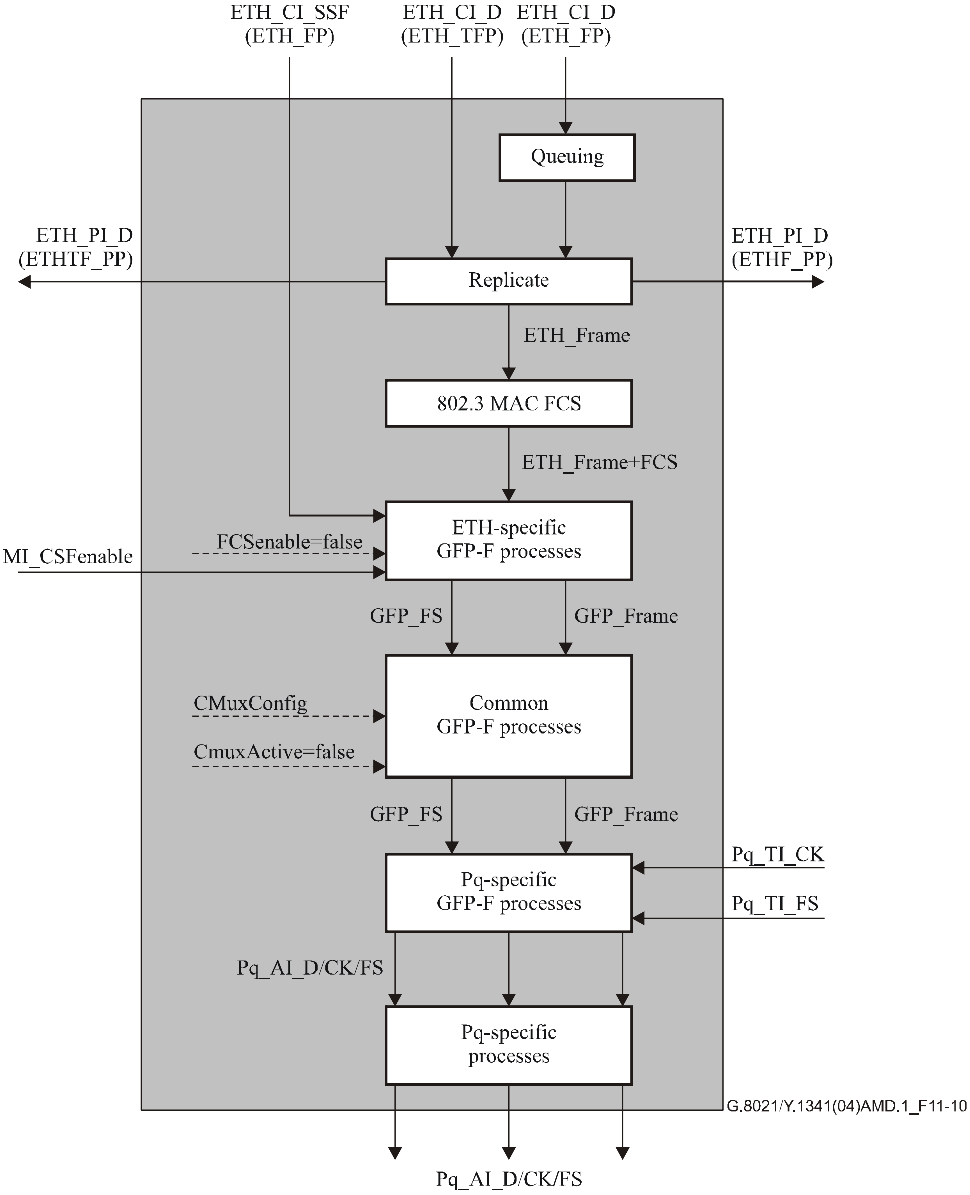


Figure 11-10 – Pq/ETH_A_So process diagram

Queuing process:

See 8.2.

Replicate process:

See 8.4.

802.3 MAC FCS generation:

See 8.9.1.

Ethernet-specific GFP-F source process:

See 8.5.4.1.1/G.806. GFP pFCS generation is disabled (FCSenable=false). The UPI value for frame-mapped Ethernet shall be inserted (Table 6-3/G.7041/Y.1303). The Ethernet frames are inserted into the client payload information field of the GFP-F frames according to 7.1/G.7041/Y.1303.

Response to ETH_CI_SSF asserted is *for further study*.

Common GFP source process:

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

Pq-specific GFP source process:

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

Pq-specific source process:

NOTE – The VLI byte is fixed stuff equal to 0x00 at the Pq_AP output of this function.

P31s specific:

MA: Signal label information is derived directly from the adaptation function type. The value for "GFP mapping" in 2.1/G.832 is placed in the payload type field of the MA byte.

Defects *None*.

Consequent actions *None*.

Defect correlations *None*.

Performance monitoring *For further study*.

11.4.1.2 Pq/ETH adaptation sink function (Pq/ETH_A_Sk)

This function extracts ETH_CI information from a Pq_AI signal (q = 11s, 12s, 31s, 32e), delivering ETH_CI to ETH_TFP and ETH_FP.

Data at the Pq_AP is a Pq (q = 11s, 12s, 31s, 32e), having a payload as described in ITU-T Rec. G.7043/Y.1343 with a value of N = 1. The VLI byte is reserved and not used for payload data.

Symbol

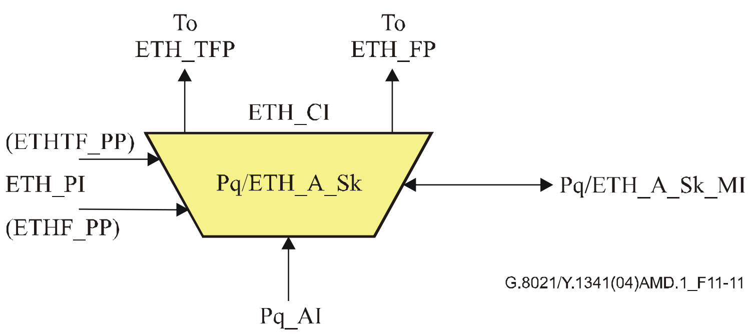


Figure 11-11 – Pq/ETH_A_Sk symbol

Interfaces

Table 11-6 – Pq/ETH_A_Sk interfaces

Inputs	Outputs
Pq_AP: Pq_AI_Data Pq_AI_ClocK Pq_AI_FrameStart Pq_AI_TSF **ETHF_PP**: ETH_PI_D ETH_PI_P ETH_PI_DE **ETHTF_PP**: ETH_PI_D ETH_PI_P **Pq/ETH_A_Sk_MP**: Pq/ETH_A_Sk_MI_FilterConfig Pq/ETH_A_Sk_MI_CSF_Reported	**ETH_TFP**: ETH_CI_D ETH_CI_P ETH_CI_DE ETH_CI_SSF **ETH_FP**: ETH_CI_D ETH_CI_P ETH_CI_DE ETH_CI_SSF **Pq/ETH_A_Sk_MP**: Pq/ETH_A_Sk_MI_AcSL Pq/ETH_A_Sk_MI_AcEXI Pq/ETH_A_Sk_MI_AcUPI Pq/ETH_A_Sk_MI_cPLM Pq/ETH_A_Sk_MI_cLFD Pq/ETH_A_Sk_MI_cUPM Pq/ETH_A_Sk_MI_cEXM Pq/ETH_A_Sk_MI_cCSF Pq/ETH_A_Sk_MI_pFCSError

Processes

A process diagram of this function is shown in Figure 11-12.

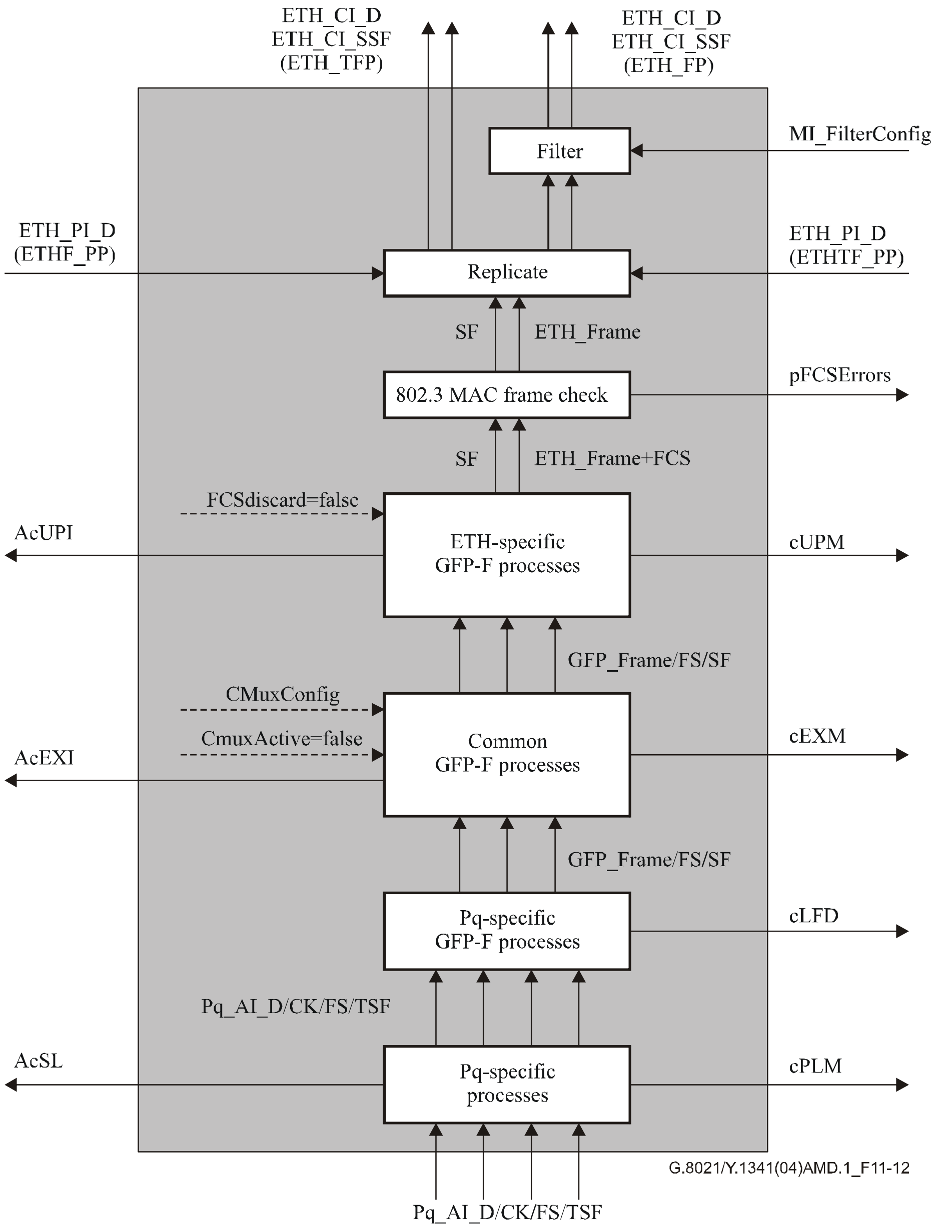


Figure 11-12 – Pq/ETH_A_Sk process diagram

Filter process:

See 8.3.

Replicate process:

See 8.4.

802.3 MAC frame check process:

See 8.9.2.

Ethernet-specific GFP-F sink process:

See 8.5.4.1.2/G.806. GFP pFCS checking, GFP p_FCSError and p_FDis are not supported (FCSdiscard=false). The UPI value for frame-mapped Ethernet shall be expected (Table 6-3/G.7041/Y.1303). The Ethernet frames are extracted from the client payload information field of the GFP-F frames according to 7.1/G.7041/Y.1303.

Common GFP sink process:

See 8.5.3.2/G.806. GFP channel multiplexing is not supported (CMuxActive=false).

Pq-specific GFP sink process:

See 8.5.2.2/G.806. The GFP frames are demapped from the Pq payload area according to ITU-T Rec. G.8040/Y.1340.

Pq-specific sink process:

NOTE 1 – The VLI byte at the Pq_AP input of this function is ignored.

P31s specific:

MA: The signal label is recovered from the payload type field in the MA byte as per 6.2.4.2/G.806. The signal label for "GFP mapping" in 2.1/G.832 shall be expected. The accepted value of the signal label is also available at the P31s/ETH_A_Sk_MP.

Defects

dPLM – See 6.2.4.2/G.806.

dLFD – See 6.2.5.2/G.806.

dUPM – See 6.2.4.3/G.806.

dEXM – See 6.2.4.4/G.806.

NOTE 2 – dPLM is only defined for q = 31s. dPLM is assumed to be false for q = 11s, 12s, 32e.

Consequent actions

The function shall perform the following consequent actions:

aSSF ← AI_TSF or dPLM or dLFD or dUPM or dEXM or dCSF

Defect correlations

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

cPLM ← dPLM and (not AI_TSF)

cLFD ← dLFD and (not dPLM) and (not AI_TSF)

cUPM ← dUPM and (not dPLM) and (not dLFD) and (not AI_TSF)

cEXM ← dEXM and (not dUPM) and (not dPLM) and (not dLFD) and (not AI_TSF)

cCSF per 8.5.4.1.2/G.806.

Performance monitoring

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

pFCSError: count of FrameCheckSequenceErrors per second.

NOTE 3 – This primitive is calculated by the MAC frame check process.

9 Additions of Appendices

Add the following text as new Appendix II:

Appendix II

AIS/RDI mechanism for an Ethernet private line over a single SDH or OTH server layer

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

In order to address fault notification for failures in either the access links or within the SDH/OTH server layer, the following functionality is required:

a) Convey fault notification for an access link failure from one side of the network to the other.

b) Convey fault notification for an SDH/OTH server layer failure to the access links.

ITU-T Rec. G.7041/Y.1303 defines client management frames (CMFs) for conveying information about the client signal from an ingress edge NE to the egress edge NE. Defined CMF indications are client signal fail (CSF) and remote fail indication (RFI).

ITU-T Rec. G.806 defines the equipment functional details of the CSF and RFI mechanisms.

This Recommendation defines the Ethernet-specific equipment functional details for the CSF and RFI mechanisms.

The combination of the above three Recommendations provides the functionality required by a) and b) above.

That basic functionality can be further enhanced by using clause 57 of [b-802.3ak] (EFM OAM) link fault flag in conjunction with the GFP-F CMF CSF and RFI indications, as illustrated in this appendix.

A simplifying assumption can be made regarding the conditioning of the Ethernet access links on either side of the SDH/OTH transport network. For an EPL application, the access link is specific to a single service, and since an Ethernet service is bidirectional, a fault in either direction should result in the access link being conditioned as "failed".

The following fault scenarios and accompanying figures illustrate the proposed interworking of the EFM OAM link fault flag with the GFP-F CMF CSF and RFI indications to appropriately condition the Ethernet access links. Only unidirectional faults are considered, the scenarios can be combined per the superposition principle to describe bidirectional faults. Further, only an SDH server layer is shown in the examples. CE = customer edge. PE = provider edge.

Scenario 1

In Figure II.1 below a unidirectional fault occurs on the east access link on ingress to the carrier network.

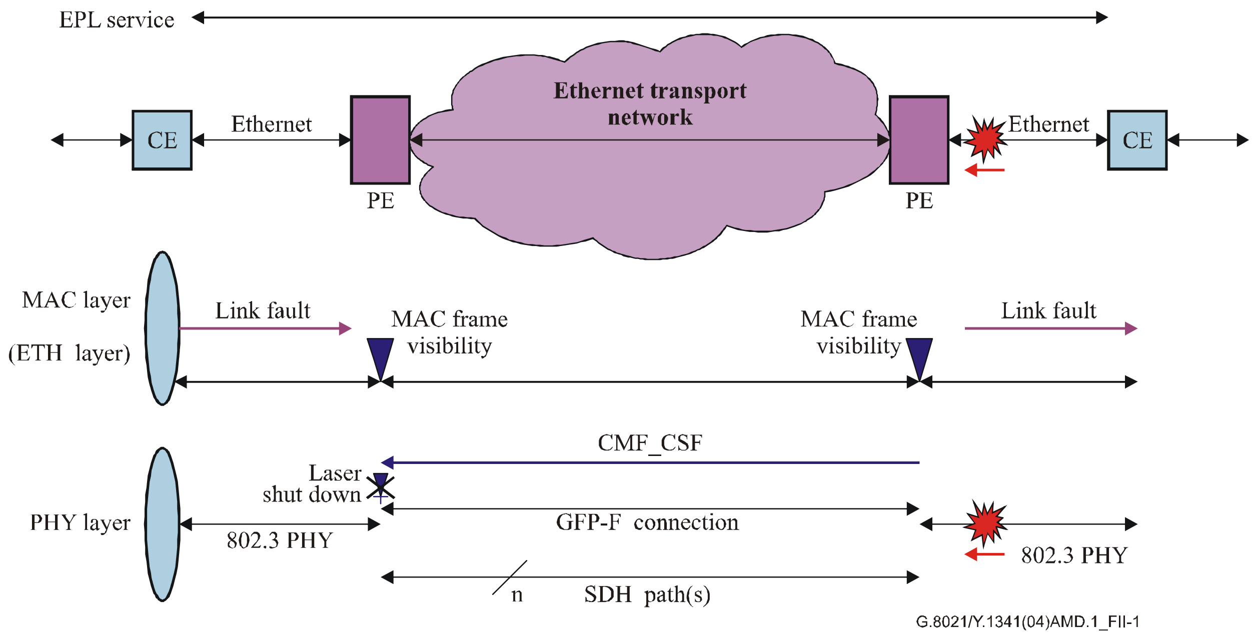


Figure II.1 – Fault on ingress

- The east PE detects loss of signal on the ingress access link:
  - 802.3ah OAM sends link fault upstream, interspersed with idles.
  - GFP-F CMF CSF indication is sent into the network.
- The east CE detects link fault:
  - Idles are sent towards the network and towards the enterprise.
- The west PE detects the GFP-F CMF CSF indication:
  - If there is no network_ETH_AIS indication available, the laser (or electrical driver) is shut down.
- The west CE detects loss of signal:
  - 802.3ah OAM sends link fault upstream, interspersed with idles.
  - Idles are sent towards the enterprise.

Scenario 2

In Figure II.2 below a unidirectional fault occurs westbound on the server layer within the carrier network.

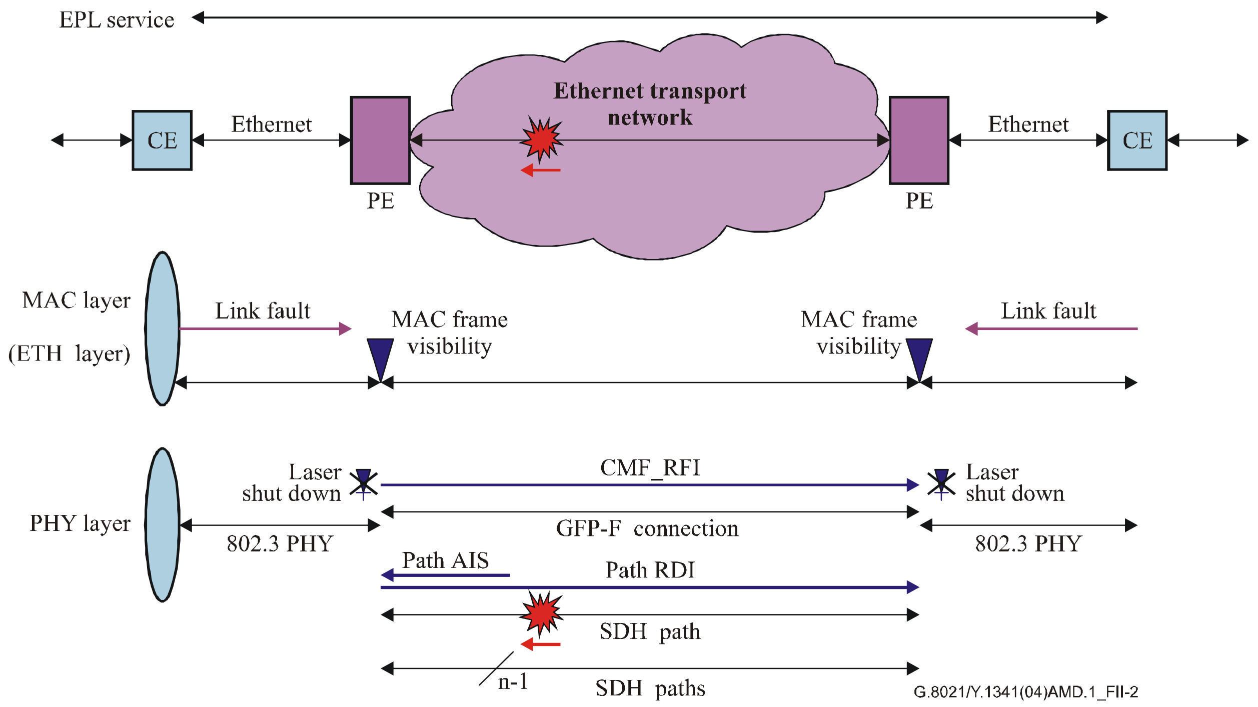


Figure II.2 – Fault within carrier network

- An NE in the carrier network detects the failure of one of the member paths of a VCAT group:
  – SDH Path AIS is generated downstream on the affected path.
- The west PE detects SDH path AIS:
  – SDH path RDI is generated back into the network on the associated path;
  – GFP-F CMF RFI is generated back into the network;
  – If there is no network_ETH_AIS indication available, the laser (or electrical driver) is shut down.
- The west CE detects loss of signal:
  – 802.3ah OAM sends link fault upstream, interspersed with idles;
  – Idles are sent towards the enterprise.
- The east PE detects the GFP-F CMF RFI indication:
  – If there is no network_ETH_RDI indication available, the laser (or electrical driver) is shut down.
- The east CE detects loss of signal:
  – 802.3ah OAM sends link fault upstream, interspersed with idles;
  – Idles are sent towards the enterprise.

Note that for a network failure affecting all member paths of the VCAT group, the same steps above apply with the addition of SDH path AIS and RDI being sent on all the member paths.

Scenario 3

In Figure II.3 below a unidirectional fault occurs on the west access link towards the enterprise network.

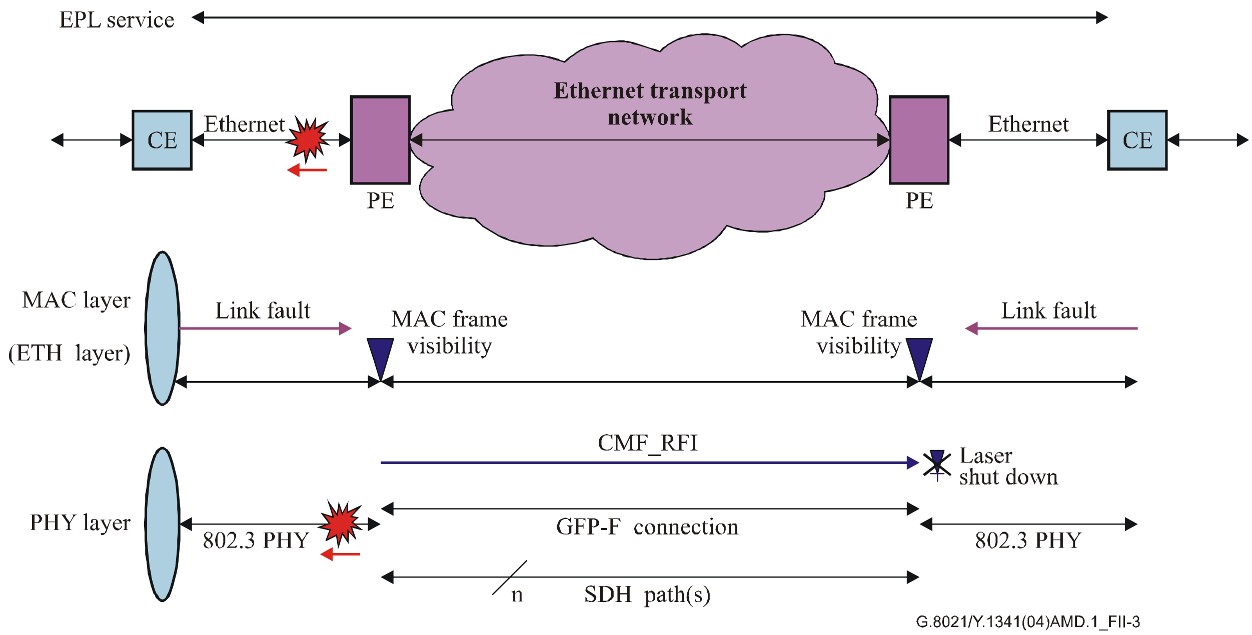


Figure II.3 – Fault on egress

- The west CE detects loss of signal:
  - 802.3ah OAM sends link fault upstream, interspersed with idles.
  - Idles are sent towards the enterprise.
- The west PE detects the link fault indication:
  - GFP-F CMF RFI indication is sent into the network.
  - Idles are sent towards the CE.
- The east PE detects the GFP-F CMF RFI indication:
  - If there is no network_ETH_RDI indication available, the laser (or electrical driver) is shut down.
- The east CE detects loss of signal:
  - 802.3ah OAM sends link fault upstream, interspersed with idles.
  - Idles are sent towards the enterprise.

Note that a PE only reacts to the reception of a link fault indication when there are no other conditioning alarms (i.e., the PE takes no further conditioning action when it receives a link fault indication in response to having shut down its own egress laser).

Bibliography

[b-802.3ak] IEEE 802.3ah-2004: *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 Amendment: Media Access Control Parameters, Physical Layers, and Management Parameters for Subscriber Access Networks.*

ITU-T Y-SERIES RECOMMENDATIONS

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

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

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

SERIES OF ITU-T RECOMMENDATIONS

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

Printed in Switzerland
Geneva, 2007