INTERNATIONAL TELECOMMUNICATION UNION

TELECOMMUNICATION STANDARDIZATION SECTOR

STUDY PERIOD 2009-2012

COM 15 – LS 145 – E

English only

Original: English

Question(s): 9/15

LIAISON STATEMENT

Source: ITU-T Study Group 15

Title: Ethernet equipment functional blocks (G.8021/Y.1341)

LIAISON STATEMENT

For action to:

For comment to: IEEE 802.1

For information to:

Approval: Agreed to at Question 9/15 meeting (Darmstadt, 1-5 March 2010)

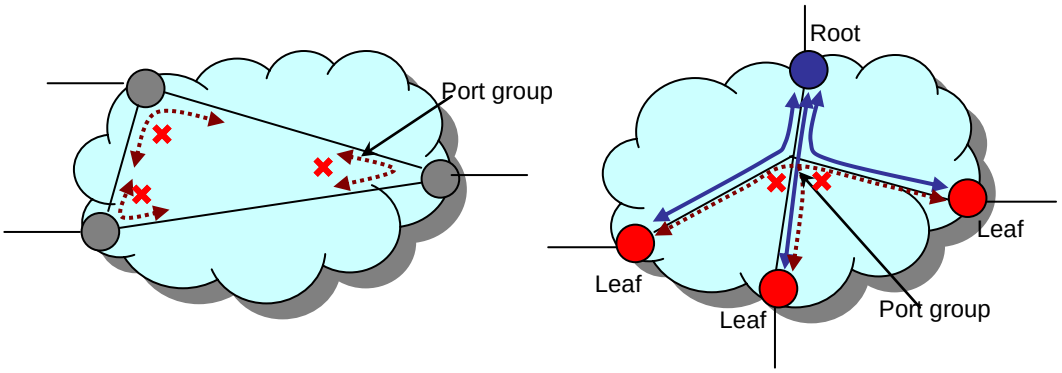
Deadline: 31 May 2010

Contact: Ghani Abbas
Ericsson
UK
Tel: +44 7710 370 367
Email: Ghani.Abbas@ericsson.com

.

ITU-T SG15 Q9 is pleased to inform you that the Amendment 2 to Recommendation G.8021/Y.1341 "Characteristics of Ethernet transport network equipment functional blocks" has been approved in February 2010.

In this amendment, an enhancement to ETH Flow Forwarding process (VLAN instance within MAC relay) is included to support the modelling of VPLS split horizon behaviour in our specification. This enhancement has introduced the concept of a "Port Group". A "Port Group" includes a number of ports on an ETH Flow Forwarding process for which the "split-horizon" behaviour applies. Frames arriving via an input port in such Port Group may be forwarded to one or more output ports on the Flow Forwarding process, with the exception of the output ports that are member of the Port Group. Frames arriving on an input port on the Flow Forwarding process that is not a member of the Port Group may be forwarded to one or more output ports on the Flow Forwarding process with exception of the port over which the frame arrived. This will allow configuring a rooted-multipoint EVC as well as a Split horizon.



(1) Split horizon (2) Rooted multipoint EVC

ITU-T SG15 Q9 would like to express its appreciation for the continued useful exchange of information with IEEE 802.1 on Ethernet technology standards. We would appreciate your comment on our Recommendation and have attached it for your information. We look forward to continuing this coordination in the future.

Attachment: 1

ITU-T Recommendation G.8021/Y.1341

Characteristics of Ethernet transport network equipment functional blocks

Draft Amendment 2

Summary

This second amendment contains additional material to be incorporated into ITU-T Recommendation G.8021/Y.1341. It presents enhancements about ETH Connection function. It also presents enhancements about ODU2P to Ethernet PP-OS adaptation function and a reference to the ODU0P to 1 GbE client adaptation function.

Amendments

1. **Add a reference to the SDL recommendation**

– ITU-T Recommendation Z.100 (11/2007), *Specification and Description Language (SDL)*.

2. **Add new Appendix V**

APPENDIX V – SDL descriptions

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

In this recommendation, detail characteristics of equipment functional blocks are described with SDL diagrams specified in ITU-T Rec. Z.100. The SDL diagrams use the following conventions.

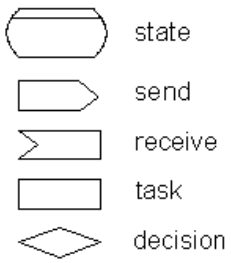


Figure V-1/G.8021/Y.13041 – SDL symbols

3. **Update figure 8-76**

Add and change the text in RED:

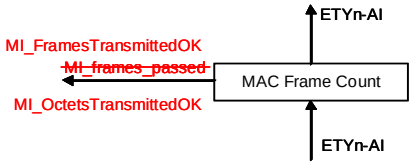


Figure 8-76/G.8021/Y.1431 – MAC Frame Count function

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

MI_pOctetsTransmittedOK[1..Np] per clause 30/IEEE802.3

MI_pFramesTransmittedOK[1..Np] per clause 30/IEEE802.3

4. **Update figure 9-3**

Added flags and Offset fields in RED:

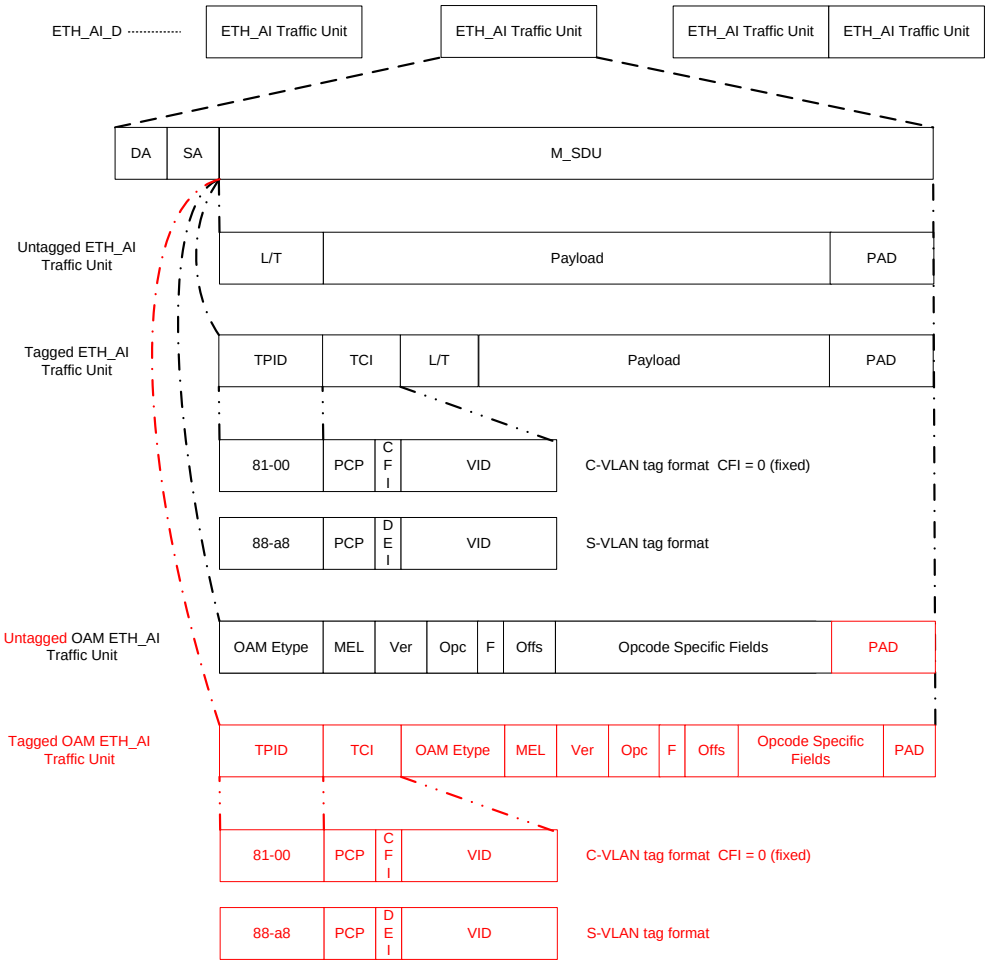


Figure 9-3/G.8021/Y.1341 – ETH Adapted Information

5. Update figure 9-4

Added flags and Offset fields in RED:

Figure 9-4/G.8021/Y.1341 – Tagged ETH_AI carrying ETH_CI OAM

6. **Reorder clause 9**

- Current text in 9.1.2 will move to 9.1 (after the title) as this is the description of the ETH connection function
- Current sub-clause 9.1.1 will remain
- Remove sub-clause 9.1.2 (moved to 9.1)
- Current sub-clause 9.1.3 will become 9.1.2
- Remove Sub-clause 9.1.4: the description of split horizon is covered by using port grouping

7. **Replace sub-clause 9.1**

Replace:

9.1 Connection Functions

By:

9.1 ETH Connection Functions (ETH_C)

The information flow and processing of the ETH_C function is defined with reference to Figures 9-8 and 9-10. The ETH_C function connects ETH characteristic information from its input ports to its output ports. As the process does not affect the nature of characteristic information, the reference points on either side of the ETH_C function are the same as illustrated in Figure 9-8.

The connection process is unidirectional and as such no differentiation in sink and source is required.

In addition, the ETH_C function supports the following sub-network connection protection schemes:

– 1+1 unidirectional SNC/S protection without APS protocol.
– 1+1 unidirectional SNC/S protection with an APS protocol.
– 1+1 bidirectional SNC/S protection with an APS protocol.
– 1:1 bidirectional SNC/S protection with an APS protocol.
– Ring protection with an APS protocol.

The protection functionality is described in sub-clause 9.1.3 and 9.1.4.

NOTE 1 – The SNC/S protection processes have a dedicated sink and source behaviour.

Symbol

The ETH Connection Function as shown in Figure 9-8 forward ETH_CI signals at its input ports to its output ports.

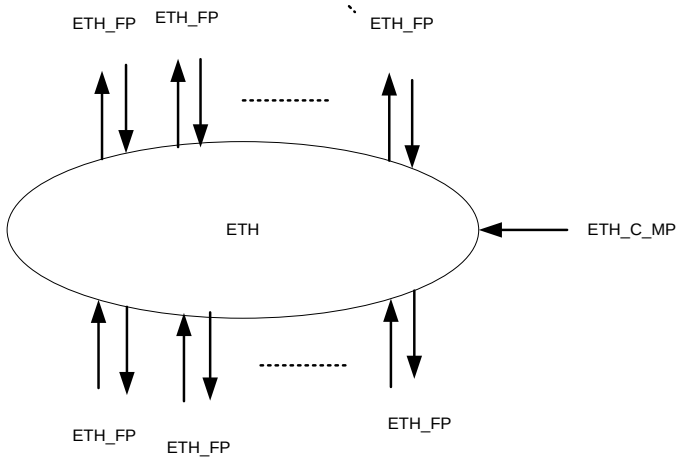


Figure 9-8/G.8021/Y.1341 – ETH Connection Function

The actual forwarding is performed using Flow Forwarding processes ETH_FF interconnecting the input and output ports.

Interfaces

Table 9-2 ETH_C Interfaces

Inputs	**Outputs**
Per ETH_FP: ETH_CI_D ETH_CI_P ETH_CI_DE ETH_CI_APS ETH_CI_SSF ETH_CI_SSD ETH_C_MP: ETH_C_MI_Create_FF ETH_C_MI_Modify_FF ETH_C_MI_Delete_FF ETH_C_MP per flow forwarding process: ETH_C_MI_FF_Set_PortIds ETH_C_MI_FF_ConnectionType ETH_C_MI_FF_Flush_Learned ETH_C_MI_FF_Flush_Config ETH_C_MI_FF_Group_Default ETH_C_MI_FF_ETH_FF ETH_C_MI_FF_Ageing ETH_C_MI_FF_Learning ETH_C_MI_FF_STP_Learning_State[i]	Per ETH_FP: ETH_CI_D ETH_CI_P ETH_CI_DE

ETH_C_MI_FF_Flow_Port_Group[j] ETH_C_MP per SNC/S protection process: ETH_C_MI_PS_WorkingPortId ETH_C_MI_PS_ProtectionPortId ETH_C_MI_PS_ProtType ETH_C_MI_PS_OperType ETH_C_MI_PS_HoTime ETH_C_MI_PS_WTR ETH_C_MI_PS_ExtCMD ETH_C_MP per Ring protection process: ETH_C_MI_RAPS_RPL_Owner_Node ETH_C_MI_RAPS_RPL_Neighbour_Node ETH_C_MI_RAPS_Propagate_TC[1…M] ETH_C_MI_RAPS_Compatible_Version ETH_C_MI_RAPS_Revertive ETH_C_MI_RAPS_Sub_Ring_Without_ Virtual_Channel	

Processes

The processes associated with the ETH_C function are as depicted in Figure 9-10.

ETH_CI traffic units are forwarded between input and output ETH flow points by means of an ETH flow forwarding process. ETH flow points may be allocated within a protection group. ETH flow points may be allocated within a flow port group (to support split-horizon).

NOTE 2 – Neither the number of input / output signals to the connection function, nor the connectivity, is specified in this Recommendation. That is a property of individual network elements.

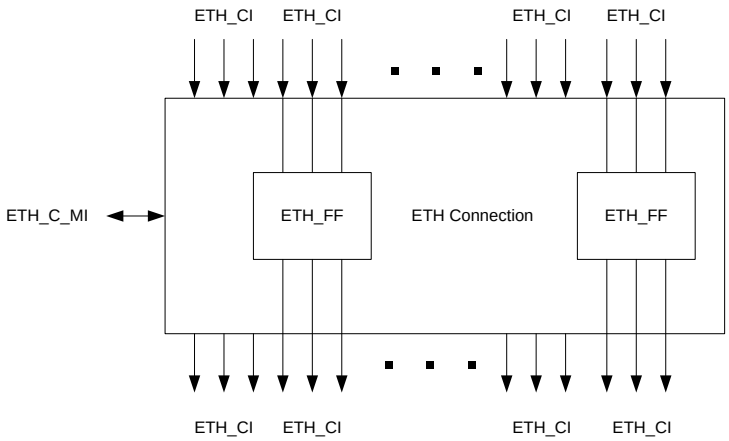


Figure 9-10/G.8021/Y.1341 – ETH Connection function with ETH_FF processes

The flow forwarding process ETH_FF is described in sub-clause 9.1.1.

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

8. Replace sub-clause 9.1.1

Replace the existing sub-clause 9.1.1 by:

9.1.1 ETH Flow Forwarding process (ETH_FF)

The ETH Flow Forwarding process, as shown in Figure 9-10, forwards ETH_CI signals at its input ports to its output ports. The forwarding may take into account the value of the DA field of the ETH_CI Traffic Unit and the flow port group of which the input flow point is a member.

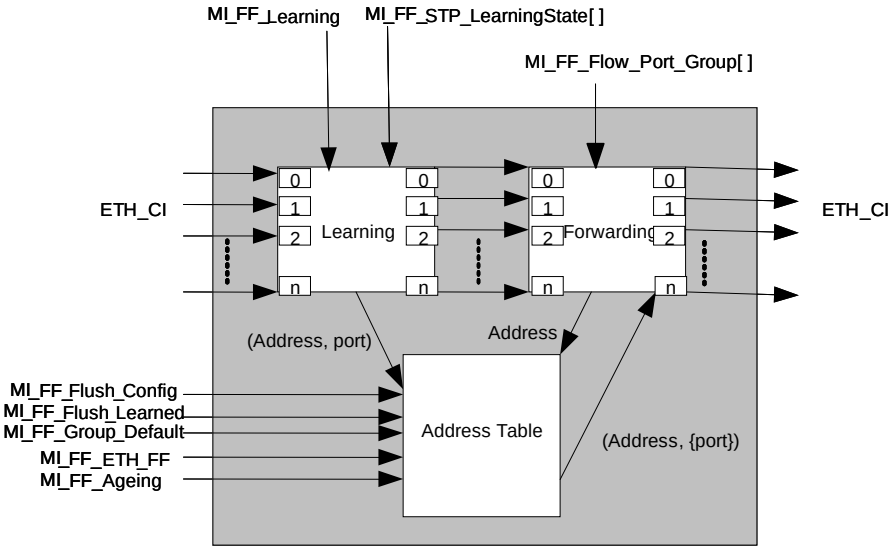


Figure 9-6/G.8021/Y.1341 – ETH Flow Forwarding Process

Figure 9-6 shows the ETH_FF in case of Individual VLAN Learning (IVL) mode. In this mode each ETH_FF has its own Address Table. Figure 9-7 shows the process for the case of Shared VLAN Learning (SVL) mode. In this mode two or more ETH_FF share the Address Table process.

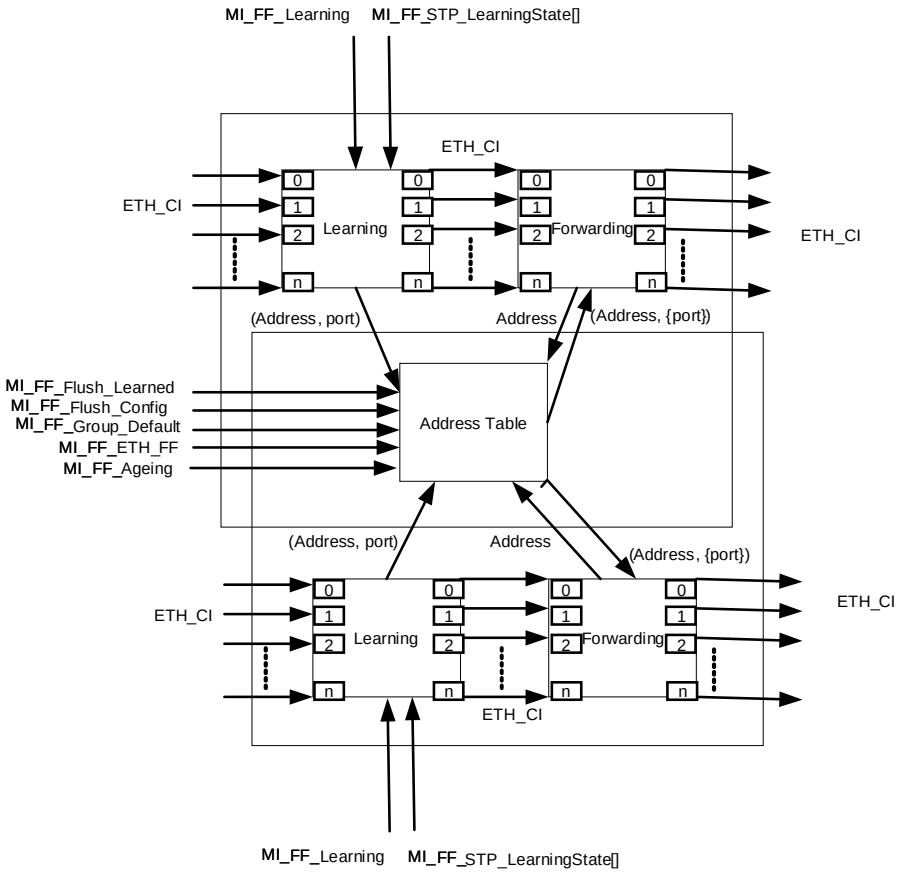


Figure 9-7/G.8021/Y.1341 – ETH Flow Forwarding Process in SVL mode

Address Table process
The AddressTable process maintains a list of tuples (Address, {ports}). This list may be configured using MI_FF_ETH_FF input signal and by the Learning process.

A tuple received from the Learning process is only stored in the Address Table process if there is no entry present for that MAC address that has been configured by the MI_FF_ETH_FF input signal.

The MI_FF_Ageing is used to provision the Ageing time period for entries configured from the Learning Process. Entries received from the Learning process are removed from the Address Table Ageing Time period after it was received. If, before the Ageing Time Period is expired, a new entry for the same MAC address is received, the Ageing Time Period starts again.

There is one specific value of MI_FF_Ageing: “never”. This means that the entries received from the Learning process are never removed.

All the tuples received from the Learning Process can be cleared using the MI_FF_Flush_Learned command.

All the tuples that are entered via the MI_FF_ETH_FF can be cleared using the MI_FF_Flush_Config command. Individual entries are removed via the MI_FF_ETH_FF signal.

The AddressTable process processes Address requests from the Forwarding process, and responds with the tuple (Address, {port}) for the specified address. For unicast MAC addresses, if the tuple does not exist the port set ({port}) is empty. For Multicast MAC addresses, if the tuple does not exist the port set ({port}) contains the ports as configured using the MI_FF_Group_Default input signal.

Learning process
If the value of MI_FF_Learning is enabled, 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. Address contains the value of the SA field of the ETH_CI Traffic Unit, and port is the port on which the Traffic Unit was received.

If the value of MI_FF_Learning is disabled, the Learning process does not submit information to the AddressTable Process.

In both cases the ETH_CI itself is forwarded unchanged to the output of the learning process.

Forwarding process

The parameters of MI_Create_FF, MI_Modify_FF, and MI_Delete_FF are used to provision the flow forwarding process. The MI_FF_Flow_Port_Group[j] parameter is used to provision the flow port group information corresponding to each port. The parameter is (port number, group).

NOTE – ETH_C functions designed according to G.8021-2007, G.8021 Amd.1-2009, or earlier versions of this recommendation may not support the flow port group.

The MI_FF_Set_PortIds parameter is used to provision TBD.

The MI_FF_ConnectionType parameter is used to provision TBD.

The MI_FF_STP_LearningState[i] input signal is provisioned per port [i]; it can be used to configure a specific port to be in the learning state. If a port is in the learning state this means that all frames received on that port will be discarded by the learning process, and therefore not forwarded to the forwarding process; however the (Address, port) tuple may be submitted to the Address Table process before the frame is dropped (depending on the value of MI_FF_Learning).

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). In all cases the ETH_CI is never forwarded on the same port and flow port group as it was received on.

9. Remove sub-clause 9.1.2

The text of sub-clause 9.1.2 has been moved to sub-clause 9.1

10. Renumber and update sub-clause 9.1.3

Replace:

9.1.3 Subnetwork Connection Protection Process

By:

9.1.2 Subnetwork Connection Protection Process

Add and change the text in RED:

Figure 9-11 shows the involved atomic functions in SNC/S. The ETH_FT_Sk provides the TSF/TSD protection switching criterion via the ETH/ETH_A_Sk function (SSF/SSD) to the ETH_C function.

Replace figure 9-11:

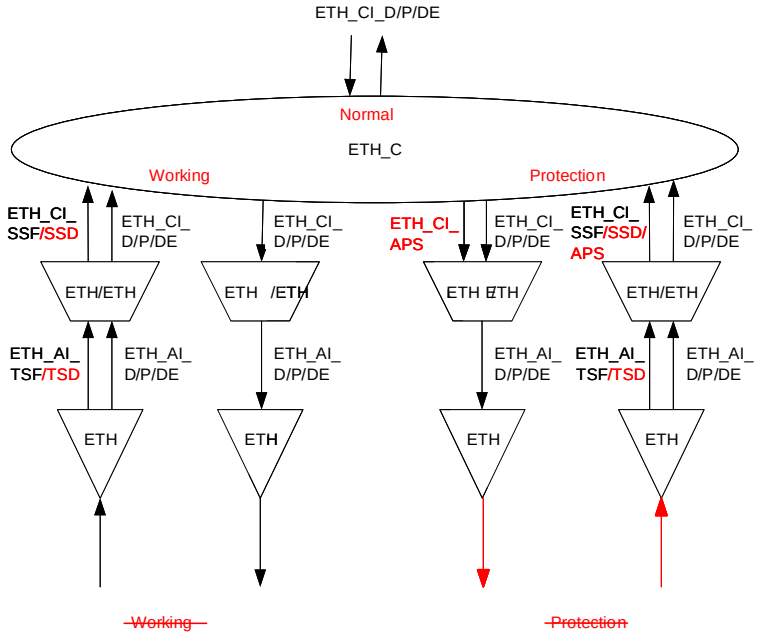


Figure 9-11/G.8021/Y.1341 – SNC/S Atomic Functions

Change the text in RED:

Configuration

The following configuration parameters are defined in G.8031/Y1342:

~~ETH_MI_PS_Enable enables the protection switching.~~

ETH_C_MI_PS_~~set~~WorkingPortId configures the Working Port.

ETH_C_MI_PS_~~set~~ProtectionPortId configures the Protection Port.

ETH_C_MI_PS_ProtType configures the protection Type.

ETH_C_MI_PS_OperType configures to be in revertive mode.

ETH_C_MI_PS_HoTime configures the Hold Off Timer.

ETH_C_MI_PS_WTR configures the Wait To Restore Timer.

ETH_C_MI_PS_ExtCMD~~cmd~~ configures the protection group command.

~~ETH_MI_PS_StateChange reports a state change.~~

11. Update figure 9-12

Add and change the text in RED:

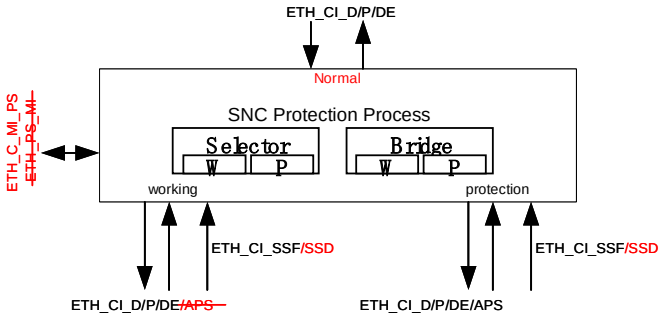


Figure 9-12/G.8021/Y.1341 – SNC/S Protection Process

12. Remove sub-clause 9.1.4

Sub-clause 9.1.4 is removed. The description of Split Horizon is included in the new sub-clause 9.1.1.

13. Update table 9-4

Change the text in RED:

Table 9-4/G.8021/Y.1341 – ETHx_FT_Sk interfaces

Inputs	Outputs
ETHx_FP: ETHx_CI_D ETHx_CI_P ETHx_CI_DE ETHx_MP: ETHx_FT_Sk_MI_LM_Enabled ETHx_FT_Sk_MI_1Second ETHx_FT_Sk_MI_LM_DEGM ETHx_FT_Sk_MI_LM_M ETHx_FT_Sk_MI_LM_DEGTHR ETHx_FT_Sk_MI_CC_Period ETHx_FT_Sk_MI_CC_Pri ETHx_FT_Sk_MI_MEG_ID ETHx_FT_Sk_MI_PeerMEP_ID[i] ETHx_FT_Sk_MI_GetSvdCCM ETHx_FT_Sk_MI_MEL ETHx_FT_Sk_MI_CC_Enable	ETHx_AP: ETHx_AI_D ETHx_AI_P ETHx_AI_DE ETHx_AI_TSF ETHx_AI_AIS ETHx_AI_TSD ETHx_RP: ETHx_FT_Sk_RI_CC_RxFC~~f~~ ETHx_FT_Sk_RI_CC_TxFC~~f~~ ETHx_FT_Sk_RI_CC_RDI ETHx_FT_Sk_RI_CC_Blk ETHx_MP: ETHx_FT_Sk_MI_cSSF ETHx_FT_Sk_MI_cLCK ETHx_FT_Sk_MI_cLOC[i] ETHx_FT_Sk_MI_cMMG ETHx_FT_Sk_MI_cUNM ETHx_FT_Sk_MI_cUNP ETHx_FT_Sk_MI_cUNPr ETHx_FT_Sk_MI_cUNL ETHx_FT_Sk_MI_cRDI[i] ETHx_FT_Sk_MI_SvdCCM ETHx_FT_Sk_MI_pN_FL ETHx_FT_Sk_MI_pN_TF ETHx_FT_Sk_MI_pF_FL ETHx_FT_Sk_MI_pF_TF ETHx_FT_Sk_MI_pF_DS ETHx_FT_Sk_MI_pN_DS

14. Update table 9-6 and Figure 9-21

Add the text in RED:

Table 9-6/G.8021/Y.1341 – 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_TSD ETHx/ETH_A_Sk_MP: ETHx/ETH_A_Sk_MI_AdminState 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 ETH_CI_SSD

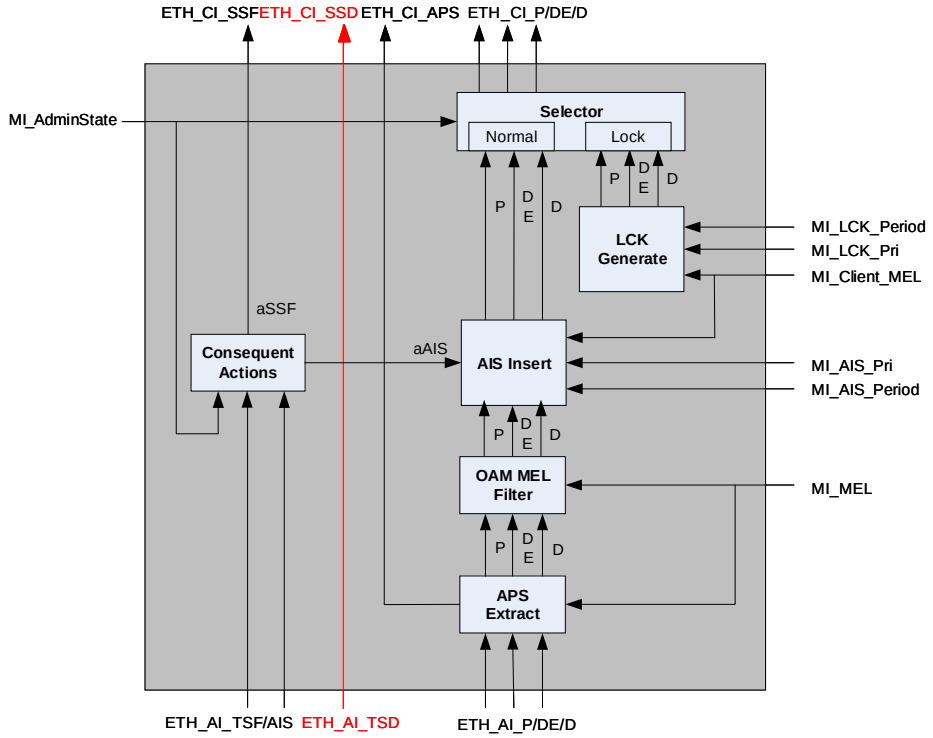


Figure 9-21/G.8021/Y.1341 – Sink direction ETHx/ETH_A_Sk process

15. Update in sub-clause 9.2.1.2

Add or change the text in RED:

Defect Generation Process

This process detects and cleares the defects (dLOC[i], dRDI[i], dLCK, dAIS, dUNL, dMMG, dUNM, dUNP, dUNPr, dDEG) as defined in Clause 6, where [i] = maintenance entity.

Defects

This function detects dLOC[i], dRDI[i], dLCK, dAIS, dUNL, dMMG, dUNM, dUNP, dUNPr, dDEG.

Defect correlations

cRDI[i] ← (dRDI[1..n]) and (MI_CC_Enable)

Performance monitoring

pN_ LF~~FLC~~ ← N_LF

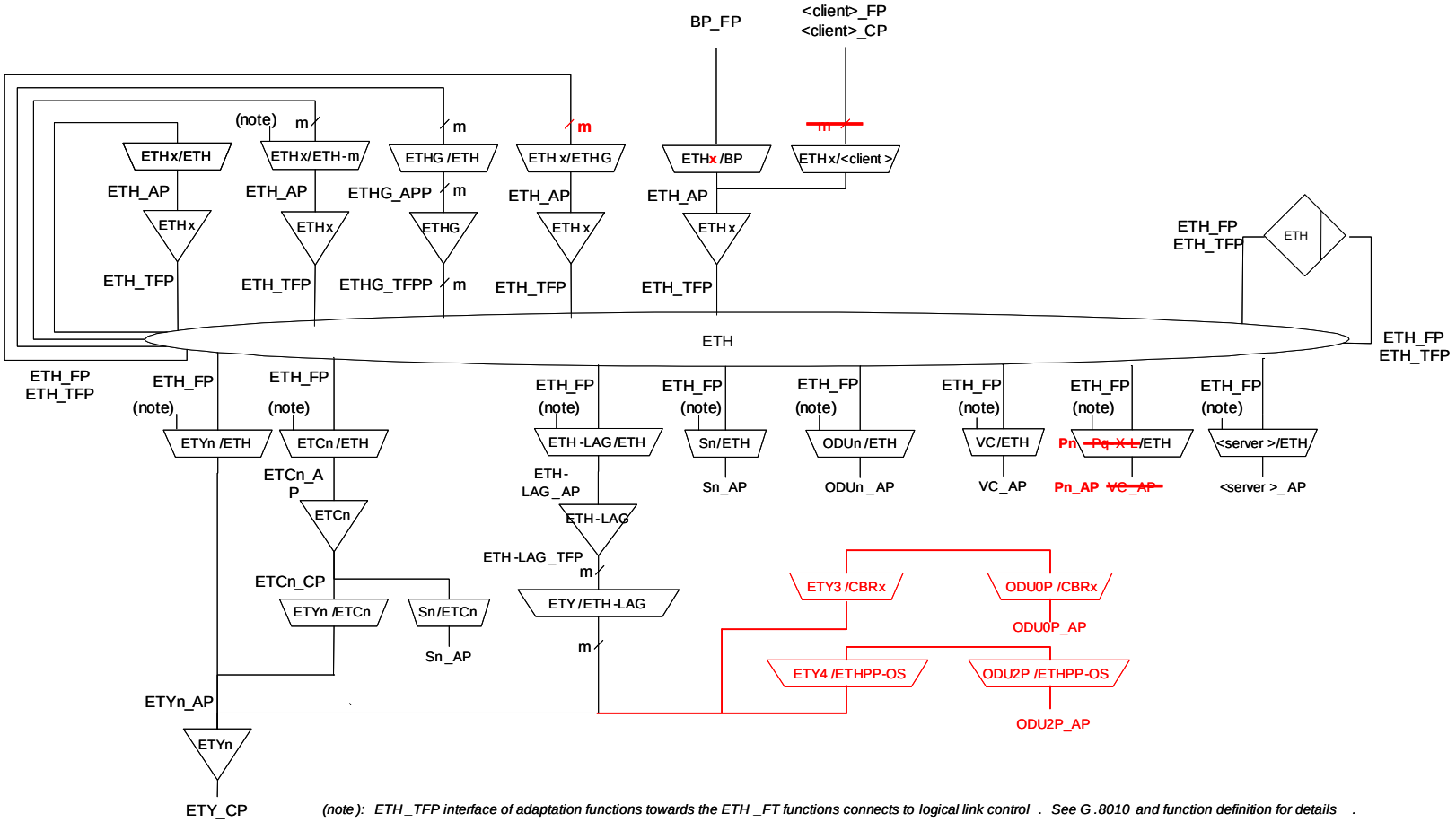
pN_TF~~C~~ ← N_TF

pF_ LF~~FLC~~ ← F_LF

pF_ TF~~NTC~~ ← F_TF

16. Update figure 1

Add or change the text/figure in RED:



(note): ETH_TFP interface of adaptation functions towards the ETH_FT functions connects to logical link control . See G.8010 and function definition for details .

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

17. Update table 9-10

Add [1…M] to ETHG/ETH_A_Sk_MI_Client_MEL.

Table 9-10/G.8021/Y.1341 – 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 ETHx/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_MEL[1…M] ETHG/ETH_A_Sk_MI_MEL[1..M]	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]

18. Update figure 9-x+2 (in Amd1)

Replace So by Sk.

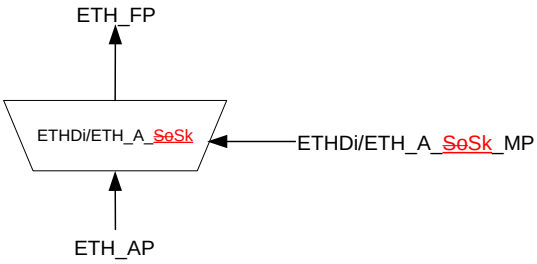


Figure 9-x+2 – ETHDi/ETH_A_Sk function

19. Update table 9-y+1 (in Amd1)

Change ETHDi/ETH_A_Sk_MI_MEL into ETHDi/ETH_A_Sk_MI_RAPS_MEL.

Table 9-y+1 – ETHDi/ETH_A_Sk Interfaces

Inputs	Outputs
ETH_AP: ETH_AI_D ETH_AI_P ETH_AI_DE ETH_AI_TSF ETHDi/ETH_A_Sk_MP: ETHDi/ETH_A_Sk_MI_RAPS_MEL	ETH_FP: ETH_CI_D ETH_CI_P ETH_CI_DE ETH_CI_RAPS ETH_CI_SSF

20. Insert the following acronym into clause 4

PP-OS Preamble, Payload, and Ordered Set information

21. Add new sub-clause 11.5.3

11.5.3 ODU2P to Ethernet PP-OS adaptation function (ODU2P/ETHPP-OS_A)

The ODU2P to Ethernet PP-OS adaptation function supports transporting Preamble and Ordered Set information of the ~~10GBE R~~10GBASE-R signals over enhanced OPU2 payload area.

It provides XGMII service over ODU2 with extended OPU2 payload area.

As shown in IEEE 802.3 Figure 46-3, the Ethernet data stream at the XGMII consists of: <inter-frame><preamble><sfd><data><efd>. For the purposes of these mappings, the client data frames include the <preamble><sfd><data> information, and the Ordered Sets include specific information carried in the <inter-frame> characters. The mapping of both client data frames and Ordered Sets into ODU2 using GFP-F frames is described in this clause.
Note that there is no Ethernet MAC termination function. Consequently, since no error checking is performed on the Ethernet MAC frames, errored MAC frames are forwarded at both the ingress and egress to the GFP adaptation functions.

11.5.3.1 ODU2P/ETHPP-OS adaptation source function (ODU2P/ETHPP-OS_A_So)

The ODU2P/ETHPP-OS_A_So function creates the ODU2P signal from a free running clock. It maps the ETHPP-OS_CI information into the payload of the OPU2P, adds OPU2P Overhead (RES, PT) and default ODU2P Overhead.

Symbol

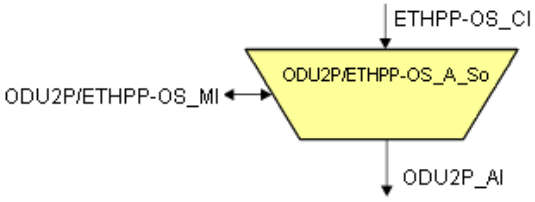


Figure 11-x/G.8021/Y.1341 – ODU2P/ETHPP-OS_A_So symbol

Interfaces *For further study.*

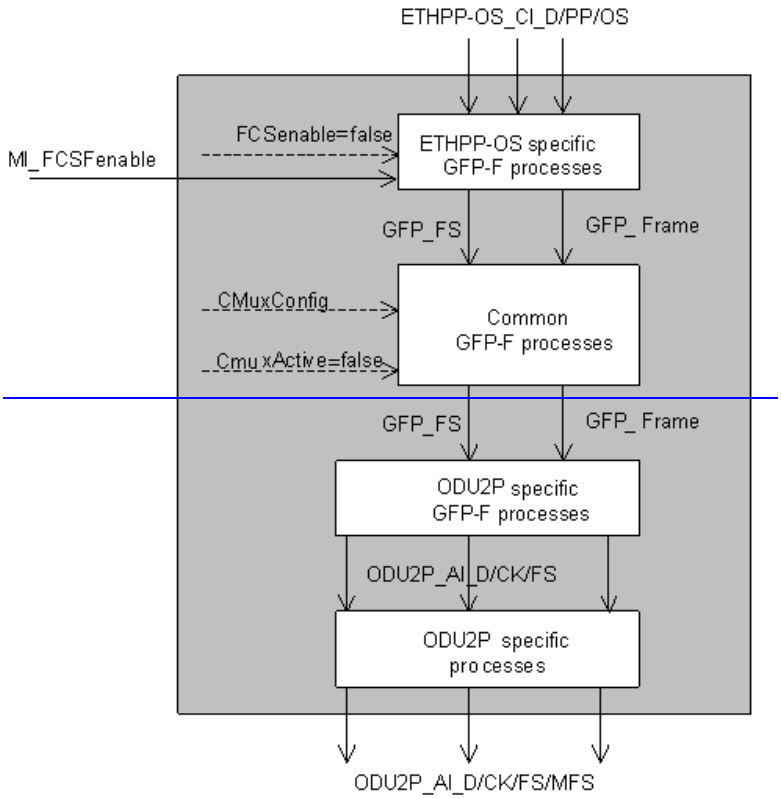
Formatted: Font: Not Bold

~~**Table 11-x/G.8021/Y.1341 – ODU2P/ETHPP-OS_A_So interfaces**~~

~~**Inputs**~~	~~**Outputs**~~
~~ETHPP-OS_TCP:~~ ~~– ETHPP-OS_CI_D~~ ~~– ETHPP-OS_CI_PP~~ ~~– ETHPP-OS_CI_OS~~ ~~ODU2P/ETHPP-OS_A_So_MP:~~ ~~– ODUkP/ETHPP-OS_A_So_MI_CSFEnable~~	~~ODU2P_AP:~~ ~~– ODU2P_AI_Data~~ ~~– ODU2P_AI_Clock~~ ~~– ODU2P_AI_FrameStart~~ ~~– ODU2P_AI_MultiframeStart~~

Processes *For further study.*

~~**A process diagram of this function is shown in Figure 11-x+1.**~~



~~**Figure 11-x+1/G.8021/Y.1341 – ODU2P/ETHPP-OS_A_So process diagram**~~

~~**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 7.9/G.7041/Y.1303.**~~

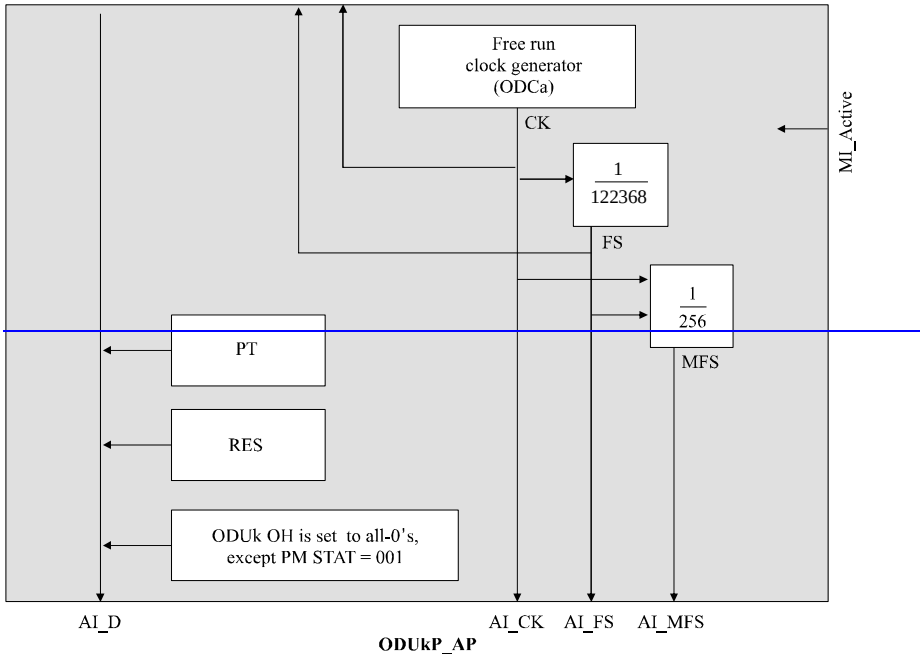
~~**Common GFP source process:**~~

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

~~**ODU2P specific GFP source process:**~~

~~**See 8.5.2.1/G.806. The GFP frames are mapped into the ODU2 payload area according to 17.3.1/G.709/Y.1331.**~~

~~**ODU2P specific source process:**~~



~~**Figure 11-x+2/G.8021/Y.1341 – ODU2P specific source processes**~~

~~**Clock and (Multi)Frame Start signal generation: The function shall generate a local ODU2 clock (ODUkP_AI_CK) of "239/(239 – k) * $4^{(k-1)}$ * 2 488 320 kHz ± 20 ppm" from a free running oscillator. The jitter and wander requirements as defined in Annex A/G.8251 (ODCa clock) apply.**~~

~~**The function shall generate the (multi)frame start reference signals AI_FS and AI_MFS for the ODU2 signal. The AI_FS signal shall be active once per 122368 clock cycles. AI_MFS shall be active once every 256 frames.**~~

~~**PT: The payload type information is derived directly from the Adaptation function type. The value for "GFP mapping" shall be inserted into the PT byte position of the PSI overhead as defined in 15.9.2.1.1/G.709/Y.1331.**~~

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

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

Defects *For further study.*

~~*None.*~~

Consequent actions *For further study.*

~~aCSF_RDI ← CI_SSFrdi and CSFrdifdiEnable and CSFEnable~~

~~aCSF_FDI ← CI_SSFfdi and CSFrdifdiEnable and CSFEnable~~

~~aCSF_LOS ← CI_SSF and CSFEnable~~

Defect correlations *For further study.*

~~*None.*~~

Performance monitoring

For further study.

Formatted: Font: Not Bold, Italic

Formatted: Heading_b

11.5.3.2 ODU2P/ETHPP-OS adaptation sink function (ODU2P/ETHPP-OS_A_Sk)

The ODU2P/ETHPP-OS_A_Sk extracts ETHPP-OS_CI information from the ODU2P payload area, delivering ETHPP-OS_CI to ETHPP-OS_TCP. It extracts the OPU2P Overhead (PT and RES) and monitors the reception of the correct payload type.

Symbol

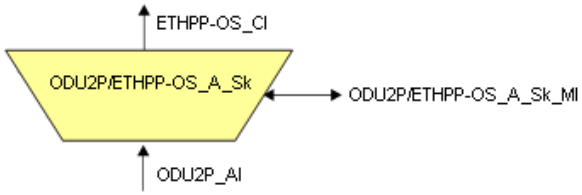


Figure 11-x+3/G.8021/Y.1341 – ODU2P/ETHPP-OS_A_Sk symbol

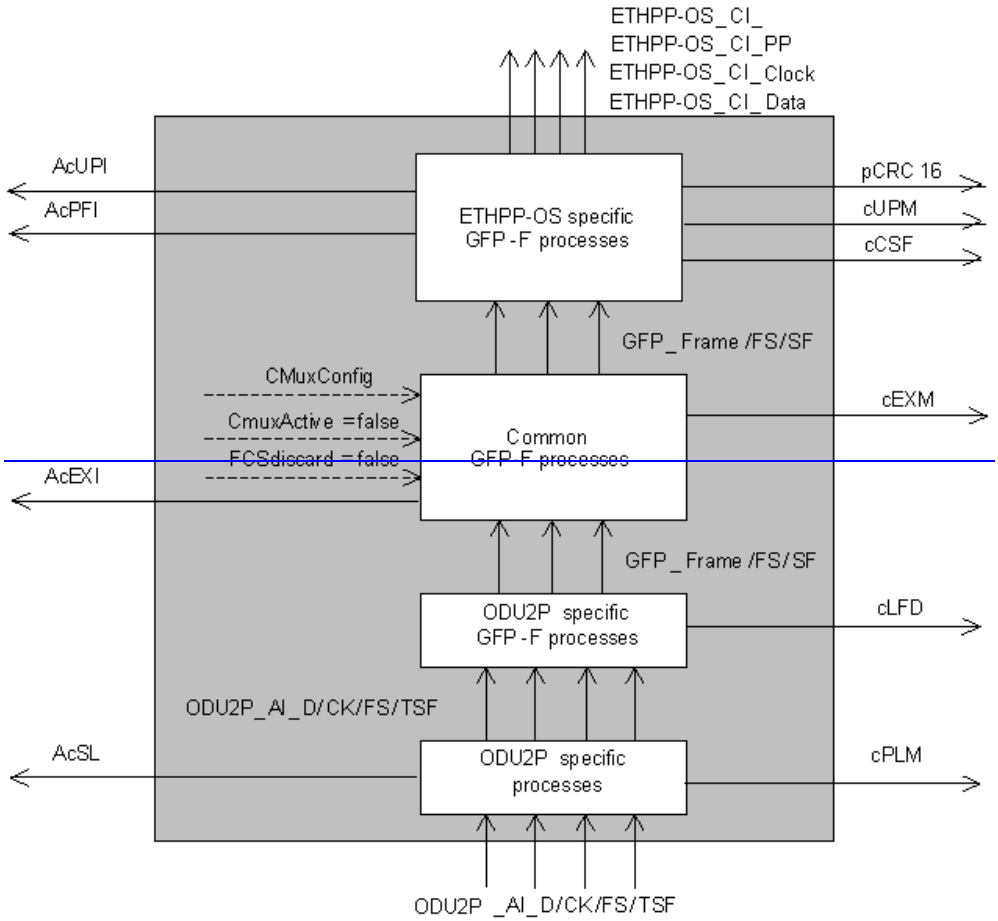
Interfaces *For further study.*

~~**Table 11-x+2/G.8021/Y.1341 – ODU2P/ETHPP-OS_A_Sk interfaces**~~

~~**Inputs**~~	~~**Outputs**~~
~~ODU2P_AP:~~ ~~- ODU2P_AI_Data~~ ~~- ODU2P_AI_ClockK~~ ~~- ODU2P_AI_FrameStart~~ ~~- ODU2P_AI_MultiframeStart~~ ~~- ODU2P_AI_TSF~~ ~~ODU2P/ETHPP-OS_A_Sk_MI:~~ ~~- ODUkP/ETHPP-OS_A_Sk_MI_Active~~ ~~- ODUkP/ETHPP-OS_A_Sk_MI_CSF_Reported~~ ~~- ODUkP/ETHPP-OS_A_Sk_MI_MAC_Length~~	~~ETHPP-OS_TCP:~~ ~~- ETHPP-OS_CI_D~~ ~~- ETHPP-OS_CI_PP~~ ~~- ETHPP-OS_CI_OS~~ ~~- ETHPP-OS_Clock~~ ~~ODU2P/ETHPP-OS_A_Sk_MI:~~ ~~- ODUkP/ETHPP-OS_A_Sk_MI_AcPT~~ ~~- ODUkP/ETHPP-OS_A_Sk_MI_AcEXI~~ ~~- ODUkP/ETHPP-OS_A_Sk_MI_AcUPI~~ ~~- ODUkP/ETHPP-OS_A_Sk_MI_cPLM~~ ~~- ODUkP/ETHPP-OS_A_Sk_MI_cLFD~~ ~~- ODUkP/ETHPP-OS_A_Sk_MI_cUPM~~ ~~- ODUkP/ETHPP-OS_A_Sk_MI_cEXM~~ ~~- ODUkP/ETHPP-OS_A_Sk_MI_cCSF~~ ~~- ODUkP/ETHPP-OS_A_Sk_MI_pFCSErrors~~ ~~- ODU2P /ETHPP-OS_A_Sk_MI_AcSL~~ ~~- ODU2P/ETHPP-OS_A_Sk_MI_AcPFI~~ ~~- ODU2P/ETHPP-OS_A_Sk_MI_pCRC16Errors~~

Processes *For further study.*

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



~~Figure 11-x+4/G.8021/Y.1341 – ODU2P/ETHPP-OS_A_Sk process diagram~~

~~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 (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 7.9/G.7041/Y.1303.~~

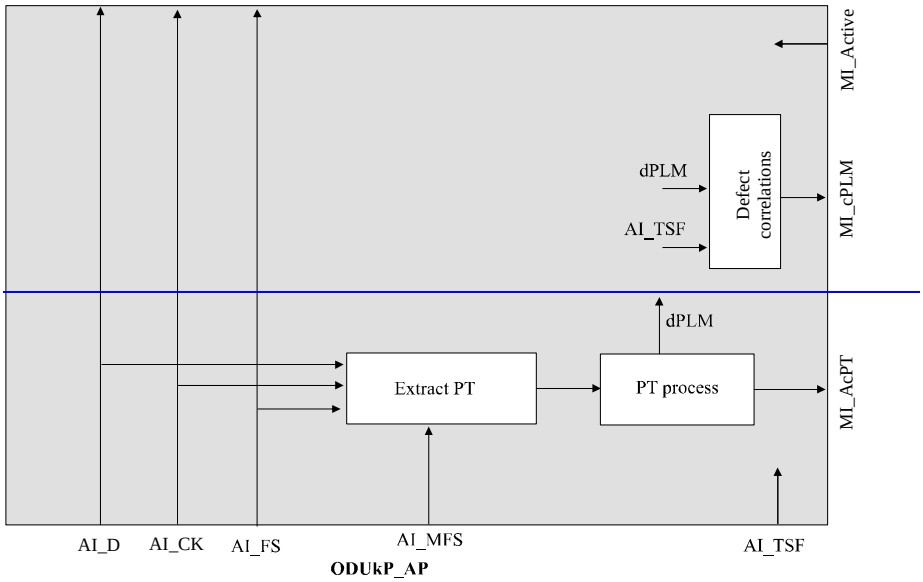
~~Common GFP sink process:~~

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

~~**ODU2 specific GFP sink process:**~~

~~See 8.5.2.2/G.806. The GFP frames are demapped from the ODU2payload area according to 17.3.1.709/Y.1331.~~

~~**ODU2P specific sink process:**~~



~~**Figure 11-x+5/G.8021/Y.1341 – ODU2P specific sink processes**~~

~~**PT**: The function shall extract the PT byte from the PSI overhead as defined in 8.7.1/G.798. The payload type value for "GFP mapping" in 15.9.2.1.1/G.709/Y.1331 shall be expected. The accepted PT value is available at the MP (MI_AcPT) and is used for PLM defect detection.~~

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

Defects *For further study.*

~~dPLM – See 6.2.4.1/G.798.~~

~~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-LOS – See 8.8.6.2~~

~~dCSF-RDI – See 8.8.6.2~~

~~dCSF-FDI – See 8.8.6.2~~

Consequent actions *For further study.*

~~The function shall perform the following consequent actions:~~

~~aSSF ← AI_TSF or dPLM or dLFD or dUPM or dEXM or dCSF_LOS~~

~~aSSFrdi ← dCSF_RDI and CSFrdifdiEnable~~

~~aSSFrdi ← dCSF_FDI and CSFrdifdiEnable~~

Defect correlations *For further study.*

~~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 dEXM) and (not dPLM) and (not dLFD) and (not AI_TSF);~~

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

~~cCSF ← (dCFS_LOS or dCSF_RDI or dCSF_FDI) and (not dEXM) and (not dUPM) and (not dPLM) and (not dLFD) and (not AI_TSF) and CSF_Reported~~

Performance monitoring

For further study.

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

~~pFCSErrors: count of FrameCheckSequenceErrors per second.~~

~~NOTE – This primitive is calculated by the MAC FCS Check process.~~

Formatted: Font: Not Bold, Italic

Formatted: Heading_b

22. Add new sub-clause 11.5.4

11.5.4 ETY4 to Ethernet PP-OS adaptation function (ETY4/ETHPP-OS_A)

For Further Study

23. Add new sub-clause 11.5.5

11.5.5. ODU0P to 1 GbE client adaptation function (ODU0P/CBRx_A)

The adaptation function that supports the transport of 1GbE signals in the OTN is the ODU0P to Client adaptation function (ODU0P/CBRx_A) (0≤x≤1.25G) described in G.798 ~~clause 14.3.7~~.

24. Add new sub-clause 11.5.6

11.5.6. ETY3 to 1 GbE client adaptation function (ETY3/CBRx _A)

For Further Study

25. Update figure 8-16

Change the text in RED:

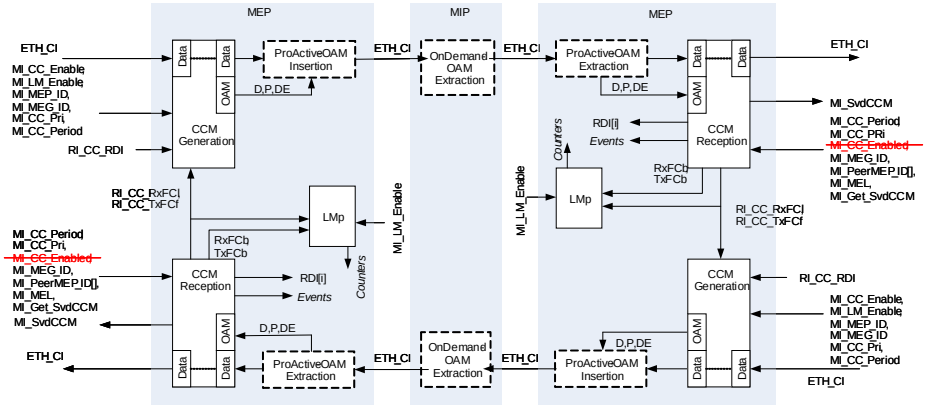


Figure 8-16/G.8021/Y.1341 – CC Overview

26. Update sub-clause 8.1.7.2

Add or change the text in RED:

Figure 8-17 shows the statediagram for the CCM Generation process. The CCM Generation Process can be enabled and disabled using the MI_CC~~M~~_Enable signal, where the default value is FALSE.

27. Update sub-clause 6.1.3.4

Add or change the text in RED:

6.1.3.4 Degraded Signal defect (dDEG)

This defect is only defined for point-to-point ~~ptp~~ ETH connections.

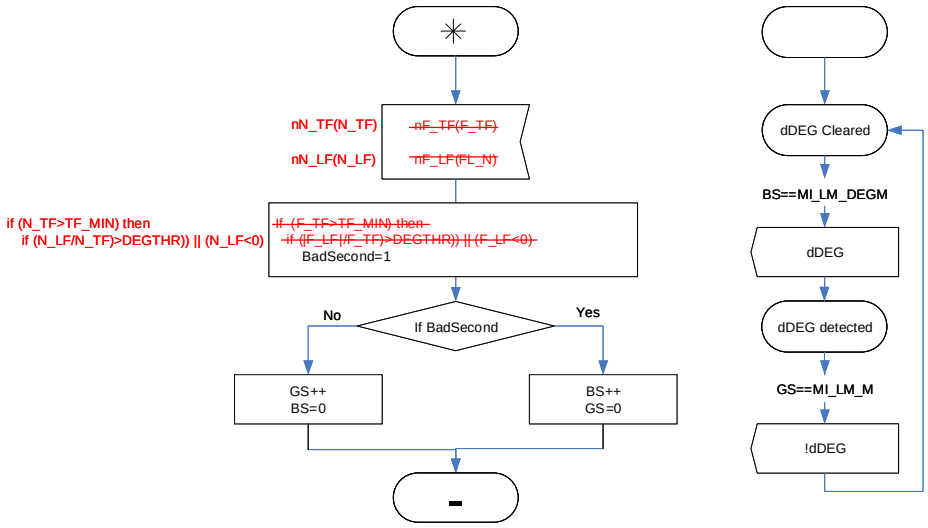


Figure 6-3/G.8021/Y.1341 – dDEG detection and clearance process

The Degraded Signal defect is calculated at the ETH layer. It monitors the connectivity of an ETH Trail.

Its detection and clearance are defined in Figure 6-3.

Every second the statemachine receives the 1 second counters for ~~far~~ near end received and transmitted frames and determines whether the second was a Bad Second. The defect is detected if there are MI_LM_DEGM consecutive Bad Seconds and cleared if there are MI_LM_M consecutive Good Seconds.

In order to declare a Bad Second the number of transmitted frames must exceed a threshold (TF_MIN). If this is true then a Bad Second is declared if either the Frame Loss is negative (i.e. there are more frames received than transmitted) or the Frame Loss Ratio (lost frames/transmitted frames) is greater than MI_LM_DEGTHR.

28. Insert the following acronym into clause 4

R-APS Ring Automatic Protection Switching

29. Update sub-clause 6.1.1

Add or change the text in RED:

The events defined for this recommendation are summarized in Table 6-1. ~~The events~~ Events, other than APS or R-APS events, are generated by processes in the ETH_FT_Sk function as defined in clause 9.2.1.2. APS events are generated by the subnetwork connection protection process as defined in clause 9.1.3. R-APS events are generated by the ring protection control process as defined in clause 9.1.4. These processes define the exact conditions for these events; Table 6-1 only provides a quick overview.

30. Add the following row to table 6-1

RAPSpm	Reception by the RPL Owner of an R-APS(NR, RB) frame with a Node ID that differs from its own.

31. In table 6-2, modify the "dFOP-PM" row as follows

dFOP-PM	APSb or RAPSpm	expAPS or #RAPSpm == 0 (K*long R-APS frame interval)

32. Update sub-clause 6.1.4.3.1

Add text in RED:

6.1.4.3.1 Linear or Ring protection Failure of Protocol Provisioning Mismatch (dFOP-PM)

The Failure of Protocol Provisioning Mismatch defect is calculated at the ETH layer. It monitors provisioning mismatch of:

- Linear protection by comparing B bits of the transmitted and the received APS protocol, or
- Ring protection by comparing the Node ID of the RPL Owner and the Node ID in a received R-APS(NR, RB) frame.

Its detection and clearance are defined in Table 6-2. dFOP-PM is detected:

- In the case of linear protection, on receipt of an APSb event and cleared on receipt of an expAPS event. These events are generated by the subnetwork connection protection process (Clause 9.1.3.), or
- In the case of ring protection, on receipt of an RAPSpm event and cleared on receipt of no RAPSpm event during K times the long R-APS frame intervals defined in G.8032/Y.1344, where K>=3.5. These events are generated by the ring protection control process (Clause 9.1.4).

33. Insert the following sub-clause after sub-clause 9.1.3

9.1.4 Ring Protection Control Process

Ring Protection with Inherent, Sub-Layer, or Test Trail monitoring is supported.

Figure 9-x shows a subset of the atomic functions involved, and the signal flows associated with the ring protection control process. This is only an overview of the Ethernet Ring Protection Control Process as specified in ITU-T Rec. G.8032/Y.1344. The ETH_FT_Sk provides the TSF protection switching criterion via the ETH/ETH_A_Sk function (SSF). G.8032 specifies the requirements, options and the ring protection protocol supported by the ring protection control process.

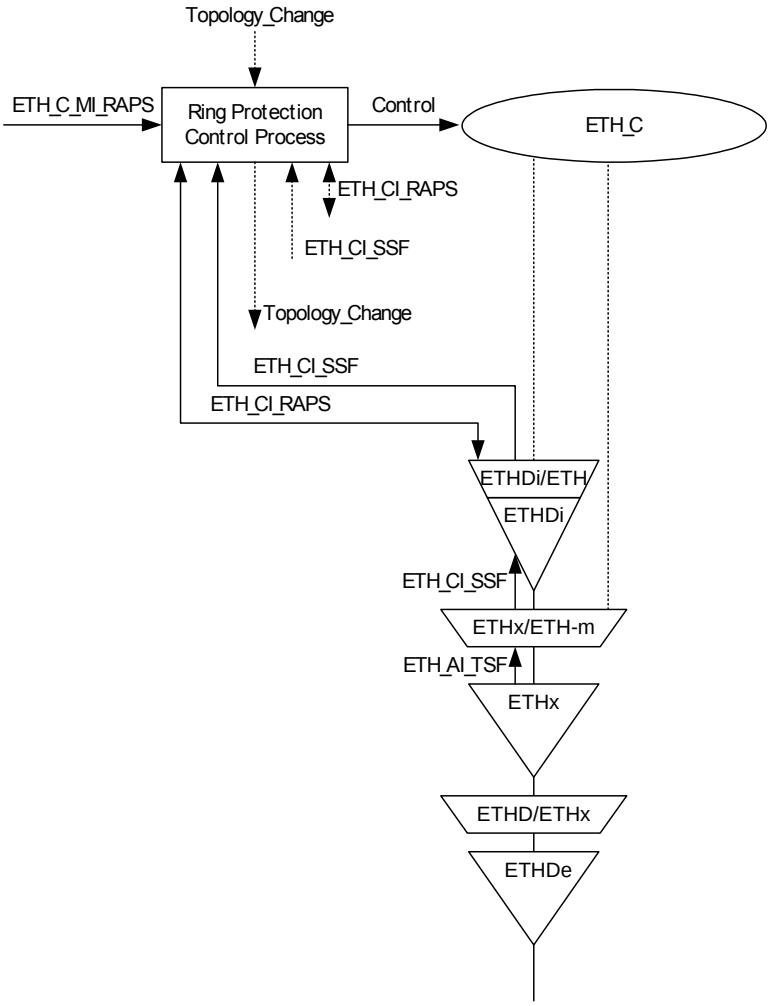


Figure 9-x/G.8021/Y.1341 – Ring Protection Atomic Functions and Control Process

Configuration

The following configuration parameters are defined in G.8032/Y.1344:

ETH_C_MI_RAPS_RPL_Owner_Node configures the node type.

ETH_C_MI_RAPS_RPL_Neighbour_Node configures the adjacency of a node to the RPL Owner.

ETH_C_MI_RAPS_Propagate_TC[1…M] configures the flush logic of an interconnection node.

ETH_C_MI_RAPS_Compatible_Version configures the Backward compatibility logic.

ETH_C_MI_RAPS_Revertive configures the revertive mode.

ETH_C_MI_RAPS_Sub_Ring_Without_Virtual_Channel configures the sub-ring type.

Defects

The function detects dFOP-PM in case the R-APS protocol is used.

Consequent Actions None.

Defect correlations

cFOP-PM ← dFOP-PM