INTERNATIONAL TELECOMMUNICATION UNION

TELECOMMUNICATION STANDARDIZATION SECTOR

STUDY PERIOD 2001-2004

STUDY GROUP 13

TD 33 (WP 3/13)

English only

Original: English

Question: 3/13 Geneva, 3-12 February 2004

TEMPORARY DOCUMENT

Source: Editor, Y.17ethoam

Title: Draft Recommendation Y.17ethoam (OAM functions and mechanisms for Ethernet based networks)

This TD provides the text for draft Recommendation Y.17ethoam (OAM functions and mechanisms for Ethernet based networks) in its Annex. This draft captures the discussions in the Q.3/13 sessions.

Contacts:

Ken-ichi Kawarai
Fujitsu Limited
Japan
Tel: +81 44 754 3877
Fax: +81 44 754 3524
Email: kawarai@jp.fujitsu.com

Ana Szpaizer
Nortel Networks
Canada
Tel: +1 613 765 1674
Fax: +1 613 763 4371
Email: janele@nortelnetworks.com

Annex

Summary

<Mandatory material>

Ke ywords

<Optional>

Introduction

This Annex contains draft Recommendation Y.17ethoam" OAM mechanisms for Ethernet based networks"

OAM Functions and Mechanisms for Ethernet based networks

1 Scope

2 References

The following ITU-T Recommendations and other references contain provisions, which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published.

The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation

NOTE: The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[1] ITU-T Recommendation Y.1730 (2004), Requirements for OAM functions in Ethernet based networks

[2] ITU-T Recommendation I.610 (1999), *B-ISDN operation and maintenance principles and functions*.

[3] CCITT Recommendation M.20 (1992), *Maintenance philosophy for telecommunications networks*.

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

[5] ITU-T Recommendation G.8010 (2003), *Architecture of Ethernet Layer Networks*.

[6] ITU-T Recommendation G.8041 (2001) *Generic Framing Procedure (GFP)*.

[EDITOR'S NOTE] TO BE COMPLETED

3 Definitions

EDITOR'S NOTE: CHECK TERMS BELOW

This Recommendation introduces some functional architecture terminology that is required to discuss the network components associated with OAM. Relevant terms are defined below.

3.1 defect: Interruption of the capability of a transport entity (e.g. network connection) to transfer user or OAM information[2].

3.2 failure: Termination of the capability of a transport entity to transfer user or OAM information. A failure can be caused by a persisting defect[2].

3.3 ETH trail: a trail in the ETH layer

3.4 ETH link: a link in the ETH layer

3.5 link: A "topological component" which describes a fixed relationship between a "subnetwork" or "access group" and another "subnetwork" or "access group"[3].

3.6 **trail**: A "transport entity" which consists of an associated pair of "unidirectional trails" capable of simultaneously transferring information in opposite directions between their respective inputs and output[3]..

3.7 CE (customer edge device), which could be an Ethernet switch, a router or a host.

3.8 PE provider Edge device, which does not do any customer MAC switching, rather it encapsulates the customer traffic into a tunnel (e.g., IP, MPLS, ATM, FR, EOS).

3.9 Point to point Ethernet connection, is an end-to-end connection between two CEs.

[EDITOR'S NOTE] TO BE COMPLETED

4 Abbreviations

EDITOR'S NOTE: CHECK ABBREVIATIONS BELOW

This Recommendation uses the following abbreviations.

CE Customer Edge device
DoS Denial of Service
ETH Ethernet
MAC Media Access Control
ME Maintenance Entity
MEP Maintenance entity End Point
MIP Maintenance entity Intermediate Point
NMS Network Management System
OAM Operation and Maintenance
PE Provider Edge device
SLA Service Level Agreement
TCP Traffic Conditioning Point

[EDITOR'S NOTE] TO BE COMPLETED

5 Conventions

Maintenance Entity End Point (MEP) is a short name for an expanded ETH flow point that includes an ETH Segment flow termination function (that marks the end point of an ETH Maintenance Entity) and an ETH Diagnostic flow termination function. The ETH Maintenance Entity is capable to initiate fault management OAM like CC and RDI and terminate OAM like CC, AIS and RDI. The ETH Diagnostic flow termination function is capable to initiate and react to diagnostic OAM like loopback and traceroute. MEP is represented by square symbol as Figure 5.1.

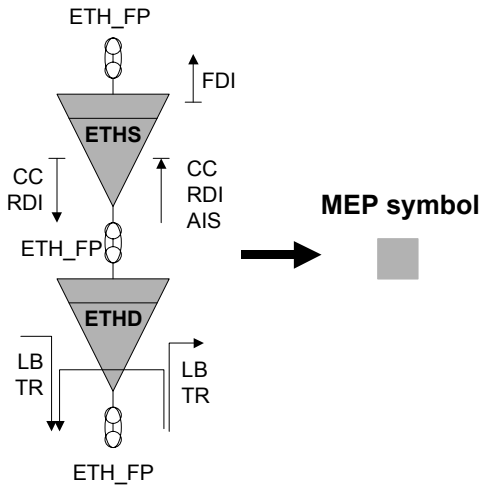


Figure 5. 1 – Maintenance entity End Point (MEP) symbol

Maintenance entity Intermediate Point (MIP) is a short name for an expanded ETH flow point including two ETH Diagnostic flow termination functions that are capable to react to diagnostic OAM like loopback and traceroute. MIP is represented by circle symbol as Figure 5.2.

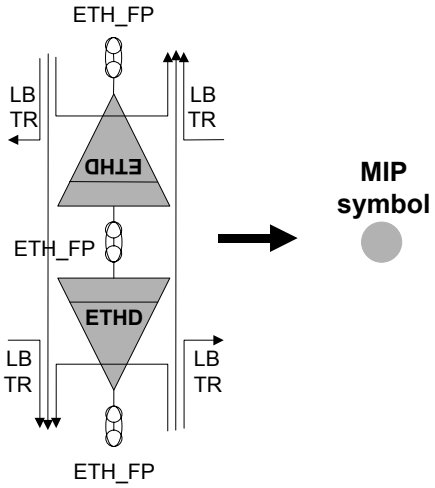


Figure 5. 2 – Maintenance entity Intermediate Point (MIP) symbol

Traffic Conditioning Point (TCP) is a short name for an expanded ETH flow point including an ETH traffic conditioning function like policing and shaping functions. TCP is represented by diamond symbol as Figure 5.3.

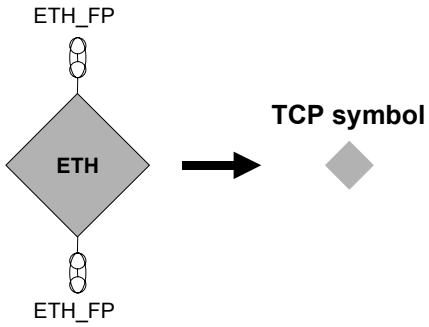


Figure 5. 3 – Traffic Conditioning Point (TCP) symbol

6 Maintenance Entities (ME)

Maintenance entities in Ethernet networks are defined in G.8010 [4] and Y.1730 [1]. The relationship between those definitions as per both Recommendations is shown in Table 6-1.

Maintenance entities are represented by the corresponding OAM flows (Y.1730) and they are defined between flow points.

Figure 6-1 shows an example of a network scenario with the corresponding Maintenance Entities End and Intermediate points (MEP and MIP respectively).

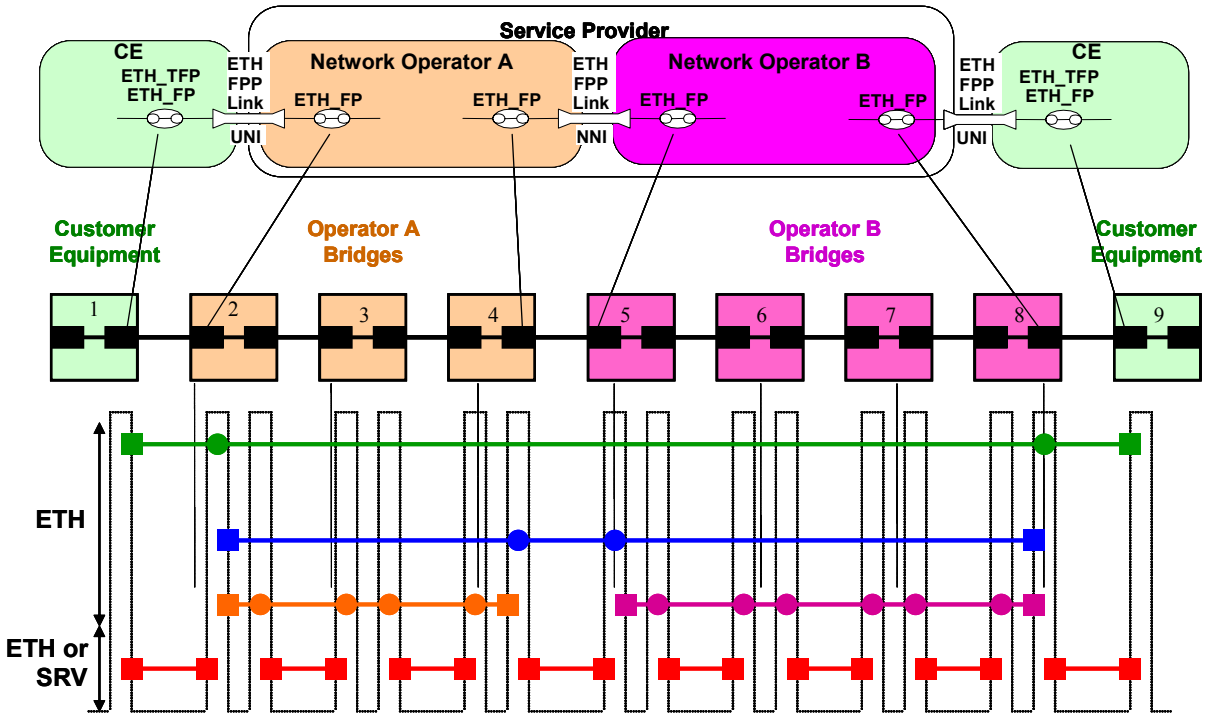


Figure 6-1: Example of ME End Point and ME Intermediate Point

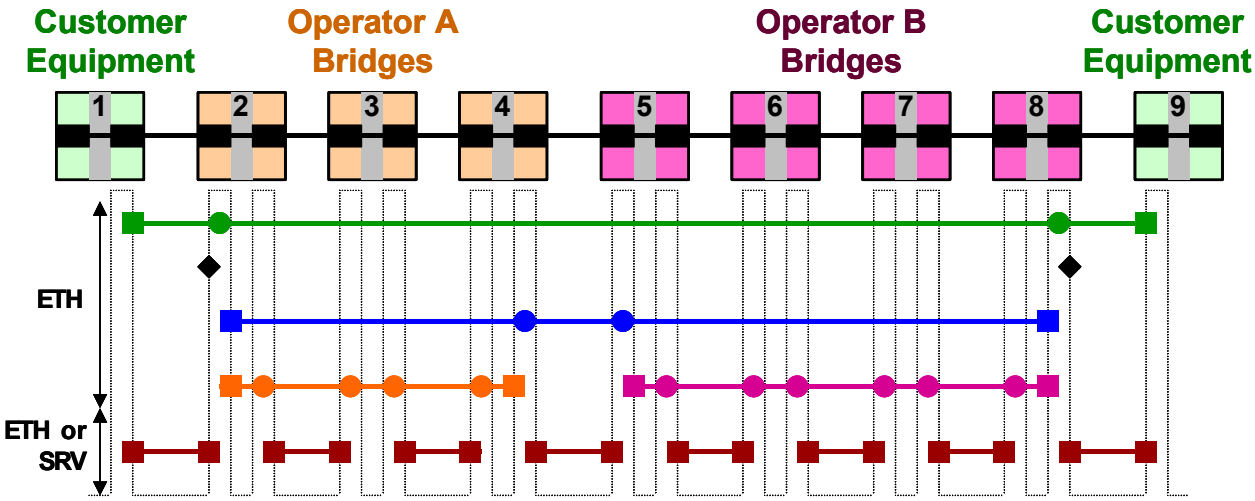


Figure 6-2 – Example of ETH MEs with their MEPs, MIPs and TCP

- p2p ETH connection between customer equipment 1 and 9 supported by a service provider and two network operators A and B
- green indicates a UNI-C to UNI-C ETH ME (customer) with MEPs in the interface ports facing the network in CE1 and CE9 and MIPs in the network interface ports facing the CEs (B2 and B8)
- blue indicates a UNI-N to UNI-N ETH ME (service provider) with MEPs at the edge of the network (B2,B8) and MIPs at the boundary of the two network operator domains (B4,B5)
- orange and mangenta indicate UNI-N to NNI ETH MEs (network operator) with MEPs at the edge of the operator networks (B2,B4 and B5,B8) and MIPs at each of the other interface ports
- brown indicates ETH link related MEs either realised as ETH MEs (sublayer monitoring) or as server layer MEs (inherent monitoring)
- black indicates location of unidirectional ETH TCPs; left TCP for direction CE1 to CE9 and right TCP for direction CE9 to CE1
- note: the TCP must be located before any of the ETH MEPs in the interface port ➔ this is not correctly illustrated at the moment in the figure!! The black TCPs should be moved to the bottom of the figure if link is sublayer (ETH ME) monitored

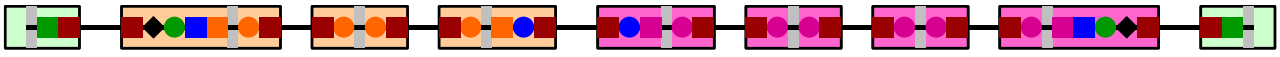


Figure 6-3 – Illustrating the order of MEPs, MIPs and TCP for example in Figure 6-2

- same p2p ETH connection as in previous figure, now represented in linear order; ETH_CI traffic units will pass through these MEPs, MIPs, TCPs in the presented order
- note: if link ME is an ETH ME (sublayer monitoring), then the TCPs must flip position with brown MEPs

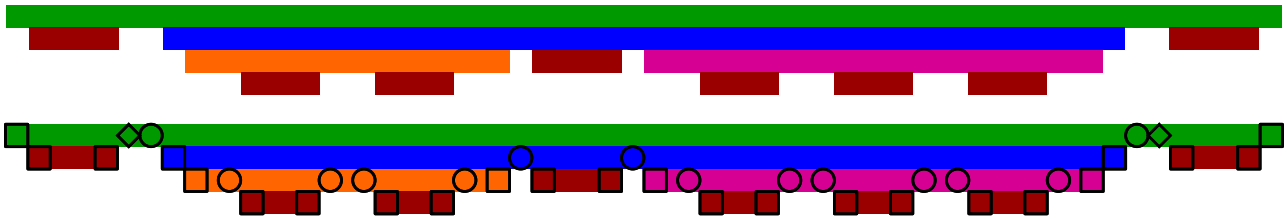


Figure 6-4 – Illustrating the stacking of ETH MEs or ETH and SRV MEs

- same p2p ETH connection as in previous two figures, now represented as a stack

6.1 MEP, MIP, TCP for Dual Relay Model and Bundling

6.1.1 Single Integrated Provider Device

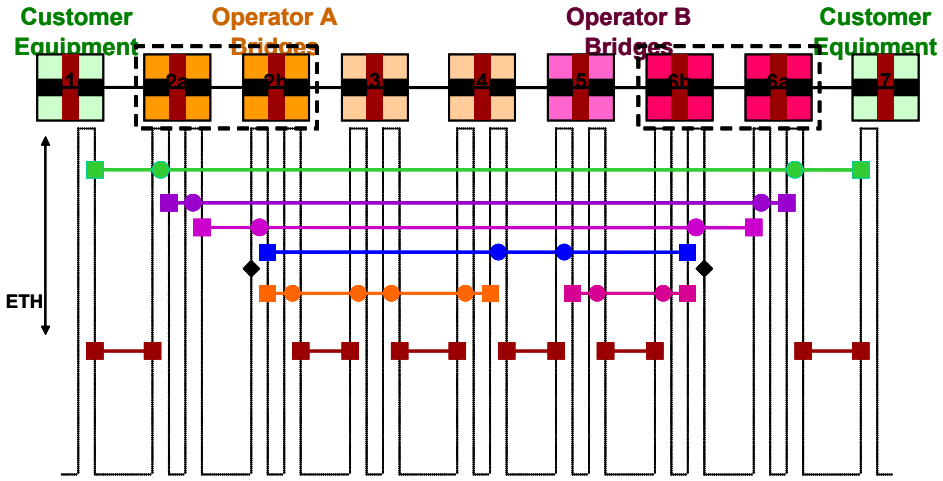


Figure 6-5: – Illustrating the location of MEPs, MIPs and TCPs in network model for the case of a one p2p connection across dual-relay provider devices

- Provider Device is represented as a dual relay model implemented with both relays. The first relay allows peering of customer L2CP protocols + multiplexing of multiple customer flows onto a single access link between the customer equipment 1 and provider bridge 2 (shown here as 2a and 2b).
- Due to the dual relay model, additional ME are introduced shown here in purple and pink between 2a and 6a. The Purple ME is associated with per customer VLAN at the provider equipment. The Pink ME is associated with per service instance (Service VLAN) that the provider applies to customer service frames.
- Between the dual relays, there are pseudo interfaces which correspond 1-to-1 with the Service VLAN or Provider Tag, which is expected to be inserted at second relay 2b.
- FFS: The positioning of the TCPs is for further study since TCPs can be positioned at customer access link level, per customer VLAN level and per provider VLAN (aka service) level.

6.1.2 Dual Relay Model with Single Relay as Provider Device

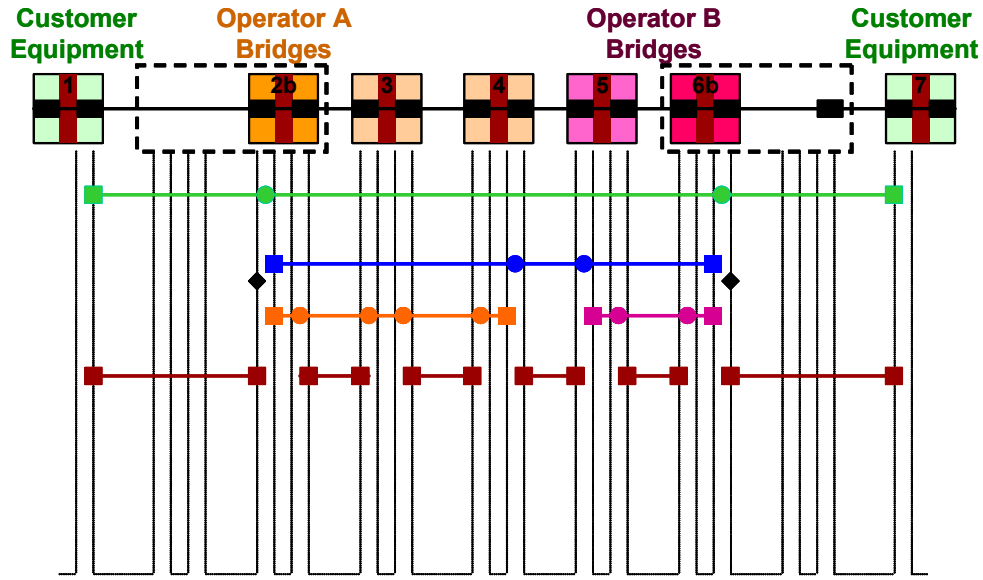


Figure 6-6: Illustrating the location of MEPs, MIPs and TCPs in network model for the case of a one p2p connection across dual-relay modelled provider device with a single relay

- Provider Device is represented as a dual relay model implemented with single relay (shown as 2b). In this case, the second relay does not allow peering of customer L2CP protocols + requires a single link for every service it supports across the customer device 1.
- Also customer device 1 is responsible for multiplexing multiple customer flows onto a single service link.
- Due to the provider using a single relay of dual relay model, additional ME that were introduced in Fig 1, are expected to be present at the customer device and are now shown here since customer is expected to manage his/her arrangements and its relationship with SP is limited to one single service instance ME marked here by Green ME between 1 and 7.
- The positioning of the TCP is clearer in this case and is shown in above figure. Customer is responsible for per customer VLAN level conditioning
- Provider Device is represented as a dual relay model implemented with single relay (shown as 2b). In this case, the second relay does not allow peering of customer L2CP protocols + requires a single link for every service it supports across the customer device 1.
- Also customer device 1 is responsible for multiplexing multiple customer flows onto a single service link.
- Due to the provider using a single relay of dual relay model, additional ME that were introduced in Fig 6-2, are expected to be present at the customer device and are now shown here since customer is expected to manage his/her arrangements and its relationship with SP is limited to one single service instance ME marked here by Green ME between 1 and 7.

- The positioning of the TCP is clearer in this case and is shown in above figure. Customer is responsible for per customer VLAN level conditioning

6.1.3 Dual Relay Model with Bundling for Single Integrated Provider Device

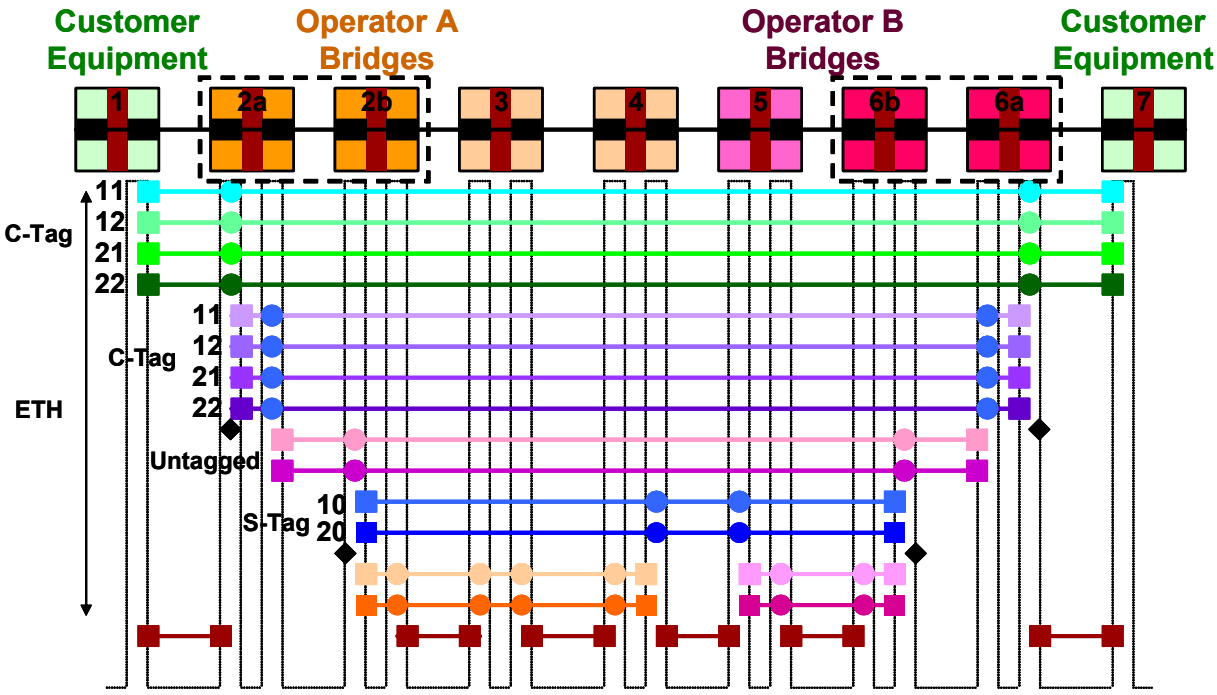


Figure 6-7: Illustrating the location of MEPs, MIPs and TCPs in network model for the case of a 2 p2p connection services with bundling across dual-relay provider devices

- Customer is shown using 4 customer VLANs (11, 12, 21, 22). It is also indicated that that customer signs up for 2 p2p connection services which the provider carries across the provider network using 2 provider VLANs (10 and 20). It is assumed that 2 customer VLANs (11 and 12) are mapped to provider VLAN 10 and other 2 customer VLANs (21 and 22) are mapped to provider VLAN 20.
- Additional ME are introduced in Figure 1 between 2a and 6a are replicated per customer VLAN and provider VLAN.
- MEs corresponding to the dual bridge pseudo interfaces which correspond 1-to-1 with the provider VLANs (10 and 20) are shown as untagged since frames from 1st relay e.g. 2a are expected to have no provider tag as they arrive at 2nd relay e.g. 2b.
- FFS: The positioning of the TCPs is for further study since TCPs can be positioned at customer access link level, per customer VLAN level and per provider VLAN (aka service) level.

6.1.4 Dual Relay Model with Bundling for Single Relay as Provider Device

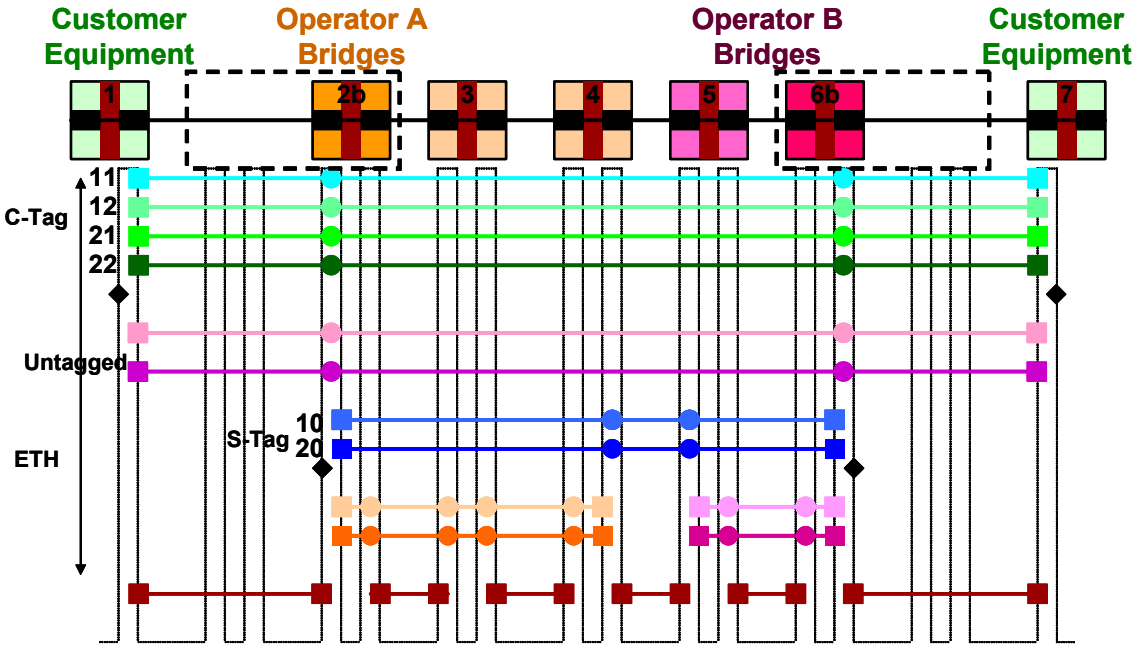


Figure 6-8: Illustrating the location of MEPs, MIPs and TCPs in network model for the case of a 2 p2p connection services with bundling across dual-relay provider devices with a single relay

- Due to the provider using a single relay of dual relay model, bundling is realized across the customer device 1 and 7.
- Additional ME is introduced at customer devices to highlight the responsibility of the customer for ME corresponding to per customer VLAN (shown here by 4 different green MEs between customer devices 1 and 7 for customer VLANs 11, 12, 21, and 22) and per service (shown here by 2 different purple MEs between customer devices 1 and 7) .
- The positioning of the TCP is clearer in this case and is shown in above figure. Customer is responsible for per customer VLAN level conditioning.

6.1.5 Dual Relay Model with all-to-one Bundling for Single Relay as Provider Device

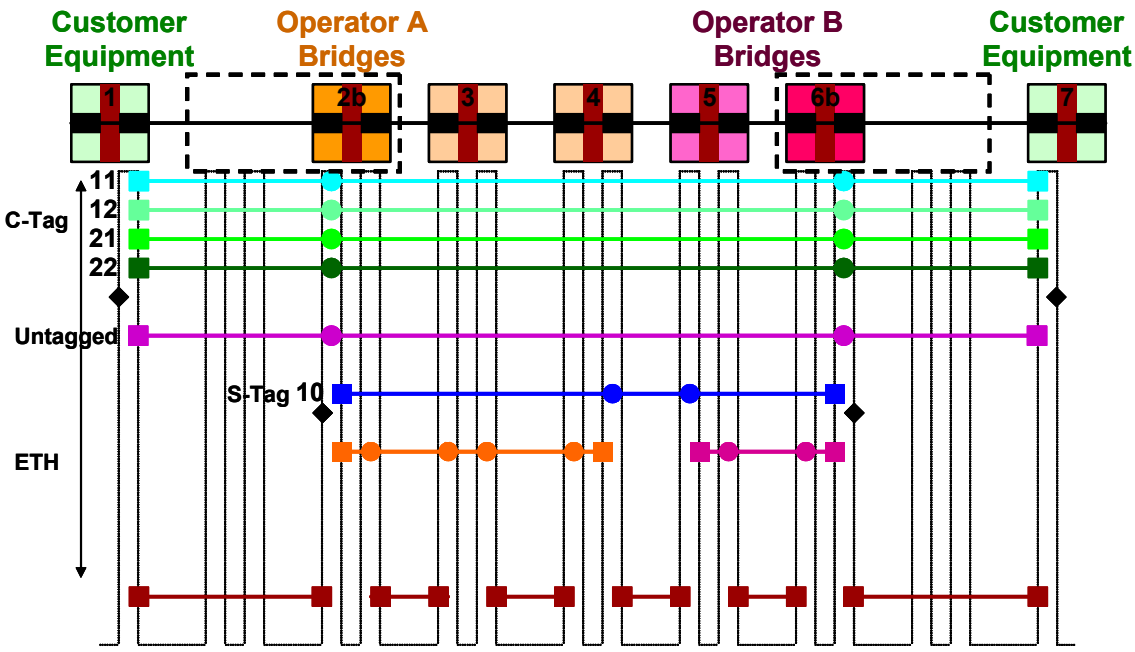


Figure 6-9: Illustrating the location of MEPs, MIPs and TCPs in network model for the case of a 1 p2p connection services with All-to-one bundling across dual-relay provider devices with a single relay

6.2 OAM Flows

Y.1730 identifies different OAM flows which represent maintenance entities. OAM flows can be inserted and extracted at the reference points, namely the flow points and termination flow points. The following OAM flows are identified:

- Customer UNI-UNI flow between reference points on the customer side of the UNI.
- Provider UNI-UNI flow between reference points on the provider side of the UNI
- Segment OAM flows:
  - Between flow points on the boundary of a provider network
  - Between flow points on the boundaries of two adjacent provider networks
  - Between any flow points as required
- ETY link OAM flow

Depending on the OAM flow, a provider may seek to limit it within its administrative boundary. For example, segment OAM flows between flow points on the boundary of a provider network may not be allowed to reach a customer network or another provider network. Similarly a segment OAM flow between flow points on the boundaries of two adjacent provider networks may not be allowed to reach a customer network or another provider network.

A mapping of the maintenance entities defined in section 9 of Y.1730 and in Figures 23 and 24 of G.8010 and their relationship with the OAM flows (Y.1730), is shown in the following table:

Table 6-1

Y.1730		G.8010
ME	**OAM flows**	**ME**
UNI-UNI (Customer)	UNI-UNI Flow	UNI_C to UNI-C ME
UNI-UNI (provider)	Transit Flow	UNI_N to UNI_N ME
Segment (PE-PE) intra-provider	Transit Flow	Intra Domain ME
Segment (PE-PE) inter-provider	Transit Flow Transit Link Flow	Inter Domain ME
Segment (any to any)	Transit Flow Transit Link Flow	
ETY Link OAM - UNI	UNI Link Flow	Access Link ME
ETY Link OAM - NNI	Transit Link Flow	Inter Domain ME

EDITOR'S NOTE: APPENDIX I INCLUDES MORE MATERIAL REGARDING OAM FLOWS THAT ARE INCLUDED THERE TO CONTINUE DISCUSSIONS.

7 OAM functions for fault management

7.1 Continuity Check(CC)(keepalive)

Ethernet Continuity Check (CC) can be applied to detect connectivity and continuity faults across Maintenance Entities between a given pair of flow termination functions. It could also be used to detect the MAC addresses of ME end-points. Continuity failures could result due to hard or soft failures, with software failure, memory corruption, or mis-configurations being some soft failures, as mentioned in Section XXX. When used in context of a specific service instance, CC can be applied to detect connectivity failures across a given pair of flow termination functions that bound a ME. that support that common service instance. Though CC can be used to detect connectivity faults across a given pair of flow termination functions. ny pair of flow points, it is particularly useful across a pair of edge flow points.

To detect connectivity failures with either a given set of flow point or all flow points meeting certain condition(s) within a boundary, CC OAM signal is generated and inserted in the ETH_FT_So and ETHS_FT_So functions. It is extracted and processed in the ETH_FT_Sk and ETHS_FT_Sk functions. Refer to Figure 1. CC is generated with either specific Unicast DAs or to a Multicast DA. Condition(s) could be that all edge flow points should receive this CC or all edge flow points participating in a service instance should receive this CC. Upon reception of the first CC from a particular flow point, the receiving flow point identifies continuity with sending flow point and expects to receive further periodic CCs. Once the receiving flow point stops receiving periodic CC from sending flow point, it detects that continuity to sending flow point is broken. Following detection of continuity failure, the detecting flow point may notify the operator, initiate fault verification followed by optional fault isolation step.

It may be noted that this mechanism has certain limitation in performing continuity failure detection. When a flow point starts participating in a network or within a network in a particular service instance for the first time, and if it has continuity failure with other flow point (s), the CC OAM frames will not reach those other flow point (s). Under such scenario, those other flow point (s) fail to detect continuity failures with this flow point. This scenario can be addressed by configuring for each flow point, a list of other flow point from which CC should be expected.

NOTE: Change flow points to functions. Also mention that configuration is always done and then this mechanism can be used to detect the continuity …..Remark that configuration…..

When the flow termination function is present, CC will be present. Periodicity of CC can be a configurable parameter?

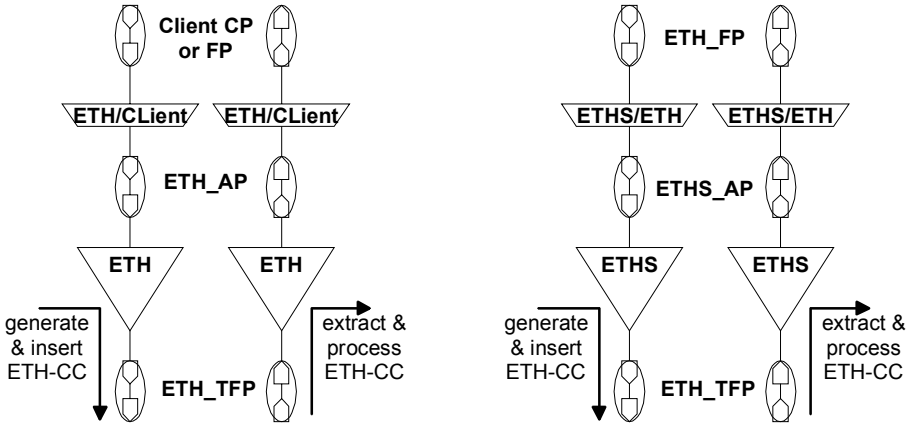


Figure 7.1-2 – Insertion/extraction & processing locations of ETH-CC OAM

In a mp connection with N endpoints there are N-1 ETH maintenance entities terminated by each ETHS_FT function. Each of these ETH maintenance entities is to be monitored for continuity and connectivity. An ETHS_FT_Sk function terminating those N-1 ETH maintenance entities should therefore expect to receive ETH-CC OAM from N-1 ETH_FT_So/ETHS_FT_So functions (Figure 2). If less than N-1 ETH-CC OAM frames are received the ETHS_FT_Sk should be able to state from which of the N-1 ETHS_FT_So functions it is not receiving the ETH-CC OAM frame(s). If it receives more than expected distinct id, then it can determine anomalies (about unexpected entities presence).

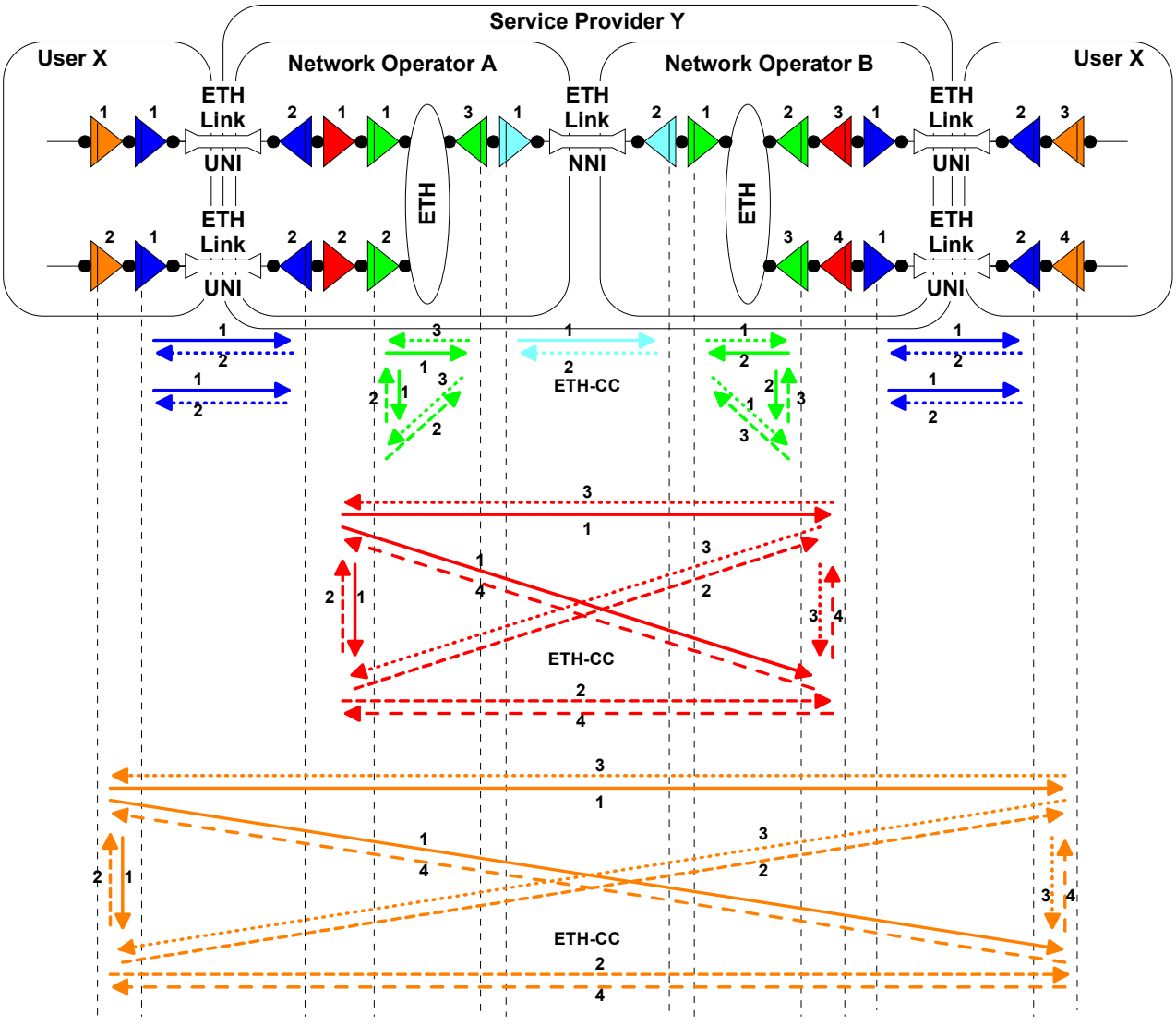


Figure 7.1-3 – ETH-CC in multi-operator ETH mp connection

NOTE: This diagram should make clear that CC message starts with Multicast DA.

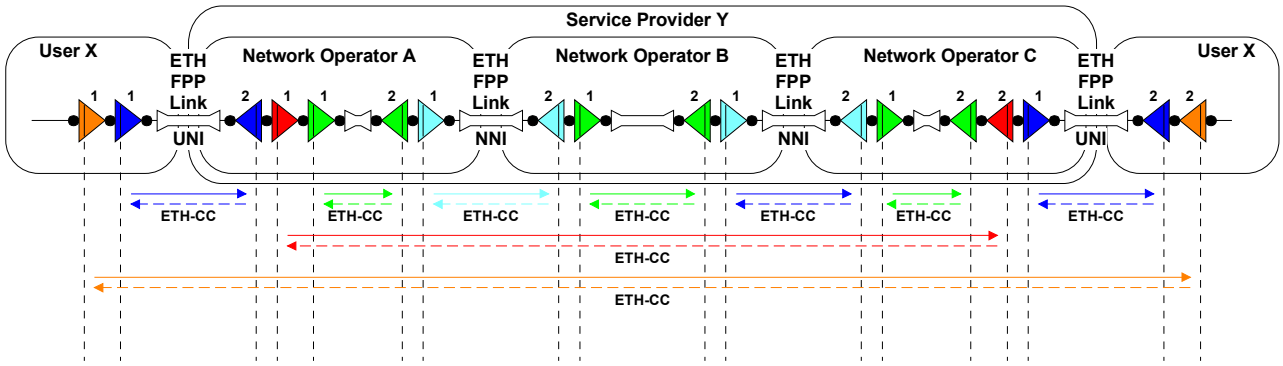


Figure 7.1-4 – ETH-CC in multi-operator ETH p2p connection

NOTE: Possible use is for loop detection using identifier information element.

7.2 Loopback

A Loopback function can be of two types:

- Intrusive Loopback
- Non-intrusive Loopback

7.2.1 Intrusive Loopback

Intrusive Loopback is used to place a remote flow point in a continuous Loopback such that all received frames would be looped back except OAM frames. Since this function results in Loopback of data frames, the data path is impacted; it is therefore considered as Intrusive Loopback. Given the nature of this function, it is expected to be always point-to-point. Intrusive Loopback OAM frames, requesting start or termination of Loopback, are expected to be Unicast (with DA = address of remote network element). Moreover, the applicability of Intrusive Loopback is expected to be limited to EPL (Ethernet Private Line) service. This function is intended for out-of-service testing.

7.2.2 Non-Intrusive Loopback

Non-intrusive Loopback is used mainly to verify continuity with remote flow point (s). Non-intrusive Loopback is performed by sending OAM frames to remote flow point (s) and expecting a response back which verifies continuity and connectivity. Since the data frames are not looped back, and the data path is not impacted; this Loopback is considered as non-intrusive. As a result, this function can be used for in-service testing.

Though a Non-intrusive Loopback may be initiated at any time, it is particularly useful when verifying continuity or connectivity once a failure is detected. Non-intrusive Loopback request may be generated either:

- automatically following detection of continuity failure, where detection could be done using connectivity check (CC) function mentioned in Section 7.6.1 of draft Recommendation Y.17ethoam, or
- On-demand via an operator initiated command, or
- Periodically.

Non-intrusive Loopback may be used for fault detection when used on a periodic basis. However, unlike CC mentioned in Section 7.6.1 of draft recommendation Y.17ethoam, Non-intrusive Loopback requires a response for each request. Response generation and response's handling by requestor require more processing in Non-intrusive Loopback as compared to CC. While a CC is suitable for detecting unidirectional connectivity failures, Non-intrusive Loopback can be used to detect bidirectional connectivity failures with single-ended maintenance entity.

A Non-intrusive Loopback can be of two types:

- Unicast Non-intrusive Loopback
- Multicast Non-intrusive Loopback

7.2.2.1 Unicast Non-intrusive Loopback

Unicast Non-intrusive Loopback request OAM frame is sent to a particular flow point (with DA = Unicast MAC address of destination network element). Upon reception of this request OAM frame, the destination flow point responds back with Non-intrusive Loopback response OAM frame (with

DA = Unicast MAC address of requesting network element, learnt from request OAM frame). Other flow points that receive this request and/or response OAM frame forward these without processing.

CC OAM signal is generated and inserted in the ETH_FT_So and ETHS_FT_So functions. It is extracted and processed in the ETH_FT_Sk and ETHS_FT_Sk functions. Refer to Figure 1.

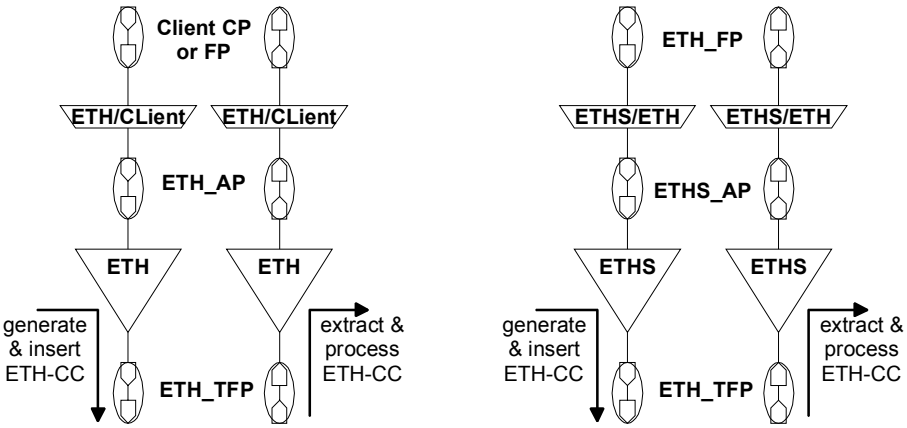


Figure 7.2-1 – Insertion/extraction & processing locations of Non-intrusive Loopback OAM

(Replace ETH-CC with Non-intrusive Loopback)

In an mp connection with N endpoints there are N-1 ETH maintenance entities terminated by an ETHS_FT function. Each of these ETH maintenance entities can be verified for continuity failures when continuity failures are detected using CC. An ETHS_FT_Sk function terminating those N-1 ETH maintenance entities can therefore expect to receive Non-intrusive Loopback OAM from N-1 ETH_FT_So functions (Figure 2). For Unicast Non-intrusive Loopback, ETHS_FT_Sk should receive OAM flow addressed to itself.

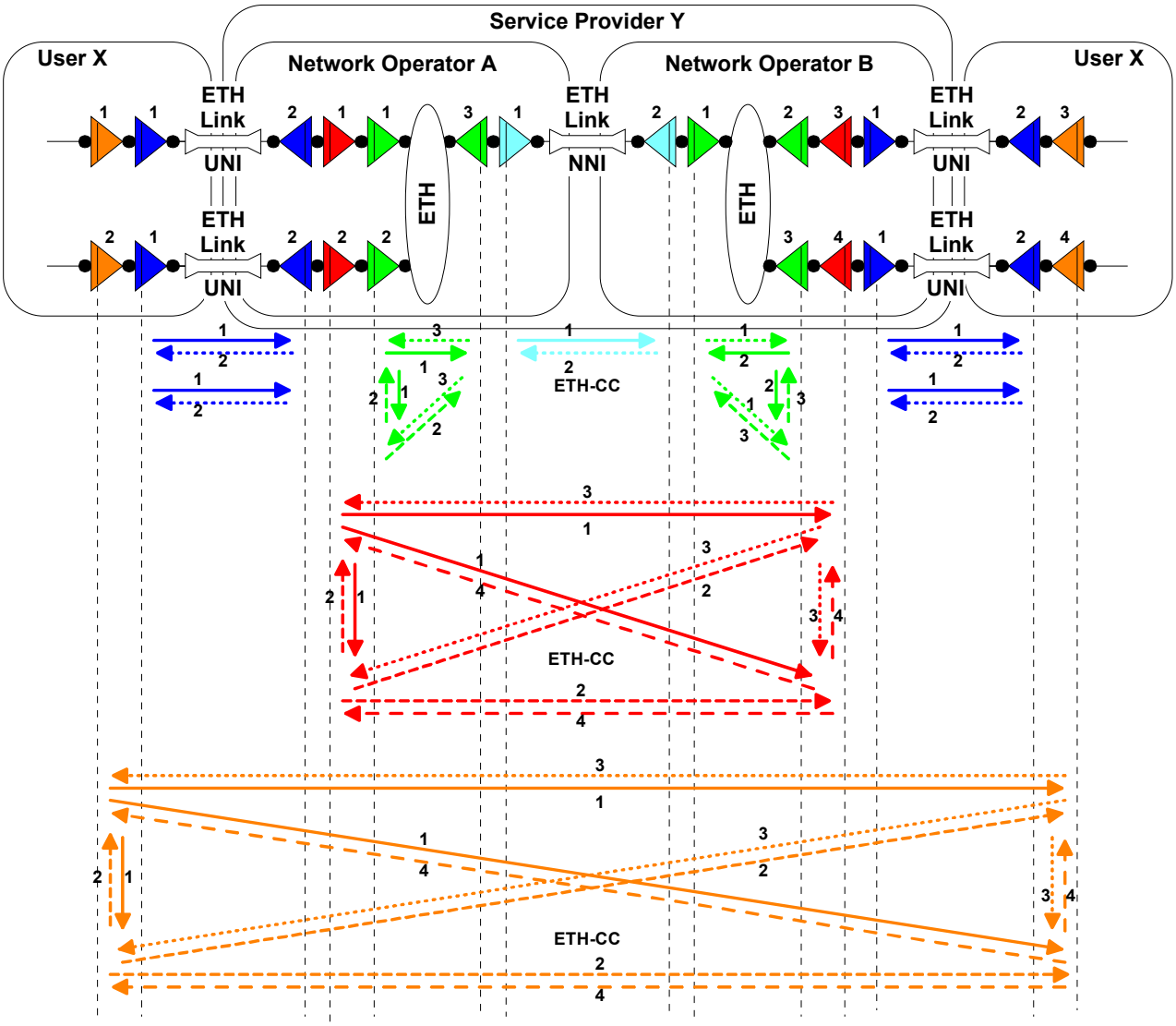


Figure 7.2-2 – Non-intrusive Loopback in multi-operator ETH mp connection

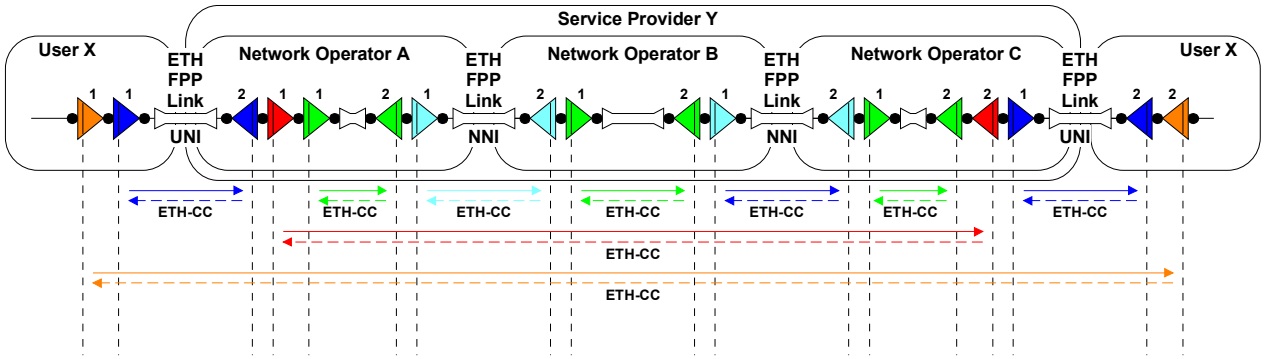


Figure 7.2-3 Non-intrusive Loopback in multi-operator ETH p2p connection

(Replace ETH-CC with Non-intrusive Loopback).

7.2.2.2 Multicast Non-intrusive Loopback

Multicast Non-intrusive Loopback request OAM frame is sent to all functions supporting Non-intrusive Loopback meeting certain condition(s) within a boundary (with DA = Multicast DA). Condition(s) could be that all edge functions should receive this request OAM frame or that all edge functions participating in a service instance should receive this request OAM frame. Upon reception of this request OAM frame, the receiving function (s) that satisfy the above mentioned condition(s) respond back with a Unicast Non-intrusive Loopback response OAM frame (with DA = Unicast MAC address of requesting network element, learnt from request OAM frame). Other functions that do not meet these conditions receive this request and/or response OAM frame and forward without processing.

7.2.3 Loopback-kind mechanism for Fault Detection

ME end points (point-to-point)

Alternative for CC

- Single sided ME
- Topology
  - Point-to-point
  - Point-to-multipoint

7.2.4 Loopback-kind mechanism for Fault Localization

- ME
- Set of loopback entities
- Loopback entities associated with a specific level (i.e. ME level)
- Loopback can be at different ME levels
- Scenarios
- Loopback response includes loopback address (i.e. SA of response is loopbacked)

7.2.5 Loopback-kind mechanism for Performance measurements

- Associated with ME (point-to-point)
- Measurements (2-way)
  - Delay
  - Jitter

7. 3 Path Trace

7.4 FDI/BDI (AIS/RDI)

8 OAM functions for performance management

8.1 Performance Parameters

Current discussions in Metro Ethernet Forum on performance parameters for Ethernet networks and services have focused on the following parameters, which are captured in an MEF working draft (January 2003)

- **Frame Loss (FL)**
  Difference between the number of service frames sent to ingress UNI and the number of service frames received at egress UNI. This is applied to Ethernet Virtual Connection (EVC) which corresponds to UNI_N to UNI_N ME.
- **Frame Delay (FD)**
  Frame delay can be specified in terms of round-trip delay, which is defined as the time elapsed since start of the transmission of the first bit of a frame by the source node until the reception of the last bit of the loop backed frame by the same source node, when the loop back is performed by the frame's destination node.
- **Frame Delay Variation (FDV)**
  Measure of the variations in the frame arrival pattern belonging to the same CoS instance compared to the arrival pattern at the ingress of the MEN.
- **Availability**
  Function of time the ME (associating service UNIs) is in available state. It is specified as a ratio of:
  Total Time ME is in Available State / Total Service Time

  where, **Total Service Time** is viewed as number of time intervals and **Available State** is viewed as interval when service meets FL, FD and FDV bounds. Unavailable state is encountered when at least one of the FL, FD or FDV measures exceed their bounds/thresholds during a time interval. These bounds/thresholds are determined by the class of service (CoS).

  Note:
  The definition of Availability should be aligned with Y.1711 [9] and/or Y.MPLSperf [10]. The details of Availability are expected to be defined in a separate Recommendation being developed by Q.6/13 – Y.17EthPerf.

Note:

For sub rate or virtual services, the frame loss can be associated with both in-profile and out-of-profile service frames.

Additional performance parameters that may be taken into consideration include:

- **Errored Frame Seconds**
  Indicates if an error (e.g., frame error due to FCS or 8B/10B coding violation) has occurred within the second. This does not take into consideration errors when frames are received error free but are not delivered.
- **Service Status**
  Indicates if the service is in-service or out-of-service. In-service or out-of-service state can be based on **Available state** defined earlier.
- **Frame Throughput**
  Number of frames and/or bytes transmitted to network interface relative to CIR

- **Frame Tx**
  Number of frames transmitted out of the customer facing interface within the (previous) time interval (e.g. 1 second).
- **Frame Rx**
  Number of frames received from the customer facing interface within the (previous) time interval (e.g. 1 second).
- **Frame Drop**
  Number of frames dropped at the customer facing interface within the (previous) time interval (e.g. 1 second).
- **Loopback Status**
  Indicates whether the customer facing interface is in an intrusive Loopback state (potentially due to OAM interactions across Access Link ME).
- **Client Signal Fail**
  Indicates state of Access Link ME.
- **Unavailable Time**
  Number of time intervals (e.g. 1 second) when the service status is unavailable.

8.2 Measurement Mechanisms

Different measurement mechanisms are possible to perform performance measurements. One significant difference across these mechanisms is the level of accuracy of measurements. These mechanisms include:

- **Management plane statistical methods**
  Statistical methods use OAM frames to estimate data path behavior. Such methods are least accurate since they apply approximation to emulate data frames.
  The limitation lies in that the behavior of actual data frames may be quite different due to different addressing, processing, transient congestion conditions etc. Also, error conditions in networks typically happens in bursts thus statistical methods can likely miss those bursts and represent different results.
- **Management plane managed objects**
  Here OAM frames use data path managed objects to calculate performance parameters and are inserted and/or extracted via management plane. These methods are fairly accurate since they use data path statistics to measure data path performance.
  Their limitation lies in that since the insertion and extraction of these OAM frames is done via management plane, in-flight frames need to be accounted for. On the egress side of OAM frame, in-flight frames refers to data frames between accessing egress data path managed objects and actual transmission of OAM frame. On ingress side of OAM frame, in-flight frames refer to data frames between reception of OAM frame and subsequent accessing of ingress data plane managed objects. However, this limitation can be addressed by averaging such measurements across multiple time intervals.
- **Data path OAM frames**
  OAM frames use data path managed objects and are inserted and/or extracted via data plane. This method tends to be most accurate since it does not have the limitation associated with the in-flight frames.
  However, the current data path hardware/chips do not support the implementation of such methods since this requires Ethernet data path processing to include automatic insertion and/or extraction of OAM frames with data plane managed object values. Moreover, it would also require changes in hardware/chips to allow ingress and egress filtering rules across OAM frames to protect service provider administrative domains from unintended OAM frames.

This contribution proposes the use of management plane managed objects mechanism. The advantage of these mechanisms is that these require no changes in the existing hardware/chips and only require change in OAM client software that needs to be implemented. The steps involved in such measurement mechanism include:

- Collection of managed object (s) information
- Calculation of performance parameter (s)

8.2.1 Performance Management Collection Method

To collect managed object information, general or specific methods can be used. When a generic method is used, it can be applied to collect information across different managed objects e.g. using TLVs as information elements instead of specific information elements. However, when specific method with specific information elements is used, a separate method is needed per managed object or per set of managed objects.

Similarly, it is possible to use either a solicited or unsolicited collection method, where solicited method requires a response after an OAM request frame is sent while unsolicited methods does not require a response to an OAM frame. Some current examples of solicited and unsolicited methods include Loopback and Continuity Check in Y17ethoam [3] respectively, though these are currently not used as performance management collection methods.

This contribution proposes to use a generic method to send/receive data path managed object information. This is similar to the variable request/response method used in IEEE 802.3ah [4 – section 57.4.3.3/.4]. The contribution also proposes to use both solicited and unsolicited methods and optionally extend the currently defined Loopback [3 – section 14.1] and Continuity Check [3 – section 10.1]. Note that this extension for PM will require additional processing and therefore should not be used for the measurement of delay.

8.2.2 Frame Loss Measurement

MEs which can support Frame Loss include:

- Service MEs for point-to-point service with dedicated UNIs
    - UNI_C to UNI_C
    - UNI_N to UNI_N
    - Access Link (UNI)
    - Inter-domain (NNI)
- Network MEs
    - Intra-domain
    - Inter-domain

8.2.3 Unsolicited Method

When applied across UNI_N to UNI_N ME, OAM frame is sent every N seconds (e.g. N=1) with **FramesTransmittedOK** value at ingress service UNI. Upon receiving this OAM frame, **FramesTransmittedOK** value is compared with **FramesReceivedOK** value at egress service UNI. Between two such consecutive OAM frames, the FL can be measured as:

Frame Loss = |CT2-CT1| - |CR2-CR1|,

where CT and CR are **FramesTransmittedOK** and **FramesReceivedOK** counts. Consecutive messages help in reducing error introduced by in-flight frames and lack of timing synchronization between sender and receiver. Within a measurement time interval, the Frame loss count can be averaged to improve the accuracy of this measurement.

8.2.4 Solicited Method

Requestor sends OAM request frame to receiver every N seconds (e.g. N=1) with its managed objects (MOs) information and expects an OAM response frame with receiver's MOs information.

When applied across UNI_C to UNI_C ME, requestor sends **FramesTransmittedOK** value at egress service UNI and requests **FramesReceivedOK** value from receiver's ingress service UNI. Similarly, when applied across UNI_N to UNI_N ME, requestor sends **FramesReceivedOK** value at ingress service UNI and requests **FramesTransmittedOK** value from receiver's egress service UNI

Upon receiving the OAM request frame, receiver compares received MO information with its corresponding MO information and sends a response OAM frame back to requestor with requested MO information. When applied across UNI_C to UNI_C ME, receiver compares received **FramesTransmittedOK** value with **FramesReceivedOK** value and responds with its **FramesTransmittedOK** value. Similarly, when applied across UNI_N to UNI_N ME, receiver compares received **FramesReceivedOK** value with its **FramesTransmittedOK** value and responds with its **FramesTransmittedOK** value.

Upon receiving OAM response frame, requestor compares original sent value with received values, similar to receiver. It is possible that receiver returns the results of frame loss instead of MO information in response, however, if the MO information is returned, the performance collection method remains generic.

Between two such consecutive OAM frames, the FL can be measured as:

Frame Loss = |CT2-CT1| - |CR2-CR1|,

where CT and CR are **FramesTransmittedOK** and **FramesReceivedOK** counts. Consecutive messages help in reducing error introduced by in-flight frames and lack of timing synchronization between sender and receiver. Within a measurement time interval, the Frame loss count can be averaged to improve the accuracy of this measurement.

The above method can be applied for measuring network level Frame Loss. The network level frame loss can be measured within the network independent of the services.

For non-dedicated point-to-point service types with multiplexed service UNI, where a UNI carries more than one service flow, it is possible to measure FL when data path MOs per service instance are supported.

8.2.5 Statistical Method

For multipoint-to-multipoint service type, statistical method across a pair of UNIs can be applied to estimate frame loss.

The requestor sends N OAM request frames to the recipient and receives M response frames back from the recipient such that M <= N. The data path frame loss can be estimated as:

Frame Loss = (N – M) per measurement time interval

As noted earlier, statistical methods are less accurate than proposed method in this contribution.

8.3 Frame Delay Measurement

Services supported include point-to-point and multipoint-to-multipoint between a given pair of UNIs.

MEs across which the frame delay can be measured are:

- Service MEs
    - UNI_C to UNI_C
    - UNI_N to UNI_N

Solicited Method – Loopback

This method measures round-trip or two-way frame delay. Requestor sends OAM request message with its timestamp to the receiver. Receiver replies copying the requestor's timestamp. At the requestor, the difference between the timestamps at the time of receiving the OAM response frame and original timestamp in the OAM response frame results in round trip frame delay.

8.4 Frame Delay Variation Measurement

Services supported include point-to-point and multipoint-to-multipoint between a given pair of UNIs.

MEs across which the frame delay variation can be measured are:

- Service MEs
    - –UNI_C to UNI_C
    - –UNI_N to UNI_N

Solicited Method – Loopback

This method measures round-trip or two-way frame delay per request and response frame. Within the period of observation, requestor keeps track of maximum frame delay (FD_{max}) and minimum frame delay (FD_{min}). Frame delay variation is calculated as:

Frame Delay Variation or Jitter = FD_{max} - FD_{min}

Information elements for FDV method in OAM Data mentioned in Y.17ethoam [3 - section 15.1] include:

- Sequence number
- Request Timestamp
    - FDV or Jitter = {FD (max) – FD (min)} per measurement time interval

- Information elements for FD method in OAM Data
  - Sequence number
  - Request Timestamp

8.5 Availability Measurement

Services supported include point-to-point with at least dedicated UNIs.

MEs across which the frame delay variation can be measured are:

- Service MEs
  - –UNI_C to UNI_C
  - –UNI_N to UNI_N

Measurement Method

Measurement is based on FL, FD and FDV methods. Availability time interval (e.g. 24hr) can be divided into measurement time intervals (e.g. 1 minute). FL, FD and FDV are measured per measurement time interval. If any of the three measures crosses its corresponding thresholds, which are dependent on the service type, the measurement time interval is considered to be unavailable else it is considered to be available.

Availability = (# of available measurement time intervals)/(# of total measurement time intervals)x100%

Note: The details of the availability are expected to be specified by Ethernet Traffic Management activities. This contribution proposes mechanisms that can be used to measure **Availability** based on how it is defined.

8.6 Other Measurements

As per the unsolicited method proposed earlier in this contribution, the following parameters can be sent every time interval (e.g. 1 second) to the peer.

8.6.1 Errored Frame Seconds

ME: Access Link ME

Within 1 second, check if any increments in (aFrameCheckSequenceErrors, aAlignmentErrors, aFramesAbortedDueToXsColls, aFramesLostDueToIntMACXmitError, aCarrierSenseErrors, aFrameLostDueToIntMACRcvError)

If yes, declare that 1 second as errored frame second

8.6.2 Service Status

ME: UNI_C to UNI_C ME or UNI_N to UNI_N ME

Within the measurement time interval (e.g. 1 min), declare whether the service is up or down as per availability measurement, explained earlier

8.6.3 Frame Throughput

ME: UNI_N to UNI_N

Within the measurement time interval, aFramesTransmittedOK at egress UNI_N relative to CIR

8.6.4 Frame Tx

ME: Access Link ME

Within 1 second, aFramesTransmittedOK at egress UNI_N

8.6.5 Frame Rx

ME: Access Link ME

Within 1 second, aFramesReceivedOK at ingress UNI_N

8.6.6 Frame Drop

ME: Access Link ME

Within 1 second, ifInDiscards at ingress UNI_N and ifOutDiscards at egress UNI_N.

8.6.7 Loopback Status

ME: Access Link ME

aLoopbackStatus at UNI_N.

8.6.8 Client Signal Fail

ME: Access Link ME

aLinkStatus at UNI_N.

8.6.9 Unavailable time

ME: UNI_N to UNI_N

This is related to availability definition with the unavailable time intervals being counted within the observation period.

9 Discovery

NOTE: AS PER DISCUSSIONS IN THE LAST INTERIM MEETING OF Q.3/13 (NOVEMBER 03) LOOPBACK WAS RULED OUT AS A MECHANISM FOR DISCOVERY.

EDITOR'S NOTE: THE FOLLOWING MATERIAL HAS TO BE REVISED SINCE IT WAS NOT AT THE LAST INTERIM MEETING ON NOVEMBER 2003.

Discovery is of two types:

- Solicited discovery
- Unsolicited discovery

NOTE: There is also a possibility of provisioned model where this information is configured as part of provisioning on each edge node.

9.1 Unsolicited Auto-discovery

This can use CC as defined in Section 7.1

Each edge node can build the information about the other edge nodes that participate in a specific service instance based on the CC messages received from these other edge nodes for that specific service instance.

9.2 Solicited Auto-discovery

The use of this is expected to be mostly done on-demand.

10 Information elements

EDITOR'S NOTE: THE COMMON INFORMATION ELEMENTS AND THE ONES EXCLUSIVE FOR CC HAVE BEEN UPDATED AT THE MEETING OF FEBRUARY 2004

10.1 Connectiviy Check

- Required Common Information Elements
  - Addressing (DA, SA MAC)
  - ME Level
  - OAM EtherType
  - Version
  - OAM OpCode = CC
  - ServiceID
  - TransactionID
- Required CC Information Elements
  - Source MEP ID

Other Functionality/Information Elements are for further study:
e.g.
  - CC Expiry Indication
  - MEP Status - CC Activation/Deactivation Indicator

CC Behavior

- Current assumptions for CC function are:
  – It does not require a reply from the receiving MEP.
  – It is periodic.
  – CC (and also other OAM packets) should be prevented from exiting from its domain and entering into other domains at the same level.

Note: For the purposes of Continuity Check, the CC message does not need to be processed at the MIP. This does not preclude the processing of CC message at the MIP for other purposes.

- For Further Study: CC with reply from receiving MEP:

  It is possible that only one end of ME i.e. only one MEP is managed by NMS (Network Management System) while the other end is not (e.g. Access link case where the customer MEP may not be under Provider's management). In this case the CC with reply may be useful.

Addressing

- SA
  - This is a Unicast MAC.
- DA
  - Current assumption is that it is a Multicast DA.
    - The source MEP may not know the MAC address of the destination MEP, but may only know the destination MEP ID.
    - Higher efficiency than Unicast
    - Discover the MAC address associated with the destination MEP ID which may be changed independently at the destination e.g. hardware replacement.
  - Regarding Multicast MAC address type, current proposals include:
    - (A) Unique Multicast MAC for each OAM Domain Level
      - <u>Advantages:</u>

      - ME Level Identification: No separate ID is needed since Multicast MAC address contains it.
      - Filtering: This allows existing bridges to be configured to allow processing (termination of OAM packet and/or discard) of OAM packets applicable at specific level. It also allows OAM packets at other levels to pass through transparently.

      - <u>Disadvantages:</u>

      - Provisioning: This requires the MEP to be provisioned for the MAC address corresponding to its ME level.

    - (B) A Single Unique Multicast MAC for all levels
      - <u>Advantages:</u>

      - Provisioning: This does not require the bridges to be provisioned specifically since this MAC addressed could be default.

- Provisioning: This is related to approach used in ME Level field. If ME levels are fixed values, provisioning is required to tell MEP which level to operate at; while if ME levels are not fixed, no provisioning is needed since MEP always operates at level 0 as per stack approach.

▪ Disadvantages:

- ME Level Identification: Separate identifier is needed.
- Filtering: This would require existing bridges to terminate OAM packets associated with all levels to differentiate between those that need to be terminated (i.e. those at the specific level that MEP belongs) and those that need to be passed through (i.e. at higher levels).

ME Level:

FFS:

There is current discussion whether the levels should be configured with fixed values or they can be approached such that configuration is not needed.

Fixed Values: Whether implicit in Multicast MAC address (A) or explicit as a separate field (B). We need to further look at how to handle add & remove of MEs.

Non-fixed values: Possible solution is Stack-approach. Here each MEP is associated with level 0.

- When MEP at ingress of ME receives OAM packets from outside the ME, it increments the level and passes the OAM packet along.
- When the MEP at egress if ME receives OAM packet from within the ME, it terminates the one with level 0 and passes other OAM packets by decrementing the level.

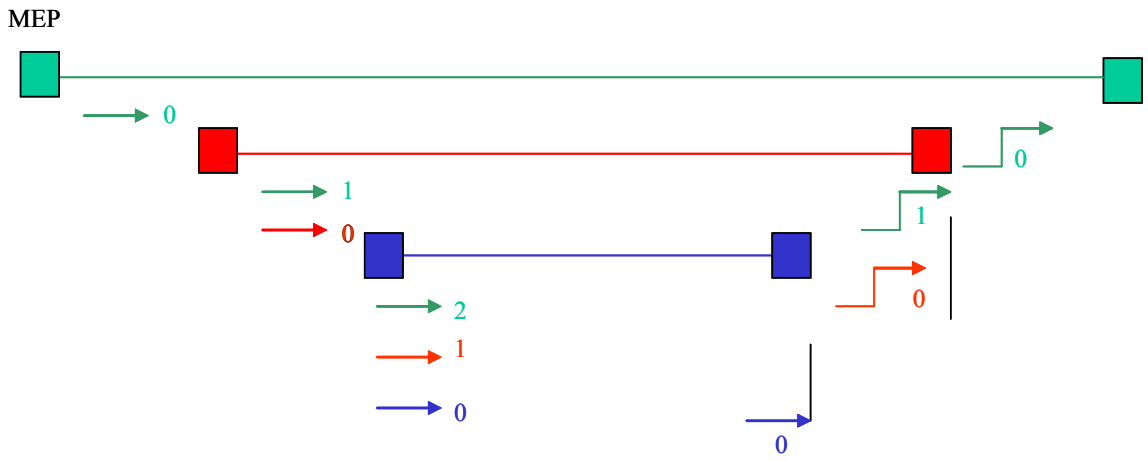
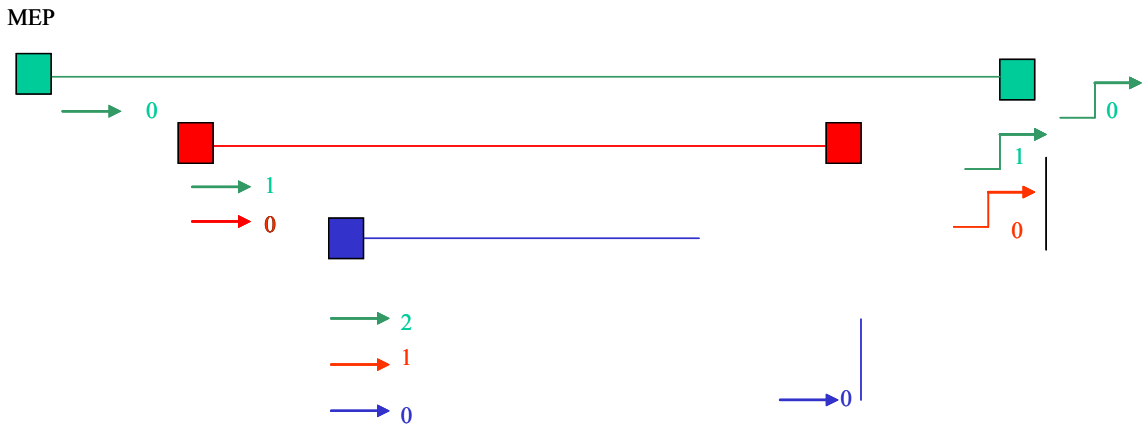


Figure 10-1: MEP Stack approach

- Limitation: There is limitation with this approach in that it can lead to OAM leaking when one MEP associated with a ME is faulty.



- Limitation: This does not allow tapping at a certain level from within the network.

Figure 10-2: Limitation of the MEP stack approach

Source MEP ID:

This is logical identifier used to identify a MEP within a service instance within a OAM domain. E.g. within a service "Foo", it identifies one of the 3 MEPs which are part of this service within the OAM domain i.e. MEP ID = "boston" or "new york" etc.

Source MEP ID is associated with the SA MAC.

The syntax of MEP ID is FFS.

CC Expiry Indication:

FFS: It identifies the period for which the information regarding the CC should be maintained at the receiving MEP. It may be required to be fixed e.g. 1s etc.

The value whether fixed or variable is for FFS and should be considered in context of applications that would use it e.g. FM, PM and/or PS (Protection Switching).

Note: The CC behavior for FM, PM and PS should be considered individually and we could combine them if these come out to be the same.

MEP Status:

Further study required regarding how MEP is:

- Added
- Removed
- Temporarily disabled

Fault Scenarios:

FFS: What happens when the MAC address associated with remote MEP ID is flip-flopping due to possible mis-configuration at the remote MEP. Does the near-end MEP simply ignore it or is expected to report mis-configuration.

10.2 Non-intrusive Loopback

- Fault Detection
  - OAM Frame Identifier – (Detection of Loops, correlation)
  - Source Port Number – (Identification of specific source port, handle)
  - Destination Port Number – (Identification of specific target port, handle)?
  - Arbitrary data part – (Stress Test: could be used to test different MTUs, pad packet to full size, put worst case patterns)?
  - Note: Addressing used for this case is option (D) i.e. Unicast Destination MAC Address. Needs validation.
- Fault Localization
  - OAM Frame Identifier – (Detection of Loops, correlation)
  - Source Port Number – (Identification of specific source port, handle)
  - Destination Port Number – (Identification of specific target port, handle)?
  - Arbitrary data part – (Stress Test: could be used to test different MTUs, pad packet to full size, put worst case patterns)?
  - Note: Addressing used for this case is option (D) i.e. Unicast Destination MAC Address. Needs validation.
- Performance – for round-trip Delay and Jitter
  - OAM Frame Identifier – (Detection of Loops, correlation)
  - Source Port Number – (Identification of specific source port, handle)?
  - Destination Port Number – (Identification of specific target port, handle)?
  - Source Timestamp
  - Note: Addressing used for this case is option (D) i.e. Unicast Destination MAC Address. Needs validation.

Extra Elements:

  - Response MAC Address
  - Randomized Delay

NOTE: loopback will not be used for discovery purposes because of potential storms in DOS scenarios due to number of replies.

10. 3 TraceRoute - Body

- OAM Frame Identifier –
- Source Port Number – (Identification of specific source port, handle)
- Destination Port Number – (Identification of specific target port, handle)
- TLV for Checksum – (checksum for part that cannot be changed)
- Target MAC Address
- Source MAC Address
- Hop Count
- Others
  - Periodicity of Loopback – (when used proactively)
  - Randomized Delay - ?

10-4 Performance Monitoring Information Elements

10.4.1 Information elements that can be applied to OAM Data for the Unsolicited Method

- Sequence number
- # of TLVs
- TLVs (Managed Object variable: **FramesTransmittedOK**, value length, value)

10.4.2 Information elements that can be applied to OAM Data for the Solicited Method

- Sequence number
- # of Transmit TLVs (value filled in by requestor, recipient simply copies it back in response)
- # of Request TLVs (value is filled in by recipient and sent back in response)
- TLVs (Managed Object variable: **FramesTransmittedOK & FramesReceivedOK**, value length, value)

10.4.3 Information elements for Frame Delay method in OAM Data

- Sequence number
- Request Timestamp

11 OAM frame formats

EDITOR'S NOTE: FRAME FORMATS WERE NOT DISCUSSED DURING THE LAST Q.3/13 INTERIM MEETING (Nov. 2003) . AS A CONSEQUENCE, THE FOLLOWING HAS TO BE REVISED AT THE NEXT MEETING.

11.1 Generic OAM Frame Format

A single generic format is defined for all Ethernet OAM messages as depicted blow. The VLAN tag is optional and if it is present, it indicates the service instance corresponding to the OAM message (e.g., only bridge nodes participating in that service instance will process/forward the OAM message). The OAM Ethernet Type has a value TBD and it indicates that the message is an OAM message for Service Provider use. It should be noted that all the OAM messages carry the same OAM Ethernet Type. It is recommended that another OAM Ethernet Type to be allocated for customer OAM usage (transparent to service provider nodes) so that there is clear differentiation between customer and provider OAM messages.

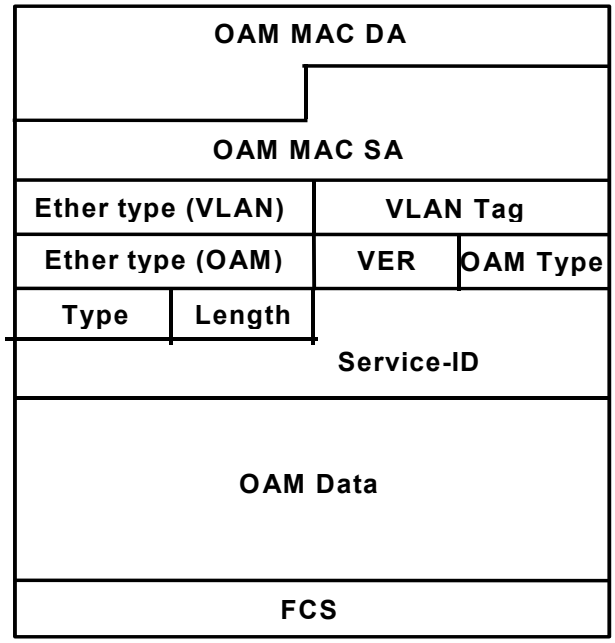


Figure 11-1: Generic OAM Message Format

The fields for the generic OAM message format are defined as follows:

- **OAM Destination MAC Address**: This MAC address can be the unicast address of a bridge or a multicast address corresponding to a group of bridges or it can be a well-defined multicast address (e.g., "bridge all" multicast address)
- **OAM Source MAC Address**: This is the MAC address of the source bridged (a unique MAC address designated for OAM functionality) or it can be the MAC address of a bridge interface over which the OAM message is sourced.
- **VLAN Ether Type and Tag**: This is an optional field and it is present when the OAM message is related to a service instance. In this case, this VLAN tag designates the associated service instance.
- **OAM Ethernet Type**: This is a unique Ethernet Type that indicates Service Provider OAM messages. It is recommended to have another Ether Type for Customer OAM messages.

OAM Type: The OAM frame type defines the type of OAM frame. The OAM frame types that are defined in this recommendation are:

- **Intrusive Loopback Request (0x00)**
- **Intrusive Loopback Release (0x01)**
- **Intrusive Loopback Reply (0x02)**
- **Non-Intrusive Loopback Request (0x03)**
- **Non-Intrusive Loopback Reply (0x04)**
- **Path Trace Request (0x05)**
- **Path Trace Response (0x06)**
- **Continuty Check (0x07)**
- **Performance Monitoring Request (0x08)**
- **Performance Monitoring Reply (0x09)**
- **AIS (0x0A)**

– **RDI (0x0B)**
– **Vendor Specific (0xFF)**. The vendor specific op-code is provided to allow vendors or other organizations to extend OAM functions in proprietary ways.
– **Other op-codes may be defined in the future**. OAM frames with unexpected unknown op-codes MUST be silently discarded.

- **OAM Version**: The Version field identifies the OAM version number. Implementations conforming to this recommendation MUST use a value of 0x01 in this field. OAM frames with a different version MUST be silently discarded
- **Service ID**: TLV for the service instance identification
- **OAM Data**: This is a data field associated with the corresponding OAM type and sub-type and its format is dependent on the OAM type/sub-type fields. The OAM frame including OAM data portion must result in an Ethernet frame with valid length. Therefore, if necessary the OAM frame must be padded with zeros for a minimum size frame.

Annex A

EDITOR'S NOTE: THE MATERIAL OF THIS ANNEX IS TAKEN FROM WD 10 OF LAST INTERIM MEETING OF Q.3/13 (NOVEMBER 2003) AND IT WAS AGREED TO INCLUDE IT IN THIS ANNEX FOR FURTHER CONSIDERATION

AIS/RDI MECHANISM FOR AN ETHERNET POINT-TO-POINT CONNECTION OVER A SINGLE SERVER LAYER (i.e. SDH or OTN)

G.8010 [4] Figure 18 (reproduced below) illustrates the architecture of an Ethernet point-to-point connection.

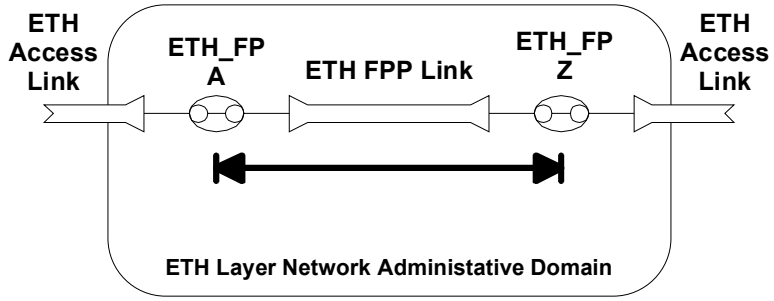


Figure 18/G.8010 – Point-to-point ETH connection (single link)

The representation of the corresponding maintenance areas is illustrated in G.8010 Figure 23 top right (reproduced below).

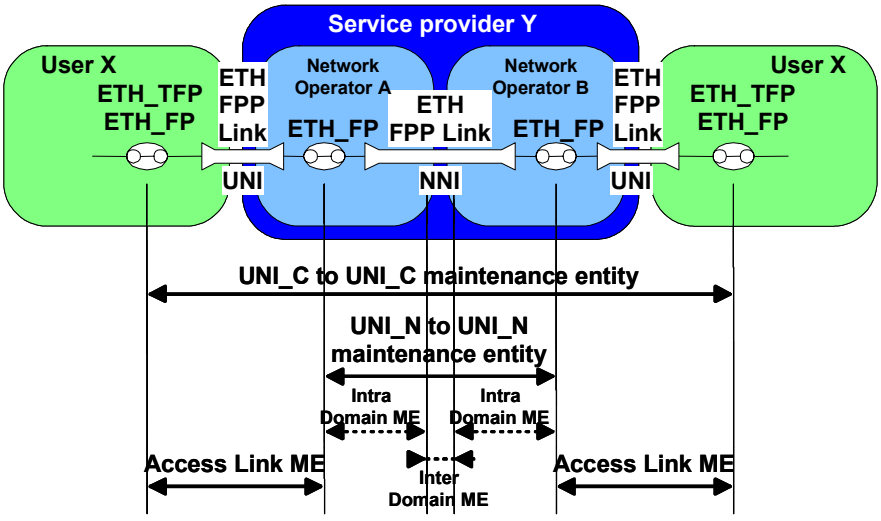


Figure 23 (top right)/G.8010 – Point-to-point ETH connection administrative domain associated maintenance entities

Of note is that there are no ETH flow points at the handoff between the two networks. So for the general case when there are multiple network operators and a single server layer, maintenance of the ETH layer is not possible within those operators' networks, and can only be performed via the server layer. That category of maintenance is called inherent monitoring, also discussed in [4].

To address the lack of AIS and network RDI functionality in EFM OAM, the issues are then:

a) How to convey an access link fault from one side of the network to the other.

b) How to convey a server layer fault to the access links.

In SG 15/Q.12, work has been progressing to define a Service Management Channel (SMC) to facilitate the provider edge NE-to-edge NE exchange of OAM information, as well as support for an intermediate provider NE to query OAM information from, and send test-related commands to, a provider edge NE.

Currently, the direction being taken in Q.12/15 proposes G.7041 [5] GFP-F Client Management Frames (CMF) for conveying the provider edge NE-to-edge NE OAM information, and a Path OH byte for the intermediate provider NE communications channel to a provider edge NE. G.7041 defines CMFs for conveying information about the client signal from an ingress edge NE to the egress edge NE. One of the defined CMF indications is Client Signal Fail (CSF). The figure below illustrates the GFP-F frame format for a CMF with a CSF indication.

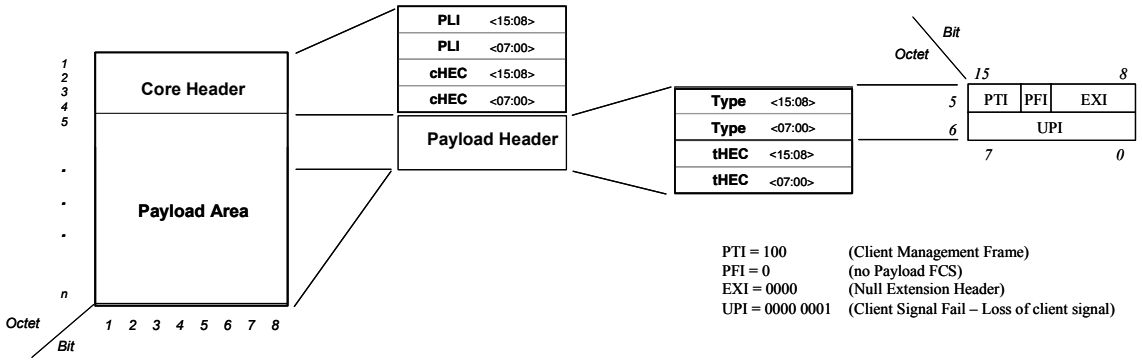


Figure A-1: GFP-F CMF CSF Frame Format

By using the EFM OAM Link Fault flag in conjunction with the GFP-F CMF CSF indication, the necessary AIS and network RDI mechanisms can be provided for an Ethernet point-to-point connection single server application.

A simplifying assumption can be made regarding the conditioning of the Ethernet access links on either side of the SDH/OTN transport network. For a dedicated point-to-point 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 indication to appropriately condition the Ethernet access links. Only uni-directional faults are considered, the scenarios can be combined per the superposition principle to describe bi-directional faults.

Scenario 1

In the figure below a uni-directional fault occurs on the east access link on ingress to the carrier network.

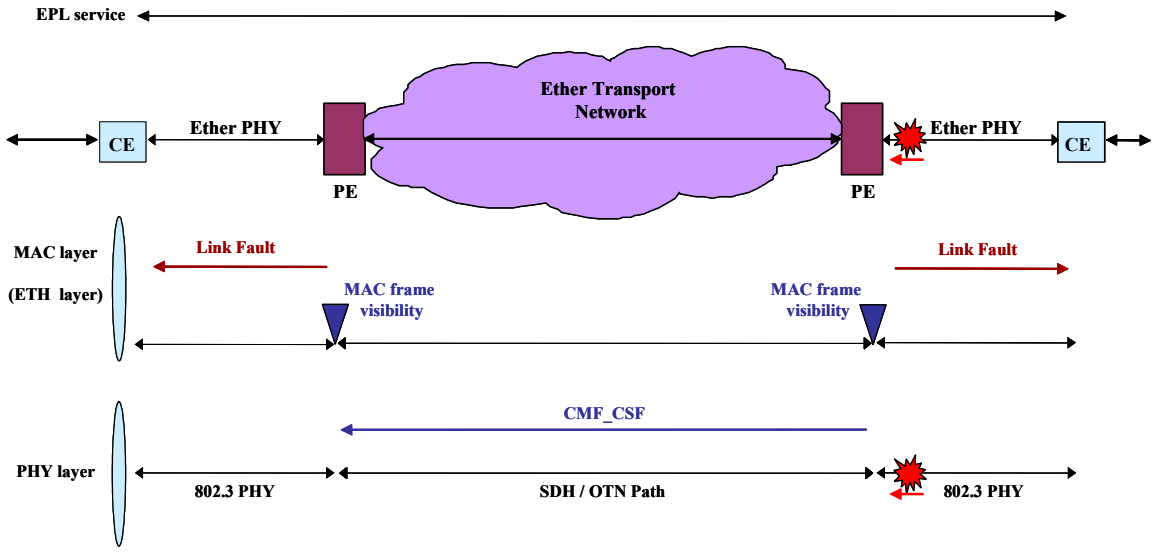


Figure A-2: Fault on Ingress

- The east PE detects the failure:
  - an 802.3ah OAM function sends Link Fault upstream, interspersed with Idles
  - a new function sends a GFP-F CMF CSF indication 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:
  - a new function translates it to 802.3ah Link Fault downstream, interspersed with Idles
- The west CE detects Link Fault:
  - Idles are sent towards the network (and towards the enterprise)

Scenario 2

In the figure below a uni-directional fault occurs westbound within the carrier network.

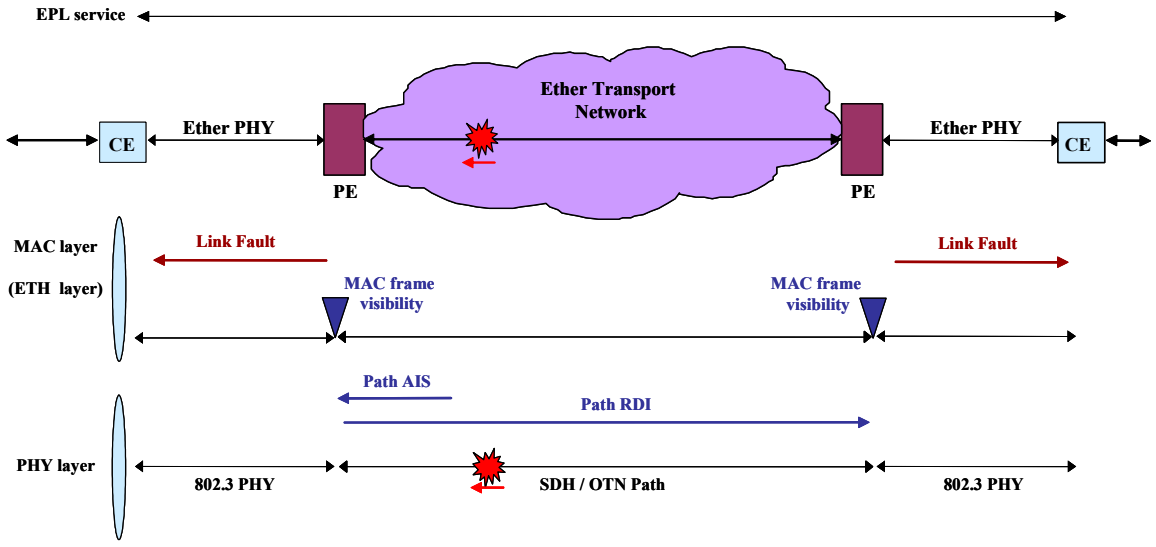


Figure A-3: Fault within Carrier Network

- An NE (or the west PE) in the carrier network detects the failure:
  - SDH Path AIS is generated downstream
- The west PE detects SDH Path AIS (or the fault directly):
  - a new function translates it to 802.3ah Link Fault downstream, interspersed with Idles
  - SDH Path RDI is generated back into the network
- The west CE detects Link Fault:
  - Idles are sent towards the network (and towards the enterprise)
- The east PE detects SDH Path RDI:
  - a new function translates it to 802.3ah Link Fault downstream, interspersed with Idles
- The east CE detects Link Fault:
  - Idles are sent towards the network (and towards the enterprise)

Scenario 3

In the figure below a uni-directional fault occurs on the west access link towards the enterprise network.

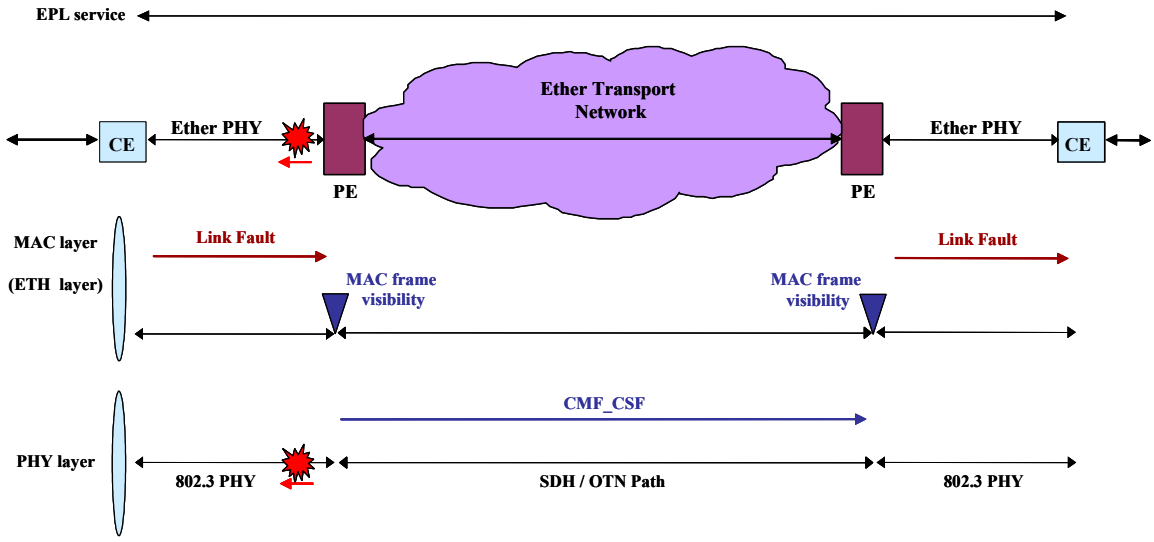


Figure A-4: Fault on Egress

- The west CE detects the failure:
  - an 802.3ah OAM function sends Link Fault upstream, interspersed with Idles
  - Idles are sent towards the enterprise
- The west PE detects Link Fault:
  - a new function translates it to a GFP-F CMF CSF indication into the network
  - Idles are sent towards the CE
- The east PE detects the GFP-F CMF CSF indication:
  - a new function translates it to 802.3ah Link Fault downstream, interspersed with Idles
- The east CE detects Link Fault:
  - Idles are sent towards the network (and towards the enterprise)

Summary

As a result, two maintenance signal translation functions, i.e. EFM OAM Link Fault flag and GFP-F CMF CSF indication can be used by the provider edge of a point-to-point single server application (i.e. SDH or OTN) in order to address network AIS and RDI mechanisms.

Appendix I

I.1 OAM Domains

Each provider can be associated with an administrative boundary, called OAM domain. A service may be carried across a single or multiple OAM domains.

As identified in Y.1730, network elements placed at the boundary of provider network serve as edge network elements and are associated with the ingress and egress of a network flow. When an edge network element of a provider performs hand-off of an ETH layer flow, while interacting with edge network element of another provider, that network element serves as an edge hand-off network element. Those network elements that are not associated with the ingress, egress or hand-off of a network flow serve as interior network elements.

It is also possible that a single provider network may have further administrative boundaries. Example is when a provider network consists of different operator networks. If this is the case, one could still identify edge, edge hand-off, and interior network elements within each such administrative boundary.

Ports on a network element in an OAM domain can be classified as interior or exterior to that OAM domain. Interior ports are those on which OAM frames, belonging to an OAM flow, are recognized and processed. Processing may result in either termination of OAM flow or relaying across other ports on the network element. Exterior ports are those on which OAM frames are not recognized and filtered. An edge network element has both interior and exterior ports to an OAM domain, while an interior network element has all its ports marked as interior ports to that OAM domain.

Within an OAM domain, OAM flows may be applicable between edge network elements only (edge hand-off network element is also an edge network element) or across all network elements (i.e. including all interior network element and edge network elements). OAM frames can be Unicast or Multicast frames. The difference between the two is based on the destination MAC address (DA). A Unicast OAM frame has a Unicast DA while a Multicast OAM frame has a Multicast DA. A Multicast OAM frame can associate itself to all edge networks elements or all network elements inside a domain based on its Multicast DA.

NOTE: Refer to G.8010 and G.805.

I.2 OAM Flows

Different OAM flows, as discussed in Section 6.1, can be identified by using an OAM flow identifier within the OAM frame. OAM flow identifier can assume the following values:

- $\text{UNI-UNI}_{\text{Customer}}$
  Customer UNI-UNI flow between reference points on the customer side of the UNI.
- $\text{UNI-UNI}_{\text{Provider}}$
  Provider UNI-UNI flow between reference points on the provider side of the UNI.
- $\text{Segment}_{\text{intra-provider}}$
  Segment OAM flow between flow points within the boundary of a provider network. This may include OAM flow between flow points on the boundary of a provider network or between any flow points within a provider network as required.
- $\text{Segment}_{\text{inter-provider}}$
  Segment OAM flow between flow points inside the boundaries of two or more provider networks. This may include OAM flow between flow points on the boundaries of two or more adjacent provider networks or between any flow points inside the boundaries of two

or more provider networks, as required. Note: Under special cases, $Segment_{inter\text{-}provider}$ may be same as $UNI\text{-}UNI_{Provider}$.

- $UNI_{Segment}$
  OAM flow between reference points (i.e. TFP and FP) on the customer side and provider side of the UNI.
- $NNI_{Segment}$
  OAM flow between flow points on two edge hand-off network elements connected to each other. Each edge hand-off network element belongs to a different provider network.
- UNI_{Link}
  If the UNI is realized using a single ETY link, this OAM flow can be used for ETY link between customer and provider network.
- $Transit_{Link}$
  Any intermediate ETY link between network elements, this OAM flow can be used.

NOTE: Both UNI_{Link} and $Transit_{Link}$ can be based on IEEE 802.3ah. [However, the reference to IEEE 802.3ah may not be possible, until it becomes a standard, though it is close to being one]

NOTE: It is worth noting that though different OAM flows have been identified, not all will be applicable for all services and/or business models; especially, there may be some limitations within multiple provider scenarios.

Y.1730		**G.8010**	**Examples**
ME	**ME**	**OAM flows**	**ME**
UNI-UNI (Customer)	UNI_C to UNI-C ME	UNI-UNI Flow	$UNI\text{-}UNI_{Customer}$
UNI-UNI (provider)	UNI_N to UNI_N ME	Transit Flow	$UNI\text{-}UNI_{Provider}$
Segment (PE-PE) intra-provider	Intra Domain ME	Transit Flow	$Segment_{Intra\text{-}provider}$
Segment (PE-PE) inter-provider	Inter Domain ME	Transit Flow Transit Link Flow	$Segment_{Inter\text{-}provider}$
Segment (any to any)		Transit Flow Transit Link Flow	$Segment_{Intra\text{-}provider}$ $Segment_{Inter\text{-}provider}$
ETY Link OAM - UNI	Access Link ME	UNI Link Flow	UNI_{Link}
ETY Link OAM - NNI	Inter Domain ME	Transit Link Flow	NNI_{Link}

It is conceivable that value of OAM Flow Identifiers can be such that filtering can be done based on whether the OAM frame entering or exiting a domain have OAM Flow Identifier value smaller than minimum OAM Flow Identifier configured on the interior and or exterior port of a domain.

For example, if the following octet values are assigned to OAM Flow identifiers:

- $UNI\text{-}UNI_{Customer}$ = 255 (0xFF)
- $UNI\text{-}UNI_{Provider}$ = 253 (0xFD)
- $Segment_{inter\text{-}provider}$ = 251 (0xFB)
- $NNI_{Segment}$ = 249 (0xF9)
- $UNI_{Segment}$ = 247 (0xF7)
- $Segment_{intra\text{-}provider}$ = 245 (0xF5)
- UNI_{Link} =
- $Transit_{Link}$ =

And, if the following minimum OAM flow Identifier values are configured across the different ports:

- NNI port = 249 (0xF9)
- UNI port = 247 (0xF7)
- Interior port = 245 (0xF5)

Filtering at edge network elements can be achieved such that OAM frames with OAM Flow identifier smaller than the minimum OAM Flow identifier are not allowed into or out of the OAM domain.

I.3 Fault Types

Two fault types are recognized in relationship with ETH OAM:

1) ETH Discontinuity
2) ETH Misconnection
3) ETH Link Faults

The two first fault types are shown in the following Figure I.3-1:

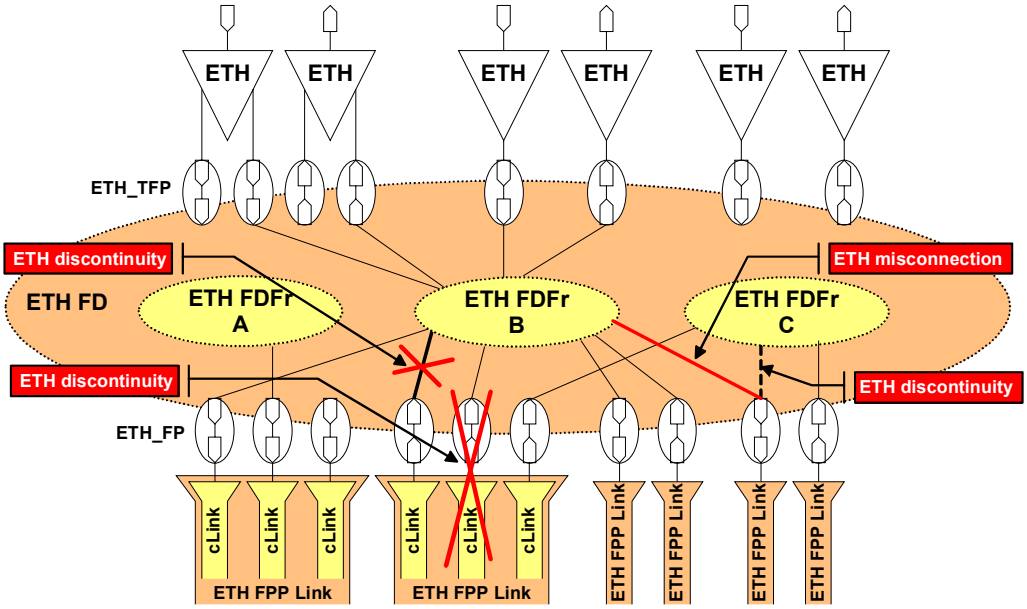


Figure I.3-1

ETH Discontinuity

Causes can be:

- Physical fault (i.e. fibre cut)
- Failure of a bridge
- Looping (customer or provider loops or due to the use of a wrong topology)
- Misconfiguration

ETH Misconnection

Can be caused by:

- Misconfiguration

Appendix II

The examples below relate to figure 6-2

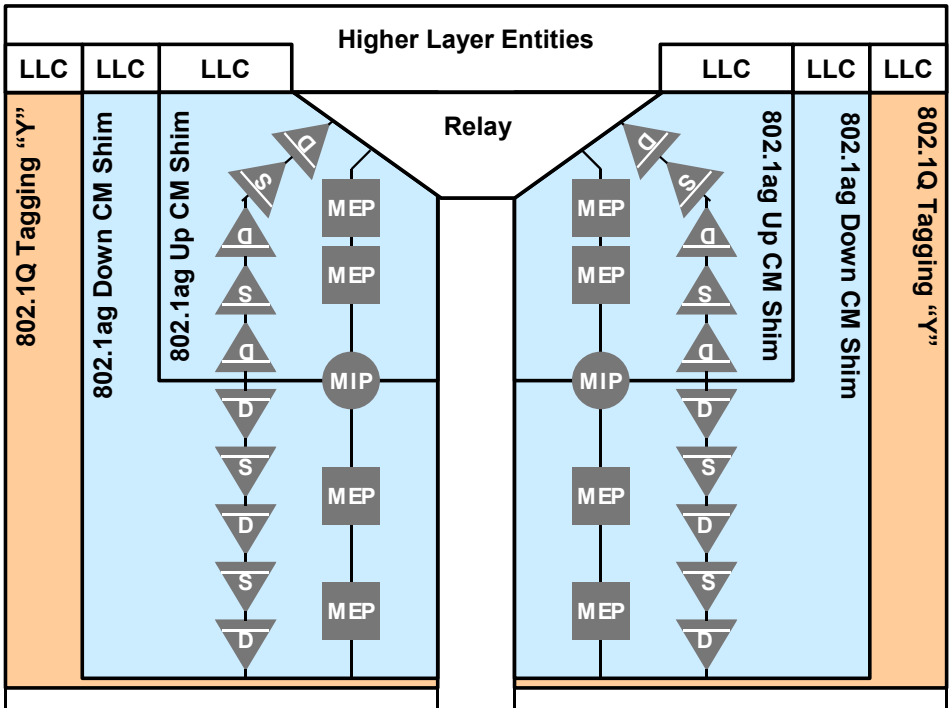


Figure II-1 – Illustrating the location of MEPs and MIPs in the 802.1ag Up and Down CM Shims in the IEEE 802.1 "baggy pants" model

- two representations are shown in parallel: the MEP and MIP shorthand symbolic representation and their associated atomic function representation
- both the up and down shims in an interface port to a bridge may support multiple MEPs; in this example each shim supports two MEPs
- the MIP is located at the junction of up and down shim as it has an ETH Diagnostic function in both the up and the down shim.
- the MEPs and MIP in an interface port have no pre-assigned knowledge of the ME level they will be operating at; this is represented by making them all colourless (i.e. grey)
- further baggy pants figures will present only MEPs and MIPs, not longer the associated atomic functions

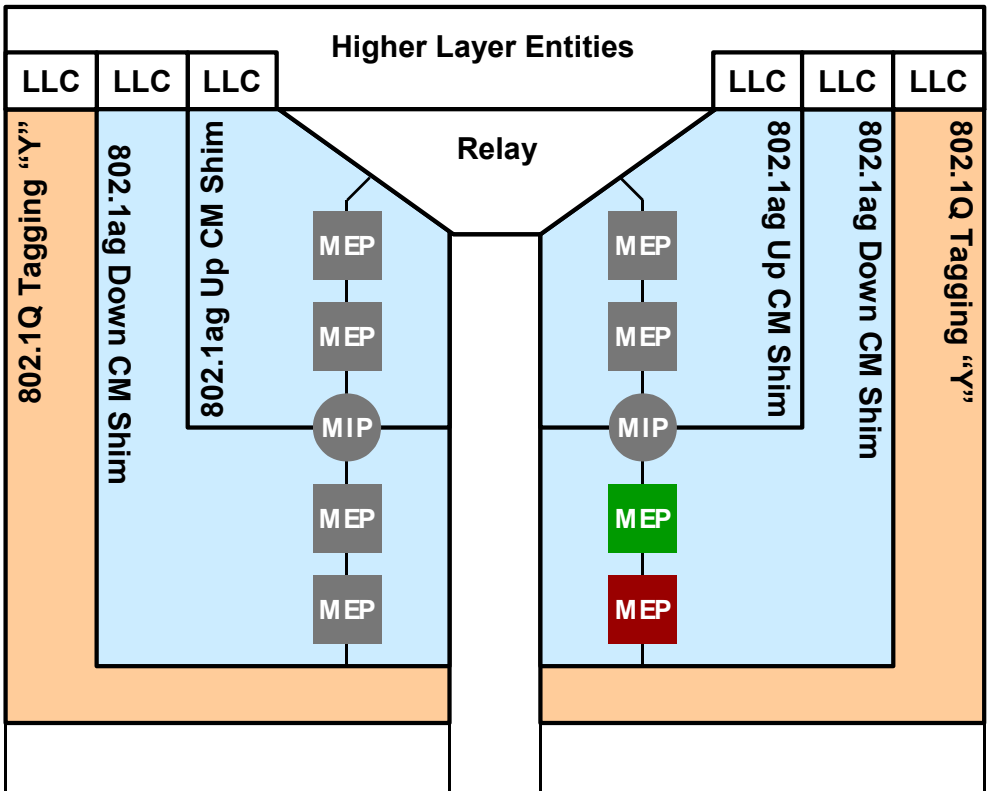


Figure II-2 – Illustrating the location of MEPs in the "baggy pants" model for CE1

- customer equipment number 1 has for the ETH connection of figure 6-2 two MEPs activated in the down shim to monitor the UNI-C to UNI-C connection and the CE1 to B2 link
- the other MEPs and MIPs are made transparent to represent that those are inactive for this connection

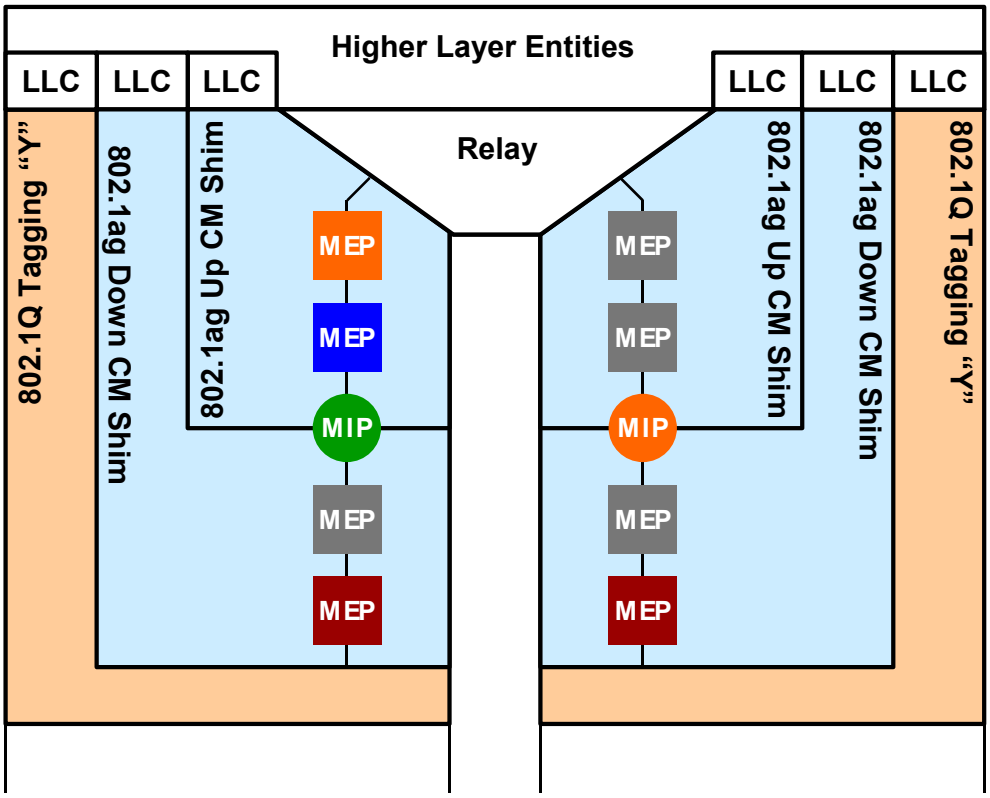


Figure II-3 – Illustrating the location of MEPs in the "baggy pants" model for B2

- operator A bridge number 2 has at its customer equipment facing port three MEPs active; one to terminate the link ME, one to terminate the service provider's ETH ME (blue) and one to terminate the operator A's ETH ME
- this port has also an active MIP for use by the customer's UNI-C to UNI-C ETH ME
- the second interface port on this bridge has an active MIP (operator A's ETH ME) and an active MEP (B2 to B3 link)
- note that the choice of the active MEP in the two down shims is arbitrary; the other MEP could have been choosen as well. Equipment should be capable to change the MEP location hitless in order to support the addition of an ME level above or below an existing ME level

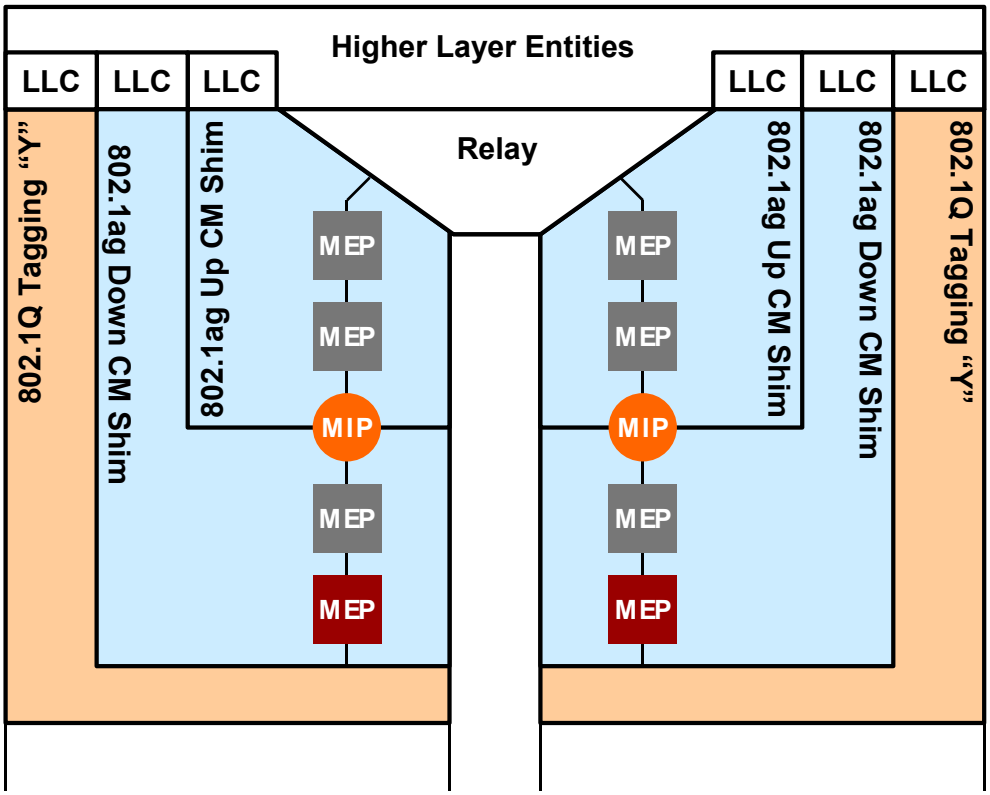


Figure II-4 – Illustrating the location of MEPs in the "baggy pants" model for B3

- operator A bridge number 3 has two ETH link related MEPs and two operator A ETH ME related MIPs active

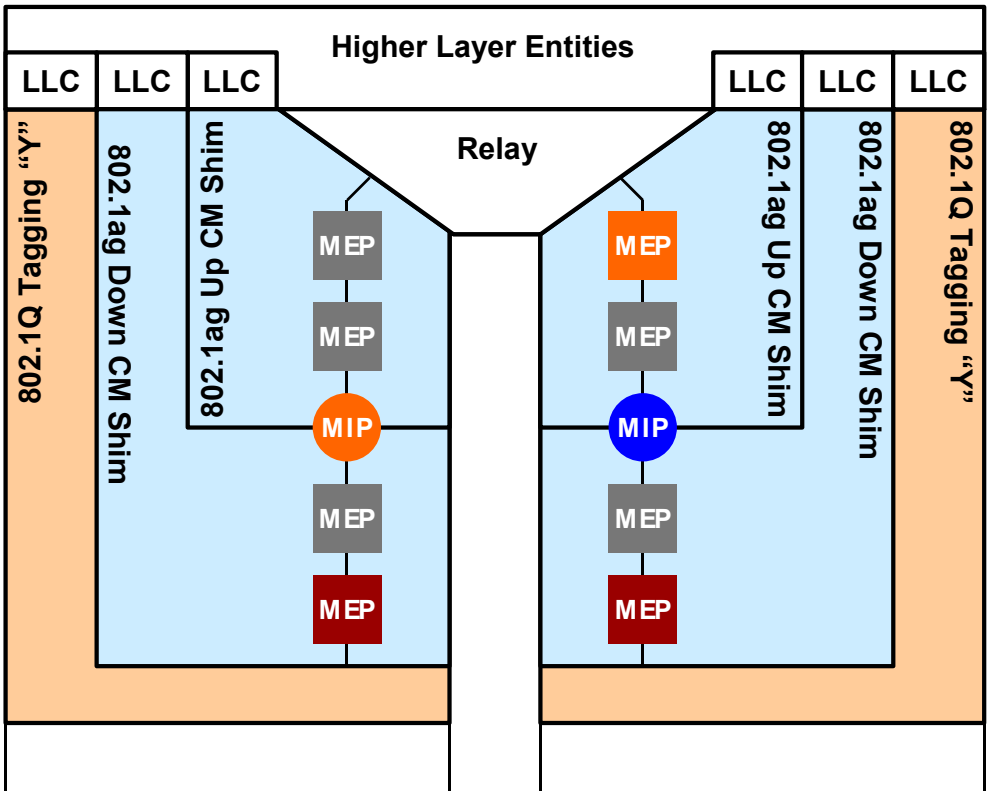


Figure II-5 – Illustrating the location of MEPs in the "baggy pants" model for B4

- operator A bridge number 4 has two ETH link related MEPs active, one in each interface port
- furthermore the operator A domain facing interface port (left) has its MIP active for use in the opertor A ETH ME level
- the operator B facing interface port (right) terminates operator A's ETH ME and has for that purpose a MEP in the up shim active
- as this right interface port is at a domain boundary, it has to support a MIP for the next higher ETH ME (service provider), to allow fault localization by this service provider (inside network of operator A, inside network of operator B or in the link between A and B)

The reader is assumed to be able to draw the MEP/MIP configurations for the other bridges at this point. Those are not shown therefore.

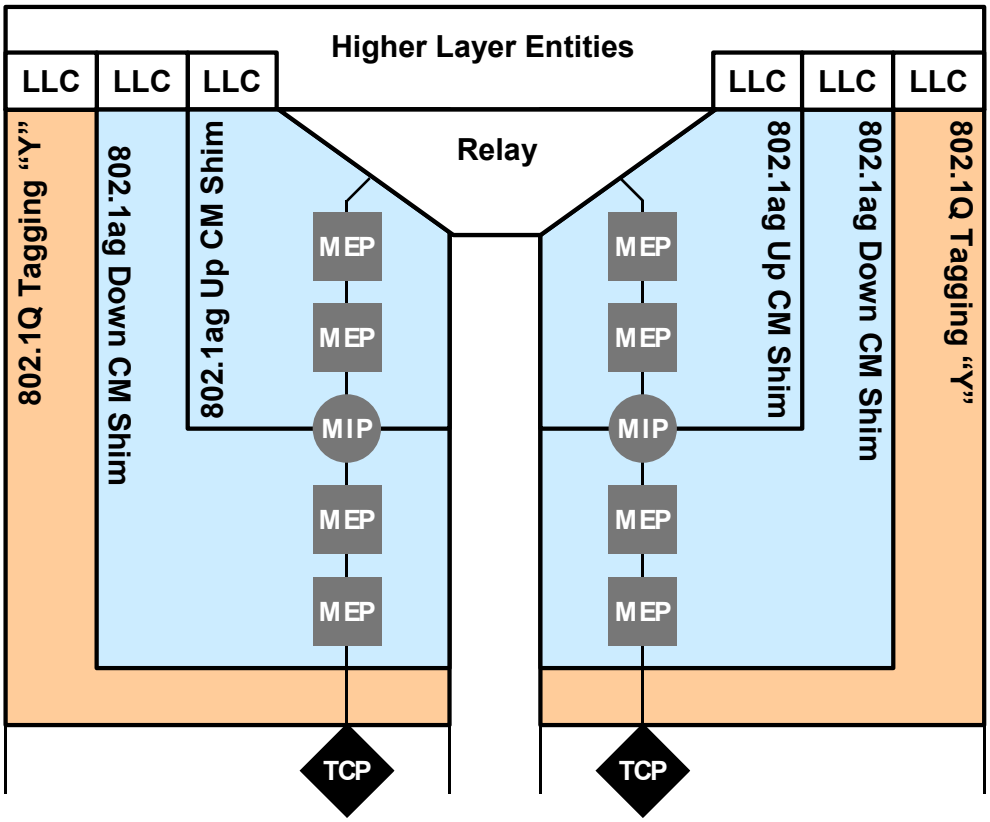


Figure II-6 – Illustrating the location of TCPs in the "baggy pants" model

- the interface port at the network side of a UNI will/may have a TCP that is located below the Down CM Shim in the baggy pants model. In this way the MEPs in the down shim will be able to register the effect (discarding) of the traffic conditioning and report this to the customer and service provider who share the responsibility for this UNI link

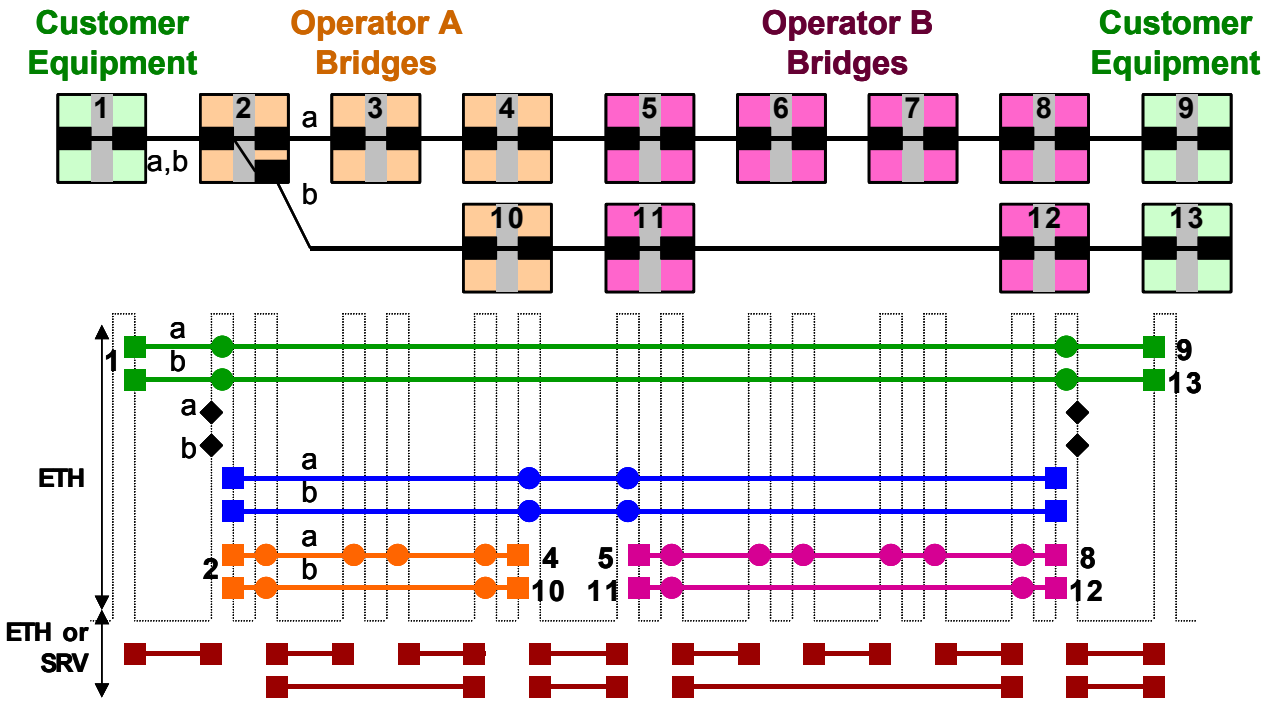


Figure II-7 – Illustrating the location of MEPs, MIPs and TCPs in the "baggy pants" model for the case of a two p2p connections with multiplexed access

- two p2p connections (a,b) with associated CM are depicted
- bridge 2 has two parallel sets of MEPs/MIP/TCP in the UNI facing port
- note: the figure depicts a single ME between CE1 and B2. This implies that this ME is a SRV ME. If it would have been an ME at ETH layer, then there should have been two ETH MEs, one for each p2p connection

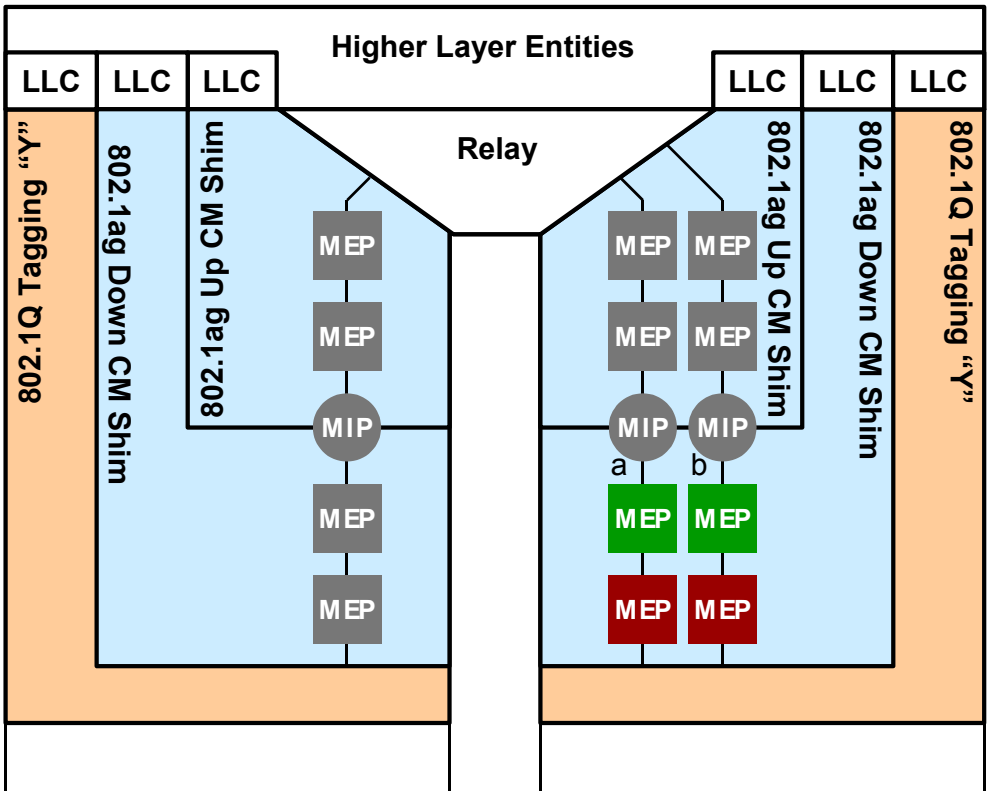


Figure II-8 – Illustrating the location of MEPs, MIPs and TCPs in the "baggy pants" model for the case of a two p2p connections with multiplexed access

- customer equipment number 1 has for the two ETH connections of figure II-7 two MEPs activated in the down shim to monitor the two UNI-C to UNI-C connections and the CE1 to B2 links

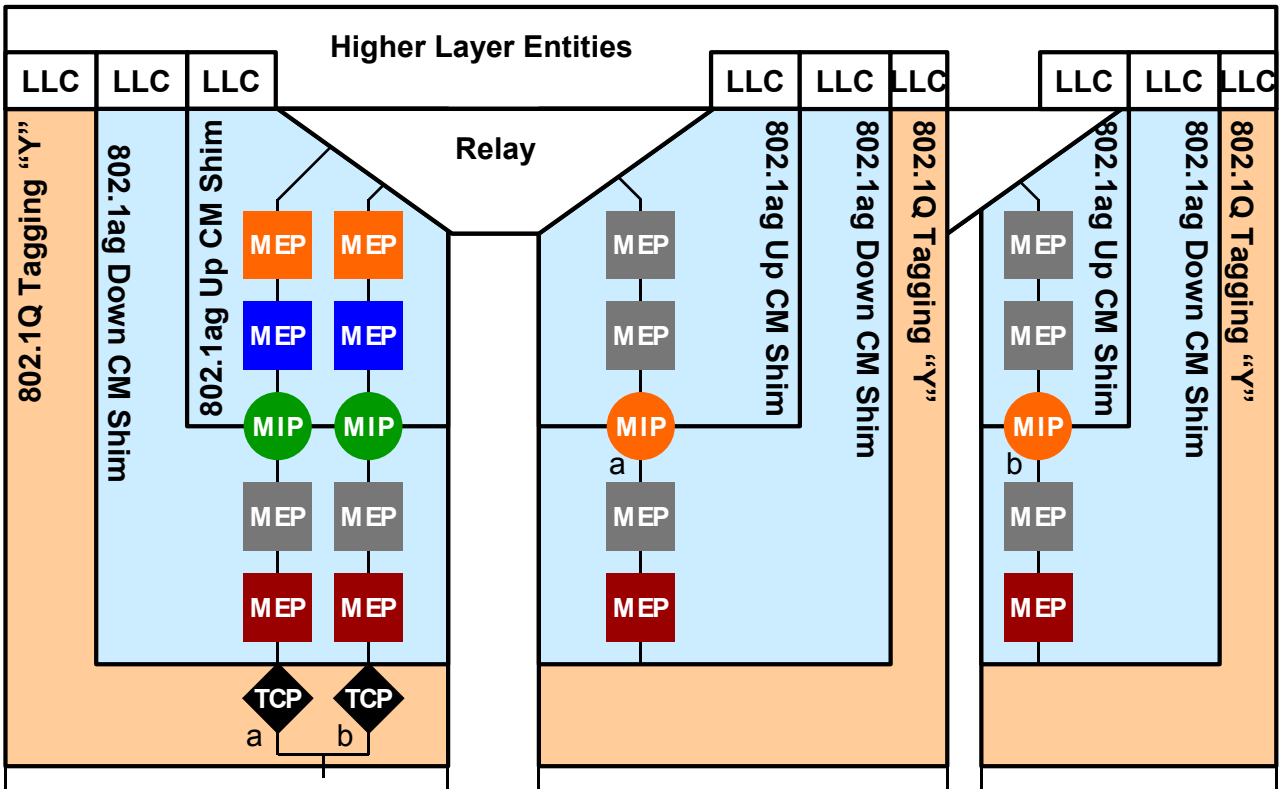


Figure II-9– Illustrating the location of MEPs, MIPs and TCPs in the "baggy pants" model for the case of a two p2p connections with multiplexed access

- operator A bridge number 2 has at its customer equipment facing port for each of the two p2p connections three MEPs active; one to terminate the link ME, one to terminate the service provider's ETH ME (blue) and one to terminate the operator A's ETH ME
- this port has also an active MIPfor each of the two p2p connections for use by the customer's UNI-C to UNI-C ETH MEs
- there are two interface ports facing the network, one for each of the two p2p connections

MEP, MIP, TCP for Dual Relay Model & Bundling MEP, MIP, TCP for Dual Relay Model & Bundling

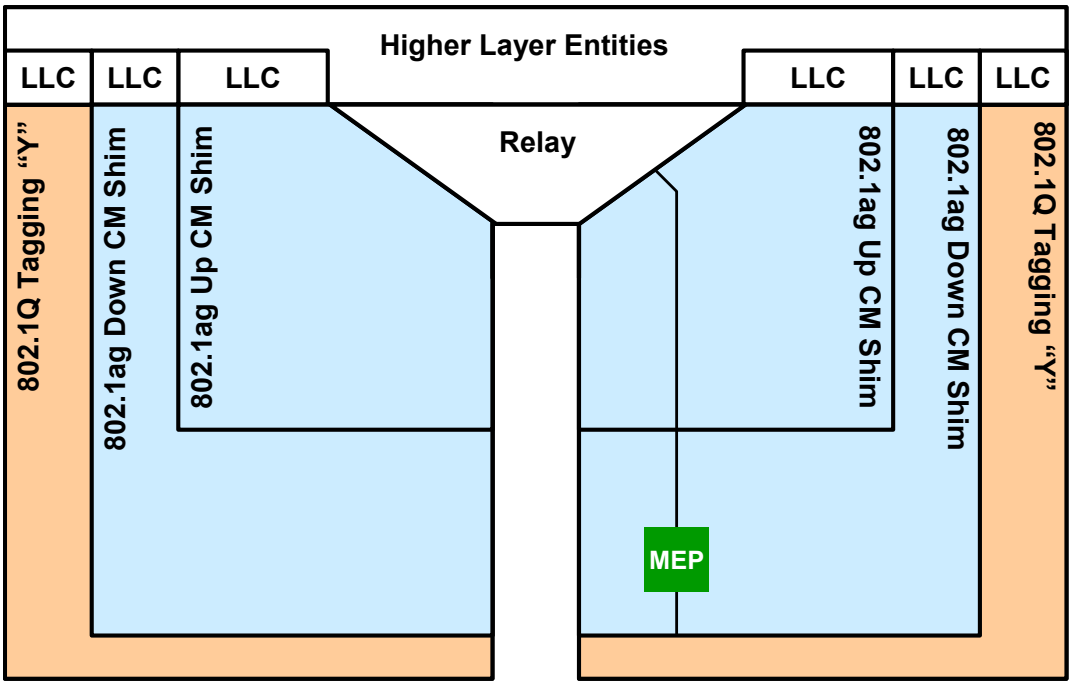


Figure II-10: Customer Bridge 1, example without ETH link ME

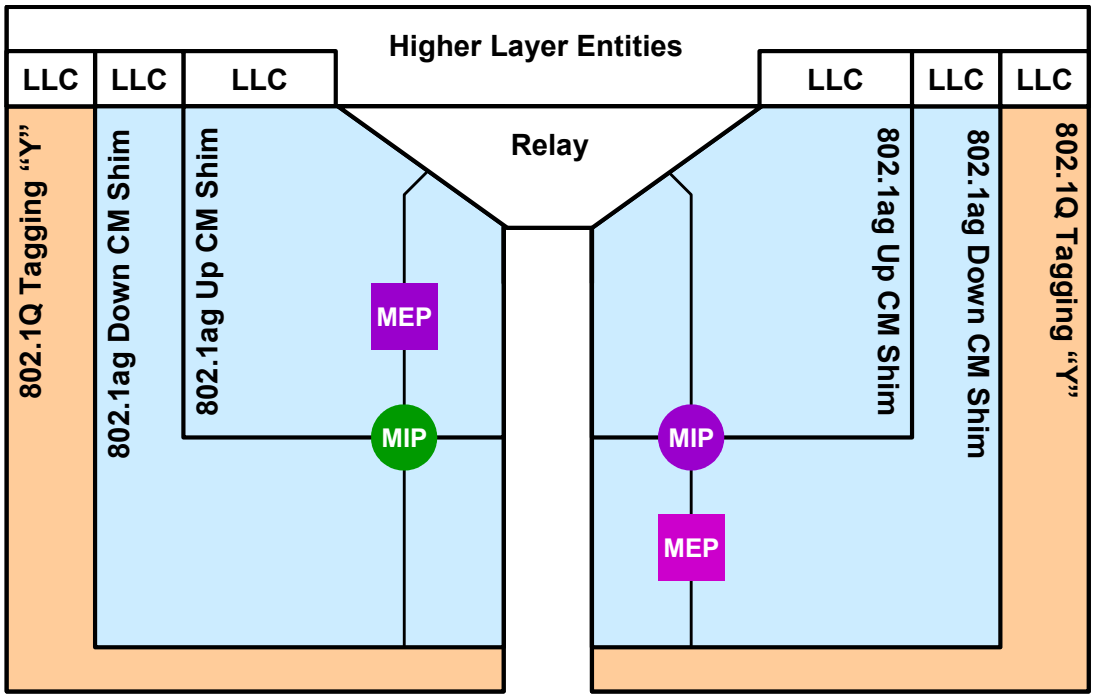


Figure II-11: Provider Bridge 2a
Example without ETH link ME

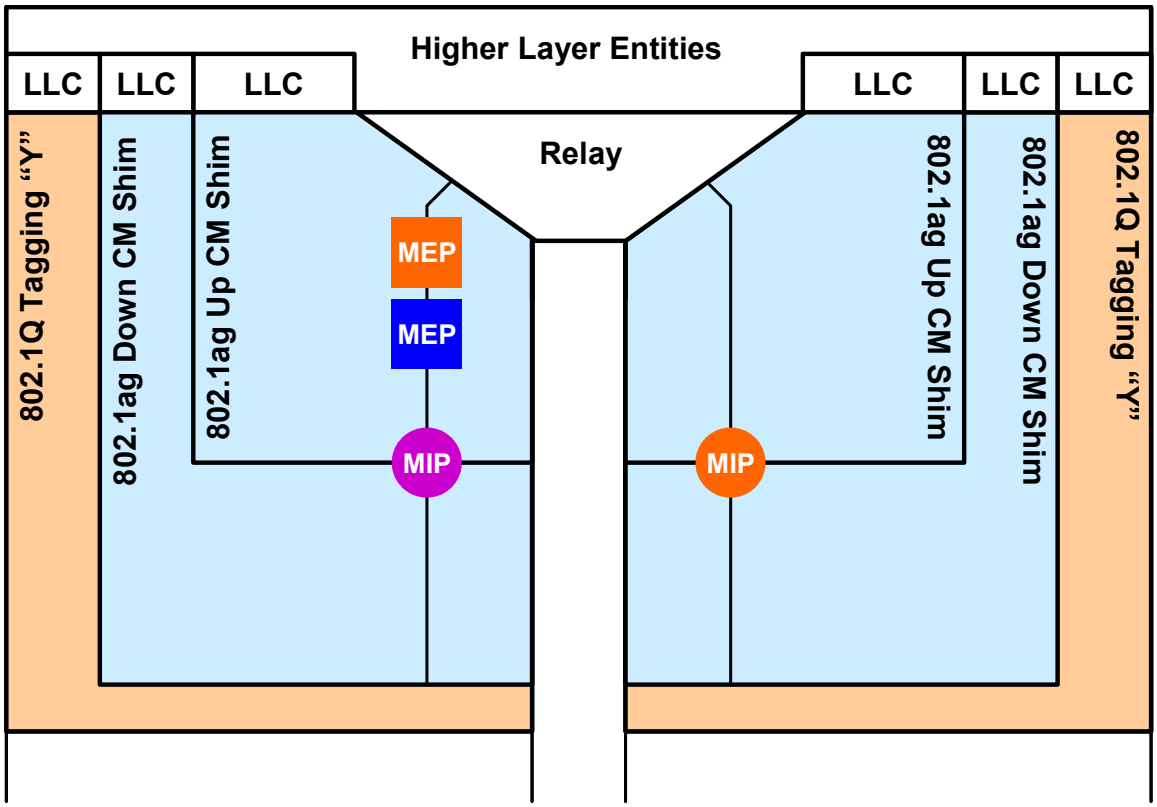


Figure II-12: Provider Bridge 2b, example without ETH link ME

Dual Relay Model with Single Relay as Provider Device

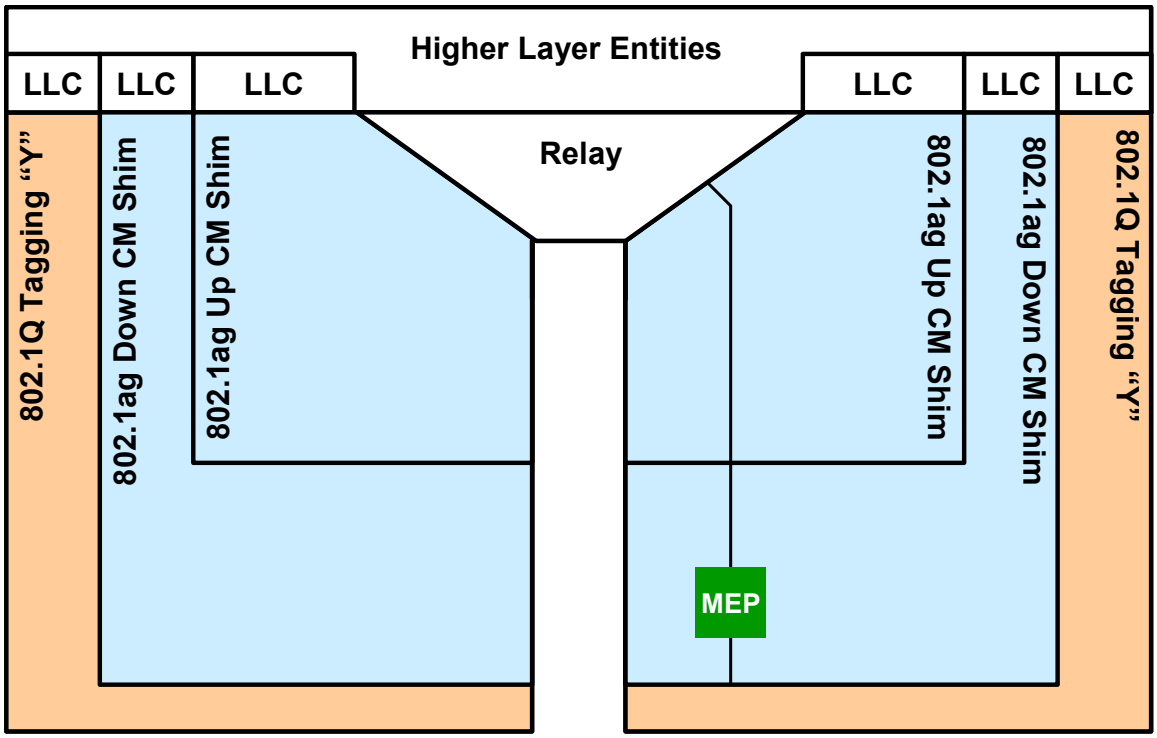


Figure II-13: Customer Bridge 1,
example without ETH link ME

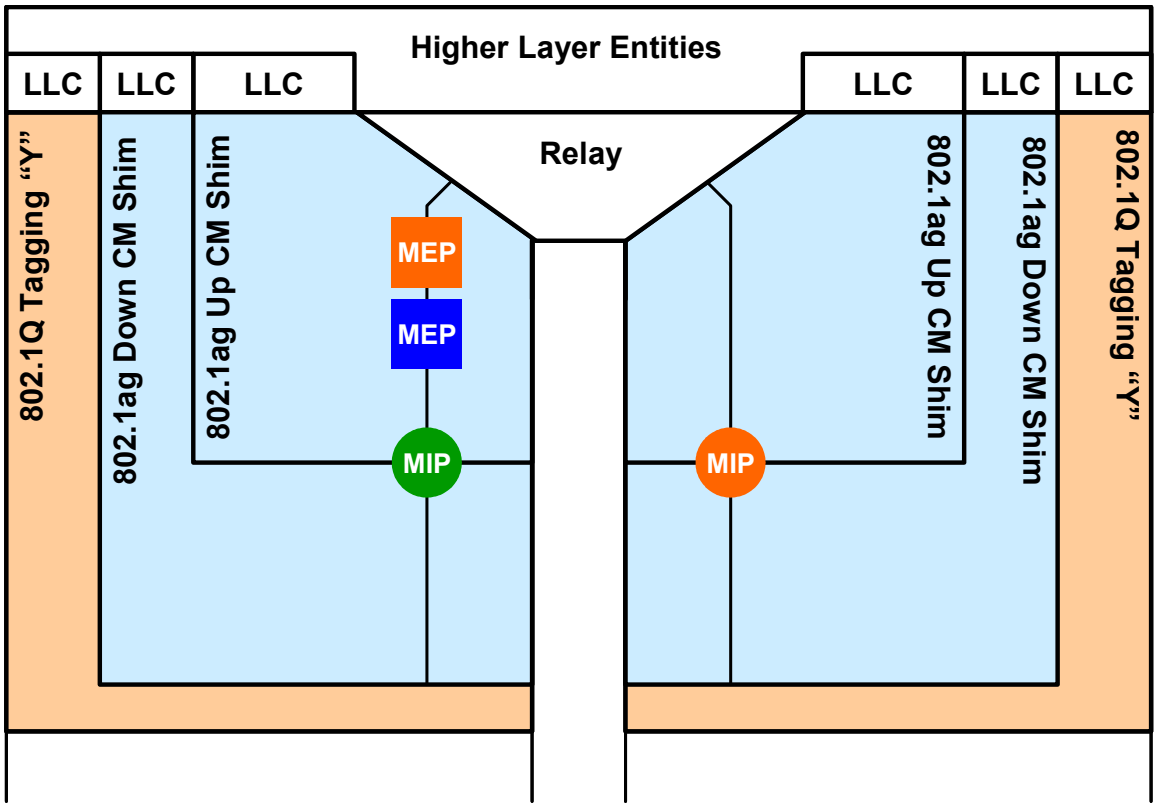


Figure II-14: Provider Bridge 2b, example without ETH link ME

Dual Relay Model with bundling for Single Integrated Provider Device

Appendix V

EDITOR'S NOTE: THIS APPENDIX WAS ADDED AT INTERIM MEETING IN NOVEMBER 2003 AND SHOULD BE REVIEWED TO SEE IF IT IS STILL APPLICABLE WAS TAKEN FROM WD 32 AS A BASIS FOR FURTHER DISCUSSIONS ON ETH ALARM SUPPRESION. THE NUMBER SHOULD BE APPENDIX III BECAUSE OTHER TWO WERE DELETED BUT IT WAS NOT DONE YET BECAUSE OF CROSS REFERENCES

ETH ALARM SUPPRESION OAM CONSIDERATIONS (ETH-AS CONSIDERATIONS)

WD27 introduces a multipoint ETH connection example in Figures 3 and 4/WD27. WD28 illustrates the ETH-AS insertion points and the ETH maintenance entities present on the ETH links. WD28 also introduces three alternatives to identify the maintenance entity level. Two of these alternatives (MELI ID, STID) are being used in this contribution to analyse the ETH-AS behaviour.

Figure 1 illustrates the maintenance entities present on some of the links in a multipoint ETH connection (see also WD28).

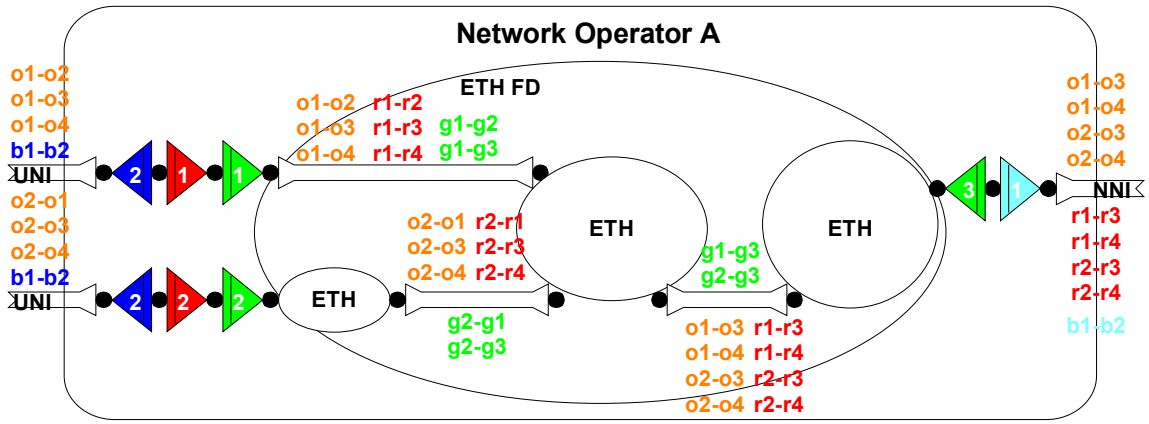


Figure V-5 – ETH maintenance entities on ETH links

ETH-AS when deploying MELI ID in ETH-CC

When deploying an ETH maintenance entity level instance ID (MELI ID) in ETH-CC OAM frames to identify the maintenance entity level the CC frame belongs to, this MELI ID information can be used at an ETH link end (and an ETH segment end) to learn the set of ETH maintenance entity levels passing through the ETH link and ETH segment. From the port identifier information present in the ETH-CC frames an ETH link end (and an ETH segment end) is able to learn the set of upstream ports that connect through the link or segment. Figure V-2 illustrates this learning at ETH link ends (Srv/ETH(-m)_A_Sk) and ETH segment ends (ETHS/ETH_A_Sk).

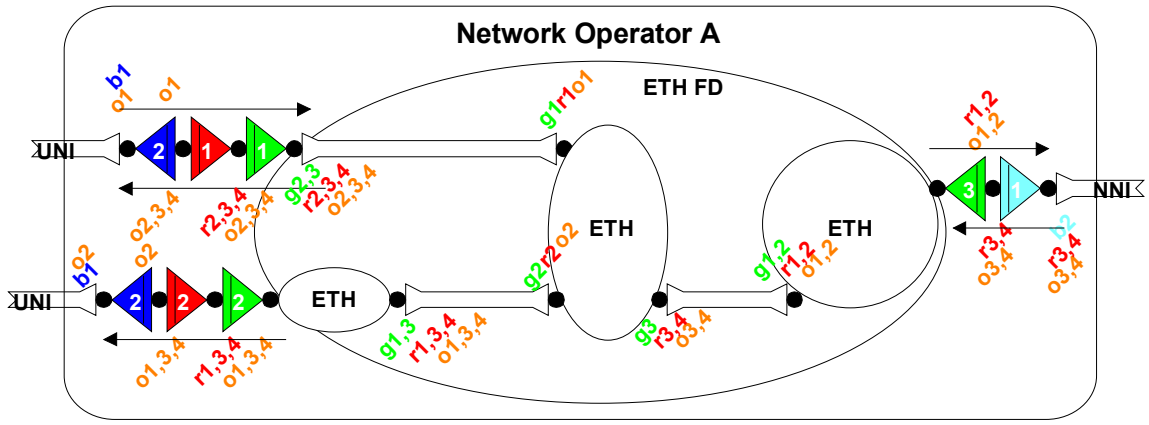


Figure V- 6 – ETH maintenance entity levels and upstream ports learned

Consider a fault occurring in an ETH link in one direction (Figure V-3), a set of maintenance entities (at multiple levels) is impacted. ETH-AS signal generation would in such case use the learned maintenance entity level and upstream port number information and generates ETH-AS frames per maintenance entity level instance, including the set of upstream port numbers.

For the case STP is present, an ETH link fault in a single direction will disable the use of the other direction of the link (for the traffic frames). At this point it is assumed that we will specify a kind of "Reverse Direction Link Down (RDLD)" maintenance signal[1] that runs between a Srv/ETH(-m)_A_So and a Srv/ETH(-m)_A_Sk function to inform the far end of the link that it is down. This signal should then result in ETH-AS signal generation at the far end of the link as well (aAIS = aSSF or dRDLD).

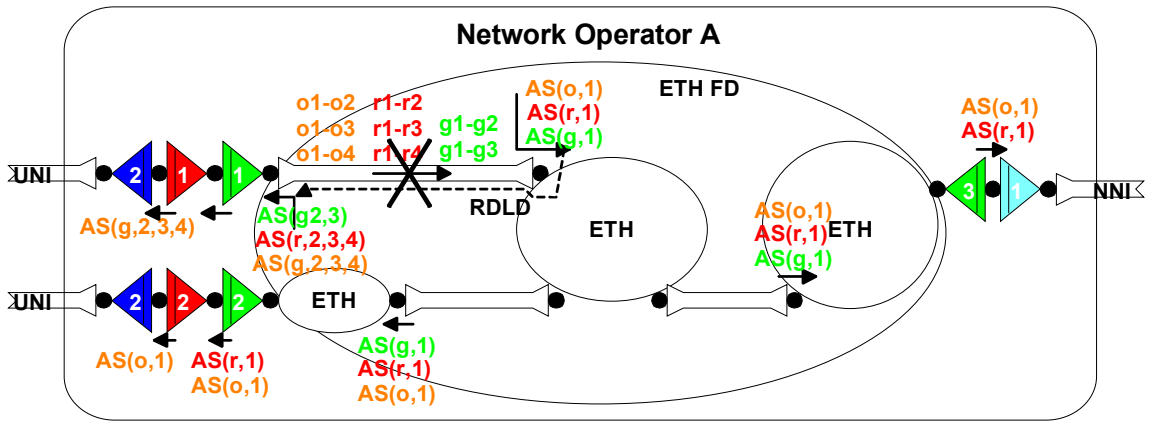


Figure V-7 – ETH-AS insertion example I

[1] As a first approximation (and perhaps already sufficient), RDI/BDI signals can be used as RDLD signal. The parameter controlling the port state MAC_Operational = CI_SSF or dRDLD. As a first and perhaps sufficient approximation MAC_Operational = AI_TSF or dRDI (from e.g. Sn_TT_Sk or Sn-X-L_TT_Sk).

The different ETH-AS signals are forwarded[2] by the ETH flow domains and each ETHS_FT_Sk function extracts the ETH-AS signals of its maintenance entity level and processes the included information (upstream port numbers that are disconnected due to fault). It will use this information to suppress the associated loss of continuity fault causes that will be detected as a consequence of the link fault.

The ETH-AS signals for other maintenance entity levels are simply passed through these ETHS_TT_Sk functions.

Figure 4 present a second example with a bi-directional ETH link fault. Figure V-5 assumes an alternative link being available in the topology, which is initially blocked by spanning tree (or network management, or …). After ETH link fault is detected e.g. STP will restore the ETH connection by taking the black link part of the active topology. At the same time it will block traffic (including ETH-AS OAM) incoming to the ETH-FDs at the end of the failed link. A blocked port will have to flush their learned set of maintenance entity level instances and upstream port numbers.

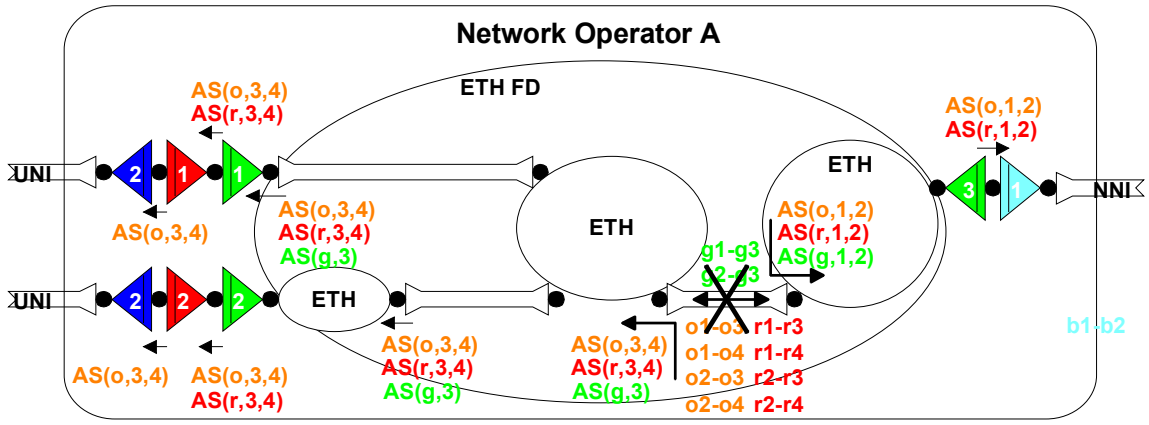


Figure V-8 – ETH-AS insertion example II

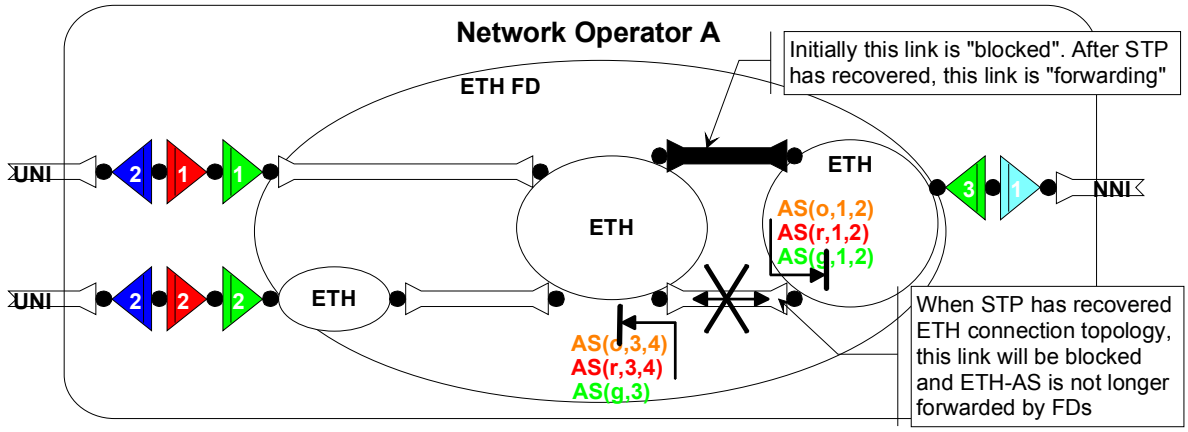


Figure V- 9 – ETH-AS insertion example II with restoration capability

---

[2] On a link fault, the port state changes as far as I understand… will this have any impact on the forwarding of these generated and inserted ETH-AS signals?

ISSUE: what if the topology only can be partially recovered…

NOTE – if instead of bridges an MPLS (VPLS) network would be used that would run Y.1711 OAM, there would be a look alike, feel alike management behaviour; the ETH maintenance entities are now replaced by MPLS maintenance entities…

ETH-AS when deploying STID in ETH-CC

Figure IV-6 illustrates the port identifiers of the maintenance entity at the top of the stack within a Srv/ETH adaptation sink function (link end) or ETHS/ETH adaptation sink function (segment end) in a multipoint ETH connection. Much less learning is required in this situation, and that is what is attractive… it also has a price…

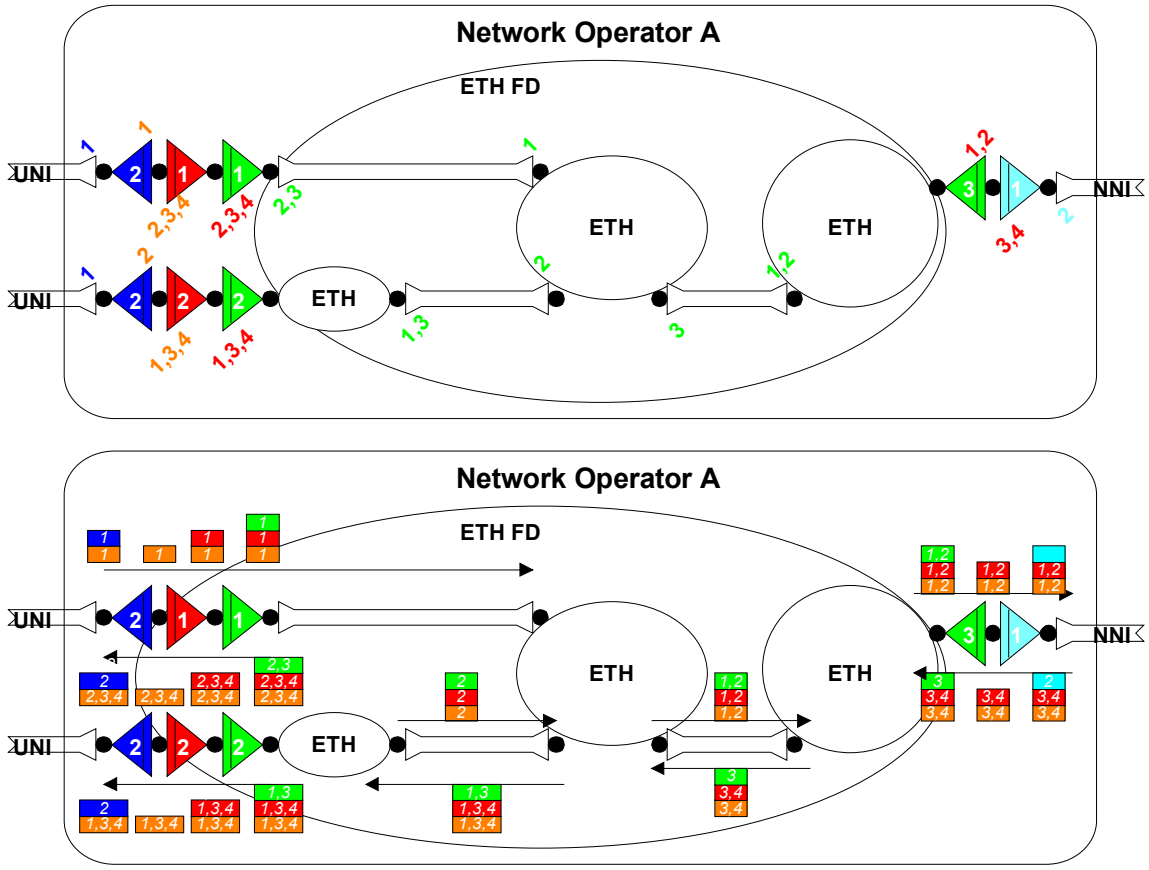


Figure V-10 – ETH maintenance entity port identifiers at the top of the stack (top) and full stack (bottom)

A link fault (Figure V-7) will now generate a single ETH-AS frame with upstream port numbers from the ETH link ends for the top level maintenance entity. Then at the first segment endpoints (green) these ETH-AS signals are extracted and processed. The signal fail status is forwarded to the adaptation sink function in the segment endpoint, where it has to trigger insertion of ETH-AS for the interrupted top level (red) maintenance entity. Unfortunately there is insufficient information at these points to generate ETH-AS frames with specific upstream port number list.

So, should we generate non-specific ETH-AS frames (then also at link ends)? The consequence is that it also will suppress the reporting of a true ETH layer continuity or connectivity fault located

elsewhere in the ETH connection… should our ETH OAM be able to detect and report a dual fault condition?

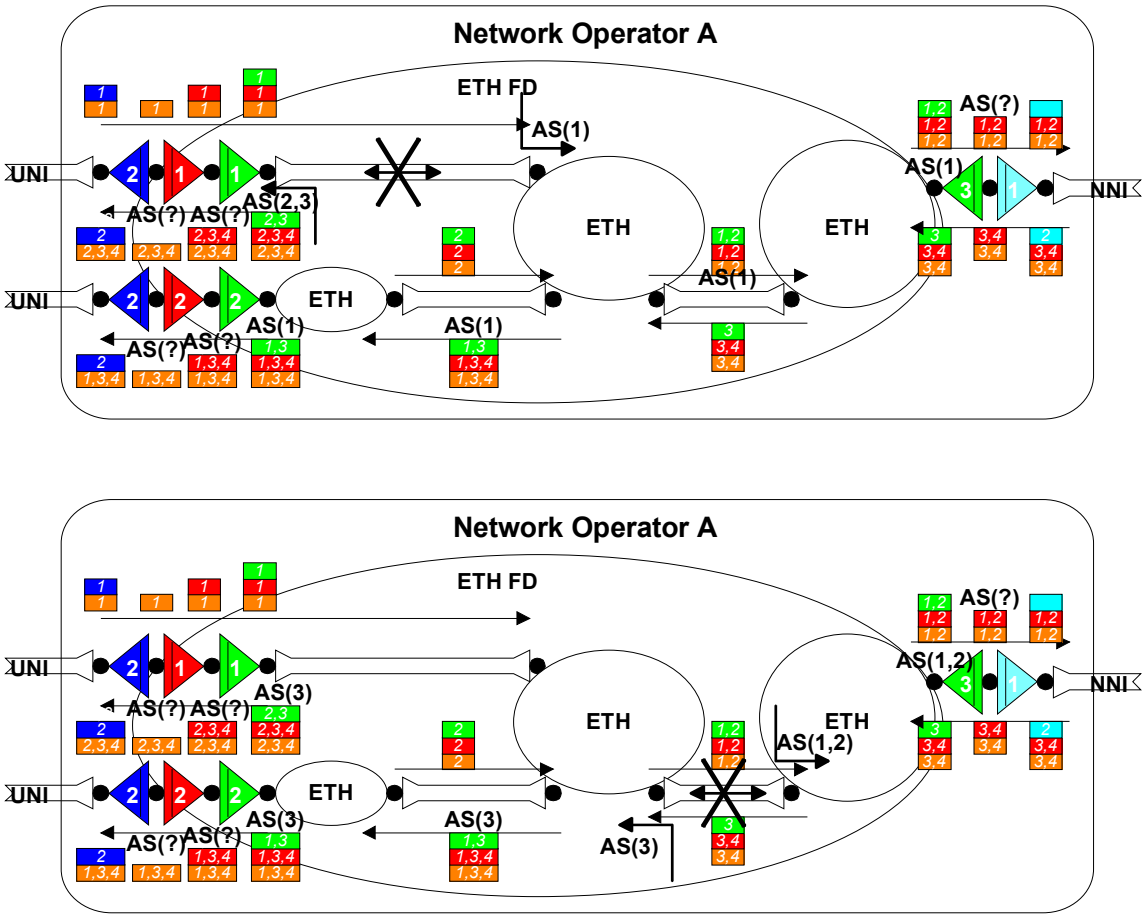


Figure V-11

Appendix VI

Some existing Management Objects(MOs) that can be used for the performance management mechanisms mentioned in Section 8 include:

- **IEEE 802.3-2002**
  - – aFramesTransmittedOK [5 – section 5.2.2.1.2]
  - – aFramesReceivedOK [5 - 5.2.2.1.5]

- **IEEE 802.1Q-2003**
  - –Frames Received [6 - 12.6.1.1.3]
  - –Frames Outbound [6 - 12.6.1.1.3]

- **RFC 3635 - Ethernet-like interface MIB (Obsoletes 2665)**
  - –IF-MIB
    - ifOutUCastPkts
    - ifOutMulticastPkts
    - ifOutBroadcastPkts
    - ifOutErrors
    - ifOutDiscards
    - ifInUCastPkts
    - ifInMulticastPkts
    - ifInBroadcastPkts
    - ifInErrors
    - ifInDiscards
  - –aFramesTransmittedOK = ifOutUCastPkts + ifOutMulticastPkts + ifOutBroadcastPkts – (ifOutErrors + ifOutDiscards)
  - –aFramesReceivedOK = ifInUCastPkts + ifInMulticastPkts + ifInBroadcastPkts + (ifInErrors + ifInDiscards)

- **RFC 2674 – VLAN Bridge MIB**
  - –dot1qPortVlanStatisticsTable
    - dot1qTpVlanPortInFrames
    - dot1qTpVlanPortOutFrames

Note: It may be noted that these managed objects values eventually wrap. This can lead to inaccurate results when such an event occurs. However, if the time interval of observation is small, the inaccuracy can be avoided. Averaging of the results over the period of observation can alleviate the in flight frames issue.

_____