INTERNATIONAL TELECOMMUNICATION UNION

# ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

# G.984.4

**Amendment 2**
(11/2009)

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

Digital sections and digital line system – Optical line systems for local and access networks

---

# Gigabit-capable Passive Optical Networks (GPON): ONT Management and Control Interface specification

## Amendment 2:

***CAUTION !***

***PREPUBLISHED RECOMMENDATION***

This prepublication is an unedited version of a recently approved Recommendation. It will be replaced by the published version after editing. Therefore, there will be differences between this prepublication and the published version.

## FOREWORD

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

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

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

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

## NOTE

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

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

## INTELLECTUAL PROPERTY RIGHTS

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

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

**Editor's note to TSB**: Paragraph styles *attribute*, *attribute follower*, *attribute list* are intended to improve readability of this amendment and should be preserved in final publication if possible. Re-formatting descriptions into tables is not preferred.

**Recommendation ITU-T G.984.4 (2008) Amendment 2**

**Gigabit-capable Passive Optical Networks (GPON):**
**ONT Management and Control Interface specification**

## 1 Summary

This amendment contains various updates to G.984.4 (2008) and G.984.4 (2008) amendment 1 (2009). A number of editorial corrections and clarifications are included, along with the following substantive changes and extensions to GPON OMCI.

—A virtual interface definition for use when an ONT is managed by two domains, OMCI and for example TR-69.

—An extended form of the OMCI message set to facilitate exchange of larger quantities of control plane information between ONT and OLT.

—An ageing time attribute on dynamically learned MAC bridge addresses.

—A means to distribute time of day from an OLT to an ONT

—Additional features to support the use of OMCI in G.986 systems

—A mechanism that supports mutual authentication of OLT and ONT and secure transport of encryption keys.

## 2 References

*Add the following reference:*

[IEEE 1588-2008] IEEE 1588 (2008), Standard for a precision clock synchronization protocol for networked measurement and control systems

[ITU-T G.986] Recommendation ITU-T G.986 (2009), *1 Gbit/s point-to-point Ethernet based optical access system*

Note to TSB: Please confirm correct name and date for this reference

## 7.3 Performance management

*Revise this clause to read as follows:*

The ONT has only *limited* performance monitoring. The OMCI supports performance monitoring using a subset of managed entities that are described throughout clause 9. These managed entities can be identified by the words "performance monitoring history data" or "extended PM" in their names.

### 7.4 Security management

*Where the text presently reads:*

[ITU-T G.984.3] specifies some mechanisms from the viewpoint of security. That includes the downstream data encryption of the ONT. The ONT2-G managed entity can enable/disable the downstream encryption function.

This Recommendation supports the protection function. The type C protection configuration that is defined in [b-ITU-T G.984.1] is considered in this Recommendation. As the switching behaviour for PON protection will be done in the TC layer, this Recommendation defines a managed entity to specify the protection capability.

*Revise it to read (clause 7.5 is new):*

### 7.4 Security management

[ITU-T G.984.3] specifies some mechanisms from the viewpoint of security. That includes the downstream data encryption of the ONT. The ONT2-G managed entity can select the downstream encryption algorithm from a list that presently includes only AES.

OMCI also supports a mechanism to allow mutual authentication of OLT and ONT and subseqent secure communication of encryption keys.

### 7.5 PON protection

This Recommendation supports the protection function. The type C protection configuration that is defined in [ITU-T G.984.1] is considered in this Recommendation. As the switching behaviour for PON protection will be done in the TC layer, this Recommendation defines a managed entity to specify the protection capability.

### 8.1 Managed entities

*Add the following entries to table 8-1 in alphabetic sequence:*

**Table 8-1. Managed entities of the G-OMCI**

| Managed entity | Required/ Optional | Description | Clause |
|---|---|---|---|
| Virtual Ethernet interface point | CR | Defines a data plane handoff to other management domains such as TR-69 or SNMP | 9.5.5 |
| ONT-E | CR | Defines the top-level ONT entity for G.986 systems | 9.1.13 |
| Enhanced security control | CR | Supports mutual authentication between OLT and ONT | 9.13.11 |
| MPLS pseudowire termination point | CR | Supports TDM pseudowires over MPLS | 9.8.14 |

*Revise the description of GEM traffic descriptor to read as follows, and relocate it into the proper alphabetic order in the table:*

**Table 8-1. Managed entities of the G-OMCI**

| Managed entity | Required/ Optional | Description | Clause |
|---|---|---|---|
| Traffic descriptor | CR | Used to specify traffic management parameters | 9.11.3 |

*Replace clause 8.2 with the following:*

## 8.2 Managed entity relation diagrams

This clause shows the relationships between managed entities. Figure 8.2-1 gives the legend of symbols used in these diagrams. The name of the managed entity, sometimes abbreviated for ease of documentation, appears in each box, with the clause in which it is defined shown in the lower right corner.

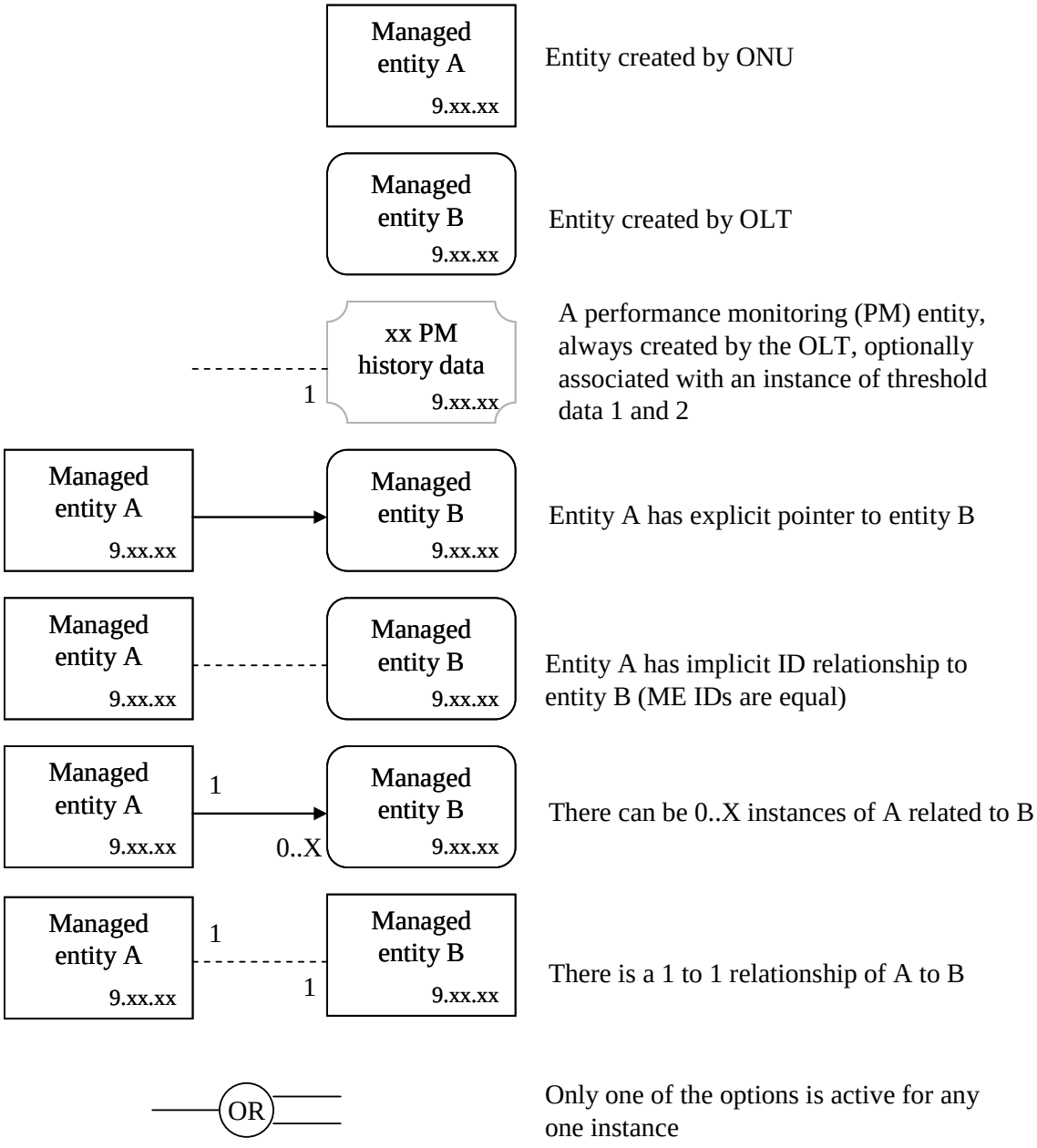



**Figure 8.2-1 – Legend for managed entity relation diagrams**

### 8.2.1 ONT common functions

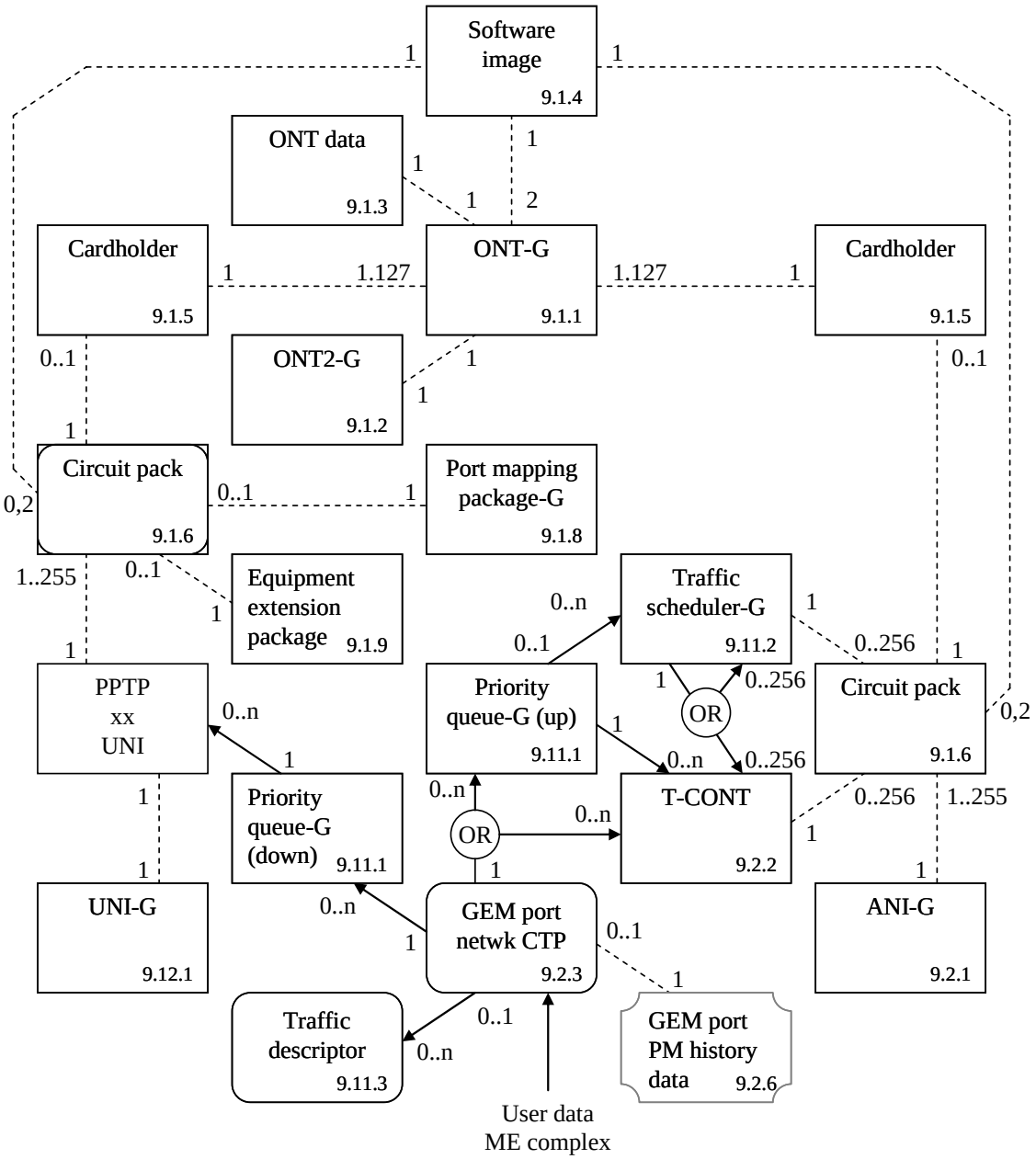



**Figure 8.2.1-1 – ONT core**

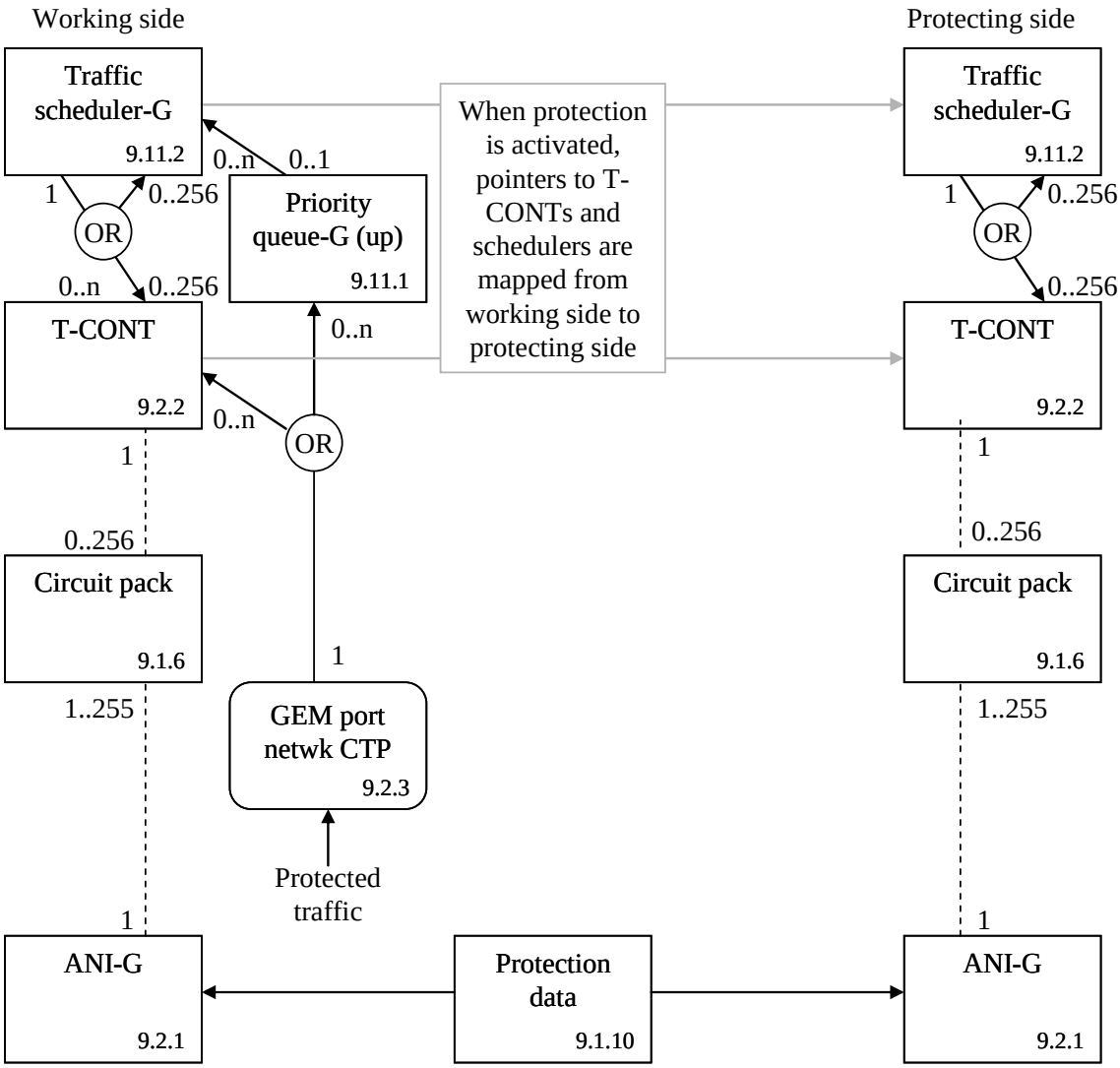



**Figure 8.2.1-2 – 1+1 PON protection**

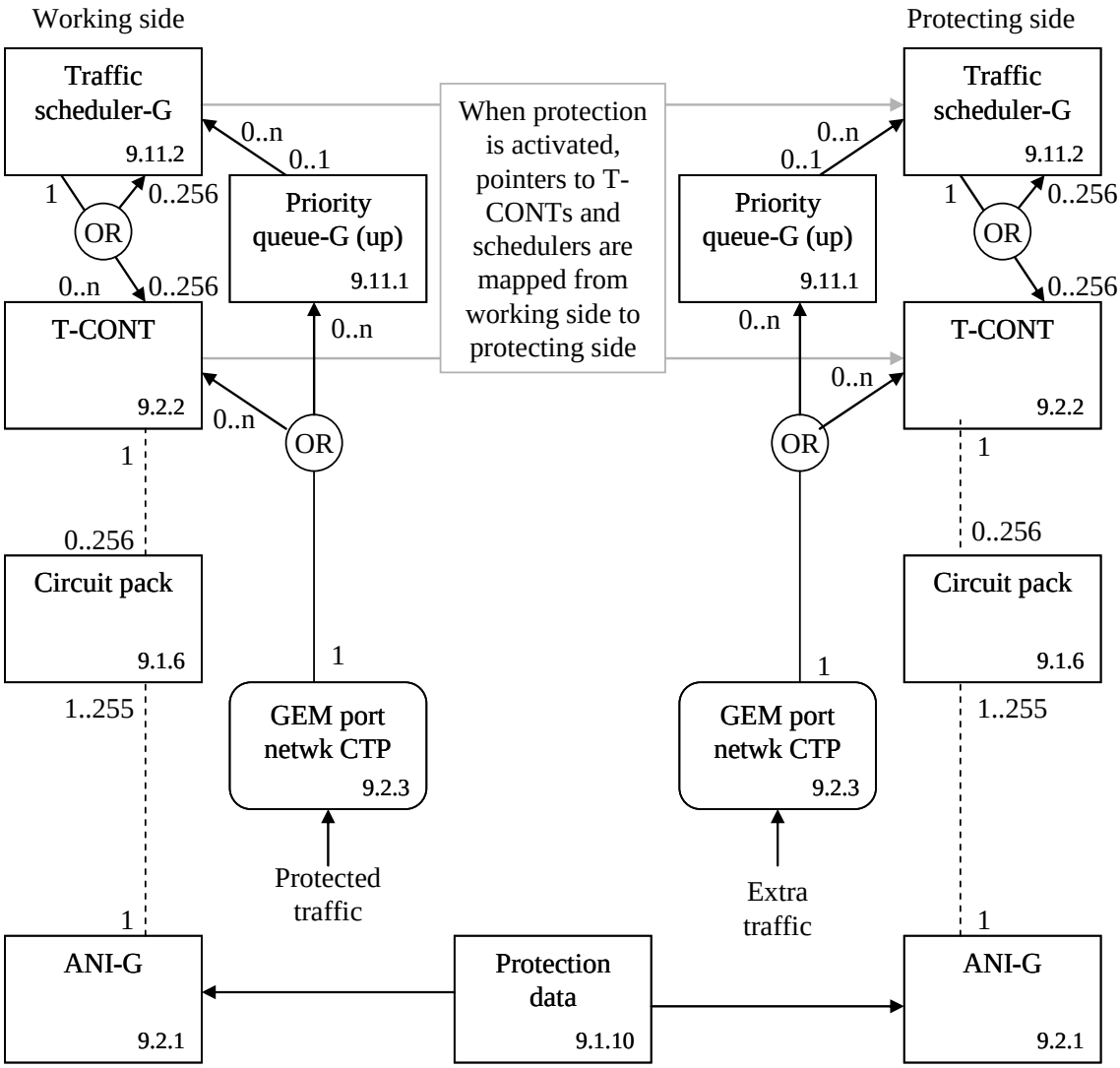



**Figure 8.2.1-3 – 1:1 PON protection**

### 8.2.2 Layer 2 functions

OMCI supports two major layer 2 traffic mapping models: MAC bridging and "802.1p mapping." MAC bridging is described in IEEE 802.1D and 802.1Q. The bridge described by figure 8.2.2-1 below has many features, and can be used to direct traffic based on MAC address (that is, true bridging) or on VLAN characteristics (using the VLAN filter feature). The mapping function describes the steering of traffic from one UNI-side entity to 1-8 ANI-side Port-IDs, as shown in figure 8.2.2-2 below. The mapper is equivalent to a MAC bridge with VLAN filters that only operate on the priority bits of the VLAN tags.

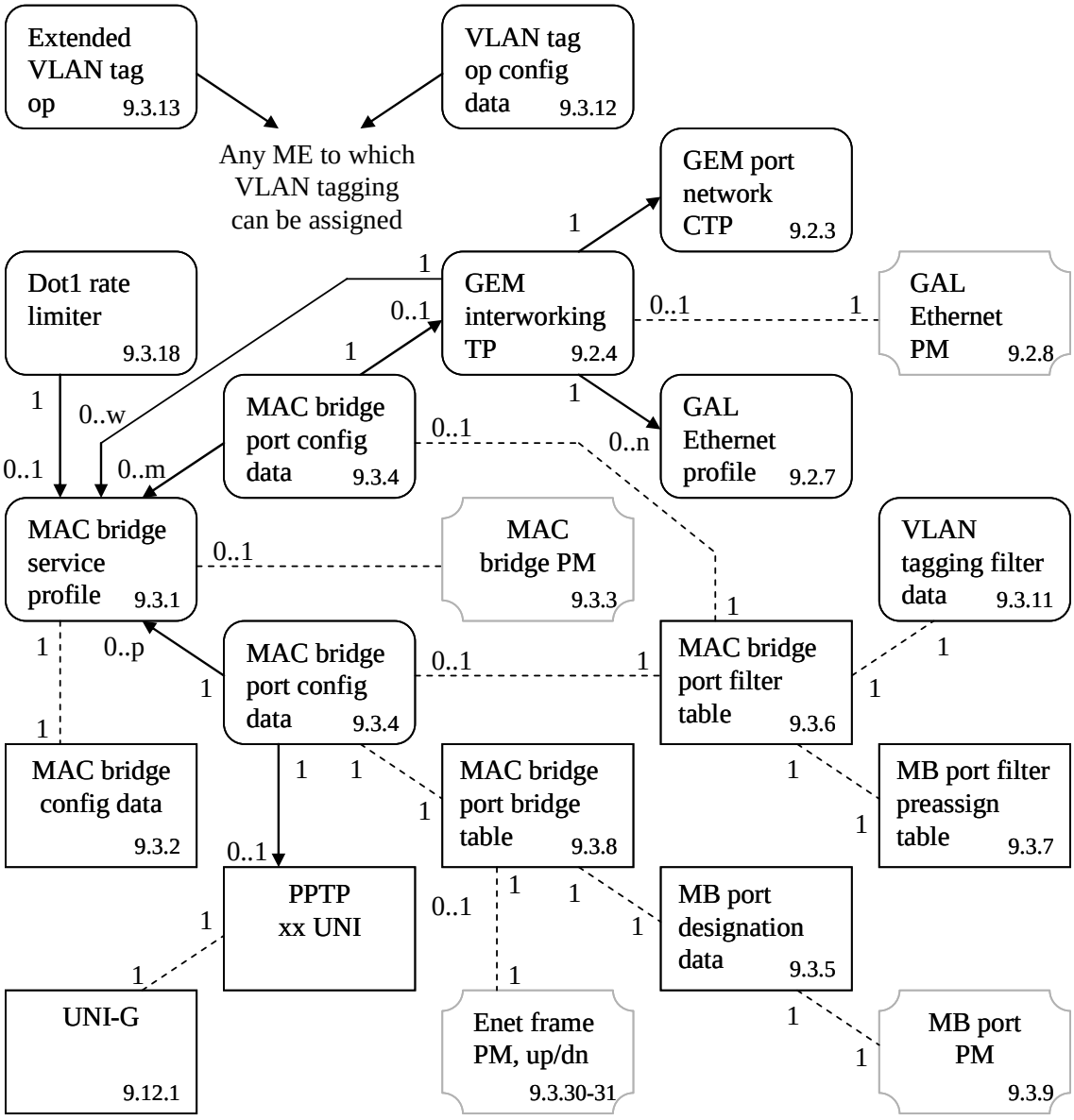



**Figure 8.2.2-1 – MAC bridged LAN**

NOTE – A bridge port may be associated with any IEEE 802.3 UNI, such as Ethernet or xDSL, or another 802.3 function such as an IP host config data ME.

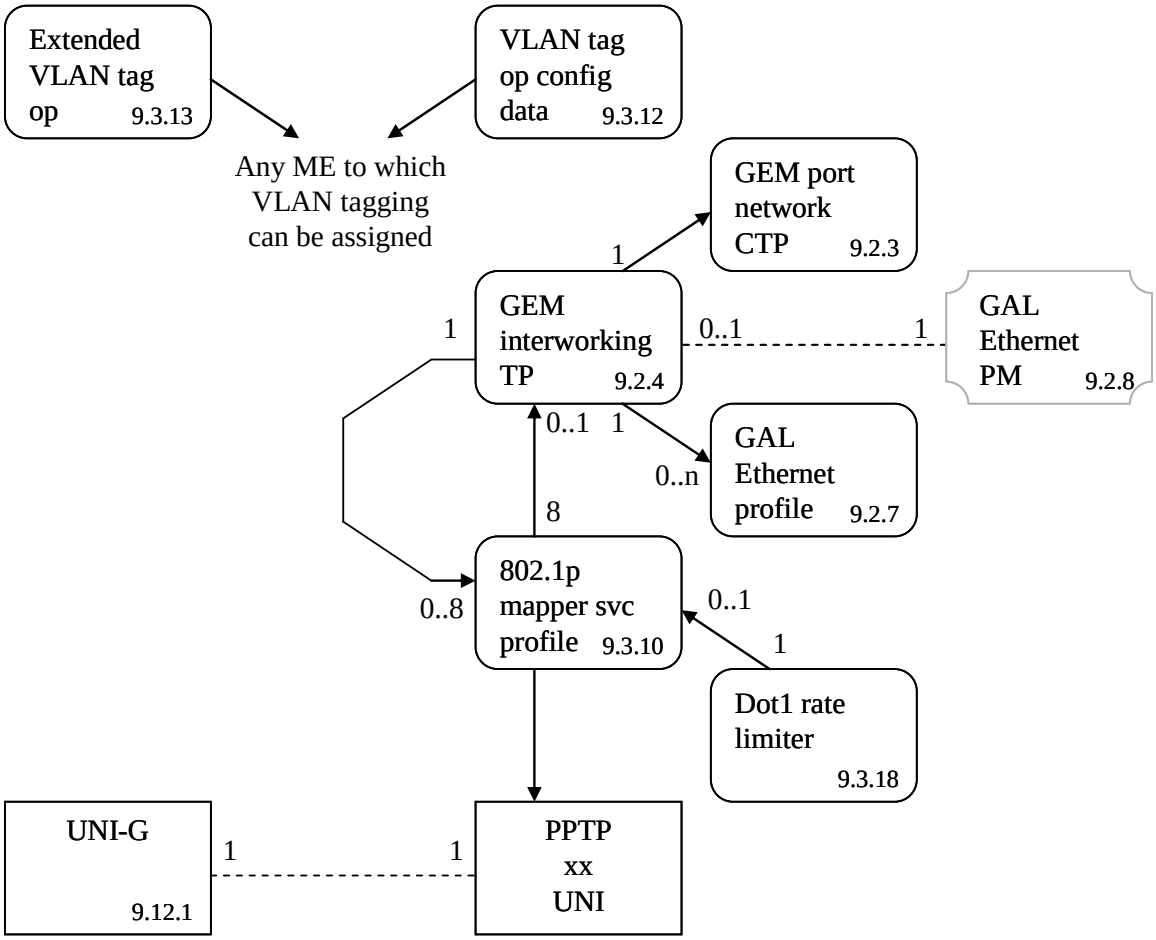



**Figure 8.2.2-2 – 802.1p mapper**

NOTE – A mapper service profile may be associated with any IEEE 802.3 UNI, such as Ethernet or xDSL, or another 802.3 function such as an IP host config data ME.

The two basic layer 2 services can be used in various combinations to achieve different overall connectivities. There are three major functional styles of layer 2 connectivity, illustrated in figures 8.2.2-3..5:

N:1 bridging, where a bridge is used to serve multiple UNI ports from a single ANI service

1:M mapping, where a mapper is used to serve a single UNI with multiple ANI connections, based on 802.1p priorities

1:P filtering, where a bridge with filters is used to serve a single UNI with multiple ANI connections, based on some VLAN information other than 802.1p priorities.

Given these three basic possibilities, there are also four more complex combinations as well, illustrated in figures 8.2.2-6..9. It is strongly encouraged that these applications be utilized before other, more exotic styles of usage.

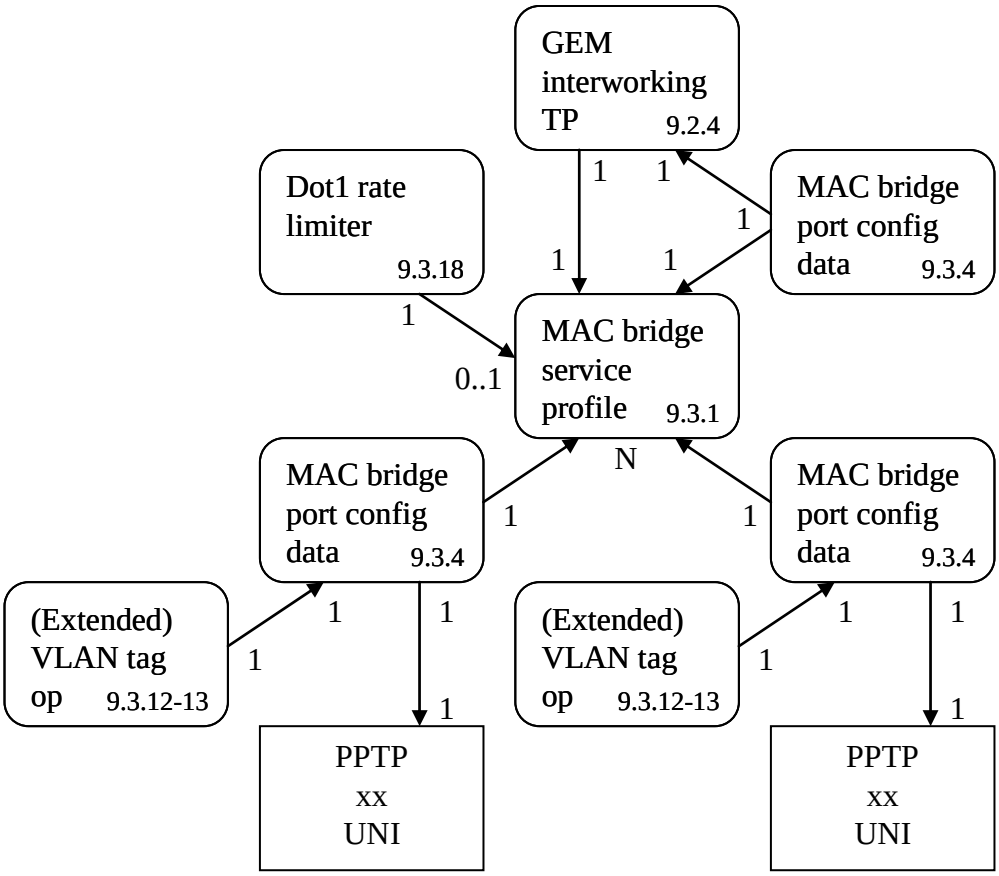



**Figure 8.2.2-3 – Illustration of N:1 bridging**

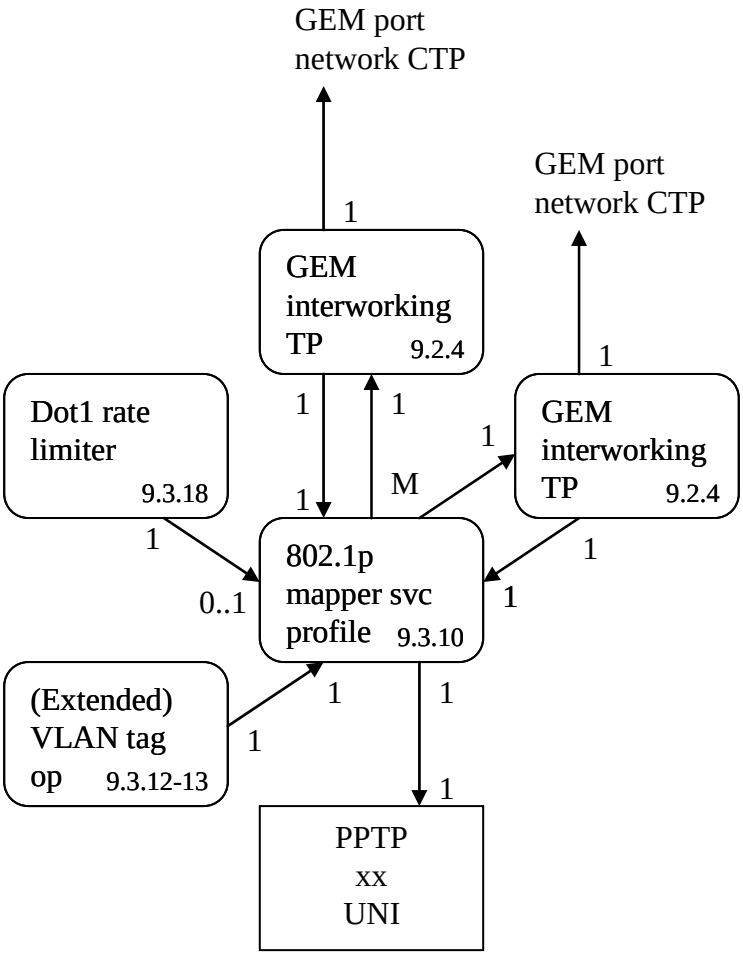



**Figure 8.2.2-4 – Illustration of 1:M mapping**

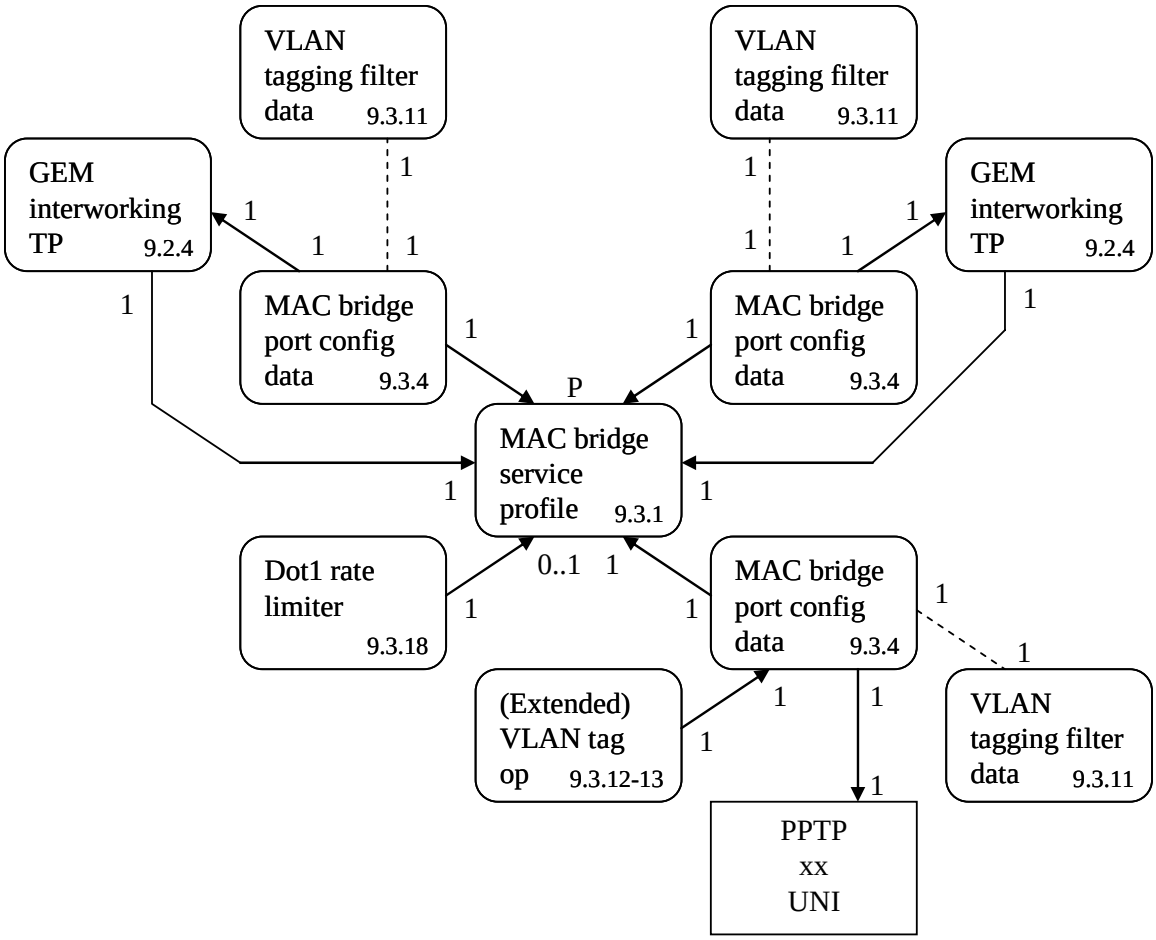



**Figure 8.2.2-5 – Illustration of 1:P filtering**

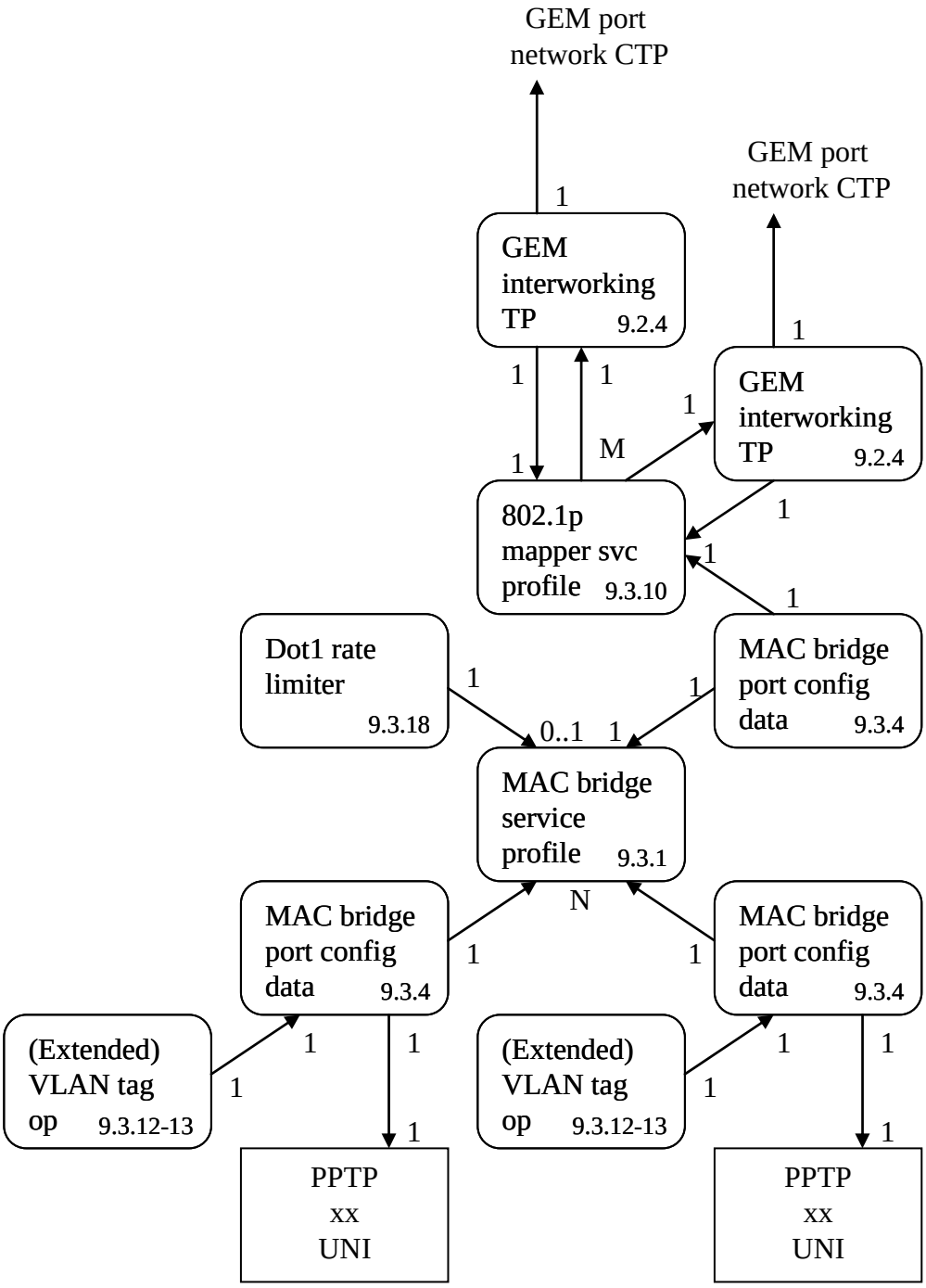



**Figure 8.2.2-6 – Illustration of N:M bridge-mapping**

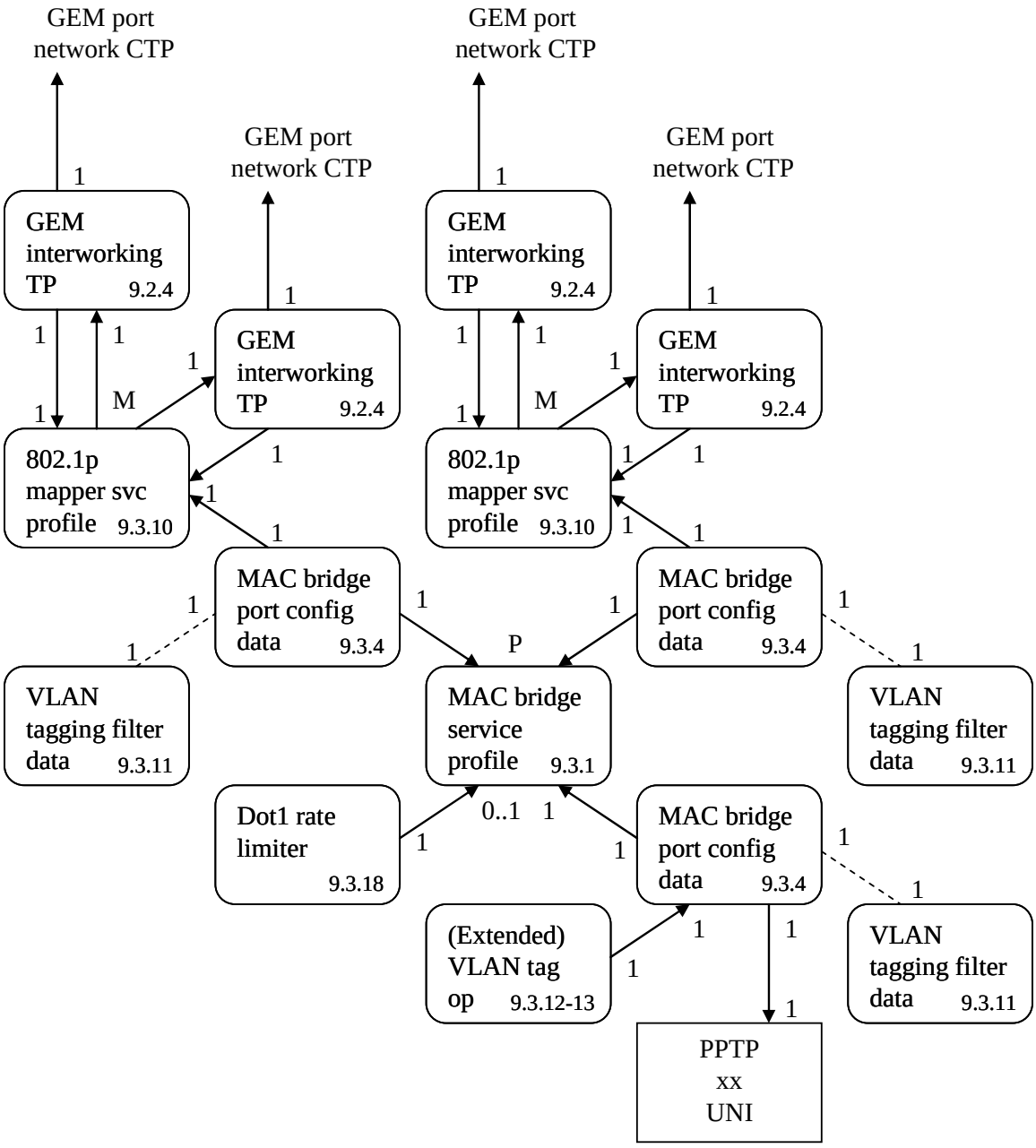



**Figure 8.2.2-7 – Illustration of 1:MP map-filtering**

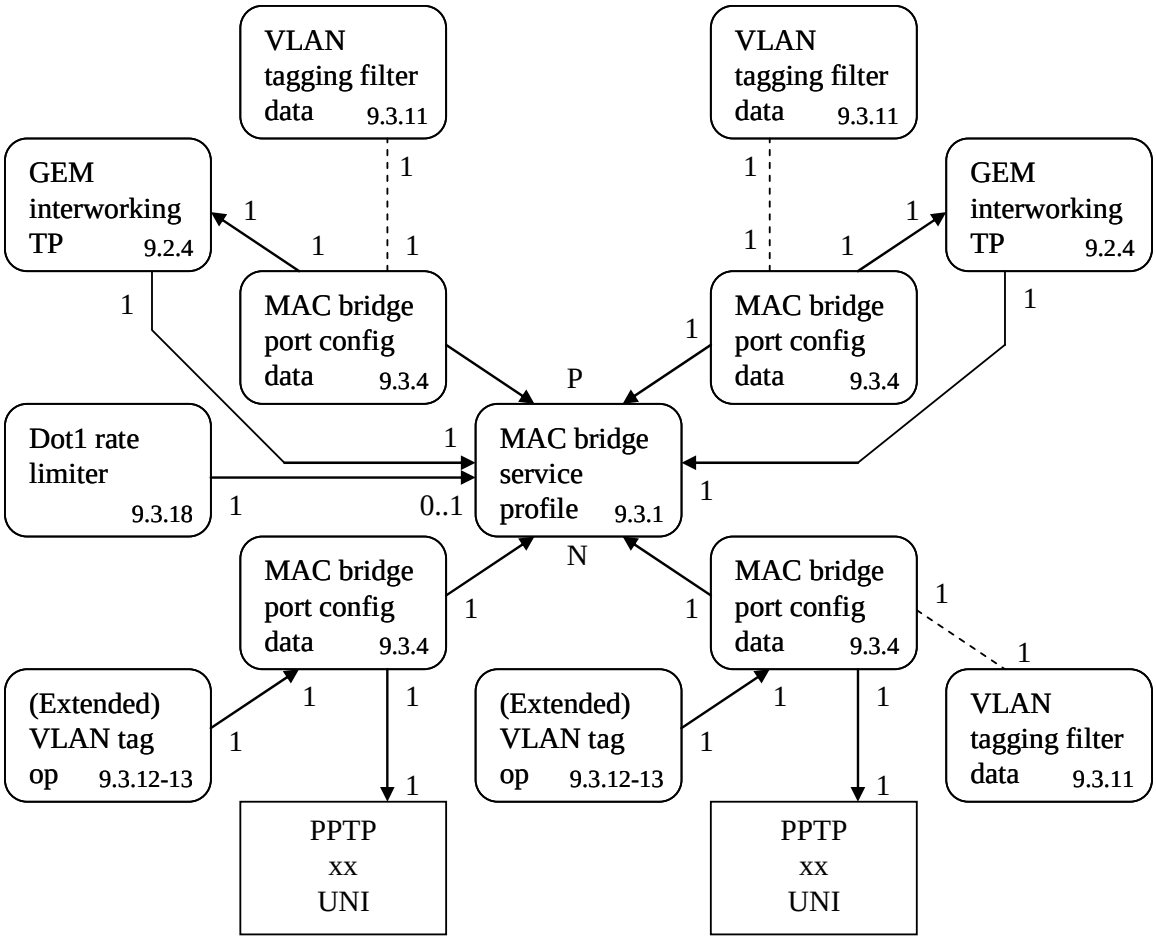



**Figure 8.2.2-8 – Illustration of N:P bridge-filtering**

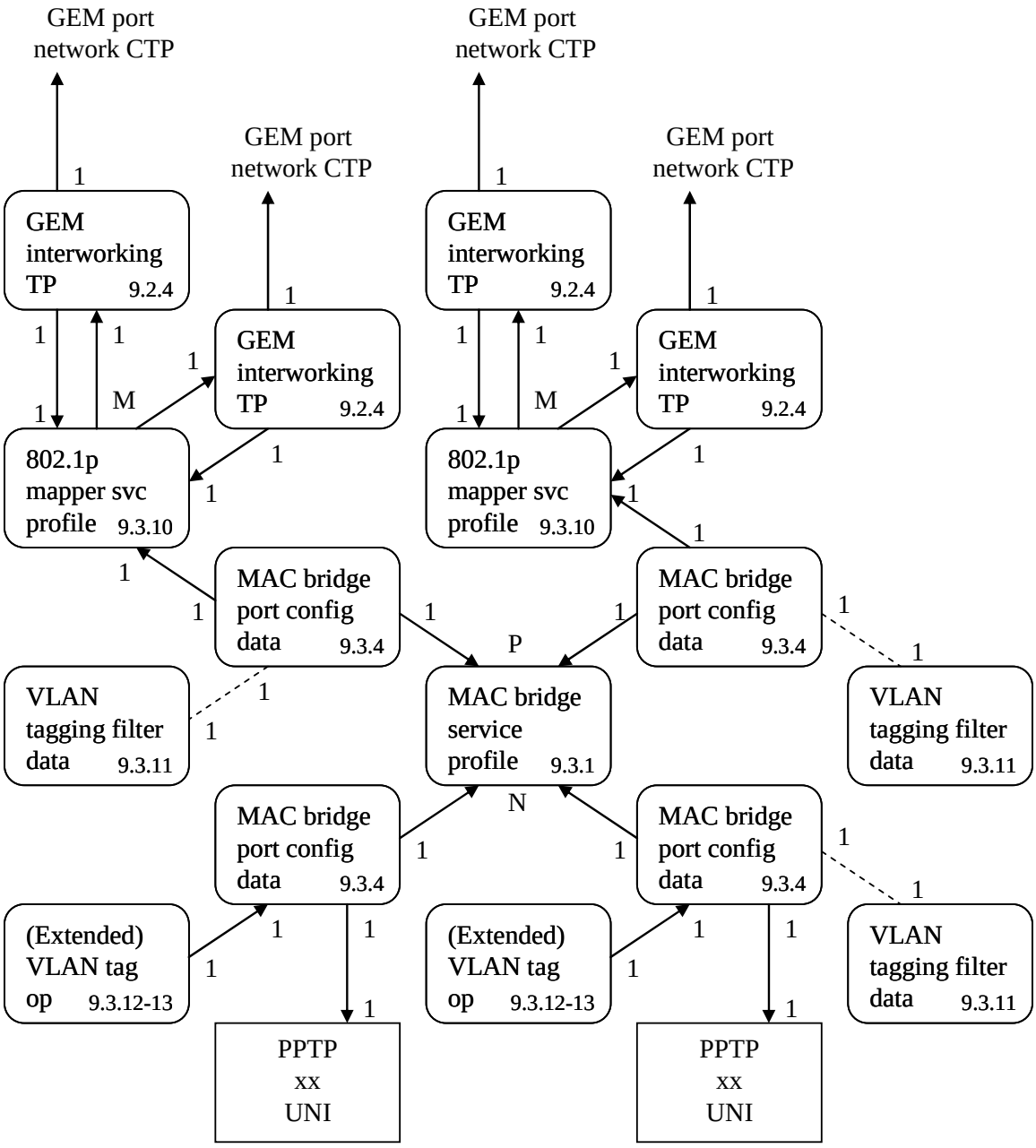



**Figure 8.2.2-9 – Illustration of N:MP bridge-map-filtering**

Figure 8.2.2-10 illustrates the use of the multicast interworking termination point. A bridge is used to multiplex the multiple ANI-side ports into the single (in this case) UNI-side port. It is essential to have a unicast path in parallel to the multicast path, because the unicast path carries the upstream signaling that is required for control of the multicast transmissions. In most scenarios, a unicast path already exists for other user communications.

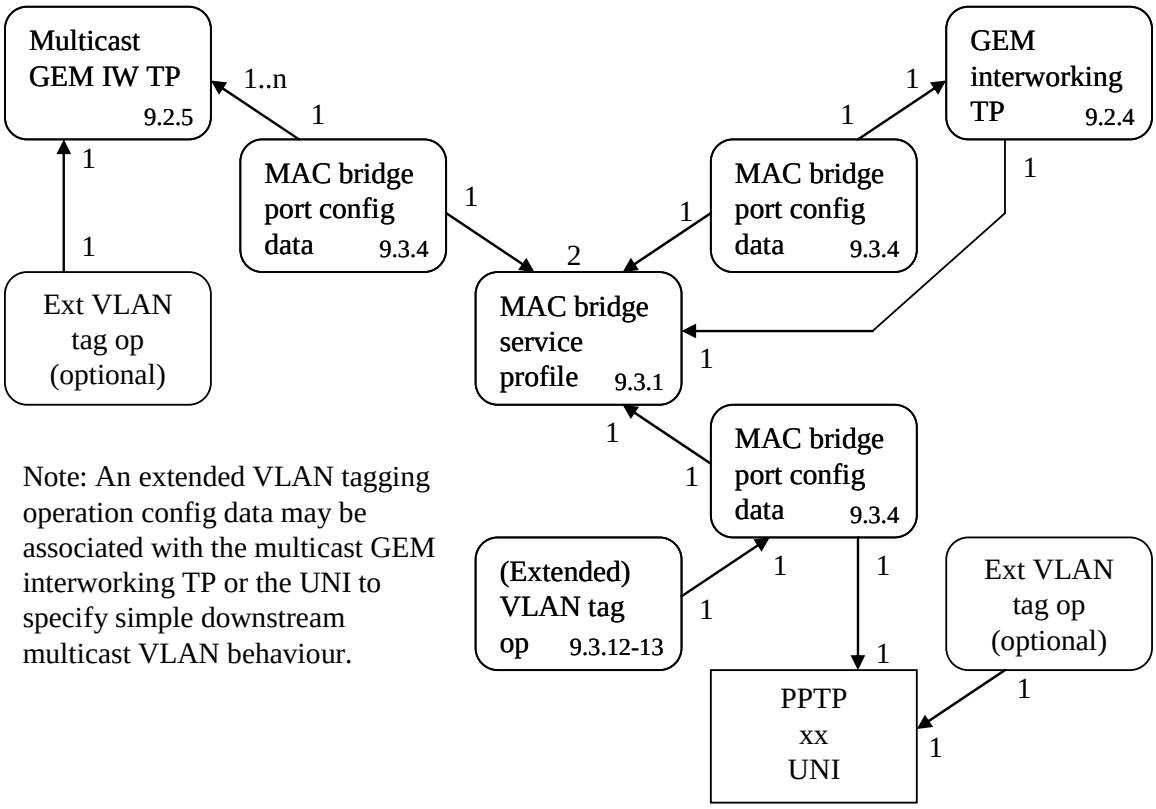



**Figure 8.2.2-10 – Illustration of multicast service**

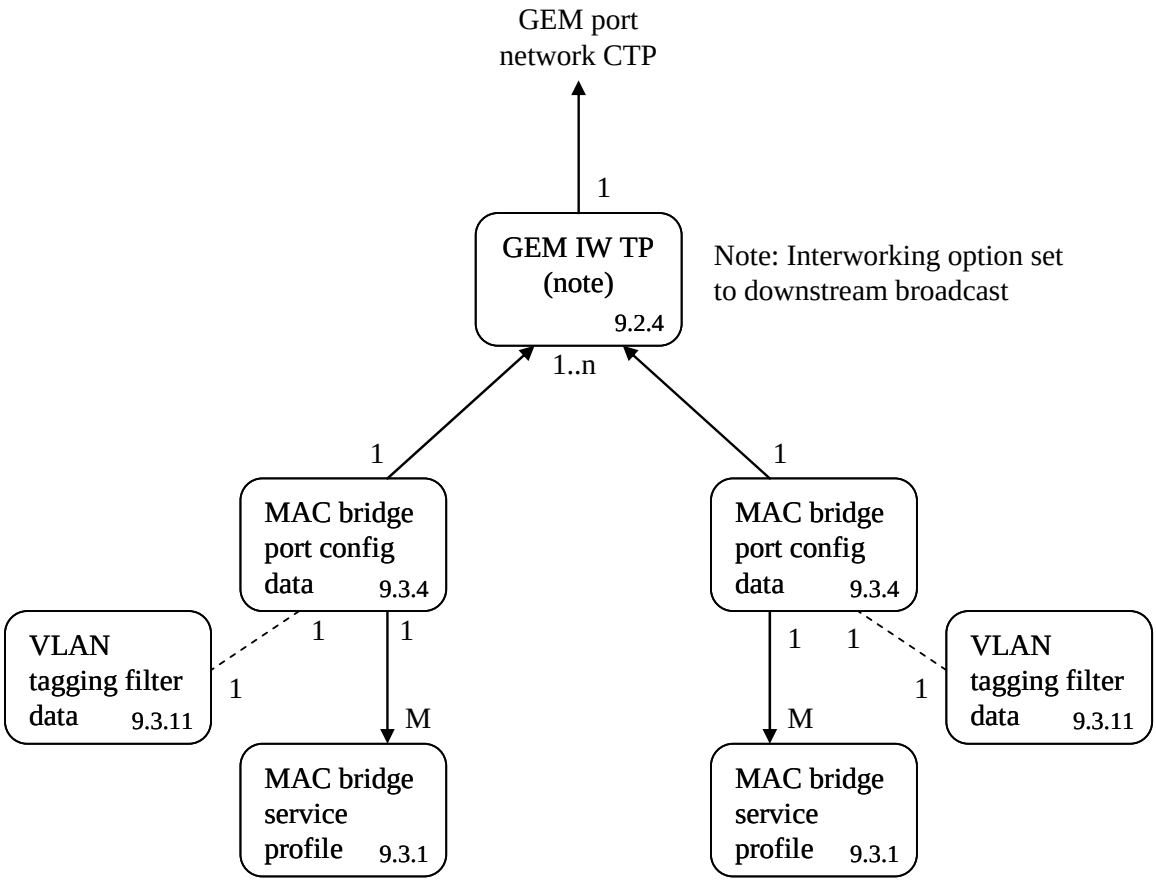



**Figure 8.2.2-11 – Illustration of downstream broadcast configuration**

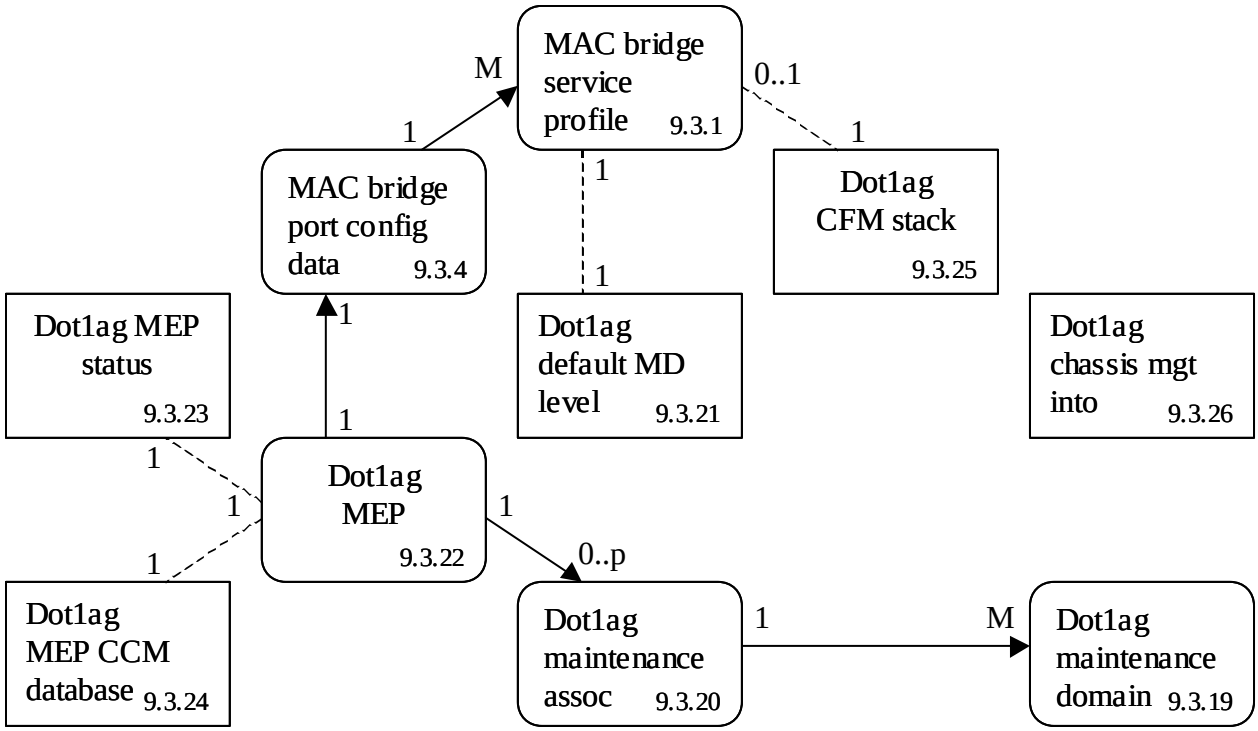



**Figure 8.2.2-12 – Illustration of 802.1ag in MAC bridge model**

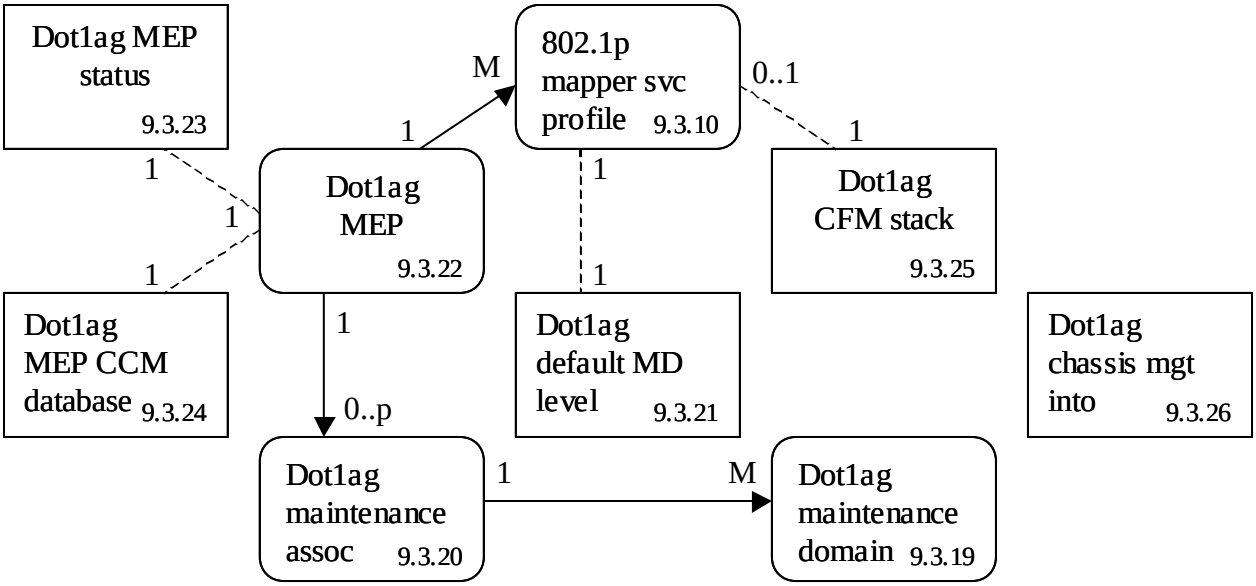



**Figure 8.2.2-13 – Illustration of 802.1ag in 802.1p mapper model**

NOTE – If a mapper is associated with ports of a bridge, the 802.1ag entities should be associated with the bridge and its ports, rather than with the mapper.

### 8.2.3 Routing

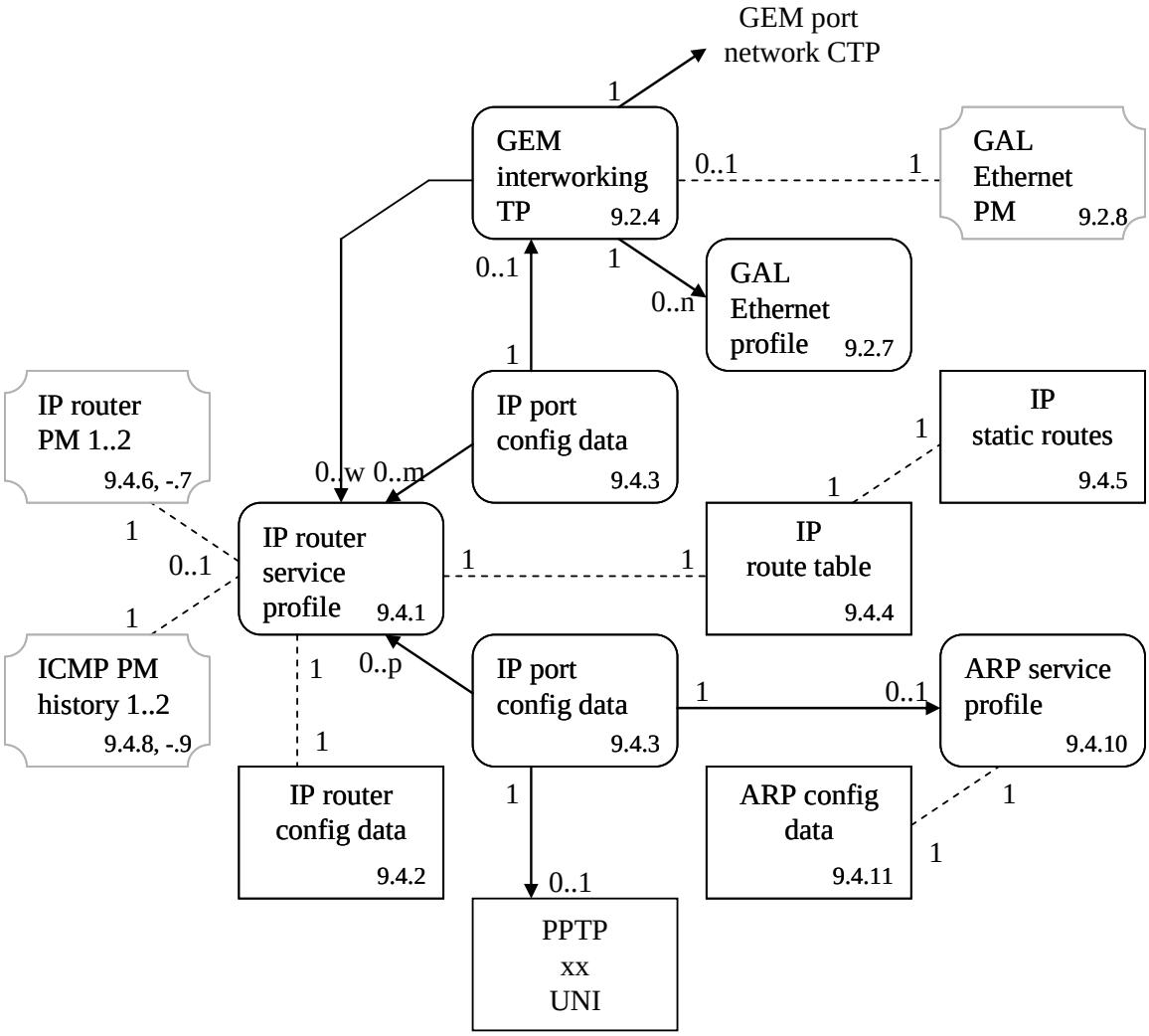



**Figure 8.2.3-1 – IP routing**

### 8.2.4 xDSL service

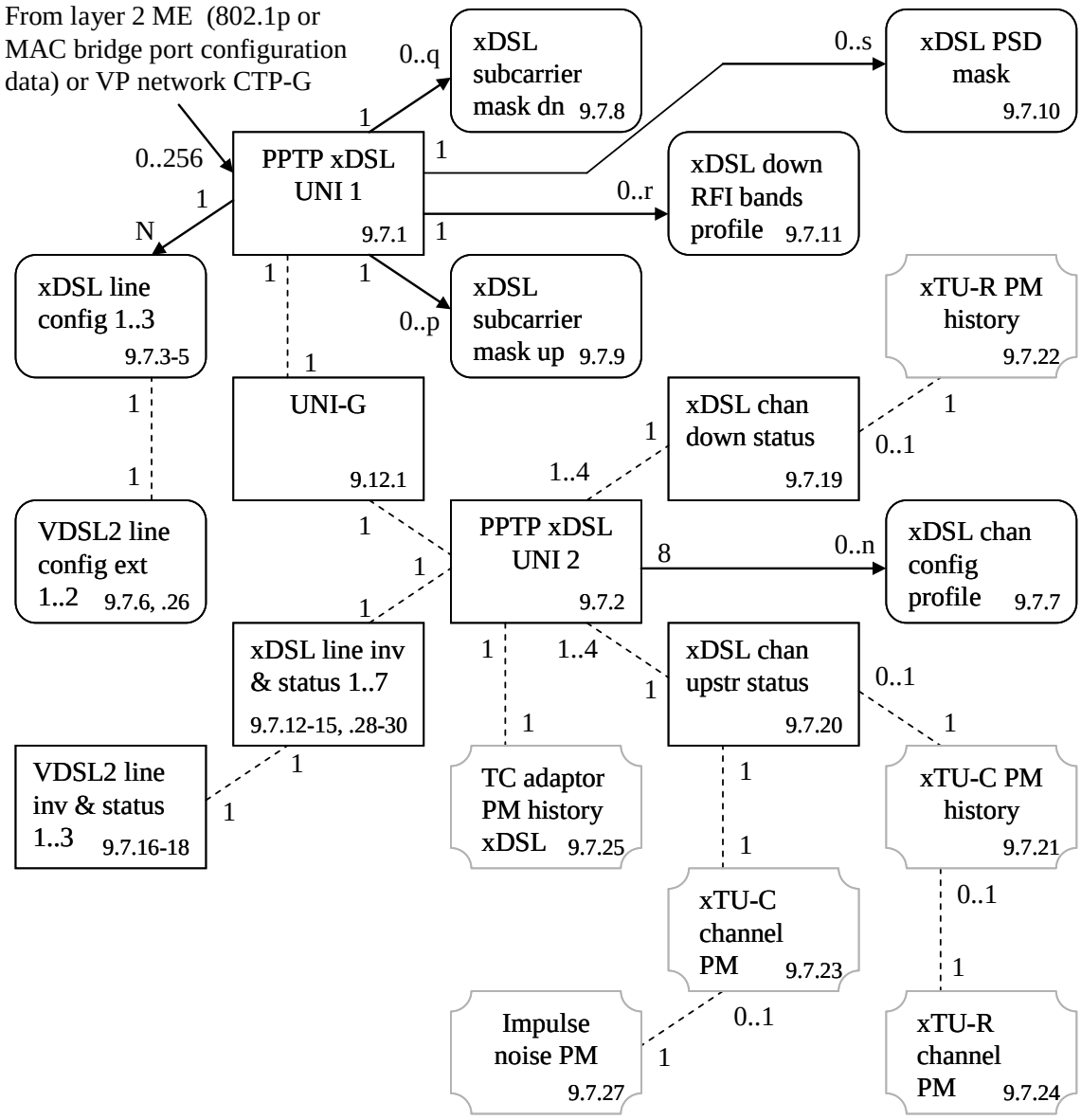



**Figure 8.2.4-1 – xDSL**

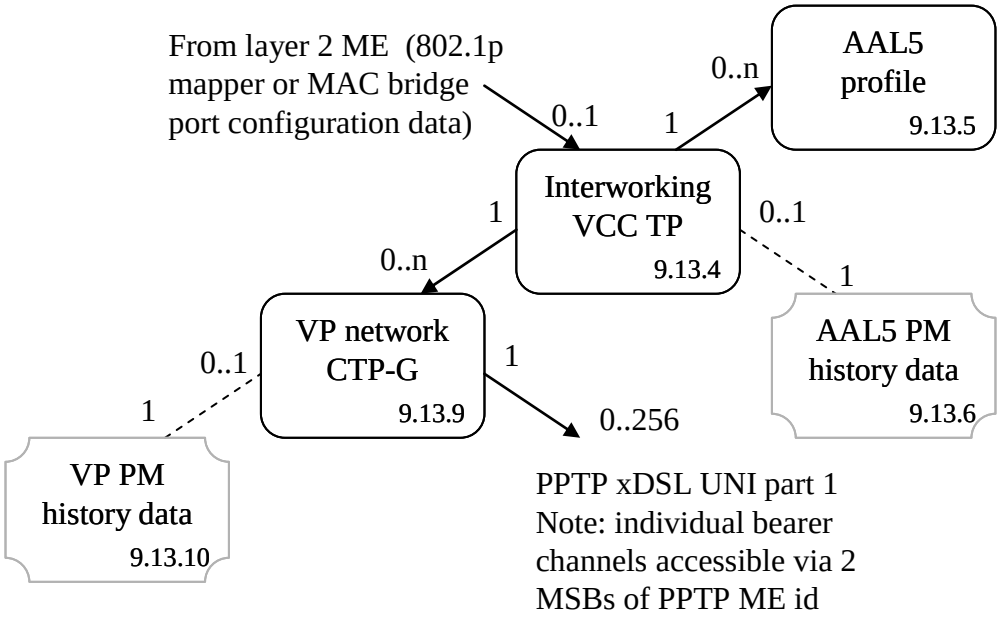



**Figure 8.2.4-2 – ATM interworking for xDSL**

### 8.2.5 802.11 service

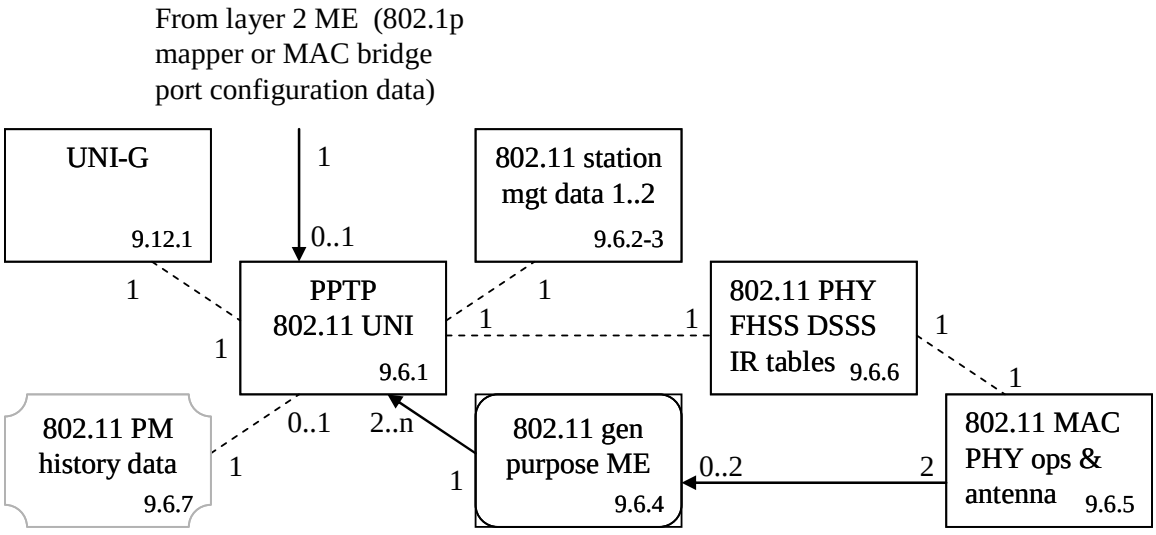



**Figure 8.2.5-1 – 802.11 wireless LAN**

### 8.2.6 MoCA service

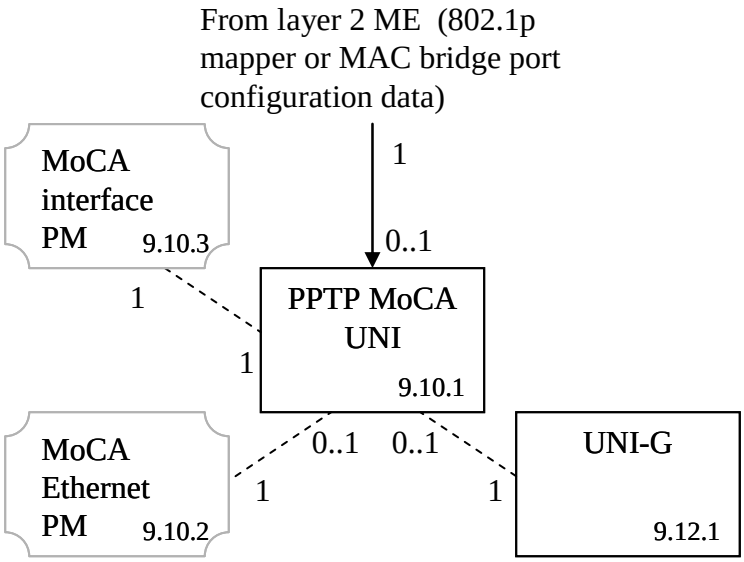



**Figure 8.2.6-1 – Multimedia over Coax Alliance**

### 8.2.7 Video return path

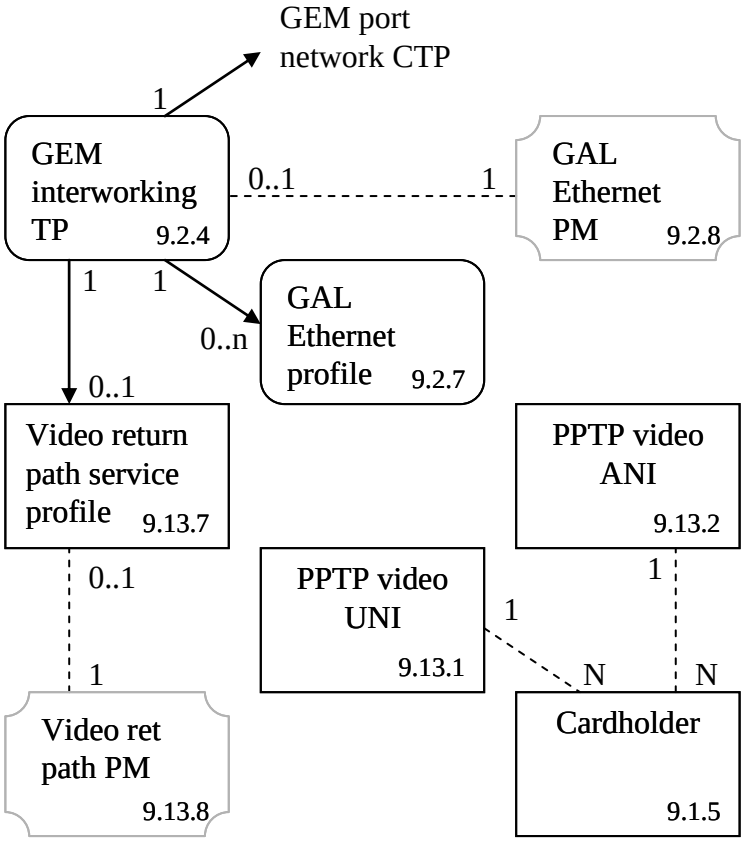



**Figure 8.2.7-1 – Video return path**

### 8.2.8 VoIP service

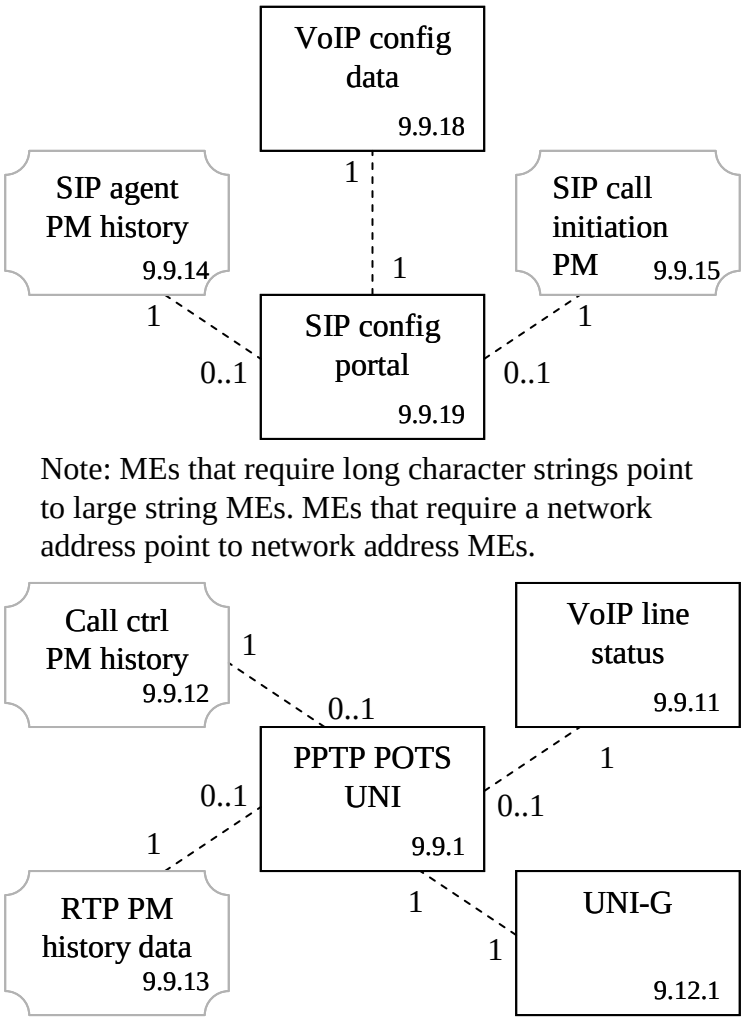
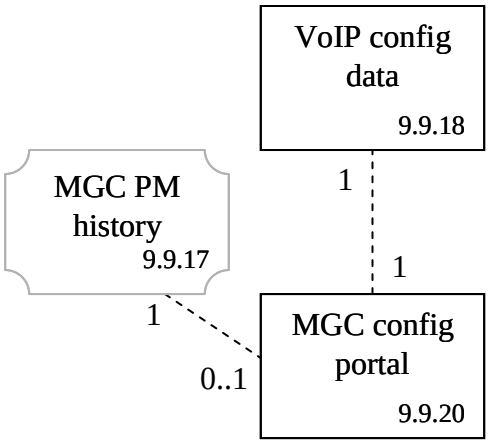



**Figure 8.2.8-1 – IP-path managed SIP VoIP**




Note: MEs that require long character strings point to large string MEs. MEs that require a network address point to network address MEs.

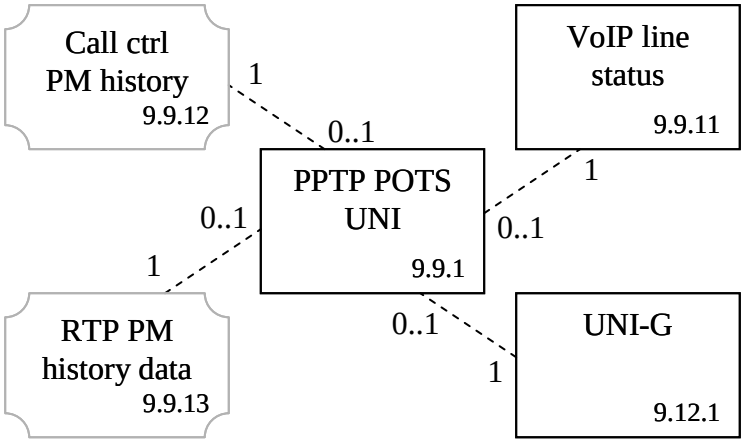
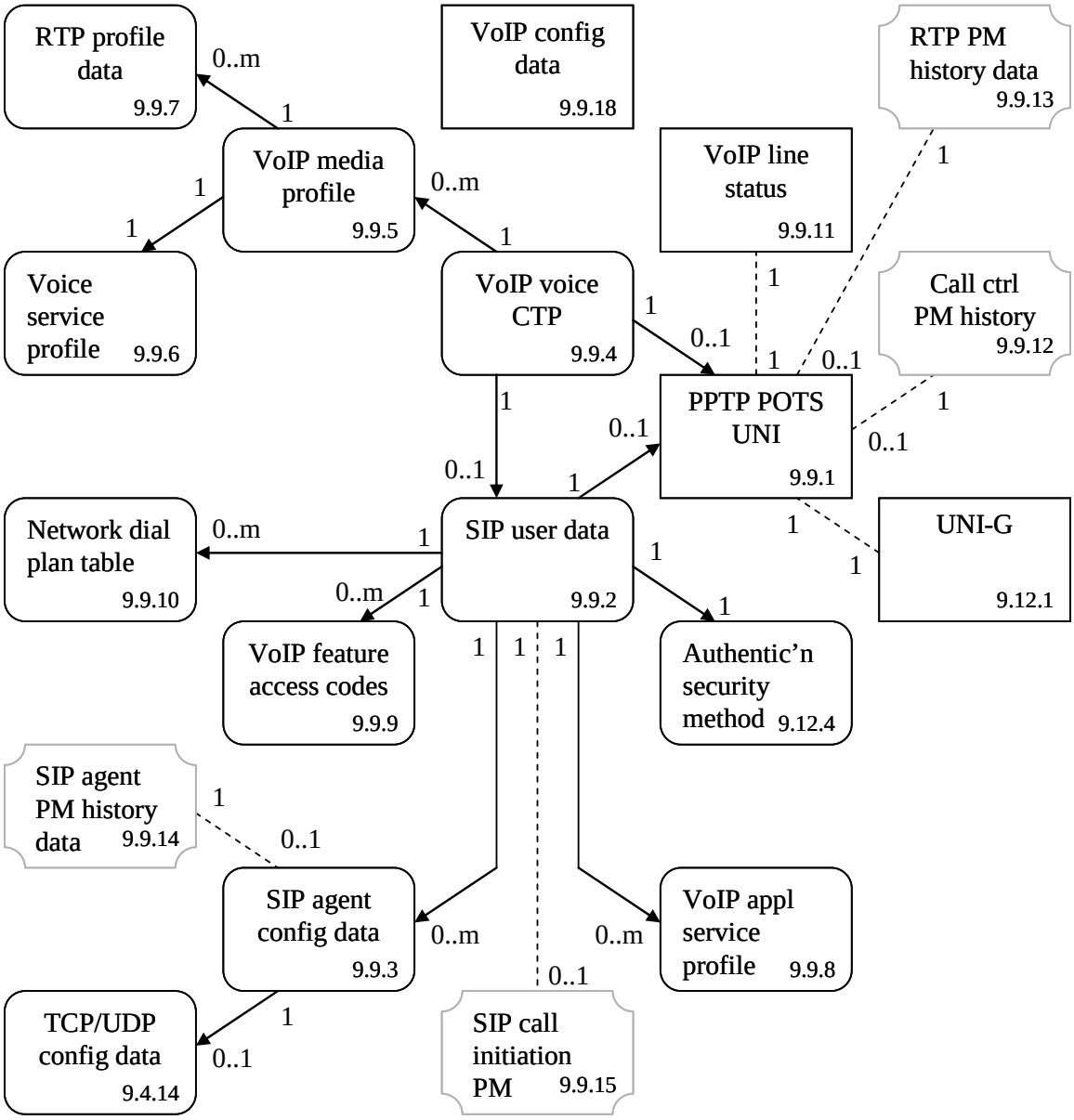
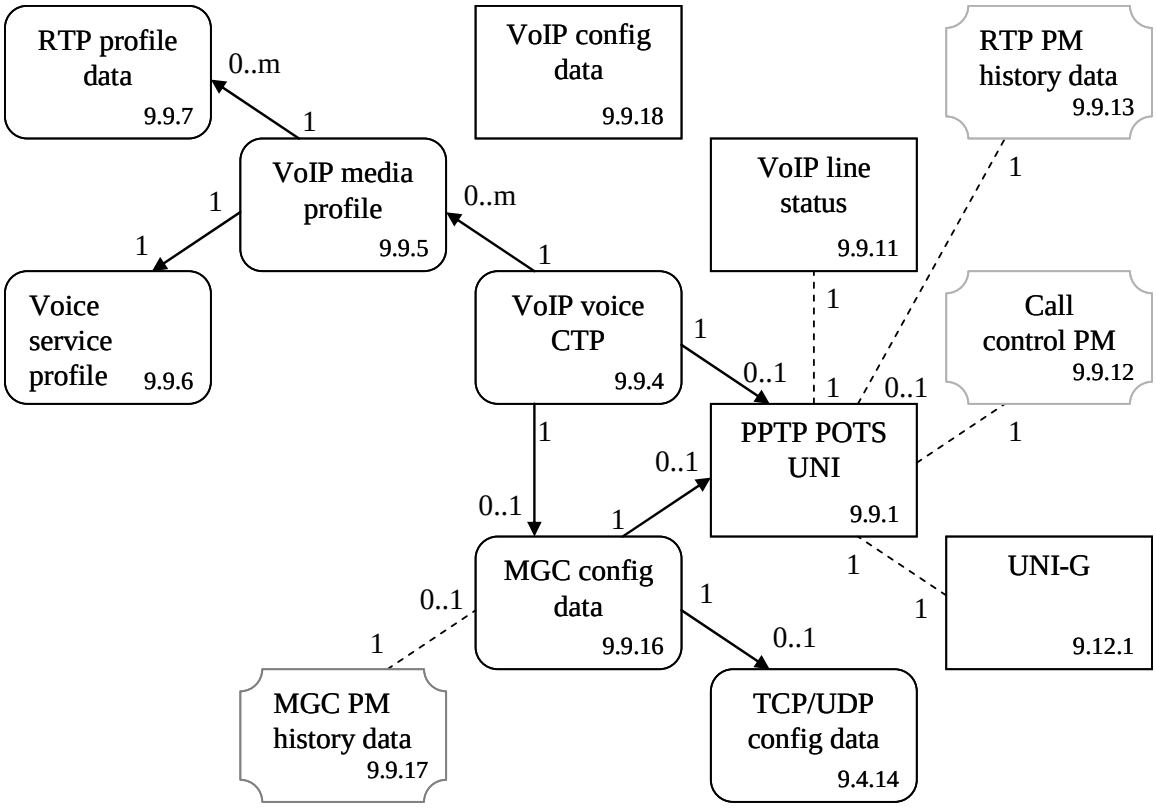



**Figure 8.2.8-2 – IP path managed H248 VoIP**




Note: MEs that require long character strings point to large string MEs. MEs that require a network address point to network address MEs.

**Figure 8.2.8-3 – OMCI managed SIP VoIP**




Note: MEs that require long character strings point to large string MEs. MEs that require a network address point to network address MEs.

**Figure 8.2.8-4 – OMCI managed H.248 VoIP**

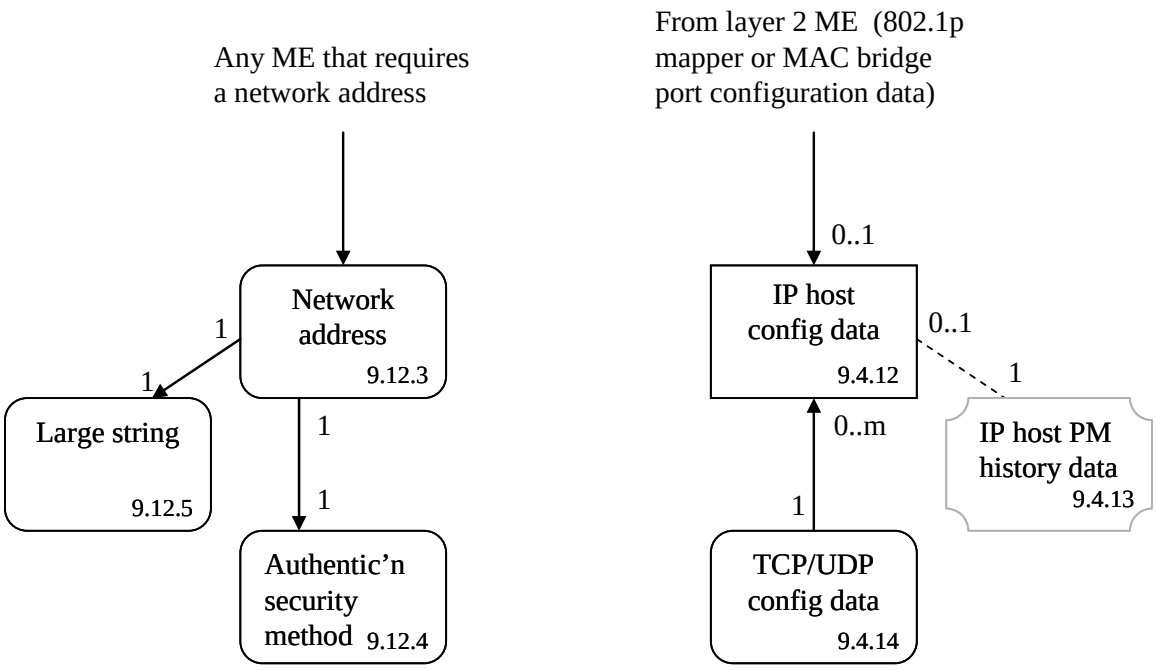



**Figure 8.2.8-5 – Common IP services**

### 8.2.9 Circuit emulation service

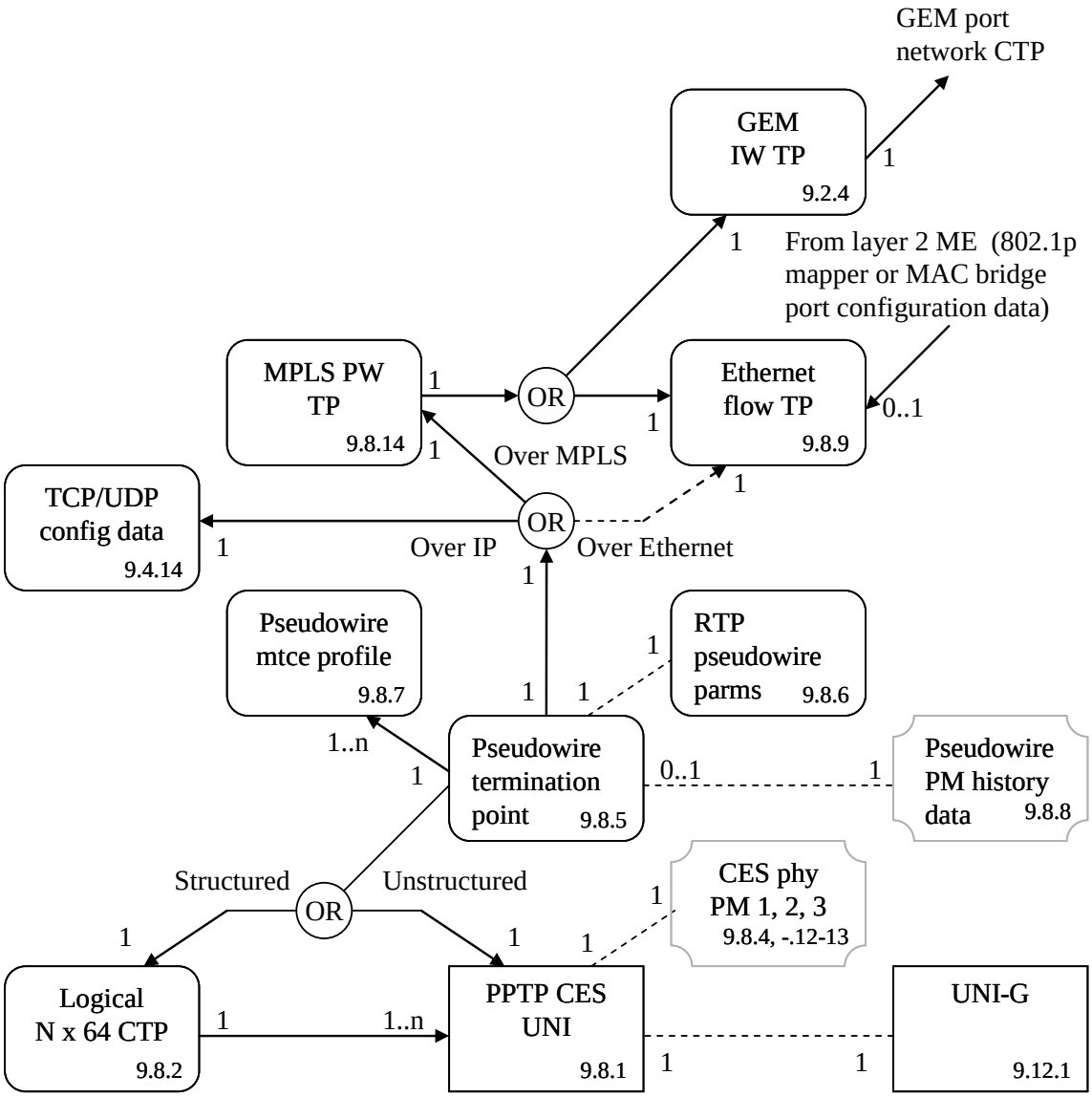



**Figure 8.2.9-1 – Pseudowire TDM**

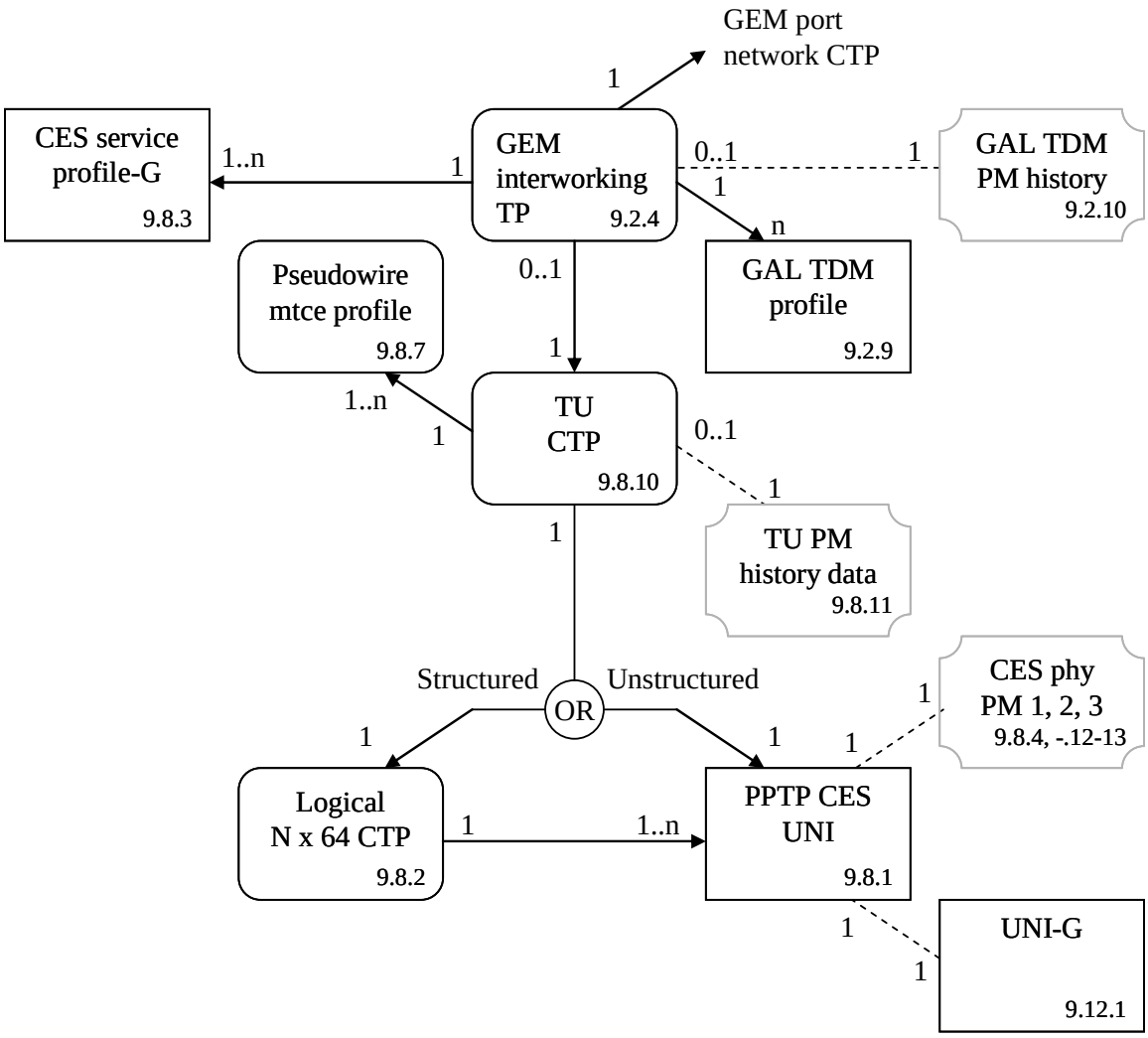



**Figure 8.2.9-2 – SDH over GEM**

### 8.2.10 Mid-span PON reach extenders

The PON reach extender is modelled as an ONT (the management entity) containing cardholders and circuit packs whose functions are to extend the reach of one or more PONs. The PON reach extender's own management ONT is understood to exist as a member of one of the extended PONs.

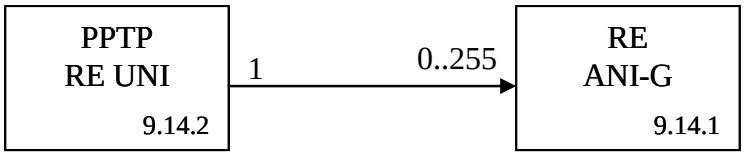



**Figure 8.2.10-1 – Mid-span PON reach extender core (repeater)**

NOTE – In many cases, the RE ANI-G and PPTP RE UNI are implemented on the same circuit pack. If so, the port mapping package can be used to create the hybrid line card.

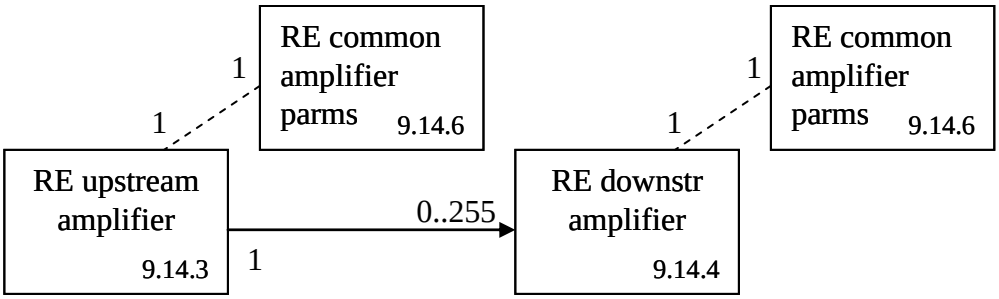



**Figure 8.2.10-2 – Mid-span PON reach extender core (optical amplifier)**

NOTE – In many cases, the RE upstream amplifier and RE downstream amplifier are implemented on the same circuit pack. If so, the port mapping package can be used to create the hybrid line card.

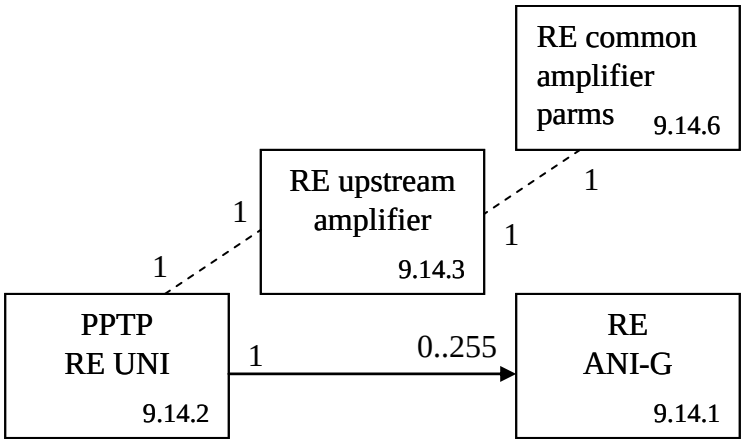



**Figure 8.2.10-3 – Mid-span PON reach extender core (hybrid)**

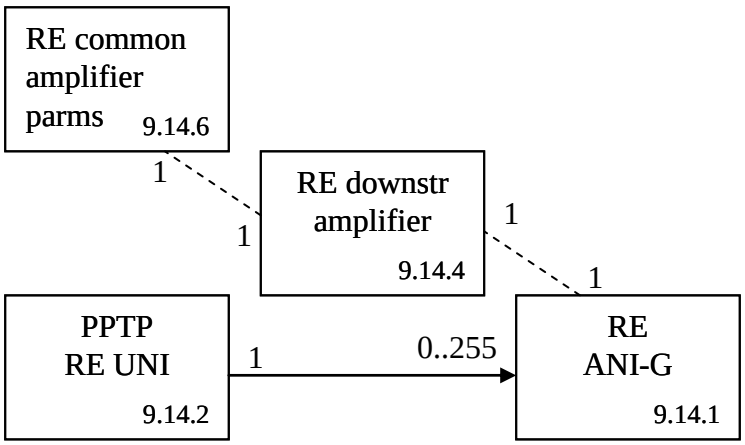



**Figure 8.2.10-4 – Mid-span PON reach extender core (hybrid)**

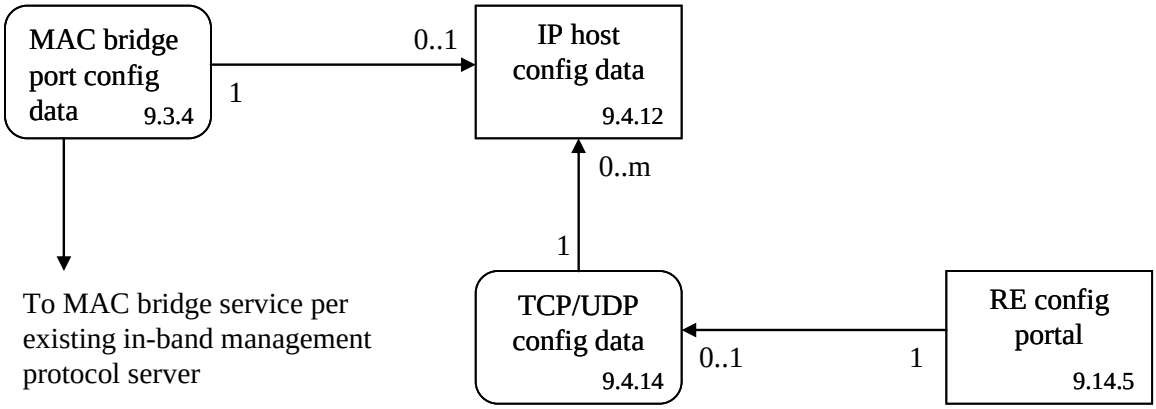



**Figure 8.2.10-5 – In-band management for mid-span PON reach extender**

### 8.2.11 Point to point gigabit Ethernet fed ONT

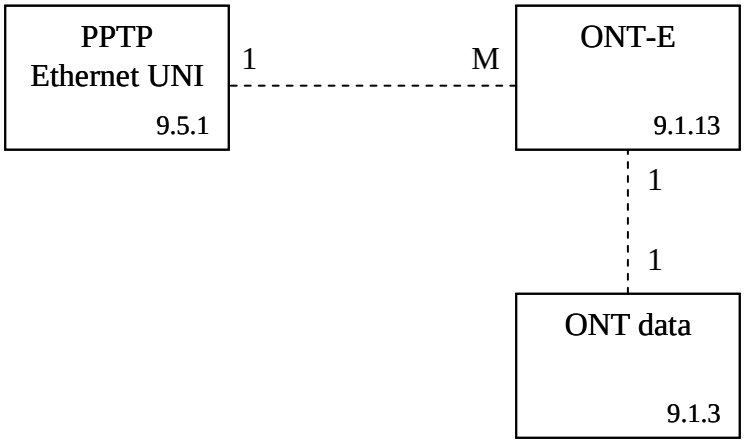



**Figure 8.2.11-1 – P2P GbE-fed ONT**

## 9 MIB description

*To the list of clauses within clause 9, add 9.14, as shown:*

These clauses are organized as follows:

…

9.14 Mid-span PON reach extender

### 9.1.1 ONT-G

*Revise the AVC table to read as follows. This removes AVCs on vendor ID, model and SN.*

**Attribute value change**

| Number | Attribute value change | Description |
|---|---|---|
| 1..7 | N/A | |
| 8 | Op state | Operational state change |
| 9 | N/A | |
| 10..16 | Reserved | |

*Revise the alarms table to read as follows.*

**Alarm**

| Number | Alarm | Description |
|---|---|---|
| 0 | Equipment alarm | Functional failure on an internal interface |
| 1 | Powering alarm | Loss of external power to battery backup unit. This alarm is typically derived through an external interface to a battery backup unit, and indicates that AC is no longer available to maintain battery charge. |
| 2 | Battery missing | Battery is provisioned but missing |
| 3 | Battery failure | Battery is provisioned and present but cannot recharge |
| 4 | Battery low | Battery is provisioned and present but its voltage is too low |
| 5 | Physical intrusion | Applies if the ONT supports detection such as door or box open |
| 6 | ONT self test failure | ONT has failed autonomous self test |
| 7 | Dying gasp | ONT is powering off imminently due to loss of power to the ONT itself. This alarm may be sent in conjunction with the powering alarm if a backup unit cannot supply power and the ONT is shutting down. |
| 8 | Temperature yellow | No service shutdown at present, but the circuit pack is operating beyond its recommended range. |
| 9 | Temperature red | Some services have been shut down to avoid equipment damage. The operational state of the affected PPTPs indicates the affected services. |
| 10 | Voltage yellow | No service shutdown at present, but the line power voltage is below its recommended minimum. Service restrictions may be in effect, such as permitting no more than N lines off-hook or ringing at one time. |
| 11 | Voltage red | Some services have been shut down to avoid power collapse. The operational state of the affected PPTPs indicates the affected services. |
| 12 | ONT manual power off | The ONT is shutting down because the subscriber has turned off its power switch. |
| 13..207 | Reserved | |
| 208..223 | Vendor-specific alarms | Not to be standardized |

### 9.1.2 ONT2-G

*Add code point 0x86 to the following attribute, as shown:*

**OMCC version**: This attribute identifies the version of the OMCC protocol being used by the ONT. This allows the OLT to manage a network with ONTs that support different OMCC versions. Release levels of G.984.4 may be supported with the following code points:

0x80 G.984.4 (06/04)

NOTE – for historic reasons, this codepoint may also appear in ONTs that support later versions of G.984.4.

0x81 G.984.4 amd 1 (06/05)

0x82 G.984.4 amd 2 (03/06)

0x83 G.984.4 amd 3 (12/06)

0x84 G.984.4 2008 (2/08)

0x85 G.984.4 2008 amd 1 (06/09)

0x86 G.984.4 2009 amd 2 (<date>). Baseline message set only, without the extended message set option

0x96 G.984.4 2009 amd 2 (<date>). Extended message set option, in addition to the baseline message set.

(R) (mandatory) (1 byte)

*Revise the total priority queue number attribute to read as follows:*

**Total priority queue number**: This attribute reports the total number of upstream priority queues that are not associated with a circuit pack, but with the ONT in its entirety. Upon ME instantiation, the ONT sets this attribute to the value that represents its capabilities. (R) (mandatory) (2 bytes)

*Append the following new attributes to the list of attributes.*

**Connectivity capability**: This attribute indicates the Ethernet connectivity models that the ONU can support. The value 0 indicates that the capability is not supported; 1 signifies support. The following codepoints are defined:

| Bit | Model |
|---|---|
| 1 (LSB) | N:1 bridging, figure 8.2.2-3 |
| 2 | 1:M mapping, figure 8.2.2-4 |
| 3 | 1:P filtering, figure 8.2.2-5 |
| 4 | N:M bridge-mapping, figure 8.2.2-6 |
| 5 | 1:MP map-filtering, figure 8.2.2-7 |
| 6 | N:P bridge-filtering, figure 8.2.2-8 |
| 7 | N:MP bridge-map-filtering, figure 8.2.2-9 |
| 8…16 | Reserved |

NOTE: It is not implied that an ONU may not support other connectivity models.

(R) (optional) (2 bytes)

**Current connectivity mode:** This attribute specifies the Ethernet connectivity model that the OLT wishes to use. The following code points are defined:

| Value | Connectivity model |
|---|---|
| 0 | No selection (default) |
| 1 | N:1 bridging |
| 2 | 1:M mapping |
| 3 | 1:P filtering |
| 4 | N:M bridge-mapping |

| Value | Connectivity model |
|---|---|
| 5 | 1:MP map-filtering |
| 6 | N:P bridge-filtering |
| 7 | N:MP bridge-map-filtering |
| 8…255 | Reserved |

NOTE: It is not implied that an ONU supports a given connectivity model only when that model is explicitly selected by this attribute. The ONU is free to support additional models at any and all times.

(R, W) (optional) (1 byte)

**QoS configuration flexibility**: This attribute reports whether various managed entities in the ONT are hard-wired or configurable. For backward compatibility, and if the ONT does not support this attribute, all such attributes are understood to be hard-wired. (R) (optional) (2 bytes)

| Bit | Interpretation when bit value = 1 |
|---|---|
| 1 (LSB) | Priority queue ME: Related port attribute is read-write and can point to any T-CONT or UNI port in the same slot |
| 2 | Priority queue ME: The traffic scheduler-G pointer is permitted to refer to any traffic scheduler-G in the same slot |
| 3 | Traffic scheduler-G ME: T-CONT pointer is read-write |
| 4 | Traffic scheduler-G ME: Policy attribute is read-write |
| 5 | T-CONT ME: Policy attribute is read-write |
| 5..16 | Reserved |

**Discussion**:

To allow for the possibility that the OLT does not support flexible configuration, the ONT vendor must assure that the priority queues and traffic schedulers are configured in a meaningful and useful way by factory default, and that this default configuration is restored upon ONT initialization and MIB reset. The specifics of such a configuration are beyond the scope of this Recommendation.

The managed entity ID of both the T-CONT and traffic scheduler contains a slot number. Even when attributes in the above list are read-write, it is never permitted to change the slot number in a reference. That is, configuration flexibility never extends across slots. It is also not permitted to change the directionality of an upstream queue to downstream, or vice versa.

### 9.1.3 ONT data

*Replace the MIB upload action description with the following:*

**MIB upload**: Latch a snapshot (i.e., copy) of the current MIB. Not every managed entity or every attribute is included in a MIB upload. Table attributes are excluded. Only the control block attributes of performance monitoring MEs are uploaded. Other MEs and attributes, such as the PPTP for the local craft terminal, are excluded as documented in their specific definitions.

### 9.1.5 Cardholder

*Add the following attributes at the end of the attributes list:*

**ARC**: See clause I.1.8. (R, W) (optional) (1 byte)

**ARC interval**: See clause I.1.8. (R, W) (optional) (1 byte)

*Revise the attribute value change table to read as follows (add ARC):*

**Attribute value change**

| Number | Attribute value change | Description |
|---|---|---|
| 1 | Actual type | Actual type of circuit pack in cardholder |
| 2..4 | N/A | |
| 5 | Actual equipment id | Actual equipment ID of circuit pack in cardholder |
| 6..7 | N/A | |
| 8 | ARC | ARC timer expiration |
| 9..16 | Reserved | |

### 9.1.10 Protection data

*Replace the present definition with the following:*

This managed entity models the capability and parameters of PON protection. An ONT that supports PON protection automatically creates one instance of this managed entity. A chassis-based reach extender could have the capability of protecting a number of PONs, possibly by way of circuit packs configured in arbitrary (rather than predefined) slots. Protection data managed entities in a multi-PON reach extender may therefore be auto-created or created by the OLT, depending on the reach extender's architecture, and the ANI-G pointers may be either populated by the ONT itself (read-only) or configured by the OLT (read-write, set-by-create). Likewise, the nature of protection may be set read-only by the ONT's architecture, or may be settable by the OLT.

Note 1: Equipment protection is modelled with the equipment protection profile and cardholder managed entities.

Note 2: For ONTs that implement reach extender functions, this ME can be used to describe OMCI protection, reach extender R'/S' protection, or both. For reach extender R'/S' protection, the protection type must be 1:1 without extra traffic, because the switching is done on a link-by-link basis, and the protection link is in cold standby mode. The instance that pertains to OMCI protection has ME ID = 0.

*Relationships*

One instance of this managed entity is associated with two instances of the ANI-G, RE ANI-G or RE upstream amplifier. One of the ANI managed entities represents the working side; the other represents the protection side.

*Attributes*

**Managed entity id**: This attribute uniquely identifies each instance of this managed entity. If there is more than one protection data ME, they are numbered in ascending order from 0. (R, Set-by-create if applicable) (mandatory) (2 bytes)

**Working ANI-G pointer**: This attribute points to the ANI-G, RE ANI-G or RE upstream amplifier managed entity that represents the working side of a protected PON. (R, W if applicable, Set-by-create if applicable) (mandatory) (2 bytes)

NOTE – It is possible, and indeed likely, that an ANI-G will have the same ME ID as the RE ANI-G or even the RE upstream amplifier that supports its physical PON interface. The ANI-G represents the embedded ONT that terminates the OMCC. Since it is not expected that protection of management communications will be implemented independently from protection of the optical layer, the ambiguity is not expected to cause a problem.

**Protection ANI-G pointer**: This attribute points to the ANI-G, RE ANI-G or RE upstream amplifier managed entity that represents the protection side of a protected PON. (R, W if applicable, Set-by-create if applicable) (mandatory) (2 bytes)

**Protection type**: This attribute indicates the type of PON protection. Valid values are:

0 1+1 protection
1 1:1 protection without extra traffic
2 1:1 protection with ability to support extra traffic

(R, W if applicable, Set-by-create if applicable) (mandatory) (1 byte)

**Revertive ind**: This attribute indicates whether protection is revertive (1) or non-revertive (0). (R, W if applicable, Set-by-create if applicable) (mandatory) (1 byte)

**Wait to restore time**: This attribute specifies the time, in seconds, to wait after a fault clear before switching back to the working side. Upon ME instantiation, the ONT sets this attribute to 3 seconds. (R, W, Set-by-create if applicable) (mandatory) (2 bytes)

**Switching guard time**: This attribute specifies the time, in milliseconds, to wait after the detection of a fault before performing a protection switch. Selection of a default value for this attribute is outside the scope of this Recommendation, as it is normally handled through supplier-operator negotiations. (R, W, Set-by-create if applicable) (optional) (2 bytes)

*Actions*

Create, delete, if applicable

Get, set

*Notifications*

None

### 9.1.12 ONT remote debug

*Revise the reply table attribute description to read as follows:*

**Reply table**: This attribute is used to pass reply information back to the OLT. Its format is defined by the command format attribute. The get, get next action sequence must be used with this attribute, since its size is unspecified. (R) (mandatory) (N bytes)

*Add new clause 9.1.13 as below:*

### 9.1.13 ONT-E

This managed entity represents a point to point gigabit Ethernet-fed ONT as equipment, as defined in [ITU-T G.986]. The ONT automatically creates an instance of this managed entity. It assigns values to read-only attributes according to data within the ONT itself.

*Relationships*

All other managed entities in this Recommendation are related directly or indirectly to the ONT-E entity.

*Attributes*

**Managed entity id**: This attribute uniquely identifies each instance of this managed entity. There is only one instance, number 0. (R) (mandatory) (2 bytes)

**Vendor id**: This attribute identifies the vendor of the ONT. Both the code set for the Vendor_ID specified in [ANSI T1.220] and Organizationally Unique Identifier (OUI) specified in Clause 9 of [IEEE 802] could be applied for this attribute.

When the code set for the Vendor_ID specified in [ANSI T1.220] is applied for this attribute, the 4 characters are mapped into the 4-byte field by concatenating the ASCII/ANSI character codes.

When the OUI is applied for this attribute, the 3 characters are mapped into the 4-byte field with 0xFF assigned to the first octet.

(R) (mandatory) (4 bytes)

| Octet | Content | |
|---|---|---|
| | **Vendor_ID in [ANSI T1.220]** | **OUI in [IEEE 802]** |
| 1 | First byte of Vendor_ID | 0xFF |
| 2 | Second byte of Vendor_ID | First byte of OUI |
| 3 | Third byte of Vendor_ID | Second byte of OUI |
| 4 | Fourth byte of Vendor_ID | Third byte of OUI |

**Version**: This attribute identifies the version of the ONT as defined by the vendor. The character value "0" indicates that version information is not available or applicable. (R) (mandatory) (14 bytes)

**Serial number**: The serial number is unique for each ONT. It is defined by the vendor. The character value "0" indicates that serial number information is not available or applicable. (R) (mandatory) (8 bytes)

*Actions*

**Get**

**Reboot**: Reboot the ONT

*Notifications*

**Alarm**

| Number | Alarm | Description |
|---|---|---|
| 0 | Equipment alarm | Functional failure on an internal interface |
| 1..207 | Reserved | |
| 208..223 | Vendor-specific alarms | Not to be standardized |

## 9.2 ANI management

*Delete the following introductory paragraph:*

~~Although the OLT maintains some of the PON related managed entities and attributes via G.984.3 PLOAM messages, there is also information to be negotiated in OMCC. Therefore, the ONT autonomously creates at least one instance of each of the managed entities ANI-G and T-CONT. These ANI management MEs are included in a MIB upload.~~

### 9.2.2 T-CONT

*In the introductory text, correct the reference to clause 11.3.3 to read 11.4.3.*

*Revise the policy attribute description to read as follows:*

**Policy**: This attribute indicates the T-CONT's traffic scheduling policy. Valid values:

0 Null.

1 HOL – Head of line queueing.

2 WRR – Weighted round robin.

(R, W) (mandatory) (1 byte)

NOTE – This attribute is read-only unless otherwise specified by the QoS configuration flexibility attribute of the ONT2-G managed entity. If flexible configuration is not supported, the ONT should reject an attempt to set it with a parameter error result-reason code.

### 9.2.3 GEM port network CTP

*Revise the descriptions of the two following attributes to read as shown (change* GEM traffic descriptor *to* traffic descriptor*).*

**Traffic descriptor profile pointer for upstream**: This attribute points to the instance of the traffic descriptor managed entity that contains the upstream traffic parameters used for this GEM port network CTP ME. This attribute is used when the traffic management option attribute in the ONT-G ME is 1 (rate controlled), specifying the PIR/PBS to which the upstream traffic is shaped. This attribute is also used when the traffic management option attribute in the ONT-G ME is 2 (priority and rate controlled), specifying the CIR/CBS/PIR/PBS to which the upstream traffic is policed. (R, W, Set-by-create) (optional) (2 bytes)

See also Appendix III.

**Traffic descriptor profile pointer for downstream**: This attribute points to the instance of the traffic descriptor managed entity that contains the downstream traffic parameters used for this GEM port network CTP ME. This attribute is used when the traffic management option attribute in the ONT-G ME is 2 (priority and rate controlled), specifying the CIR/CBS/PIR/PBS to which the downstream traffic is policed. (R, W, Set-by-create) (optional) (2 bytes)

See also Appendix III.

### 9.2.4 GEM interworking termination point

*Modify the following attributes as shown.*

**Managed entity id**: This attribute uniquely identifies each instance of this managed entity. ~~It must be unique over all interworking VCC termination point and GEM interworking termination point MEs.~~ (R, Set-by-create) (mandatory) (2 bytes).

**Interworking option**: This attribute identifies the type of non-GEM function that is being interworked. The options are:

0 Unstructured TDM.

1 MAC bridge LAN.

2 Reserved for future use.

3 IP data service.

4 Video return path.

5 802.1p mapper.

6 Downstream broadcast.

7 MPLS data service.

(R, W, Set-by-create) (mandatory) (1 byte)

**Service profile pointer**: This attribute points to an instance of a service profile, such as:

| | |
|---|---|
| CES service profile-G | if interworking option = 0 |
| MAC bridge service profile | if interworking option = 1 |
| IP router service profile | if interworking option = 3 |
| Video return path service profile | if interworking option = 4 |
| 802.1p mapper service profile | if interworking option = 5 |
| Null pointer | if interworking option = 6 |
| Null pointer | if interworking option = 7 |

(R, W, Set-by-create) (mandatory) (2 bytes)

**GAL profile pointer**: This attribute points to an instance of the GAL profile. The relationship between the interworking option and the related GAL profile is:

| Interworking option | GAL profile type |
|---|---|
| 0 | GAL TDM profile |
| 1 | GAL Ethernet profile |
| 2 | Reserved for future use |
| 3 | GAL Ethernet profile for data service |
| 4 | GAL Ethernet profile for video return path |
| 5 | GAL Ethernet profile for 802.1p mapper |
| 6 | Null pointer |
| 7 | Null pointer |

(R, W, Set-by-create) (mandatory) (2 bytes)

### 9.2.5 Multicast GEM interworking termination point

*To the list of actions, add*

Set table (optional)

### 9.3.1 MAC bridge service profile

*Add the following new attribute at the end of the attributes list:*

**Dynamic filtering ageing time**: This attribute specifies the age of dynamic filtering entries in the bridge database, after which unrefreshed entries are discarded. In accordance with [IEEE 802.1D] clause 7.9.2 and [IEEE 802.1Q] clause 8.8.3, the range is 10..1 000 000 seconds, with a resolution of 1 second and a default of 300 seconds. (R, W, Set-by-create) (optional) (4 bytes)

### 9.3.2 MAC bridge configuration data

*Remove* set *from the list of actions.*

### 9.3.4 MAC bridge port configuration data

*Add code point 11 to the TP type attribute:*

**TP type**: This attribute identifies the type of termination point associated with this MAC bridge port. Valid values are:

1 Physical path termination point Ethernet UNI
2 Interworking VCC termination point
3 802.1p mapper service profile
4 IP host config data
5 GEM interworking termination point
6 Multicast GEM interworking termination point
7 Physical path termination point xDSL UNI part 1
8 Physical path termination point VDSL UNI
9 Ethernet flow termination point
10 Physical path termination point 802.11 UNI
11 Virtual Ethernet interface point

(R, W, Set-by-create) (mandatory) (1 byte)

*Revise the descriptions of the two following attributes to read as shown (change* GEM traffic descriptor *to* traffic descriptor*).*

**Outbound TD pointer**: This attribute points to a traffic descriptor that limits the traffic rate leaving the MAC bridge. (R, W) (optional) (2 bytes)

**Inbound TD pointer**: This attribute points to a traffic descriptor that limits the traffic rate entering the MAC bridge. (R, W) (optional) (2 bytes)

*Add the following attribute:*

**MAC learning depth**: This attribute specifies the maximum number of MAC addresses to be learned by this MAC bridge port. The default value 0 specifies that there is no administratively-imposed limit. (R, W, Set-by-create) (optional) (1 byte)

NOTE – If this attribute is not zero, its value overrides the value set in the MAC learning depth attribute of the MAC bridge service profile.

*Revise the notifications section to read:*

*Notifications*

**Alarm**

| Number | Alarm | Description |
|---|---|---|
| 0 | Port blocking | This port has been blocked due to loop detection in accordance with 802.1D. |
| 1..207 | Reserved | |
| 208..223 | Vendor-specific alarms | Not to be standardized |

NOTE – To determine the state of a MAC bridge port, the OLT can read the port state attribute of the MAC bridge port designation data.

### 9.3.6 MAC bridge port filter table data

*To the list of actions, add*

Set table (optional)

### 9.3.10 802.1p mapper service profile

*Add code points 7 and 8 to the following attributes:*

**TP pointer**: This attribute points to an instance of the associated termination point. The termination point type is determined by the TP type attribute:

| **TP type** | **TP pointer points to…** |
|---|---|
| 0 | Bridging mapping. Pointer should be set to 0xFFFF by OLT and ignored by ONT. |
| 1 | PPTP Ethernet UNI |
| 2 | IP host config data |
| 3 | Ethernet flow TP |
| 4 | PPTP xDSL UNI |
| 5 | PPTP 802.11 UNI |
| 6 | PPTP MoCA UNI |
| 7 | Virtual Ethernet interface point |
| 8 | Interworking VCC termination point |
| Not supported | Bridging mapping if TP pointer value is 0xFFFF. TP pointer may also point to PPTP Ethernet UNI. |

NOTE – When the TP type is xDSL, the two most significant bits may be used to indicate a bearer channel.

(R, W, Set-by-create) (mandatory) (2 bytes)

**TP type**: This attribute identifies the type of termination point associated with the mapper.

0 Mapper used for bridging-mapping

1 Mapper directly associated with a PPTP Ethernet UNI

2 Mapper directly associated with an IP host service

3 Mapper directly associated with an Ethernet flow termination point

4 Mapper directly associated with a PPTP xDSL UNI

5 Mapper directly associated with a PPTP 802.11 UNI

6 Mapper directly associated with a PPTP MoCA UNI

7 Mapper directly associated with a virtual Ethernet interface point

8 Mapper directly associated with an interworking VCC termination point

(R, W, Set-by-create) (optional) (1 byte)

### 9.3.12 VLAN tagging operation configuration data

*Add code point 11 to the association type attribute:*

**Association type**: This attribute specifies the type of the ME that is associated with this VLAN tagging operation configuration data ME. Values are assigned in accordance with the following list.

0 (default) Physical path termination point Ethernet UNI (for backward compatibility, may also be an IP host config data ME; they must not have the same ME ID). The associated ME instance is implicit; its identifier is the same as that of this VLAN tagging operation configuration data.

1 IP host config data
2 802.1p mapper service profile
3 MAC bridge port configuration data
4 Physical path termination point xDSL UNI
5 GEM interworking termination point
6 Multicast GEM interworking termination point
7 Physical path termination point MoCA UNI
8 Physical path termination point 802.11 UNI
9 Ethernet flow termination point
10 Physical path termination point Ethernet UNI
11 Virtual Ethernet interface point

The associated ME instance is identified by the associated ME pointer.
(R, W, Set-by-create) (optional) (1 byte)

### 9.3.13 Extended VLAN tagging operation configuration data

*Add code point 10 to the association type attribute:*

**Association type**: This attribute identifies the type of the ME associated with this extended VLAN tagging ME. Values are assigned as follows:

0 MAC bridge port configuration data
1 802.1p mapper service profile
2 Physical path termination point Ethernet UNI
3 IP host config data
4 Physical path termination point xDSL UNI
5 GEM interworking termination point
6 Multicast GEM interworking termination point
7 Physical path termination point MoCA UNI
8 Physical path termination point 802.11 UNI
9 Ethernet flow termination point
10 Virtual Ethernet interface point

(R, W, Set-by-create) (mandatory) (1 byte)

*Where the text presently reads:*

When the table is created, the ONT should predefine three entries that list the default treatment (of normal forwarding) for untagged, single tagged, and double tagged frames. As an exception to the rule on ordered processing, these default rules are always considered as a last resort for frames that don't match any other applicable rule. Best practice dictates that these entries not be deleted; however, they can be modified to produce the desired default behavior.

15, x, x, 15, x, x, x, (0, 15, x, x, 15, x, x)

15, x, x, 14, x, x, x, (0, 15, x, x, 15, x, x)

14, x, x, 14, x, x, x, (0, 15, x, x, 15, x, x)

*Revise it to read:*

When the table is created, the ONT should predefine three entries that list the default treatment (of normal forwarding) for untagged, single tagged, and double tagged frames. As an exception to the rule on ordered processing, these default rules are always considered as a last resort for frames that don't match any other applicable rule. Best practice dictates that these entries not be deleted; however, they can be modified to produce the desired default behavior.

15, x, x, 15, x, x, 0, (0, 15, x, x, 15, x, x) – no tags, Ethertype don't care

15, x, x, 14, x, x, 0, (0, 15, x, x, 15, x, x) – 1 tag, Ethertype don't care

14, x, x, 14, x, x, 0, (0, 15, x, x, 15, x, x) – 2 tags, Ethertype don't care

*Where the text presently reads:*

**Filter inner priority**: (4 bits)

0..7 Filter received frames on this inner priority value.

8 Do not filter on inner priority.

14 This is the default filter when no other one-tag rule applies.

15 This entry is a no-tag rule; ignore all other filter fields.

Other values: reserved

*Revise it to read:*

**Filter inner priority**: (4 bits)

0..7 Filter received frames on this inner priority value.

8 Do not filter on inner priority.

14 This is the default filter when no other one-tag rule applies.

15 This entry is a no-tag rule; ignore all other VLAN tag filter fields.

NOTE – the Ethertype filter field is a valid criterion.

Other values: reserved

*To the list of actions, add*

Set table (optional)

### 9.3.16 Dot1X performance monitoring history data

*Replace the threshold crossing alert table with the following:*

**Threshold crossing alert**

| Number | Threshold crossing alert | Threshold value attribute # (Note) |
|---|---|---|
| 0..3 | Reserved | |
| 4 | Invalid EAPOL frames received | 5 |
| 5..8 | Reserved | |
| 9 | EAP length error frames received | 10 |
| NOTE – This number associates the TCA with the specified threshold value attribute of the threshold data 1 managed entity. | | |

### 9.3.17 Radius performance monitoring history data

*Replace the threshold crossing alert table with the following:*

**Threshold crossing alert**

| Number | Threshold crossing alert | Threshold value attribute # (Note) |
|---|---|---|
| 0 | Reserved | |
| 1 | Retransmission count | 2 |
| 2..4 | Reserved | |
| 5 | Invalid radius packets received | 6 |
| NOTE – This number associates the TCA with the specified threshold value attribute of the threshold data 1 managed entity. | | |

### 9.3.18 Dot1 rate limiter

*Revise the descriptions of the three following attributes to read as shown (change* GEM traffic descriptor *to* traffic descriptor*).*

**Upstream unicast flood rate pointer**: This attribute points to an instance of the traffic descriptor that specifies the maximum rate of upstream unicast packets whose destination address is unknown to the bridge. A null pointer specifies that no administrative limit is to be imposed. (R, W, Set-by-create) (optional) (2 bytes)

**Upstream broadcast rate pointer**: This attribute points to an instance of the traffic descriptor that specifies the maximum rate of upstream broadcast packets. A null pointer specifies that no administrative limit is to be imposed. (R, W, Set-by-create) (optional) (2 bytes)

**Upstream multicast payload rate pointer**: This attribute points to an instance of the traffic descriptor that specifies the maximum rate of upstream multicast payload packets. A null pointer specifies that no administrative limit is to be imposed. (R, W, Set-by-create) (optional) (2 bytes)

### 9.3.21 Dot1ag default MD level

*To the list of actions, add*

Set table (optional)

### 9.3.22 Dot1ag MEP

*Revise the description of the following attribute to read:*

**CCM and LTM priority**: Ranging from 0..7, this attribute permits CCM and LTM frames to be explicitly prioritized, which may be needed if flows are separated eg by 802.1p priority. The priority specified in this attribute is also used in LTR frames originated by this MEP. The value 0xFF selects the 802.1ag default, whereby CCM and LTM frames are transmitted with the highest Ethernet priority available. (R, W, Set-by-create) (mandatory) (1 byte)

*Revise the description of the test action to read as follows:*

Test: The test operation causes the MEP to originate one or more loopback messages (LBMs) or a linktrace message (LTM) in accordance with the message format defined in clauses II.2.27 and II.2.45 (baseline format) and clauses II.3.27 and II.3.45 (extended format).

The link trace test returns its results in a general purpose buffer ME, which must have been created in advance by the OLT. (The general purpose buffer is designated by a pointer in the test message itself.) Upon completion of the linktrace operation, the general purpose buffer contains a sequence of LTR entries in the order they were received:

*Remainder of test action description unchanged.*

### 9.3.27 Multicast operations profile

*To the list of actions, add*

Set table (optional)

### 9.4.3 IP port configuration data

*Add code point 10 to the TP type attribute:*

**TP type**: This attribute specifies the type of termination point associated with this IP port.

1 PPTP Ethernet UNI
2 GEM interworking TP
3 MAC bridge port configuration data
4 802.1p mapper service profile
5 Physical path termination point xDSL UNI
6 Multicast GEM interworking termination point
7 Physical path termination point MoCA UNI
8 Physical path termination point 802.11 UNI
9 Ethernet flow termination point
10 Virtual Ethernet interface point

(R, W, Set-by-create) (mandatory) (1 byte)

### 9.4.12 IP host config data

*Replace the following attribute descriptions to read as shown:*

**Managed entity id**: This attribute uniquely identifies each instance of this managed entity. The ONT creates as many instances as there are independent IP stacks on the ONT. To facilitate discovery by the OLT, IP host config data MEs should be numbered from 0 upward. (R) (mandatory) (2 bytes)

**IP options**: This attribute is a bit map that enables or disables IP related options. The value 1 enables the option while 0 disables it. The default value of this attribute is 0.

0x01 Enable DHCP
0x02 Respond to pings
0x04 Respond to traceroute messages
0x08 Enable IP stack
0x10..0x80 Reserved

(R, W) (mandatory) (1 byte)

Several attributes of this managed entity may be paired together into two categories, manual settings and current values.

| Manual settings | Current values |
|---|---|
| IP address | Current address |
| Mask | Current mask |
| Gateway | Current gateway |
| Primary DNS | Current primary DNS |
| Secondary DNS | Current secondary DNS |

While the IP stack is disabled, there is no IP connectivity to the external world from this managed entity instance.

While DHCP is disabled, the current values are always the same as the manual settings. While DHCP is enabled, the current values are those assigned by DHCP, or undefined (0) if DHCP has never assigned values.

**IP address**: The address used for IP host services, this attribute has default value 0. (R, W) (mandatory) (4 bytes)

**Mask**: The subnet mask for IP host services, this attribute has default value 0. (R, W) (mandatory) (4 bytes)

**Gateway**: The default gateway address used for IP host services, this attribute has default value 0. (R, W) (mandatory) (4 bytes)

**Primary DNS**: The address of the primary DNS server, this attribute has default value 0. (R, W) (mandatory) (4 bytes)

**Secondary DNS**: The address of the secondary DNS server, this attribute has default value 0. (R, W) (mandatory) (4 bytes)

**Current address**: Current address of the IP host service. (R) (optional) (4 bytes)

**Current mask**: Current subnet mask for the IP host service. (R) (optional) (4 bytes)

**Current gateway**: Current default gateway address for the IP host service. (R) (optional) (4 bytes)

**Current primary DNS**: Current primary DNS server address. (R) (optional) (4 bytes)

**Current secondary DNS**: Current secondary DNS server address. (R) (optional) (4 bytes)

## 9.5 Ethernet services

*Replace the introductory section with the following:*

This clause defines managed entities associated with physical and virtual Ethernet UNIs, as shown in figure 9.5-1.

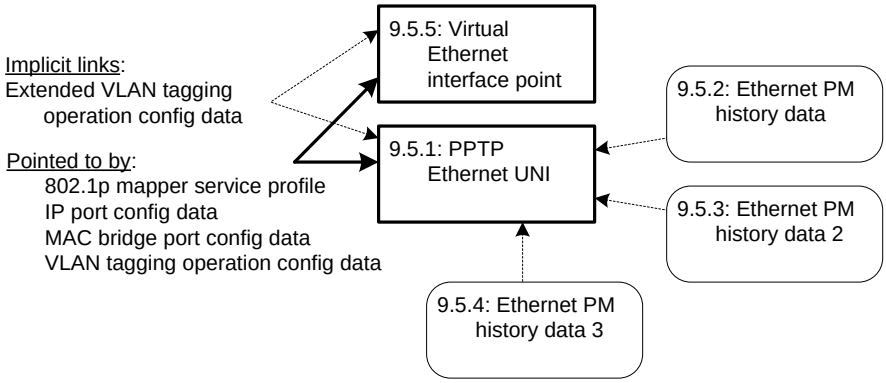



**Figure 9.5-1 – Managed entities associated with physical Ethernet UNIs**

### 9.5.1 Physical path termination point Ethernet UNI

*Revise the following attribute description to read as shown:*

**Max frame size**: This attribute denotes the maximum frame size allowed across this interface. Upon ME instantiation, the ONT sets the attribute to 1518. (R, W) (mandatory for GPON, optional for GbE) (2 bytes)

**DTE or DCE ind**: This attribute specifies the Ethernet interface wiring:

0 DCE or MDI-X (default).

1 DTE or MDI.

2 Automatic selection

(R, W) (mandatory) (1 byte)

### 9.5.5 Virtual Ethernet interface point

*Add the following new clause:*

This managed entity represents the data plane hand-off point in an ONT or ONU to a separate (non-OMCI) management domain. The virtual Ethernet interface point is managed by OMCI, and is potentially known to the non-OMCI management domain. One or more Ethernet traffic flows are present at this boundary.

Instances of this managed entity are automatically created and deleted by the ONT. This is necessary because the required downstream priority queues are subject to physical implementation constraints. The OLT may use one or more of the virtual Ethernet interface points created by the ONT.

It is expected that the ONT would create one virtual Ethernet interface point for each non-OMCI management domain.

*Relationships*

An instance of this managed entity is associated with an instance of a virtual Ethernet interface between OMCI and non-OMCI management domains.

*Attributes*

**Managed entity id**: This attribute uniquely identifies each instance of this managed entity. To facilitate easy discovery, the ONT should assign IDs in the sequence 1, 2, …. The values 0 and 0xFFFF are reserved. (R) (mandatory) (2 bytes)

**Administrative state**: This attribute locks (1) and unlocks (0) the functions performed by this managed entity. When the administrative state is set to lock, all user functions of this UNI are blocked, and alarms, TCAs and AVCs for this managed entity and all dependent managed entities are no longer generated. (R, W) (mandatory) (1 byte)

**Operational state**: This attribute indicates whether or not the managed entity is capable of performing its function. Valid values are enabled (0) and disabled (1). (R) (optional) (1 byte)

**Interdomain name**: This attribute is a character string that provides an optional way to identify the virtual Ethernet interface point to a non-OMCI management domain. The interface may also be identified by its managed entity ID, IANA assigned port and possibly other ways. If the vendor offers no information in this attribute, it should be set to a sequence of null bytes. (R, W) (optional) (25 bytes)

**TCP/UDP pointer**: This attribute points to an instance of the TCP/UDP config data managed entity, which provides for OMCI management of the non-OMCI management domain's IP connectivity. If no OMCI management of the non-OMCI domain's IP connectivity is required, this attribute may be omitted or set to its default, a null pointer. (R, W) (optional) (2 bytes)

**IANA assigned port**: This attribute contains the TCP or UDP port value as assigned by IANA for the management protocol associated with this virtual Ethernet interface. This attribute is to be regarded as a hint, not as a requirement that management communications use this port; the actual port and protocol are specified in the associated TCP/UDP config data managed entity. If no port has been assigned, or if the management protocol is free to be chosen at run-time, this attribute should be set to 0xFFFF. (R) (mandatory) (2 bytes)

*Actions*

Get, set

*Notifications*

**Attribute value change**

| Number | Attribute value change | Description |
|---|---|---|
| 0..1 | N/A | |
| 2 | Op state | Operational state |
| 3 | N/A | |
| 4..16 | Reserved | |

**Alarm**

| Number | Alarm | Description |
|---|---|---|
| 0 | Connecting function fail | Indicates a failure of the connecting function. May be used to signal faults from the non-OMCI management domain into OMCI. |
| 1..207 | Reserved | |
| 208..223 | Vendor specific alarms | Not to be standardized |

### 9.7.1 Physical path termination point xDSL UNI part 1

*Revise NOTE 1 in the Alarm table to read:*

NOTE 1 – The data rate upshift and downshift alarms are deprecated. They are not defined in [ITU-T G.997.1].

### 9.7.6 VDSL2 line configuration extensions

*To the list of actions, add*

Set table (optional)

### 9.7.10 xDSL PSD mask profile

*To the list of actions, add*

Set table (optional)

### 9.7.11 xDSL downstream RFI bands profile

*To the list of actions, add*

Set table (optional)

### 9.7.17 VDSL2 line inventory and status data part 2

*Revise the introductory text and relationships sections to read:*

This managed entity extends the xDSL line configuration MEs. The ME name was chosen because its attributes were initially unique to G.993.2 VDSL2. Due to continuing standards development, some attributes – and therefore this managed entity – have also become applicable to other recommendations, specifically G.992.3 and G.992.5.

This ME contains upstream attributes.

*Relationships*

This is one of the status data MEs associated with an xDSL UNI. It is meaningful if the PPTP supports G.992.3, G.992.5 or G.993.2. The ONT automatically creates or deletes an instance of this managed entity upon creation and deletion of a physical path termination point xDSL UNI part 1 that supports these attributes.

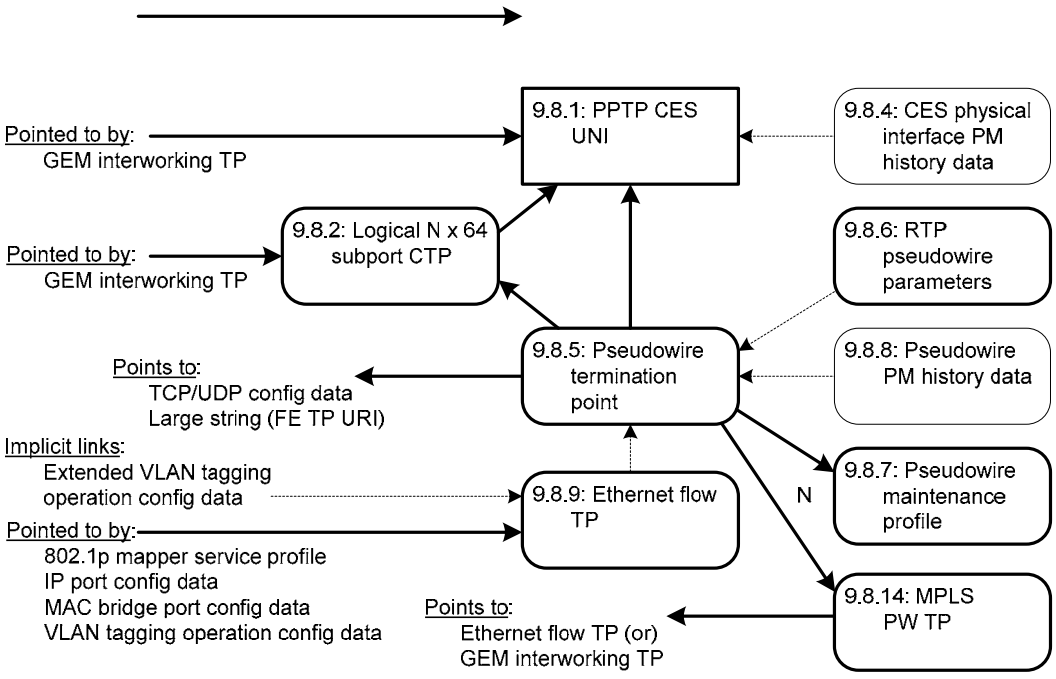
### 9.7.26 VDSL2 line configuration extensions 2

*To the list of actions, add*

Set table (optional)

## 9.8 TDM services

Replace the introductory figure with the following:



### 9.8.1 Physical path termination point CES UNI

*Revise the relationships section to read :*

> An instance of this managed entity is associated with each real or preprovisioned CES port. It can be linked from a pseudowire or a TU TP, or a logical N x 64 kb/s CTP.

### 9.8.2 Logical N x 64kbit/s sub-port connection termination point

*Delete the second paragraph of the introductory text of this clause in G.984.4 (2008), ie the text that reads as follows:*

An instance of this managed entity is created by the OLT before the creation of an associated interworking termination point (see clause 9.3.6/G.984.4, GEM interworking termination point).

*Revise the relationships section to read :*

> Zero or more instances of this ME are associated with an instance of the physical path termination point CES UNI. It can be linked from a pseudowire or a TU TP.

### 9.8.5 Pseudowire termination point

*Revise the introductory paragraph to read:*

The pseudowire termination point supports packetized (rather than TDM) G-PON transport of TDM services, transported either directly over Ethernet, over UDP/IP or over MPLS. Instances of this managed entity are created and deleted by the OLT.

*Revise the underlying transport attribute to read as shown:*

> **Underlying transport**:
>
> 0 Ethernet, MEF8.
>
> 1 UDP/IP.
>
> 2 MPLS.
>
> (R, W, Set-by-create) (mandatory) (1 byte)

*Replace the near-end IP info attribute with the north-side pointer attribute:*

> **Near-end IP info**: When the pseudowire service is transported via IP, this attribute points to an instance of the TCP/UDP config data managed entity. The default value 0 is applicable if the pseudowire is not transported via IP. (R, W, Set-by-create) (mandatory for IP transport) (2 bytes)
>
> **North-side pointer**: When the pseudowire service is transported via IP, as indicated by the underlying transport attribute, the north-side pointer attribute points to an instance of the TCP/UDP config data managed entity. When the pseudowire service is transported directly over Ethernet, the north-side pointer attribute is not used – the linkage to the Ethernet flow termination point is implicit in the ME IDs. When the pseudowire service is transported over MPLS, the north-side pointer attribute points to an instance of the MPLS PW TP. (R, W, Set-by-create) (mandatory) (2 bytes)

*Add the following new ME:*

### 9.8.14 MPLS pseudowire termination point

This managed entity contains the underlying transport configure data of a pseudowire which underlying transportation method is MPLS. Instances of this managed entity are created and deleted by the OLT.

*Relationships*

Zero or one instance of this managed entity is associated with each instance of pseudowire termination point managed entity.

*Attributes*

**Managed entity id**: This attribute uniquely identifies each instance of this managed entity. (R, Set-by-create) (mandatory) (2 bytes)

**TP type**: This attribute specifies the type of ANI-side termination point associated with this managed entity.

1 Ethernet flow termination point

2 GEM interworking TP

(R, W, Set-by-create) (mandatory) (1 byte)

**TP pointer**: This attribute points to the instance of the TP associated with this MPLS PW TP. The type of the associated TP is determined by the TP type attribute. (R, W, Set-by-create) (mandatory) (2 bytes)

**MPLS label indicator**: This attribute specifies the MPLS label stacking situation.

0 Single MPLS labelled

1 Double MPLS labelled

(R, W, Set-by-create) (mandatory) (1 byte)

**MPLS PW direction**: This attribute specifies the inner MPLS direction.

0 Upstream only

1 Downstream only

2 Bidirectional

(R, W, Set-by-create) (mandatory) (1 byte)

**MPLS PW uplink label**: This attribute specifies the label of the inner MPLS pseudowire upstream. The attribute is not meaningful for unidirectional downstream PWs. (R, W, Set-by-create) (mandatory) (4 bytes)

**MPLS PW downlink label**: This attribute specifies the label of inner MPLS pseudowire downstream. The attribute is not meaningful for unidirectional upstream PWs. (R, W, Set-by-create) (mandatory) (4 bytes)

**MPLS PW EXP**: This attribute specifies the inner MPLS EXP value in the upstream direction. The attribute is not meaningful for unidirectional downstream PWs. (R, W, Set-by-create) (mandatory) (1 byte)

**MPLS tunnel direction:** This attribute specifies the direction of the (outer) MPLS tunnel.

0 Upstream only

1 Downstream only

2 Bidirectional

(R, W, Set-by-create) (mandatory) (1 byte)

**MPLS tunnel uplink label:** This attribute specifies the (outer) label for the upstream MPLS tunnel. If the MPLS tunnel is downstream only, this attribute should be set to 0. (R, W, Set-by-create) (mandatory) (4 bytes)

**MPLS tunnel downlink label:** This attribute specifies the (outer) label for the downstream MPLS tunnel. If the MPLS tunnel is upstream only, this attribute should be set to 0. (R, W, Set-by-create) (mandatory) (4 bytes)

**MPLS tunnel EXP:** This attribute specifies the EXP value of the upstream MPLS tunnel. If the MPLS tunnel is downstream only, this attribute should be set to 0. (R, W, Set-by-create) (mandatory for double MPLS labelled case) (1 byte)

*Actions*

Create, delete, get, set

*Notifications*

None

### 9.9.1 Physical path termination point POTS UNI

*Revise the description of the following attribute to read:*

**Interworking TP pointer**: This attribute points to the associated instance of the GEM interworking termination point managed entity. The value 0 is a null pointer. This attribute is not used in VoIP and should not be supported. (R, W) (optional) (2 bytes)

*Revise the description of the test action to read as follows:*

**Test**: Request that the ONT perform one or more MLT tests or a dial tone make/break test. Vendor-specific tests are also supported by the test and test result message layouts in II.2.27 and II.2.45 (baseline message format) and in II.3.27 and II.3.45 (extended message format).

### 9.9.3 SIP agent config data

*Add the following attributes:*

**SIP response table**: This attribute specifies the tone and text to be presented to the subscriber upon reception of various SIP messages (normally 4xx, 5xx, 6xx message codes). The table is a sequence of entries, each of which is defined as follows:

**SIP response code** (2 bytes): This field is the index into the SIP response table. When a set operation is performed with the value 0 in this field, the table is cleared.

**Tone** (1 byte): This field specifies one of the tones in the tone pattern table of the associated VoIP application service profile. The specified tone is played to the subscriber.

**Text message** (2 bytes): This field is a pointer to a large string that contains a message to be displayed to the subscriber. If the value of this field is a null pointer, text pre-associated with the tone may be displayed, or no text at all.

(R, W) (optional) (N * 5 bytes)

**SIP option transmit control**: This Boolean attribute specifies that the ONT is (true) or is not (false) enabled to transmit SIP options. The default value is false. (R, W, Set-by-create) (optional) (1 byte)

**SIP URI format**: This attribute specifies the format of the URI in outgoing SIP messages. The default value 0 specifies TEL URIs; the value 1 specifies SIP URIs. Other values are reserved. (R, W, Set-by-create) (optional) (1 byte)

*To the list of actions, add*

Set table (optional)

### 9.9.4 VoIP voice CTP

*Revise the description of the user protocol pointer attribute to read as follows (underlined text is new):*

**User protocol pointer**: This attribute points to signalling protocol data. If the signalling protocol used attribute of the VoIP config data managed entity specifies that the ONT's signalling protocol is SIP, this attribute points to a SIP user data ME, which in turn points to a SIP agent config data. If the signalling protocol is H.248, this attribute points directly to an MGC config data ME. (R, W, Set-by-create) (mandatory) (2 bytes)

### 9.9.6 Voice service profile

*Revise the following attribute description to read*

**Jitter target**: This attribute specifies the target value of the jitter buffer in milliseconds. The system tries to maintain the jitter buffer at the target value. The value 0 specifies dynamic jitter buffer sizing. (R, W, Set-by-create) (optional) (2 bytes)

*Add the following attributes:*

**DTMF digit levels**: This attribute specifies the power level of DTMF digits that may be generated by the ONT toward the subscriber set. It is a 2s complement value referred to 1 mW at the 0 TLP (dBm0), with resolution 1 dB. The default value 0x8000 selects the ONT's internal policy. (R, W, Set-by-create) (optional) (2 bytes)

**DTMF digit duration**: This attribute specifies the duration of DTMF digits that may be generated by the ONT toward the subscriber set. It is specified in milliseconds. The default value 0 selects the ONT's internal policy. (R, W, Set-by-create) (optional) (2 bytes)

**Hook flash minimum time**: This attribute defines the minimum duration recognized by the ONT as a switchhook flash. It is expressed in milliseconds; the default value 0 selects the ONT's internal policy. (R, W, Set-by-create) (optional) (2 bytes)

**Hook flash maximum time**: This attribute defines the maximum duration recognized by the ONT as a switchhook flash. It is expressed in milliseconds; the default value 0 selects the ONT's internal policy. (R, W, Set-by-create) (optional) (2 bytes)

**Tone pattern table**: This attribute is a table each of whose entries specifies a complex tone (or silence) and a duration. By linking tones and silence together, possibly cyclically, continuous, varying or interrupted tone sequences, repetitive or not, may be defined. A tone sequence is initiated by pointing to the first tone pattern table entry that defines its parameters. Each entry is a vector comprising the following components:

**Index** (1 byte): This component is simply an index into the table. It ranges from 1..255. In a set operation, the value 0 in this field clears the table.

**Tone on** (1 byte): This boolean component controls whether the tone is on (true) or off. If the tone is off, the frequency and power fields are not meaningful.

**Frequency 1** (2 bytes): This component specifies the frequency of one of the tone components in Hz.

**Power 1** (1 byte): This component specifies the power level of the corresponding frequency component. It ranges from 0..25.5 dBm0 with 0.1 dB resolution.

Three additional pairs of frequency-power components may be specified to define a complex tone. If a pair of possibilities is not to be used, its frequency field should be set to 0.

**Frequency 2** (2 bytes)

**Power 2** (1 byte)

**Frequency 3** (2 bytes)

**Power 3** (1 byte)

**Frequency 4** (2 bytes)

**Power 4** (1 byte)

The following pair of frequency-power components allows the composite tone to be modulated (warble effect). If this effect is not to be used, the frequency should be set to 0.

**Modulation frequency** (2 bytes), Hz

**Modulation power** (1 byte), 0..25.5 dBm0

**Duration** (2 bytes): This component specifies the duration of the phase, in milliseconds. The value 0 specifies that the phase endures indefinitely, that is, until terminated by other events such as call abandonment.

**Next entry** (1 byte): This component is a pointer to another entry in this same table, which permits sequences of tones to be defined, possibly cyclically. A reference to a non-existent table entry, or the value 0, indicates that the sequence should be terminated.

(R, W) (optional) (N * 20 bytes).

**Tone event table**: This attribute is a table each of whose entries specifies an event for which a tone is defined. If the tone can be synthesized by a sequence of complex tones and silence, the event refers to an entry in the tone pattern table. Otherwise, the event refers to a file name that is expected to be recognized by the ONT environment. Each entry in the tone event table is a vector comprising the following components:

**Event** (1 byte): This component is an emumeration of the events for which a tone may be defined. The event component also serves as the index for the table. A set operation to event 0 causes the table to be cleared.

| Value | Tone event |
|---|---|
| 0 | Not used for get operation; clears table under set operation |
| 1 | Busy |
| 2 | Confirmation |
| 3 | Dial |
| 4 | Message waiting |
| 5 | Off hook warning (receiver off hook) |
| 6 | Ringback (audible ring) |
| 7 | Reorder |
| 8 | Stutter dial |
| 9 | Call waiting 1 |
| 10 | Call waiting 2 |
| 11 | Call waiting 3 |
| 12 | Call waiting 4 |
| 13 | Alerting signal |
| 14 | Special dial |
| 15 | Special info |
| 16 | Release |
| 17 | Congestion |
| 18 | User defined 1 |
| 19 | User defined 2 |
| 20 | User defined 3 |
| 21 | User defined 4 |
| 22..32 | Reserved |
| 33 | Intrusion |
| 34 | Dead tone |
| 35..223 | Reserved |
| 224..255 | Vendor specific codes, not to be standardized |

**Tone pattern** (1 byte): This component specifies an entry point into the tone pattern table attribute, to be invoked when the specified event occurs. The value 0 indicates that no tone from the tone pattern table is to be played.

**Tone file** (2 bytes): This component points to a large string managed entity that contains the path and name of a file containing a codec sequence to be

played out. If no file is found after traversing these links, no tone is played. The behaviour is unspecified if both tone pattern and tone file are specified.

**Tone file repetitions** (1 byte): This component specifies the number of times the tone file is to be repeated. The value 0 means that the file is to be repeated indefinitely until terminated by some external event such as call abandonment.

**Reverved** (2 bytes)

(R, W) (optional) (N * 7 bytes).

**Ringing pattern table**: This attribute is a table each of whose entries specifies a ringing pattern and a duration. By linking ringing and silence together, possibly cyclically, continuous or interrupted ringing sequences, repetitive or not, may be defined. A ringing sequence is initiated by pointing to the first ringing pattern table entry that defines its parameters. Each entry is a vector comprising the following components:

**Index** (1 byte): This component is simply an index into the table. It ranges from 1..255. In a set operation, the value 0 in this field clears the table.

**Ringing on** (1 byte): This boolean component controls whether ringing is on (true) or off during this interval.

**Duration** (2 bytes): This component specifies the duration of the ringing phase, in milliseconds. The value 0 specifies that the phase endures indefinitely, that is, until terminated by other events such as call abandonment.

**Next entry** (1 byte): This component is a pointer to another entry in this same table, which permits sequences of ringing bursts to be defined, possibly cyclically. A reference to a non-existent table entry, or the value 0, indicates that the sequence should be terminated.

(R, W) (optional) (N * 5 bytes).

**Ringing event table**: This attribute is a table each of whose entries specifies an event for which a ringing sequence is defined. If the ringing sequence can be generated as a sequence of power ringing and silent intervals, the event refers to an entry in the ringing pattern table. Otherwise, the event refers to a file name that is expected to be recognized by the ONT environment. Each entry is a vector comprising the following components:

**Event** (1 byte): This component is an emumeration of the events for which a ringing sequence may be defined. The event component also serves as the index for the table. A set operation with the value 0 in this field causes the table to be cleared.

| Value | Tone event |
|---|---|
| 0 | Not used for get operation; clears table under set operation |
| 1 | Default |
| 2 | Splash |
| 3..223 | Reserved |
| 224..255 | Vendor specific codes, not to be standardized |

**Ringing pattern** (1 byte): This component specifies an entry point into the ringing pattern table attribute, to be invoked when the specified event occurs. The value 0 indicates that no ringing sequence is defined in the ringing pattern table.

**Ringing file** (2 bytes): This component points to a large string managed entity that contains the path and name of a file containing a ring tone to be played out. If no file is found after traversing these links, no ringing is played. The behaviour is unspecified if both ringing pattern and ringing file fields are specified.

**Ringing file repetitions** (1 byte): This component specifies the number of times the ringing file is to be repeated. The value 0 means that the file is to be repeated indefinitely until terminated by some external event such as call abandonment.

**Ringing text** (2 bytes): This component points to a large string managed entity that contains a text string to be displayed on the CPE device in conjunction with this event. A null pointer indicates that no text is to be displayed.

(R, W) (optional) (N * 7 bytes).

*To the list of actions, add*

Set table (optional)

### 9.9.9 VoIP feature access codes

*Add the following attributes:*

**Unattended/blind call transfer**: (R, W) (optional) (5 bytes)

**Attended call transfer**: (R, W) (optional) (5 bytes)

### 9.9.10 Network dial plan table

*To the list of actions, add*

Set table (optional)

### 9.9.13 RTP performance monitoring history data

*Replace the list of TCAs as shown:*

**Threshold crossing alert**

| ~~**Number**~~ | ~~**Threshold crossing alert**~~ | ~~**Threshold value attribute # (Note 2)**~~ |
|---|---|---|
| ~~0~~ | ~~RTP-PM RTP packet loss (Note 1)~~ | ~~1~~ |
| ~~1~~ | ~~RTP-PM packet jitter~~ | ~~2~~ |
| ~~2~~ | ~~RTP-PM no RTCP packet~~ | ~~3~~ |
| ~~3~~ | ~~RTP-PM timeout~~ | ~~4~~ |
| ~~4~~ | ~~RTP-PM buffer underflows~~ | ~~6~~ |
| ~~5~~ | ~~RTP-PM buffer overflows~~ | ~~7~~ |

**Threshold crossing alert**

| Number | Threshold crossing alert | Threshold value attribute # (Note 2) |
|---|---|---|
| 0 | RTP errors (note 1) | 1 |
| 1 | Packet loss | 2 |
| 2 | Maximum jitter | 3 |
| 3 | Max time between RTCP packets | 4 |
| 4 | Buffer underflows | 6 |
| 5 | Buffer overflows | 7 |

### 9.9.21 Physical path termination point ISDN UNI

*Revise the description of the test action to read as follows:*

**Test**: Request that the ONT perform one or more MLT tests. See test and test result message layouts in clauses II.2.27 and II.2.45 (baseline message format) and in II.3.27 and II.3.45 (extended message format).

### 9.11.1 Priority queue-G

*Replace the introductory paragraphs with the following:*

This managed entity specifies the priority queue used by a GEM port network CTP. In the upstream direction, a priority queue-G ME is also related to a T-CONT ME. By default, this relationship is fixed by the ONT hardware architecture, but some ONTs may also permit the relationship to be configured through OMCI, as indicated by the QoS configuration flexibility attribute of the ONT2-G managed entity.

In the downstream direction, priority queues are associated with UNIs. Again, the association is fixed by default, but some ONTs may permit the association to be configured through OMCI.

The OLT can find all the queues by reading the priority queue-G managed entity instances. If the OLT tries to retrieve a non-existent priority queue, the ONT denies the get action with an error indication.

See also Appendix III.

Upstream priority queues can be added to the ONT. Moreover, priority queues can exist in the ONT core and circuit packs serving both UNI and ANI functions.

In the upstream direction, the weight attribute permits configuring an optional traffic scheduler. Several attributes support back pressure operation, whereby a back pressure signal is sent backward and causes the attached terminal to temporarily suspend sending data.

The yellow packet drop thresholds are used to specify the packet drop probability for a packet that has been marked yellow (drop eligible) by a traffic descriptor or by external equipment such as a residential gateway. If the current queue occupancy is less than the min threshold, the yellow packet drop probability is zero. If the current queue occupancy is greater than or equal to the max threshold, the yellow packet drop probability is one. Otherwise, the yellow drop probability increases linearly between 0 and max_p as the current queue occupancy increases from the min to the max threshold.

Drop precedence colour marking indicates the method by which a packet is marked as drop eligible (yellow). For DEI and PCP marking, a drop eligible indicator is equivalent to yellow colour,

otherwise the colour is green. For DSCP AF marking, the lowest drop precedence is equivalent to green, otherwise the colour is yellow.

*Revise the relationships section to read as follows:*

*Relationships*

One or more instances of this managed entity are associated with the ONT-G managed entity to model the upstream priority queues if the traffic management option attribute in the ONT-G ME is 0 or 2.

One or more instances of this managed entity are associated with a physical path termination point UNI managed entity as downstream priority queues. Downstream priority queues may or may not be provided for a virtual Ethernet interface point.

*Revise the following attribute descriptions to read as follows:*

**Related port**: This attribute represents the slot, port/T-CONT and priority information associated with the instance of priority queue-G ME. This attribute comprises four bytes.

In the upstream direction, the first two bytes are the ME ID of the associated T-CONT, the first byte of which is a slot number, the second byte a T-CONT number. In the downstream direction, the first byte is the slot number and the second byte is the port number of the queue's destination port.

The last two bytes represent the priority of this queue. The range of priority is 0 to 0x0FFF. The value 0 indicates the highest priority and 0x0FFF indicates the lowest priority. (R, W) (mandatory) (4 bytes)

NOTE 1 – If flexible configuration is supported, the related port attribute is meaningful only if the traffic scheduler-G pointer attribute value is null. Otherwise, the related port attribute is ignored.

NOTE 2 – The related port attribute is read-only unless otherwise specified by the QoS configuration flexibility attribute of the ONT2-G managed entity. Even if flexibility is supported, only the second byte, the port or T-CONT number, may be changed. The OMCI set command must contain four bytes to match the attribute size, but the ONT must ignore all except the second byte.

If flexible configuration is not supported, the ONT should reject an attempt to set the related port with a parameter error result-reason code.

**Traffic scheduler-G pointer**: This attribute points to the traffic scheduler-G ME instance that is associated with this priority queue. This pointer is used when this priority queue is connected with a traffic scheduler. The default value is null (0). (R, W) (mandatory) (2 bytes)

NOTE 3 – When the QoS configuration flexibility attribute of the ONT2-G managed entity allows flexible assignment of the traffic scheduler, the OLT may configure the traffic scheduler-G pointer to refer to any traffic scheduler in the same slot.

If traffic scheduler flexibility is not permitted by the QoS configuration flexibility attribute, the OLT may use the traffic scheduler-G pointer attribute only by pointing to another traffic scheduler-G ME that is associated with the same T-CONT as the priority queue itself.

The ONT should reject an attempt to violate these conditions with a parameter error result-reason code.

*Revise the following attribute description to read as shown (change* GEM traffic descriptor *to* traffic descriptor*).*

**Drop precedence colour marking**: This attribute specifies how the drop precedence is marked on the ingress packets to the priority queue. The default value is 0.

0 No marking (treat all packets as green)

1 Internal marking (from traffic descriptor ME)

2 DEI (802.1ad)

3 PCP 8P0D (802.1ad)

4 PCP 7P1D (802.1ad)

5 PCP 6P2D (802.1ad)

6 PCP 5P3D (802.1ad)

7 DSCP AF class (RFC 2597 [b-~~ITU-T G.984.4.4~~RFC 2597])

(R, W) (optional) (1 byte)

### 9.11.2 Traffic scheduler-G

*Revise the T-CONT pointer attribute description to read as follows:*

**T-CONT pointer**: This attribute points to the T-CONT ME instance associated with this traffic scheduler. This pointer is used when this traffic scheduler is connected to the T-CONT directly. It is null (0) otherwise. (R, W) (mandatory) (2 bytes)

NOTE – This attribute is read-only unless otherwise specified by the QoS configuration flexibility attribute of the ONT2-G managed entity. If flexible configuration is not supported, the ONT should reject an attempt to set the T-CONT pointer attribute with a parameter error result-reason code.

*Revise the policy attribute description to read as follows:*

**Policy**: This attribute represents scheduling policy. Valid values include

0 Null

1 HOL (head of line)

2 WRR (weighted round robin)

(R, W) (mandatory) (1 byte)

NOTE – This attribute is read-only unless otherwise specified by the QoS configuration flexibility attribute of the ONT2-G managed entity. If flexible configuration is not supported, the ONT should reject an attempt to set the policy attribute with a parameter error result-reason code.

### 9.11.3 GEM traffic descriptor

*Revise the heading of this clause to read*: **9.11.3 Traffic descriptor**

*Revise the first introductory paragraph to read as shown (change* GEM traffic descriptor *to* traffic descriptor*).*

The traffic descriptor allows for traffic management. A priority controlled ONT can point from a MAC bridge port configuration data ME to a traffic descriptor in order to implement traffic management (marking, policing). A rate controlled ONT can point to a traffic descriptor from either

a MAC bridge port configuration data ME or GEM port network CTP to implement traffic management (marking, shaping).

### 9.12.1 UNI-G

*Add the following note at the end of the Administrative state attribute description:*

NOTE – PPTP MEs also have an administrative state attribute. The user port is unlocked only if both administrative state attributes are set to unlocked.

*Add the following new attributes:*

**Management capability**: An ONT may support the ability for some or all of its PPTPs to be managed either directly by OMCI or from non-OMCI management environment such as TR-69. This attribute advertises the ONT's capabilities for each PPTP.

This attribute is an enumeration with the following code points:

0 OMCI only

1 Non-OMCI only. In this case, the PPTP may be visible to OMCI, but only in a read-only sense, eg for PM collection.

2 Both OMCI and non-OMCI

(R) (optional) (1 byte)

**Non-OMCI management identifier**: If a PPTP can be managed either directly by OMCI or a non-OMCI management environment, this attribute specifies how it is in fact to be managed. This attribute is either 0 (default = OMCI management), or it is a pointer to a virtual Ethernet interface point, which in turn links to a non-OMCI management environment. (R, W) (optional) (2 bytes)

### 9.12.2 OLT-G

*Add the following new attribute:*

**Time of day information:** This attribute provides the information required to achieve time of day synchronization between a reference clock at the OLT and a local clock at the ONT. This attribute comprises two fields: the first field (4 bytes) is the sequence number of the specified GEM superframe. The second field (10 bytes) is $Tstamp_N$ as defined in G.984.3, clause 10.4.6, using the timestamp format of IEEE 1588-2008, clause 5.3.3. The value 0 in all bytes is reserved as a null value. (R, W) (optional) (14 bytes)

**Ed note:** check and correct reference to 984.3 clause

### 9.12.4 Authentication security method

*Where the text presently reads:*

**Validation scheme**: This attribute specifies the validation scheme used when the ONT validates a challenge. Validation schemes are defined as follows:

…

1 Validate using MD5 digest authentication as defined in RFC 2069 (recommended)

*Revise it to read:*

**Validation scheme**: This attribute specifies the validation scheme used when the ONT validates a challenge. Validation schemes are defined as follows:

…

1 Validate using MD5 digest authentication as defined in RFC 2617 (recommended)

*RFC 2617 obsoletes RFC 2069. RFC 2069 is also therefore to be removed from the reference list in clause 2.*

### 9.12.6 Threshold data 1

*Delete the last sentence of the first paragraph of the relationships section, ie the text in G.984.4 (2008). The resulting paragraph should read as follows:*

An instance of this managed entity may be related to multiple instances of performance monitoring history data type managed entities.

### 9.12.9 Managed entity

*Where the relationships section reads:*

One or more managed entity are related to the OMCI object entity.

*Revise it to read:*

One or more managed entity entities are related to the OMCI object entity.

| Ed note to TSB: “entity entities” is correct. |
|---|

### 9.12.13 File transfer controller

*Where the text presently reads:*

**Network address pointer**: This attribute is a pointer to a network address ME that specifies optional authentication information, along with the URI to be used for the file transfer. The URI should specify the protocol, one from the list of protocols supported by the ONT, and optionally a port. For unidirectional multicast download (eg DSM-CC), the URI should specify the multicast IP address (as a text string). (R, W) (mandatory) (2 bytes)

**File transfer trigger**: This attribute causes the file transfer to begin. If a given set operation writes values to several attributes of this managed entity, the ONT should apply the file transfer trigger after updating all other attributes. Some operations may not be applicable to some files; the ONT should deny commands that request unsupported actions. (R, W) (mandatory) (1 byte)

| Value | Meaning |
|---|---|
| 0 | Reserved |
| 1 | Initiate file download (to the ONT) |
| 2 | Initiate file upload (from the ONT) |
| 3 | Abort current file transfer |
| 4 | Delete target file |
| 5..255 | Reserved |

**MC GEM IWTP pointer**: This attribute is a pointer that specifies the multicast GEM interworking termination point to be used for the transfer, assuming multicast protocol. (R, W) (optional) (2 bytes)

*Revise it to read:*

**Network address pointer**: This attribute is a pointer to a network address ME that specifies optional authentication information, along with the URI to be used for the file transfer. The URI should specify the protocol, one from the list of protocols supported by the ONT, an IP address or a string that can be resolved into an IP address, and optionally a port. For unidirectional multicast download (eg DSM-CC), the URI should specify a multicast IP source address. (R, W) (mandatory) (2 bytes)

**File transfer trigger**: This attribute causes the file transfer to begin. If a given set operation writes values to several attributes of this managed entity, the ONT should apply the file transfer trigger after updating all other attributes. Some operations may not be applicable to some files; the ONT should deny commands that request unsupported actions. (R, W) (mandatory) (1 byte)

| Value | Meaning |
|---|---|
| 0 | Reserved |
| 1 | Initiate file download (to the ONT) |
| 2 | Initiate file upload (from the ONT) |
| 3 | Abort current file transfer |
| 4 | Delete target file |
| 5 | Perform a directory listing operation. The scope of the directory is not specified; at the vendor's option, the listing may be filtered by matching some or all of file type, file instance and local file name attributes. |
| 6..255 | Reserved |

**GEM IWTP pointer**: This attribute is a pointer that specifies a unicast or multicast GEM interworking termination point, depending on whether the transfer protocol to be used is unicast or multicast. (R, W) (optional) (2 bytes)

*Add the following new attribute:*

**Directory listing table**: When a directory listing is complete, this attribute contains the result of a directory listing operation. The content and format of the table is not specified. (R) (optional) (N bytes)

*Add an AVC as shown:*

**Attribute value change**

| Number | Attribute value change | Description |
|---|---|---|
| 1..6 | N/A | |
| 7 | File transfer status | |
| 8..1610 | Reserved | |
| 11 | Directory listing table | This AVC signals the OLT that a directory listing operation is complete and may be retrieved with a get, get next sequence. |
| 12..16 | Reserved | |

### 9.12.14 Generic status portal

*Add the following new clause:*

The generic status portal managed entity provides a way for the OLT to discover the status and configuration information of a non-OMCI management domain within an ONT. The non-OMCI management domain is indicated by the virtual Ethernet interface point associated with this generic status portal.

The generic status portal ME uses two table attributes to convey status and configuration from a non-OMCI managed domain to OMCI. Each of these attributes uses an XML document to present this information. These XML documents are not required to be understood by the OLT or EMS. The schema for the documents may be used in the creation of tools that parse and interpret the contents of the document.

*Relationships*

One instance of this ME is created by the OLT for each separate non-OMCI management domain whose information is desired to be visible.

*Attributes*

**Managed entity ID**: This attribute uniquely identifies each instance of this managed entity. Through an identical ID, the generic status portal ME is implicitly linked to an instance of the virtual Ethernet interface point ME. (R, Set-by-create) (mandatory) (2 bytes). (R)

**Status document table**: This attribute is used to pass a textual representation of the non-OMCI managed domain status back to the OLT. The contents are vendor specific and formatted as an XML document. The first element of the document must contain an XML declaration indicating the version of XML and encoding used in the remainder of the document. The second element of the document must contain a schema reference to the vendor supplied schema used by the remainder of the document. The get, get next sequence must be used with this attribute since its size is unspecified. (R) (mandatory) (x bytes)

**Configuration document table**: This attribute is used to pass a textual representation of the non-OMCI managed domain configuration back to the OLT. The contents are vendor specific and formatted as an XML document. The first element of the document must contain an XML declaration indicating the version of XML and encoding used in the remainder of the document. The second element of the document must contain a schema reference to the vendor supplied schema used by the remainder of the document. The get, get next sequence must be used with this attribute since its size is unspecified. (R) (mandatory) (x bytes)

**AVC report rate**: This attribute governs the rate at which the generic status portal generates attribute value change notifications. The default value 0 disables AVCs, while the highest value 3, which may be useful for debugging, generates an AVC on every change seen in the non-OMCI management domain. As a guideline, the value 1 should collect changes into not more than one notification in ten minutes, while value 2 should generate an AVC not more than once per second. (R, W, Set-by-create) (optional) (1 byte)

*Actions*

Create, delete

Get, get next

*Notifications*

**Attribute value change**

| Number | Attribute value change | Description |
|---|---|---|
| 1 | Status document table | Indicates an update to the status document table from a non-OMCI interface. Because the attribute is a table, the AVC contains no information about its value. The OLT must use the get, get next action sequence if it wishes to obtain the updated attribute content. |
| 2 | Configuration document table | Indicates an update to the configuration document table from a non-OMCI interface. Because the attribute is a table, the AVC contains no information about its value. The OLT must use the get, get next action sequence if it wishes to obtain the updated attribute content. |
| 3..16 | Reserved | |

### 9.13.3 Physical path termination point LCT UNI

*Revise clause 9.13.3 to read as follows. The text in existing clause 9.13.3 after the administrative state attribute remains unchanged.*

This managed entity models debug access to the ONT from any physical or logical port, for example via a dedicated local craft terminal UNI, via ordinary subscriber UNIs, or via the IP host config ME.

The ONT automatically creates an instance of this managed entity per port:

- When the ONT has an LCT port built into its factory configuration.
- When a cardholder is provisioned to expect a circuit pack of LCT type.
- When a cardholder provisioned for plug-and-play is equipped with a circuit pack of LCT type. Note that the installation of a plug-and-play card may indicate the presence of LCT ports via equipment ID as well as its type, and indeed may cause the ONT to instantiate a port mapping package that specifies LCT ports.
- When the ONT supports debug access through some other physical or logical means.

The ONT automatically deletes an instance of this managed entity when a cardholder is neither provisioned to expect an LCT circuit pack, nor is it equipped with an LCT circuit pack, or if the ONT is reconfigured in such a way that it no longer supports debug access.

LCT instances are not reported during a MIB upload.

*Relationships*

An instance of this managed entity is associated with an instance of a real or virtual circuit pack managed entity classified as LCT type. An instance of this managed entity may also be associated with the ONT as a whole, if the ONT supports debug access through means other than a dedicated physical LCT port.

*Attributes*

**Managed entity id**: This attribute uniquely identifies each instance of this managed entity. This two-byte number indicates the physical position of the UNI. The first

byte is the slot id (defined in clause 9.1.5). The second byte is the port ID, with range 1..255. If the LCT UNI is associated with the ONT as a whole, its managed entity ID should be 0. (R) (mandatory) (2 bytes)

**Administrative state**: This attribute locks (1) and unlocks (0) the functions performed by this managed entity. When the administrative state is set to lock, debug access through all physical or logical means is blocked, except that the operation of a possible ONT remote debug ME is not affected. Administrative lock of ME instance 0 overrides administrative lock of any other PPTP LCT UNIs that may exist. Selection of a default value for this attribute is outside the scope of this Recommendation. (R, W) (mandatory) (1 byte)

### 9.13.4 Interworking VCC termination point

*Modify the Managed entity id attribute as shown.*

**Managed entity id**: This attribute uniquely identifies each instance of this managed entity. ~~It must be unique over all interworking VCC termination point and GEM interworking termination point MEs.~~ (R, Set-by-create) (mandatory) (2 bytes).

*Add the following ME.*

### 9.13.11 Enhanced security control

This managed entity contains the capabilities, parameters and controls of enhanced G-PON security features. The attributes in this ME are intended to be used to implement a symmetric-key-based three step authentication process as described in the supplemental information section below.

*Relationships*

One instance of this managed entity is associated with the ONT managed entity.

*Attributes*

**Managed entity id**: This attribute uniquely identifies each instance of this managed entity. There is only one instance, number 0. (R) (mandatory) (2 bytes)

**OLT crypto capabilities**: This attribute specifies the cryptographic mechanisms available at the OLT and is written by the OLT during authentication step 1. It is formatted as a bit map, where a 1 bit indicates that the particular algorithm is supported, and a 0 bit indicates it is not supported.

| Bit position | Algorithm supported |
|---|---|
| 1 (lsb=1, msb=128) | AES-CMAC-128 (support is mandatory) |
| 2 | HMAC-SHA-256 |
| 3 | HMAC-SHA-512 |
| 4-128 | Reserved |

(W) (mandatory) (16 bytes)

**OLT random challenge table**: This attribute specifies the random challenge issued by the OLT during authentication step 1. It is structured as a table, with each entry being 17 bytes. The first byte is the entry index, and the remaining 16 bytes are the content of the entry. In normal use, the OLT will write all the entries

in the table, and then trigger the ONT's processing of the entire table using the OLT challenge status attribute. The table size is known by the maximum index set by the OLT. The OLT can clear the table with a set operation to index 0. (R, W) (mandatory) (17*N bytes)

**OLT challenge status**: This Boolean attribute is used during authentication step 1. It controls and reports the status of the OLT crypto capabilities and the OLT random challenge table attributes. As a status report, the value false indicates that the values represented in those MEs are not complete. The value true indicates that they are complete. This attribute behaves as follows:

- If the OLT writes to either the OLT crypto capabilities or the OLT random challenge table, then the OLT challenge status attribute becomes false.
- If the OLT challenge status attribute is false and the OLT sets the OLT challenge status attribute to true, the ONT begins processing the contents of the OLT crypto capabilities and OLT random challenge table using the selected cryptographic hash algorithm.

The ONT initializes this attribute to the value false. (R, W) (mandatory) (1 byte)

**ONT selected crypto capabilities**: This attribute specifies the cryptographic capability selected by the ONT in authentication step 2. Its value specifies one of the bit positions that has the value 1 in the OLT crypto capabilities attribute. (R) (mandatory) (1 byte)

**ONT random challenge table**: This attribute specifies the random challenge issued by the ONT during authentication step 2. It is structured as a table, with each entry being 16 bytes of content. Once the OLT triggers a response to be generated using the OLT challenge status attribute, the ONT generates the response, and then writes the table (in a single operation). The AVC generated by this attribute signals the OLT that the challenge is ready, so that the OLT can commence a get/get-next sequence to obtain the table's contents. (R) (mandatory) (16*P bytes)

**ONT authentication result table**: (authentication step 2) This attribute contains the result of the authentication computation from the ONT, according to the ONT selected crypto capabilities attribute. The value of the ONT authentication result table is equal to

SelectedHashFunction (PSK | ONT_selected_crypto capabilities | OLT_random_challenge_table | ONT_random_challenge_table| 0x0000 0000 0000 0000),

where "|" denotes concatenation.

This attribute is structured as a table, with each entry being 16 bytes of content. Once the OLT triggers a response to be generated using the OLT challenge status attribute, the ONT generates the response, and then writes the table (in a single operation). The AVC generated by this attribute signals the OLT that the response is ready, so that the OLT can commence a get/get-next sequence to obtain the table's contents. (R) (mandatory) (16*Q bytes)

**OLT authentication result table**: This attribute is used in authentication step 3. It specifies the result of the authentication computation from the OLT. The OLT response is equal to

SelectedHashFunction (PSK | ONU_selected_crypto capabilities | ONT_random_challenge_table | OLT_random_challenge_table | ONT_serial_number).

The ONT_serial number is the serial number attribute of the ONT-G managed entity.

This attribute is structured as a table, with each entry being 17 bytes. The first byte is the entry index; the remaining 16 bytes are content. The OLT writes all entries in the table, and then triggers the ONT's processing of the table using the OLT result status attribute. The table size is known by the maximum index set by the OLT. The OLT can clear the table with a set operation to index 0. (W) (mandatory) (17*R bytes)

**OLT result status**: (authentication step 3) This Boolean attribute controls and reports the status of the OLT authentication result table attribute. As a status report, the value false indicates that the value represented in the OLT authentication result table is not complete. The value true indicates that it is complete. This attribute behaves as follows:

- If the OLT writes to the OLT authentication result table, then OLT result status becomes false.
- If the OLT result status is false and the OLT sets the OLT result status to true, the ONT begins processing the contents of the OLT authentication result table using the specified algorithm.

(R, W) (mandatory) (1 byte)

**ONT authentication status**: This attribute indicates the status of the authentication relationship from the perspective of the ONT. It has the following values:

0 Inactive: the authentication procedure is not currently active.

1 OLT challenge pending: the authentication procedure is under way, and is between message step 1 and 2.

2 ONT challenge pending: the authentication procedure is under way, and is between message step 2 and 3.

3 Authentication success: the procedure has completed, and the ONT has authenticated the OLT

4 Authentication failure: the procedure has completed, and the ONT has not authenticated the OLT

5 Error: the authentication procedure was started but could not be completed

When the ONT authentication status has the value 3, encryption keys exchanged in the TC layer will be encrypted using the session key. The OLT should check the value of of this attribute before initiating a key switch.

(R) (mandatory) (1 byte)

**Session key name**: Following successful authentication, this register contains the "name" of the current session key. The session key is defined as:

SelectedHashFunction (PSK | OLT random_challenge | ONT random challenge)

The session key name is defined as:

SelectedHashFunction (PSK | ONT random challenge | OLT random challenge | 0x 3141 5926 5358 9793 3141 5926 5358 9793)

If the selected hash function generates more than 128 bits, the result is truncated to the leftmost (most significant) 128 bits.

Upon termination of a session key (eg due to an ONT reset or due to an ONT-local decision that the session key has expired), the ONT sets the session key name to all zeros. (R) (mandatory) (16 bytes)

*Actions*

Get, set, get next

*Notifications*

**Attribute value change**

| Number | Attribute value change | Description |
|---|---|---|
| 1..4 | Reserved | |
| 5 | ONT random challenge table | A new ONT challenge has been loaded into the table for the OLT to retrieve |
| 6 | ONT authentication result table | A new ONT response has been loaded into the table for the OLT to retrieve |
| 7..9 | Reserved | |
| 10 | ONT authentication status | The ONT authentication status has changed |
| 11..16 | Reserved | |

**Supplementary information**

This managed entity contains the facilities to perform a conventional three step hash-based authentication sequence found in ISO/IEC 9798-4 (used in DSL systems that employ MS-CHAPv2 and elsewhere) using get and set messages.

The logical structure of the conventional three step sequence is as follows:

Message 1: (Peer 1 → peer 2) my_cryptographic_capabilities, random_challenge_1

Message 2: (Peer 2 → peer 1): selected_cryptographic_capabilities, random_challenge_2, MsgHash (PSK, selected_cryptographic_capabilities, random_challenge_1, random_challenge_2, peer_1_identity)

Message 3: (Peer 1 → peer 2): MsgHash (PSK, selected_cryptographic_capabilities, random_challenge_2, random_challenge_1, peer_2_identity)

Where:

MsgHash () is a keyed hash function of the message

PSK is the pre-shared key known only to the peers of the session
Peer_1_identity is always “0x0000 0000 000 0000”
Peer_2_identity is the ONT serial number

The prerequisite is the availability of a pre-shared secret: PSK. A PSK of 128 bits simplifies the application of security algorithms based on AES-128 (e.g. AES-CMAC-128). A PSK is associated with a particular ONT and is stored at that ONT and at the operator infrastructure. On the operator side, the PSK for a particular ONT might be stored in the physically-connected OLT, or at a central server that the OLT accesses during authentication. Configuration of the PSK into the ONT and into the operator infrastructure may be done in any manner that satisfies these requirements.

In OMCI, the authentication message sequence follows the steps illustrated in figure 9.13.11-1.

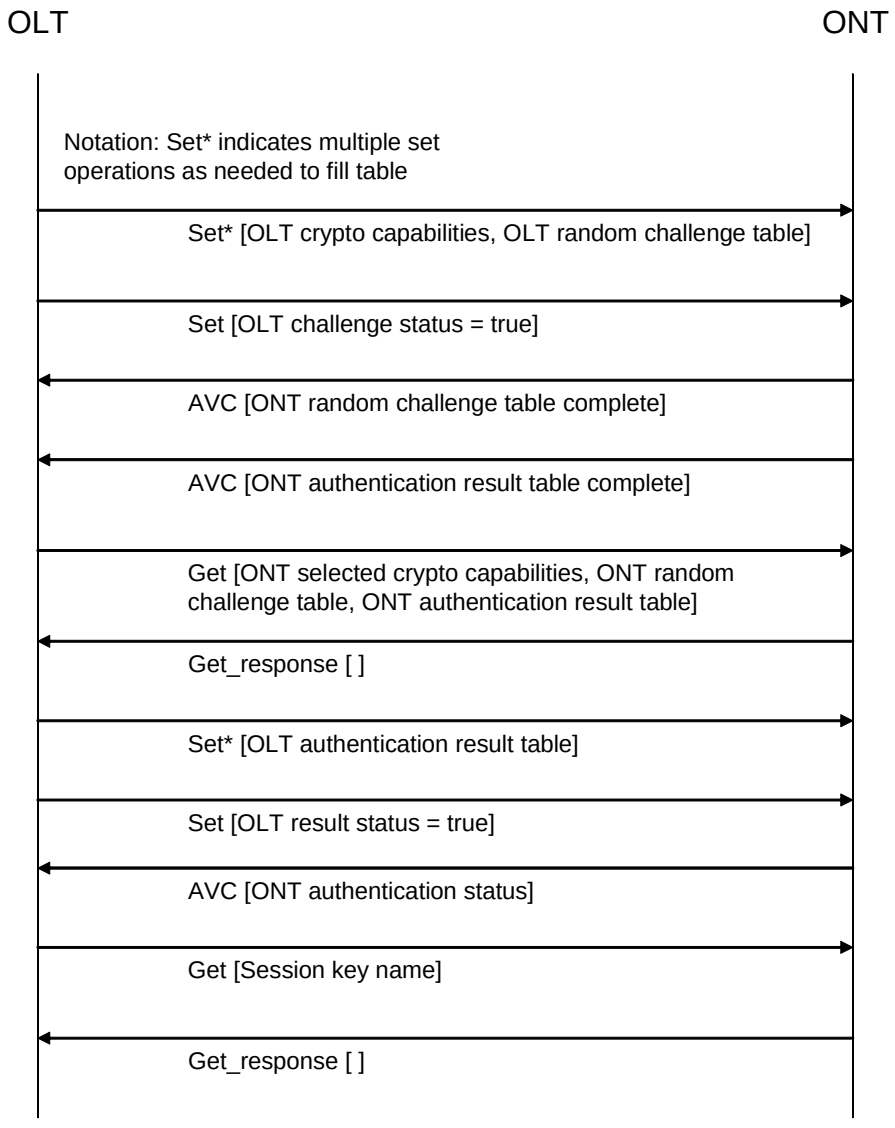



**Figure 9.13.11-1 – Authentication message exchange sequence**

**ONT states**

As tracked by the ONT authentication status attribute and illustrated in figure 9.13.11-2, the mutual authentication functionality in the ONT exists in the state machine described below. It is to be

understood that this state machine runs entirely in O5, the operation state defined in [ITU-T G.984.3].

a. Inactive – S0

   After ONT registration, the ONT is in inactive state. The OLT initiates the authentication process by writing a challenge into the OLT random challenge table.

b. OLT challenge pending – S1

   After the OLT writes its challenge to the OLT random challenge register, the ONT enters this state.

   The ONT transitions to ONT challenge pending state after it has chosen the ONT random challenge and calculated the ONT authentication result table.

   If the ONT was unable to perform the operations necessary to transition to ONT challenge pending, it enters the authentication error state S5.

   While the ONT is in OLT challenge pending state, the OLT should not write new values into the OLT random challenge table.

c. ONT challenge pending – S2

   In this state the ONT waits for the OLT to read the relevant registers and write the result of the ONT's authentication challenge to the OLT authentication result table.

   If no response to the ONT's challenge is received in the OLT authentication hash register within three seconds, the ONT transitions to the authentication error state S5. If a response is received, the ONT validates it and transitions either to authenticated state S3 or to deauthenticated state S4.

   While the ONT is in the ONT challenge pending state, the OLT should not write new values into the OLT random challenge table.

d. Authentication success – S3

   The ONT must set a valid value for the session key name attribute before it enters the authentication success state. The OLT will typically read the session_key name attribute when it receives the AVC indicating that the ONT authentication status attribute has changed to state S3 to ensure that the ONT is synchronized and the new key is ready to be utilized within the TC layer PLOAM function.

e. Authentication failure – S4

   The ONT enters this state from the ONT challenge pending state whenever the authentication procedure fails, due to e.g. a PSK mismatch (OLT authentication result table differs from the MsgHash value computed by the ONT).

f. Authentication error – S5

   This state indicates that the authentication procedure was started but could not be completed.

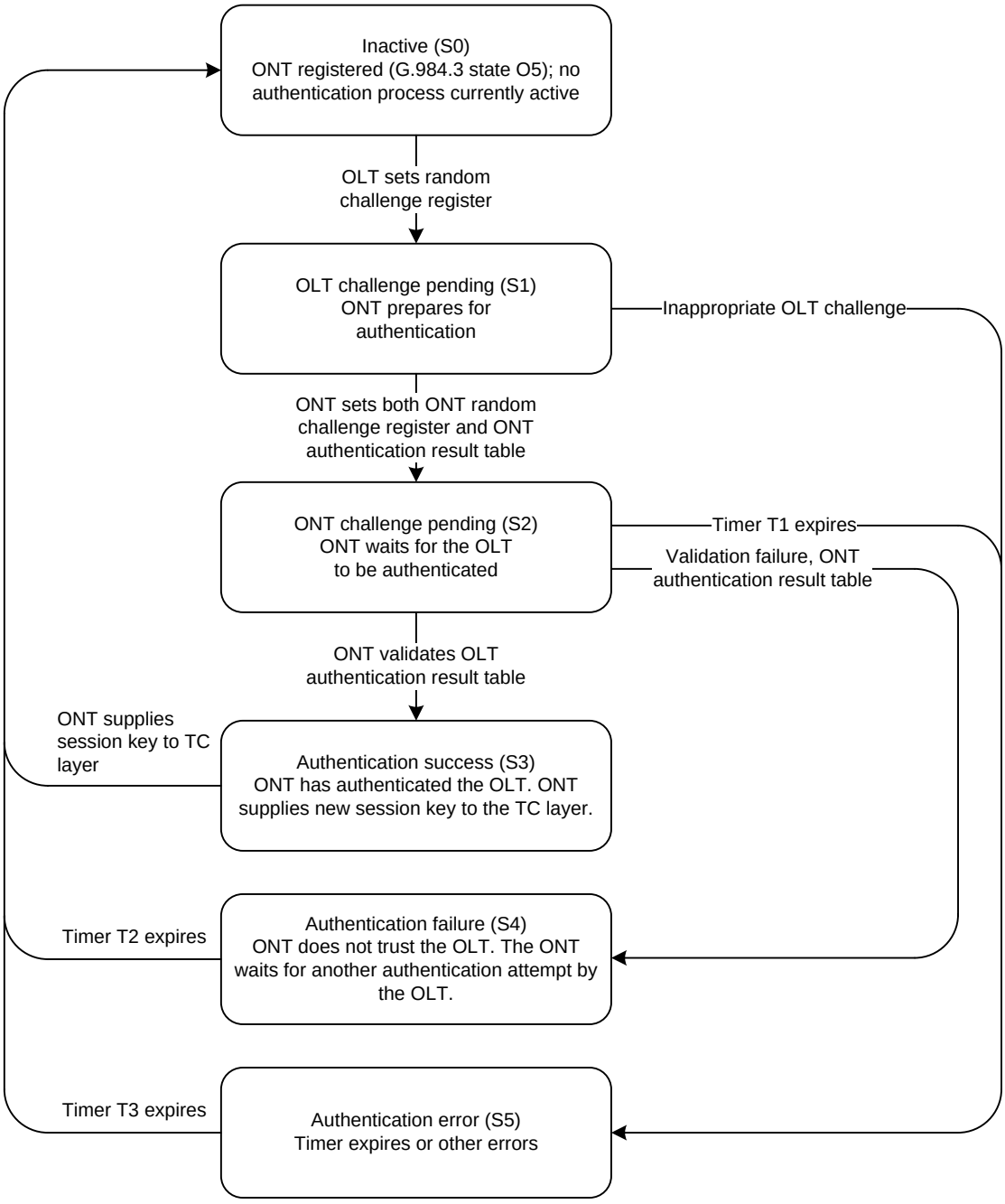



**Figure 9.13.11-2 – ONT state diagram**

Notes to figure 9.13.11-2:

Timer T1 – ONT challenge pending timer. Timer T1 is used to abort an unsuccessful OLT authentication attempt by limiting the overall time an ONT can remain in state S2. The recommended value of T1 is 3 seconds.

Timer T2 – Authentication failure timer. Timer T2 is used to assert a mismatch of the session key condition by limiting the time an ONT can remain in state S4. The recommended value of T2 is 1 second.

Timer T3 – Authentication error timer. Timer T3 is used to assert a failure of authentication condition by limiting the time an ONT can remain in state S5. The recommended value of T3 is 1 second.

**Synchronization with TC layer and security considerations**

When the ONT is in authenticated state, it uses its session key to encrypt the encryption key transmitted in the encryption_key PLOAM message.

The session key is defined as:

SessionKey = SelectedHashFunction (PSK | OLT random_challenge | ONT random challenge)

where SelectedHashFunction () is the hash function selected by the ONT in the ONT selected crypto capabilities attribute from the list supplied by the OLT.

The encryption of the encryption key shall be performed using AES-128 in electronic codebook (ECB) mode.

Since the integrity of the encryption key carried in the encryption key PLOAM message is not protected, there is the possibility that the key can be forged or replayed by an attacker. A forged key is a form of denial-of-service attack, and both forged and replayed keys can be detected with key synchronization mechanisms. A replay attack, however, could force the OLT to use an old encryption key, which violates the security requirements of the downstream data encryption. Consequently, an OLT designed to resist a replay attack should ensure that the ONT does not send a previously used encryption key between authentication cycles.

### 9.14 Mid-span PON reach extender

*Revise the second paragraph of the introductory material, and add a note. The second paragraph, with the note, then reads as follows:*

The RE model includes one built-in ONT, which serves for management of the RE itself, as well as for optional subscriber or craft UNIs. The ONT is therefore able to use any of the managed entities defined elsewhere in this recommendation, including the ANI-G and T-CONT MEs, which represent the built-in ONT's individual uplink.

NOTE – Although the built-in management ONT is physically contained within a physical reach extender equipment, the management model perspective is that the ONT software controls the entire equipment. In terms of the model, therefore, the management ONT contains the reach extender equipment and functionality.

#### 9.14.2 Physical path termination point RE UNI

*Revise the per burst receive signal level table attribute description to read as follows:*

**Per burst receive signal level table**: This table attribute reports the most recent measurement of received burst upstream optical signal level at 1310 nm. Each table entry has a two byte ONU-ID field (note) (most significant end), and a two byte power measurement field. The power measurement field is a 2s complement integer referred to 1 mW (ie dBm), with 0.002 dB granularity. (R) (optional) (4N bytes, where N is the number of distinct ONTs connected to the S'/R' interface.)

NOTE – In its initial definition, only one byte was assigned to ONU-ID. This was changed to two bytes in anticipation that future PON technologies may support a split ratio greater than 256.

### 9.14.3 RE upstream amplifier

*Revise the per burst receive signal level table attribute description to read as follows:*

**Per burst receive signal level table**: This table attribute reports the most recent measurement of received burst upstream optical signal level at 1310 nm. Each table entry has a two byte ONU-ID field (note) (most significant end), and a two byte power measurement field. The power measurement field is a 2s complement integer referred to 1 mW (ie dBm), with 0.002 dB granularity. (R) (optional) (4N bytes, where N is the number of distinct ONTs connected to the S'/R' interface.)

NOTE – In its initial definition, only one byte was assigned to ONU-ID. This was changed to two bytes in anticipation that future PON technologies may support a split ratio greater than 256.

## 10 ONT management and control channel (OMCC)

## 11 ONT management and control protocol

*Revise clause 11 as shown below.*

### 11.1 Baseline and extended message formats

Two OMCI message formats are defined in clause 11.2, the baseline and the extended message formats.

Baseline messages have 48-byte fixed length PDUs, while extended messages have variable length PDUs. A receiver that does not support extended messages may therefore reject extended message based on nothing more than their length.

Both baseline and extended messages carry an I.363.5 CRC code in their final four bytes. This facilitates ad hoc recovery of both message types by a receiver.

Baseline and extended messages are distinguished from one another by the device identifier field, which is in the same byte location in both message types. Baseline messages are addressed to device identifier 0x0A, while extended messages employ device identifier 0x0B.

All ONTs and all OLTs are required to support the baseline format. During initialization, and whenever the ONT is re-ranged onto the PON, both entities use baseline format to establish communications and to negotiate their capabilities. If both endpoints support extended messages, they may or may not choose to conduct all or some subsequent communications in the extended message set. Baseline messages may be used for any transaction, that is, any exchange of one or more related messages such as a get/get-next sequence.

Figure 11.1-1 illustrates the negotiation and the exchange of messages in one or the other message format.

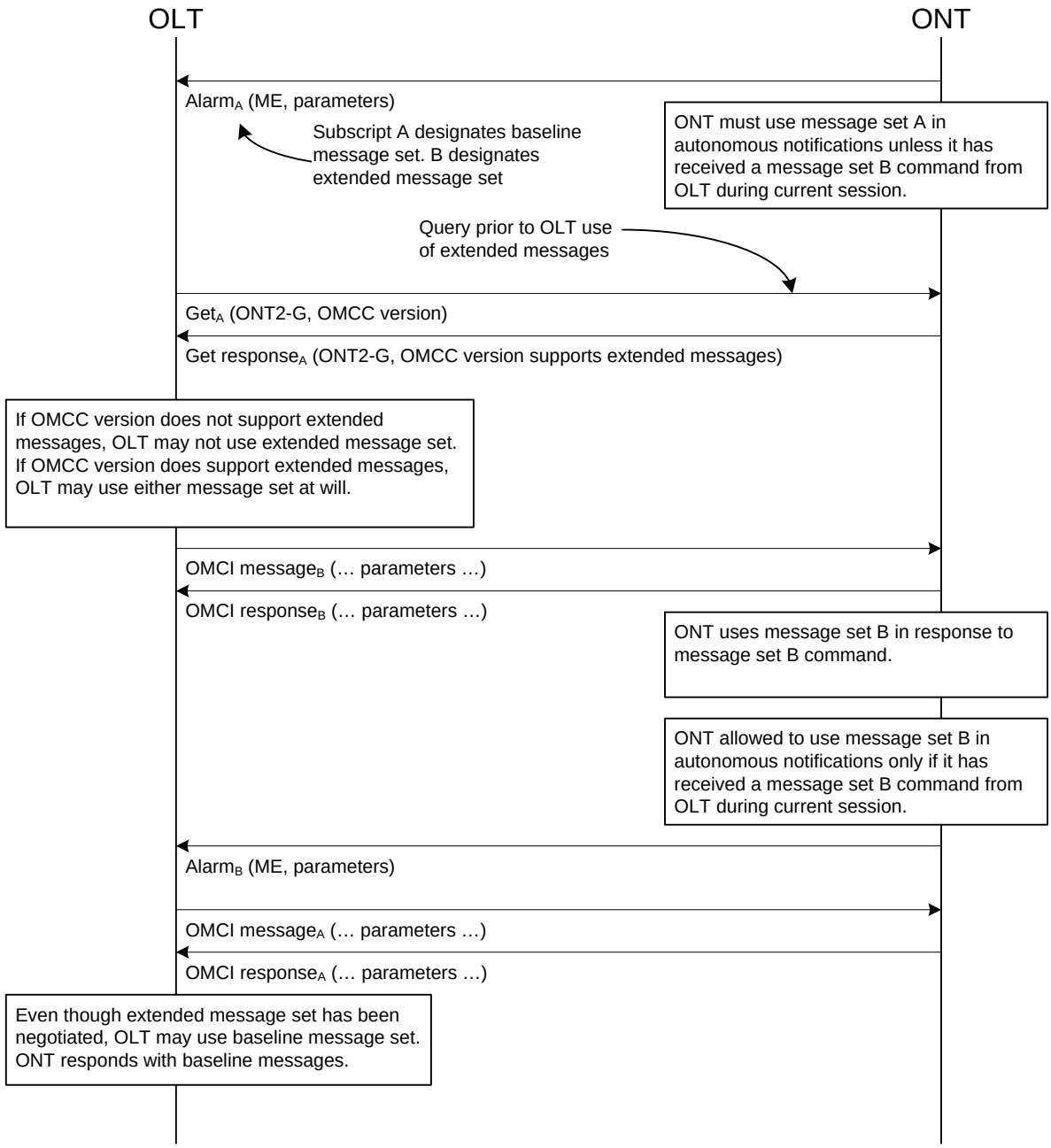



**Figure 11.1-1 – OMCI message set negotiation**

If the OLT has a priori knowledge that the ONT supports the extended message set, it may choose to omit the query step. However, an ONT may not transmit extended messages, including autonomous notifications, until it has received at least one extended message from the OLT during the current session (since initialization or re-ranging on the PON).

NOTE – It is anticipated that the baseline message format will be deprecated or removed from the G.987 recommendations. G.984 systems are therefore encouraged to use the extended message set if they support it.

## 11.2 Packet format

Each OMCI protocol packet is encapsulated directly in one GEM frame, which may be fragmented in accordance with normal fragmentation rules. The GEM frame header contains the port ID of the OMCC for the addressed ONT, with a header PTI of 000 or 001.

Two message formats are defined, a baseline format and an extended format. The following clauses discuss the details.

The baseline packet format is shown in figure 11.2-1. This fixed length 48-byte format reflects the ATM heritage of PON.

| Transaction correlation identifier (2 bytes) | Message type (1 byte) | Device identifier (1 byte) | Message identifier (4 bytes) | Message contents (32 bytes) | OMCI trailer (8 bytes) |
|---|---|---|---|---|---|

**Figure 11.2-1 – ONT management and control protocol baseline packet format**

Figure -11.2-2 shows the extended packet format. The packet has variable length, up to 1980 bytes.

Ed note: If IEEE defines OMCI carriage (eg via new Ethertype), additional bytes may need to be reserved for header.

| Transaction correlation identifier (2 bytes) | Message type (1 byte) | Device identifier (1 byte) | Message identifier (4 bytes) | Message contents length (2 bytes) | Message contents (variable) | CRC (4 bytes) |
|---|---|---|---|---|---|---|

**Figure -11.2-2 – ONT management and control protocol extended packet format**

The following clauses specify each field of these messages.

### 11.2.1 Transaction correlation identifier

The transaction correlation identifier is used to associate a request message with its response message. For request messages, the OLT selects a transaction identifier. A response message carries the transaction identifier of the message to which it is responding. The transaction identifier of messages generated autonomously by the ONT is 0.

As explained in clause 11.4.1, and for the baseline message format only, the most significant bit of the transaction correlation identifier indicates the priority of the message. The following coding is used: 0 = low priority, 1 = high priority. The OLT decides whether a command should be executed with low or high priority. The extended message format does not recognize priorities.

The remainder of the transaction identifier may be arbitrary, but should be chosen to avoid the possibility of ambiguous responses from ONTs.

### 11.2.2 Message type

The message type field is subdivided into four parts. These are shown in figure -11.2.2-1.

| 8 | 7 | 6 | 5 … 1 |
|---|---|---|---|
| DB | AR | AK | MT |

**Figure 11.2.2-1 – Message type field subdivision**

Bit 8, the most significant bit, is reserved (DB). This bit is always 0. This bit is not used in the baseline message format, but is available for possible future use in the extended message format.

Bit 7, acknowledge request (AR), indicates whether or not the message requires an acknowledgement. An acknowledgement is a response to an action request, not a link layer handshake. If an acknowledgement is expected, this bit is set to 1. If no acknowledgement is expected, this bit is 0. In messages sent by the ONT, this bit is always 0.

Bit 6, acknowledgement (AK), indicates whether or not this message is an acknowledgement to an action request. If a message is an acknowledgement, this bit is set to 1. If the message is not a response to a command, this bit is set to 0. In messages sent by the OLT, this bit is always 0.

Bits 5..1, message type (MT), indicate the message type, as defined in table 11-1. Values not shown in the table are reserved.

**Table 11-1 – OMCI message types**

| MT | Type | Purpose | AK | Increment MIB data sync |
|---|---|---|---|---|
| 4 | Create | Create a managed entity instance with its attributes | Yes | Yes |
| 6 | Delete | Delete a managed entity instance | Yes | Yes |
| 8 | Set | Set one or more attributes of a managed entity | Yes | Yes |
| 9 | Get | Get one or more attributes of a managed entity | Yes | No |
| 11 | Get all alarms | Latch the alarm statuses of all managed entities and reset the alarm message counter | Yes | No |
| 12 | Get all alarms next | Get the active alarm status of the next managed entity | Yes | No |
| 13 | MIB upload | Latch the MIB | Yes | No |
| 14 | MIB upload next | Get latched attributes of a managed entity instance | Yes | No |
| 15 | MIB reset | Clear the MIB, re-initialize it to its default, and reset the MIB data sync counter to 0 | Yes | No |
| 16 | Alarm | Notification of an alarm | No | No |
| 17 | Attribute value change | Autonomous notification of an attribute value change | No | No |
| 18 | Test | Request a test on a specific managed entity | Yes | No |
| 19 | Start software download | Start a software download action | Yes | Yes |
| 20 | Download section | Download a section of a software image | Note | No |
| 21 | End software download | End of a software download action | Yes | Yes |
| 22 | Activate software | Activate the downloaded software image | Yes | Yes |
| 23 | Commit software | Commit the downloaded software image | Yes | Yes |
| 24 | Synchronize time | Synchronize the time between OLT and ONT | Yes | No |
| 25 | Reboot | Reboot ONT or circuit pack | Yes | No |
| 26 | Get next | Get the latched attribute values of the managed entity within the current snapshot | Yes | No |
| 27 | Test result | Notification of result initiated by a test command | No | No |

**Table 11-1 – OMCI message types**

| MT | Type | Purpose | AK | Increment MIB data sync |
|---|---|---|---|---|
| 28 | Get current data | Get current counter value associated with one or more attributes of a managed entity | Yes | No |
| 29 | Set table | Set one or more rows of a table | Yes | Yes |
| Note – The download section action is acknowledged only for the last section within a window. See Appendix I.2.15. | | | | |

### 11.2.3 Device identifier

In baseline OMCI messages, this field is defined to be 0x0A.

In extended OMCI messages, this field is defined to be 0x0B.

### 11.2.4 Message identifier

*Revise table 11-2 code point 280 to read as follows (change* GEM traffic descriptor *to* Traffic descriptor*).*

**Table 11-2 – Managed entity class identifiers**

| Managed entity class value | Managed entity |
|---|---|
| 280 | Traffic descriptor (NOTE – Formerly GEM traffic descriptor) |

*Add the following code points to table 11-2 in numeric sequential order and adjust the reserved codepoints as shown:*

**Table 11-2 – Managed entity class identifiers**

| Managed entity class value | Managed entity |
|---|---|
| 329 | Virtual Ethernet interface point |
| 330 | Generic status portal |
| 331 | ONT-E |
| 332 | Enhanced security control |
| 333 | MPLS pseudowire termination point |
| 334-65279 | Reserved for future standardization |
| 65280-65535 | Reserved for vendor specific use |

### 11.2.5 Message contents length, extended message format

These two bytes contain the length, in bytes, of the message contents field. Its value lies between 0 and 1966, for a 1980-byte PDU limit.

Eleven bits suffice to specify this range. The five most significant bits of this field are reserved for future use.

### 11.2.6 Message contents

The layout of the message contents field is message specific. The detailed layout of all messages appears in Appendix II.

### 11.2.7 OMCI trailer, baseline message format

The AAL5 trailer definition is reused in this field. The eight bytes of this field are as follows:

a) The first two bytes correspond to CPCS-UU and CPI. They are set to 0 at the transmitter and ignored at the receiver.

b) The length of the CPCS-SDU field is set to 0x0028 (40 decimal).

c) The 32-bit CRC is as specified in [ITU-T I.363.5].

### 11.2.8 CRC, extended message format

The integrity of the extended message format is verified by this four-byte field, a 32-bit CRC as specified in [ITU-T I.363.5].

### 11.2.9 Message limits

Baseline OMCI messages impose limits on the size of attributes. Table 11.2.9-1 lists the important limits.

**Table 11.2.9-1 – OMCI baseline message limitations**

| Item | Limited by | Maximum size, bytes |
|---|---|---|
| Total size of set-by-create attributes (including ME ID) | Create | 34 |
| Size of (R) or (R,W) simple attribute | Get response | 25 |
| Size of (R) or (R,W) structured table entry (note) | Set | 30 |
| Total size of a get | Get response | 25 |
| Total size of a get current data | Get current data response | 25 |
| NOTE – A structured table is one that contains distinct and separable rows, each row of which has the same syntax as the others. Long strings of bytes are also designated tables in clause 9, because the mechanism for retrieval is the same: get, followed by a number of get next commands. Such a byte string could be regarded as a table with but a single row, the length of which is limited only by the number of get next commands that can be specified. There is no way to set a value into such a byte string, however, so these attributes are necessarily read-only. | | |

Extended messages are limited by the total size of the PDU. For backward compatibility, attribute definitions remain within the size limits of baseline messages, but more attributes may be contained within a single extended message. The following considerations apply to baseline messages only. The larger PDU eliminates the possibility of message length violation in the extended message set.

It is important that OLT and ONT implementations take size limits into account. For example, it is easy to form a (baseline) get command that asks the ONT to return more attributes than can fit into a (baseline) get response message. If the OLT asks for too many attributes in a get request, the ONT may respond with as many attributes as fit into the space available. From the attribute-present mask, the OLT can parse the attributes that were sent correctly, and can issue another get to retrieve the attributes that did not fit.

While this is the preferred behaviour, an alternate interpretation may be that the ONT would return a parameter error code when it receives a (baseline) get request whose response does not completely

fit into one (baseline) get response message. For the sake of interoperability, the expected behaviour between an OLT and ONT with different interpretations is provided below:

Case 1. The ONT reports a parameter error, and the OLT expects a partial list. If this happens, the OLT should react by simplifying its get request. The ONT then responds without an error.

Case 2. The ONT provides a partial list, while the OLT expects to get an error. The OLT receives a normal message and processes it normally. The OLT asks again for any attributes it did not get.

### 11.2.10 Test result enumeration

*No change to this clause.*

## 11.3 Message flow control and error recovery

NOTE – Prioritized message handling is defined only for the baseline message set. In the context of the extended message set, this clause should be read with the understanding that there is only one priority class.

The flow control/error recovery procedures for message exchange over the OMCC are based on a simplex acknowledged transaction stop-and-wait mechanism that can be extended to support concurrent execution of multiple transaction requests of different priority levels. These flow control procedures ensure that a low level acknowledged transaction request transmitted from the OLT has been properly received and processed to completion by the ONT before the next message of the same priority level is sent by the OLT. The stop-and-wait protocol uses the transaction correlation identifier field, retry counter(s), and applicable transaction request timer(s) to control the message flow rate while relying on a CRC calculation to verify the data integrity of all received messages.

A transaction request timer $T_i$ with expiration time $Tmax_i$ is started when a transaction request message of priority level i is sent to an ONT, and is stopped upon receipt of an acknowledgement message containing the same transaction correlation identifier value. If a valid acknowledgement message is not received by the OLT after timer $T_i$ expires, the OLT re-sends the original transaction request message.

A retransmitted acknowledged transaction request message carries the same transaction correlation ID as the original message. Each time an acknowledged transaction request message is retransmitted by the OLT, the transmitter increments the retry counter $R_i$ (the counter associated with priority level i acknowledged transaction requests). When a retry counter $R_i$ (initialized to 0 upon the first transmission of each new command) reaches the maximum retry value, $Rmax_i$, the transmitter stops re-transmitting the message and declares an OMCC link state error.

These timers ($T_i$) and retry counters ($R_i$) are only maintained within the OLT controller and do not exist within the ONT. Threshold values for timer expiration ($Tmax_i$) and number of retries ($Rmax_i$) are not subject to standardization. It is suggested that the default threshold values of both Tmax and Rmax be independently configurable for each priority level. The default value for $Tmax_1$ (i.e. high priority threshold) should account for the typical message transmission delay plus the command message response time.

These flow control/error recovery procedures are illustrated in figure -11.3-1 for a case where the OMCC link is not permanently broken. First the OLT sends an acknowledged transaction request (message 1) with priority level 0. Subsequently (while message 1 is still outstanding), the OLT issues an additional acknowledged transaction request (message 2) with priority level 1. Both of these commands are received and executed with the associated response (acknowledgement messages) returned to the OLT by the ONT. The acknowledgement for message 1 is received by the OLT in time, however the response to message 2 is lost and never received. The OLT detects that something went wrong because timer $T_1$ expires, and the OLT therefore retransmits the original

command (message 2). From the identical transaction ID, the ONT detects that this retransmitted command is identical to the last received command (for priority level 1) and therefore does not re-execute it. The ONT simply retransmits the original response from the previous execution of message 2, which reaches the OLT in time.

The final transaction in the example is the case where the OLT sends an acknowledged transaction request (message 3) with priority level 0, but the message itself gets lost and is never properly received by the ONT. After the associated timer ($T_0$) expires, the OLT retransmits the command and now all goes well.

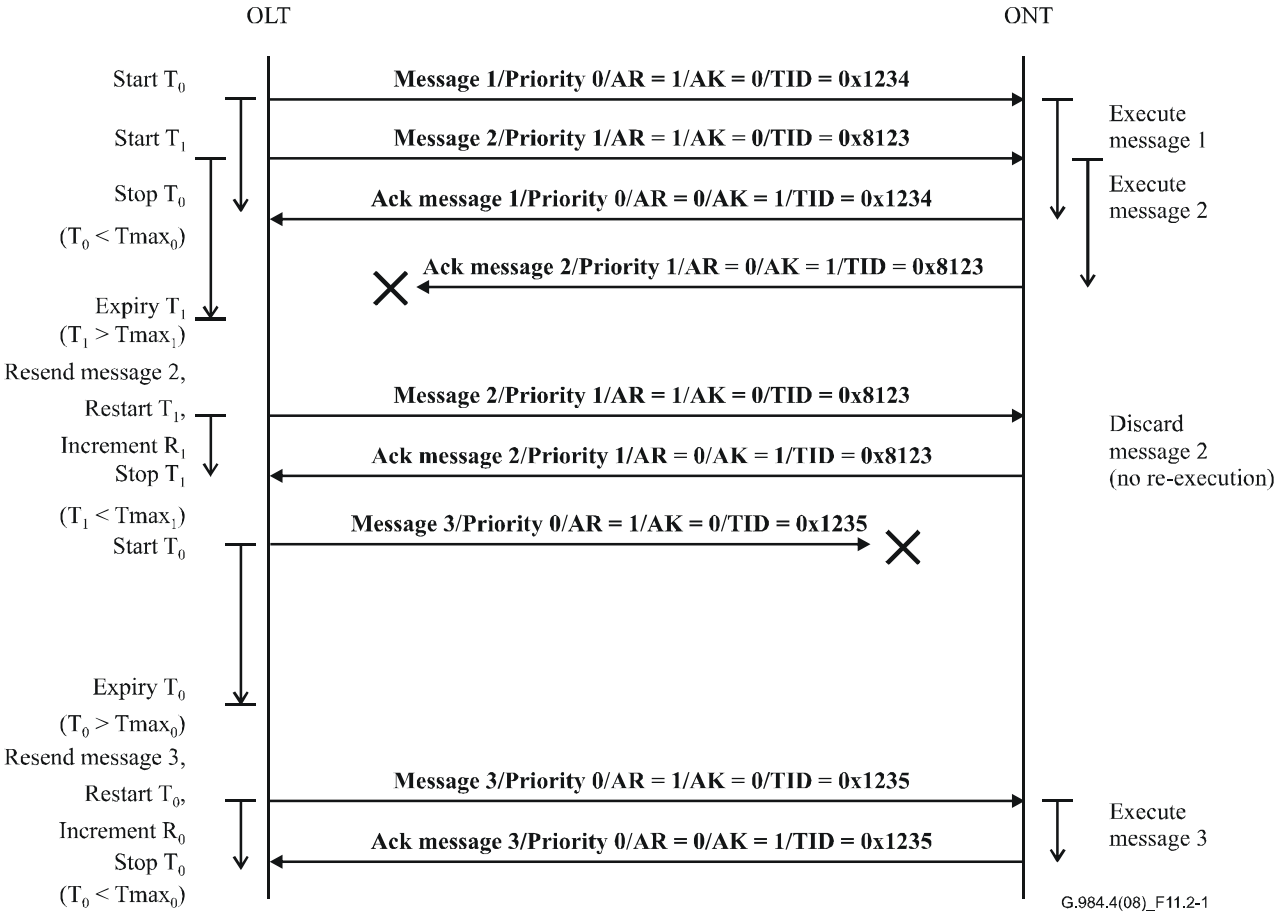



**Figure 11.3-1- – Concurrent message exchange with error recovery**

A case where the OMCC link is effectively broken (link down) is shown in figure -11.3-2.

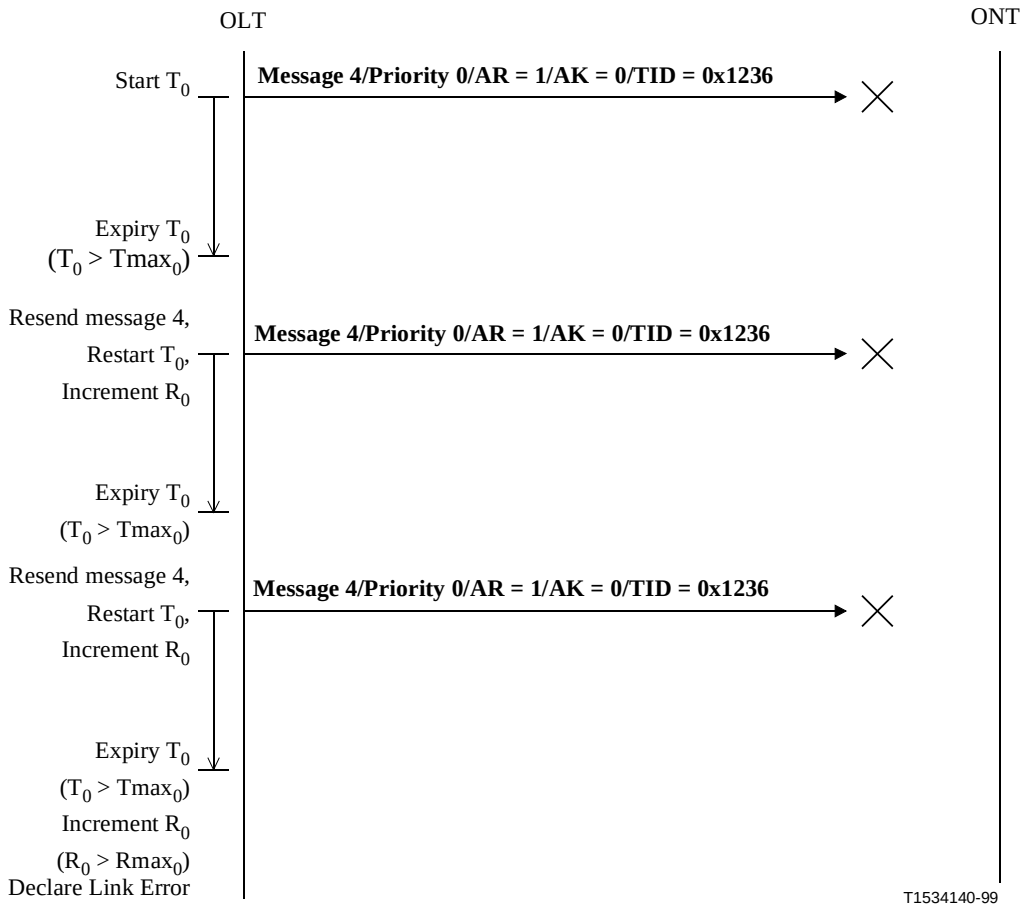



**Figure 11.3-2 – OMCC link error detection**

## 11.4 OMCI handling within the ONT

### 11.4.1 Prioritized protocol entities

NOTE – Prioritized message handling is defined only for the baseline message set. In the context of the extended message set, this clause should be read with the understanding that there is only one priority class.

This clause specifies the behaviour of the ONT more precisely than in the preceding clause with respect to the prioritized request mechanism of the OMCC.

Conceptually, the way the ONT handles the OMCC requests can be illustrated by referring to the dual priority level implementation example shown in figure 11.4.1-1.

When the ONT receives a GEM frame via the GEM port associated with the management channel, it shall calculate the CRC and compare it with the value found in the OMCI trailer. If the values do not match, the ONT shall discard the message. It is recommended that this event be logged by the ONT and possibly communicated to the OLT by some out-of-band mechanism but, as far as the protocol is concerned, the message is discarded silently.

Messages with a correct CRC are then placed into either of two distinct incoming FIFO-based message queues, according to the high or low priority level of the associated command encoded in the most significant bit of the transaction correlation identifier field. If the associated incoming message queue is already full, the ONT must simply discard the message. It is recommended that

this event be logged by the ONT and possibly communicated to the OLT by some out-of-band mechanism but, as far as the protocol is concerned, the message is discarded silently.

There are two distinct incoming command processing protocol entities, one associated with each priority level, that serve messages sequentially from independent incoming FIFO queues. Each protocol entity can execute concurrently. If a message is a one-way (unacknowledged) command, the protocol entity simply executes the command. If a message is an acknowledged command, the protocol entity must first look at the transaction correlation identifier. If it is not equal to the transaction correlation identifier of the last executed command with the same priority level, the protocol entity executes the command and places the response/acknowledgement, with an identical transaction correlation identifier, in the outgoing FIFO queue of the same priority level. If the transaction correlation identifier is equal to that of the last executed command with the same priority level (the case where the OLT retransmits a command due to lack of proper acknowledgement), the protocol entity does not actually execute the command but simply places the response from the last execution of that command in the outgoing FIFO queue to re-send the previous acknowledgement response. In both cases, the command processing protocol entity for a given priority level should block until there is room in the associated outgoing FIFO queue for the response message.

In the other direction, requests by ONT applications to send autonomous event notifications simply result in the corresponding messages being directed to an event notification protocol entity for transmission to the OLT. The event notification protocol entity forwards these event notification messages to the low priority outgoing FIFO queue. In this case as well, the event notification protocol entity should block until there is room in the low priority outgoing FIFO queue to hold the notification message. The CRC generator removes messages from the outgoing FIFO queues using a strict priority discipline (that is, the low-priority queue is served only when the high-priority queue is empty), generates a CRC, formats the packet as either a baseline or an extended message, depending on which form is in use, and transmits the message to the OLT.

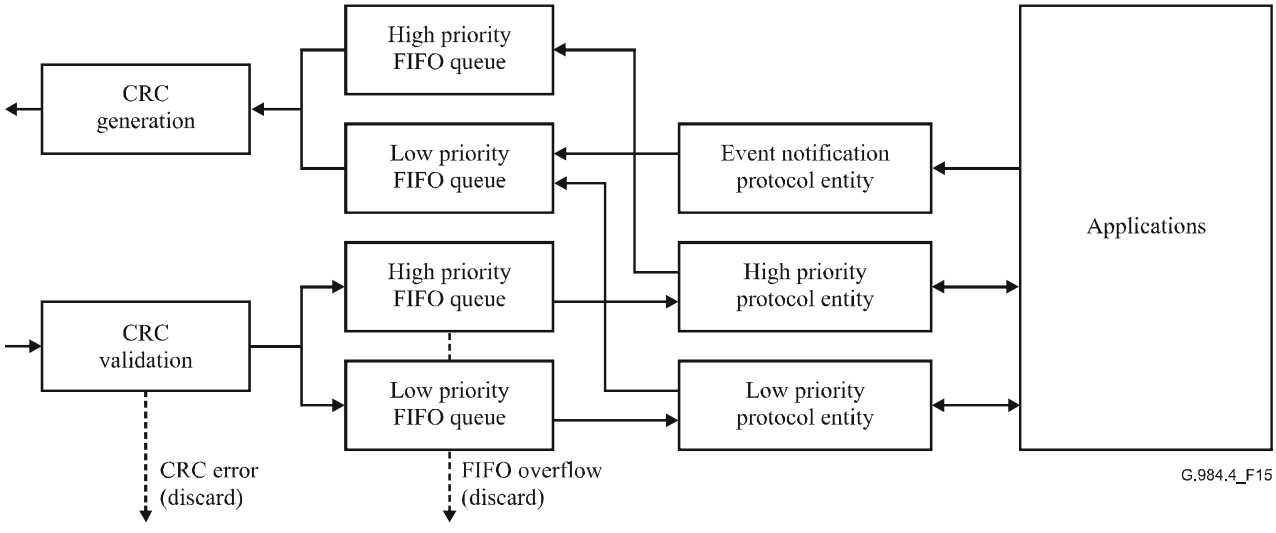



**Figure 11.4.1-1– Protocol entities within the ONT**

### 11.4.2 Restrictions on the actions in relation to the protocol entities

To reduce the complexity and the amount of memory necessary in the ONT, the OLT is not allowed to issue a MIB upload or a software download of a certain priority level while a similar action in the other priority level is in progress.

### 11.4.3 Use of the default alloc-ID

*No change to this clause.*

*Revise clause Appendix I to read as shown below.*

# Appendix I

## OMCI common mechanisms and services

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

This appendix describes the common mechanisms of OMCI, for example, MIB resynchronization, and OMCI services, for example, equipment management or connection management.

NOTE: Some information in this appendix should be modified appropriately when applied to G.986.

### I.1 Common mechanisms

The common mechanisms consist of:

a) MIB data sync increase.

b) MIB audit and resynchronization.

c) Alarm sequence number increase.

d) Alarm audit and resynchronization.

e) Get an attribute that is larger than the OMCI message contents field.

f) Create an instance of a managed entity with an attribute that is larger than the OMCI message contents field.

g) Reporting of test result.

h) Alarm reporting control.

These common mechanisms are explained with the aid of scenario diagrams.

#### I.1.1 MIB data sync increase

The revised content of this clause now appears in the G.984.4 implementers' guide [ref].

#### I.1.2 MIB audit and resynchronization

The revised content of this clause now appears in the G.984.4 implementers' guide [ref].

#### I.1.3 Alarm sequence number increase

The revised content of this clause now appears in the G.984.4 implementers' guide [ref].

#### I.1.4 Alarm audit and resynchronization

The revised content of this clause now appears in the G.984.4 implementers' guide [ref].

#### I.1.5 Table attributes

Normal attributes are coded such that they do not exceed the maximum OMCI attribute size, as limited by the baseline message format. However, there are cases where attributes need to be larger because they comprise arrays of data. In other cases, the attribute may be unstructured, but

nevertheless be too large to be represented as a conventional attribute. Both types of large attributes are known as tables, and can be identified by the word *table* in their names.

A table entry may be short enough that more than one row would fit into a (baseline) set command. However, the set command has no deterministic way to specify how many such rows are present. Therefore, the set action is permitted to set only a single entry in the table, the size of which is specified in clause 9 for the particular attribute in question.

The set operation on a table row is possible only when individual table entries have fixed size that does not exceed the maximum that can be conveyed in the (baseline) set message. A table attribute with variable-length rows or longer fixed-length rows is restricted to being read-only.

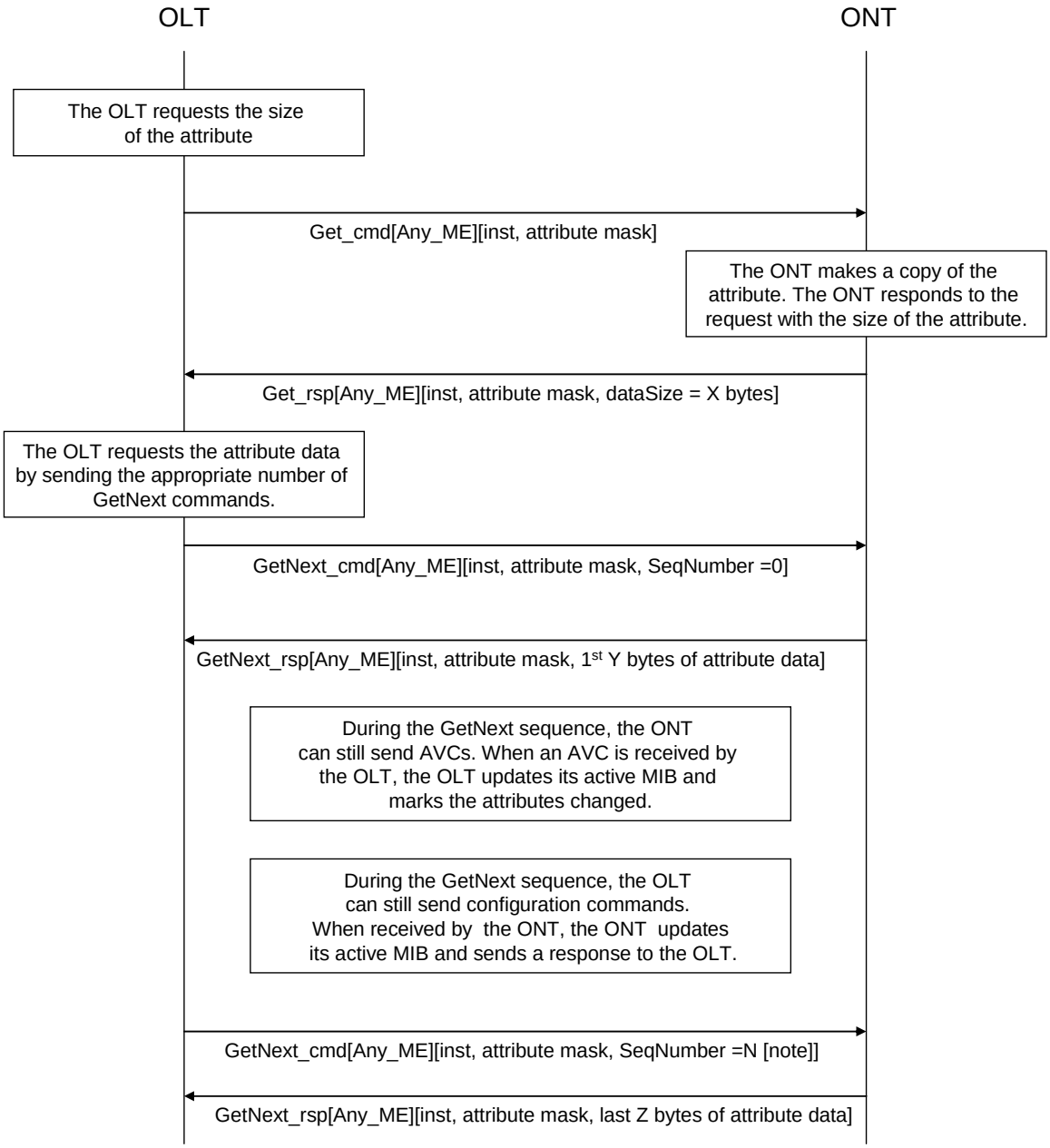
An optional set table command is defined in the extended message set. Functionally, the set table command is the equivalent of an ordered sequence of set commands, each directed to the same table attribute of a given managed entity. As with the set command, table rows must have fixed length, and because of the backward compatibility requirement, no table row may exceed the baseline length limit.

The actual size of any given table attribute instance at any given time may be smaller than the OMCI single-message limit. Regardless of its actual size, however, the following sequence governs the retrieval of all table attributes.

Figure I.1.5-1 shows how the OLT retrieves a table attribute. The OLT sends a get command, just as for any other attribute. The ONT latches a copy of the table for the anticipated get next sequence. In the get response, the ONT returns, not the value of the table attribute, but a four-byte field containing the table's size, expressed in bytes.

NOTE – Zero is a valid size for many table attributes.

The OLT then requests the attribute data from the ONT via the appropriate number of get next commands. There is no structure in the get next response; it simply regards the table as a byte string.




NOTE: N+1 is the number of get next commands as derived by the OLT to retrieve the complete table. For baseline OMCI messages, Y is 29 bytes; for extended OMCI messages, Y = 1966 Bytes (1980 PDU size – 14 Bytes of OMCI header).

**Figure I.1.5-1 – Get a table attribute**

The OLT issues as many get next requests as are needed to accommodate the size of the table attribute. The maximum time between two get next requests is 60 seconds. If the OLT does not send a get next request within this time after the previous get next request or after the initial get request, the ONT assumes the get attribute transaction has terminated, discards its copy of the table attribute, and denies further get next requests directed to that attribute.

Capturing snapshots of multiple large tables could exhaust the limited memory resources of the ONT. Within any one ONT, the OLT should get and get-next only one table attribute at a time. If more than one table attribute is selected in the get command attribute mask, the ONT may reject the command with an attributes failed or unknown result-reason code.

If more than one bit is set in the get next command attribute mask, or if the specified attribute is not a table, the ONT should respond with a parameter error result code.

In each get next command, the OLT generates a sequence number, starting from 0. The sequence number resets to 0 for each attribute, even if successive attributes are part of the same managed entity parent.

### I.1.6 This clause intentionally left blank

(no subordinate text)

### I.1.7 Report test result

*No change to this clause*

### I.1.8 Alarm reporting control

Alarm reporting control allows for the suppression of alarms from physical path termination points and cardholders, under the control of the management system. [ITU-T M.3100] completely describes ARC from a generic viewpoint. OMCI provides for ARC functions using two attributes of the parent managed entity: ARC and ARC interval. These two attributes are described below.

**ARC:**

This attribute allows the activation of alarm reporting control (ARC) for this PPTP or cardholder. The attribute works in concert with the ARC_interval attribute. A value of 0 indicates disabled, while a value of 1 indicates enabled. The default value is disabled. When the ARC attribute is set to disabled, the PPTP or cardholder is in the M.3100 ALM state. Alarms are reported normally in the ALM state. When the ARC attribute is set to enabled, the PPTP or cardholder is in the M.3100 NALM-QI state, in which alarms are suppressed.

The PPTP or cardholder moves from state ALM to state NALM-QI when the OLT changes the ARC attribute to enabled. The PPTP or cardholder moves from the NALM-QI state to the ALM state when either 1) the PPTP or cardholder is trouble free and the ARC_interval timer expires, or 2) the ARC attribute is set to disabled by the OLT. Continuation or recurrence of a fault resets the timer. If the ARC_interval timer expires, the ONT sets the ARC attribute to disabled autonomously, and sends an AVC to notify the OLT. Refer to [ITU-T M.3100] for more extensive discussion.

The ARC_interval attribute can assume normal timing values of 0 to 254 minutes. The value 0 implies that a PPTP or cardholder in the NALM-QI state goes immediately to the ALM state upon detection of a problem-free state. An ARC_interval value of 255 has the special meaning that the timer never expires. The PPTP or cardholder remains in the NALM-QI state until the OLT sets the ARC attribute to disabled. This behaviour is equivalent to the NALM state, which is another generic behaviour of the ARC function in M.3100.

Note that there is no support for the NALM-TI sub-function in the OMCI system. (R,W) (optional) (1 byte)

**ARC_interval:**

This attribute defines the interval to be used with the ARC function for this PPTP. The values 0 through 254 give the duration in minutes for the NALM-QI timer. The special value 255 means that the timer never expires. The default value is zero. (R, W) (optional) (1 byte)

ARC suppresses alarm reporting on the parent managed entity and all dependent entities, but does not suppress the alarm conditions themselves. Therefore, if an alarm condition develops during an

ARC interval, the ONT should maintain the internal indication of the alarm, and if the OLT gets all alarms regardless of ARC, it should be reported.

### I.1.9 Performance monitoring

*Replace clause I.1.9 with the following:*

This Recommendation defines a number of performance monitoring managed entities. They share a number of characteristics, as described here. Exceptions to the generic behaviour are defined in the specific managed entity affected. Two groups of PM managed entities are defined:

1. Classical PM, whose members are identified by names containing the string, "performance monitoring history data."
2. Extended PM, whose members are identified by names containing the string, "extended PM."

The remainder of this clause describes behaviour common to the two classes of PM MEs. Sub-clauses then discuss aspects unique to each PM ME class.

All PM managed entities are created and deleted by the OLT.

Classical PM, and optionally extended PM, are based on the assumption of a continuing sequence of 15-minute intervals. This sequence of 15-minute intervals is synchronized by the synchronize time action, issued by the OLT against the ONT-G managed entity. The synchronize time action establishes a 15-minute tick boundary and starts numbering the intervals from 0. The interval number is returned in the interval end time attribute. In the PM ME template, the interval end time is a single byte, which rolls over from 255 to 0.

The synchronize time action is the only mechanism guaranteed to reset either the phase or the interval number. For example, neither ONT re-boot nor MIB reset can be expected to have these effects (the performance monitoring consequence of the latter events is undefined). In the absence of a synchronize time message, an ONT would be expected to maintain 15-minute intervals asynchronously with the outside world, and with arbitrary interval end-time identifiers.

The ONT performs no archival; archival is the function of the OLT or a management system. In 15-minute accumulation mode, the ONT conceptually has only two storage bins: a current accumulator and a history bin. At 15-minute intervals, they switch roles. History is discarded at age thirty minutes, when the previous history bin is initialized into its role as the new current accumulator. The previous accumulator, now in its role as the history bin, retains its totals for 15 minutes, so that the OLT can upload them if desired.

In 15-minute accumulation mode, the get action on a PM managed entity returns the values of attributes in the history bin. An ONT may also support an optional action in this mode, get current data. The effect of this action is to return the value of attributes in the current accumulator. When a PM attribute is an average, it acquires a value only at the close of a 15-minute interval. The value returned by a get current data operation is undefined (0xFF in every byte would be reasonable).

OMCI supports PM thresholds and threshold crossing alerts (TCAs). Not all PM attributes need to be thresholded; threshold definition and assignment is part of the specification of each PM attribute.

Most performance attributes are counters. During the accumulation interval, the PM managed entity collects counter statistics in accordance with each PM attribute definition, and continuously compares the accumulated values with any thresholds that may exist. When an accumulated value first equals or exceeds the threshold, the ONT originates a TCA. At the end of the current 15-minute interval, the ONT issues a second TCA, cancelling the first.

If a counter PM attribute should fill up during the interval, it remains at its maximum possible value, rather than rolling over.

When a PM attribute is an average, its value is computed only at the end of the interval. A TCA on such an attribute can therefore be declared only at the end of the interval. The TCA would then be immediately cleared as the accumulator was reset for the next interval. The definition of a given PM attribute may specify different or more detailed behaviour.

When a PM attribute is a high-water mark, a TCA is declared when the monitored parameter equals or exceeds the threshold value from below; conversely for a low-water mark attribute. There is no general definition of the mechanism to clear the TCA, nor specification of delay or hysteresis to avoid TCA storms for a parameter fluctuating near the threshold value. These should be defined in the specification of each PM attribute.

It should be noted that TCAs are reported in OMCI alarm messages. There is no overlap between TCA codepoints and alarm codepoints, because a given ME class declares either alarms or TCAs, but not both.

Even when thresholding is defined on a PM managed entity, it is the option of the OLT whether to provision it (see clause I.1.10 regarding optional pointers). The PM managed entity template includes a placeholder for a pointer to an instance of the threshold data 1/2 managed entities. Threshold configuration is performed by means of threshold data MEs, which are created and deleted by the OLT. If no assigned threshold attribute number exceeds seven in a PM ME's definition, the existence of a threshold data 2 ME is optional.

The control block of a PM ME contains persistent data that can be set by the OLT. Setting the value of a control block attribute therefore increments MIB sync. In addition, the control block attribute of a PM ME is included in MIB upload. Other PM attributes are transient and are not included in MIB uploads.

**Template for the definition of a PM ME**

Existing PM generally follows the outline below. Significant exceptions are discussed here; an implementation is advised to be aware that the definitions of individual MEs and attribues may contain other exceptions.

<Description>

*Relationships*

<Relationships>

*Attributes*

**Managed entity id**: This attribute is discussed further in the separate clauses below.

**Interval end time**: This attribute identifies the most recently finished 15-minute interval. (R) (mandatory) (1 byte)

**Control block**: This attribute is discussed further in the separate clauses below. In classical PM, it is just a pointer to threshold data MEs.

**PM1**: Definition of first PM accumulation attribute, in most cases a counter, in other cases, an average, a high-water mark or a low-water mark. Four bytes is the recommended size of a PM accumulation attribute, but definitions vary. (R) (mandatory) (4 bytes)

**PM2…**: Definition of additional PM accumulation attributes, maximum not to exceed 14. There is no particular preference on the order in which parameters are defined.

*Actions*

Create, delete, get, set

Get current data (optional)

*Notifications*

**Threshold crossing alert**

| Number | Threshold crossing alert | Threshold data counter # (Note) |
|---|---|---|
| 1 | PM1 | 1 |
| 2 | PM2 | 2 |
| 3 | … | 3 |
| 4 | | 4 |
| Note – This number associates the TCA with the specified threshold value attribute of the threshold data 1 managed entity. | | |

The **Threshold crossing alert** column lists thresholded parameters in order. The **Number** column merely identifies a row in the table and should not be considered significant.

The **Threshold data counter** column assigns threshold attributes of the threshold data 1 and if required, the threshold data 2 MEs. In all cases, the assignment is monotonic, but existing PM definitions may or may not skip a threshold attribute for each PM attribute that is to be thresholded. In future PM definitions, it is recommended that threshold attributes be assigned sequentially, without gaps. If, for example, the only thresholded PM attributes were the first, third and sixth in order of PM ME definition, they would still be assigned threshold attributes 1, 2, 3.

### I.1.9.1 Classical PM

In classical PM, the managed entity ID attribute takes the same value as the parent managed entity's ME ID, so that no explicit pointer to the parent ME is required. The ME class of the parent is fixed in the definition of the classical PM ME.

**Managed entity id**: This attribute uniquely identifies each instance of this managed entity. Through an identical ID, this managed entity is implicitly linked to an instance of a <parent managed entity class>. (R, Set-by-create) (mandatory) (2 bytes)

In classical PM, the control block attribute is always a simple pointer to a threshold data ME, and is designated as such. The attribute value may be set to a null pointer if no thresholding is desired. If no assigned threshold number exceeds 7, it is the OLT's option whether to create a threshold data 2 ME or not. The template text reads as follows, depending on the highest threshold attribute assigned:

**Threshold data 1/2 id**: This attribute points to an instance of the threshold data 1 and 2 managed entities that contains PM threshold values. (R, W, Set-by-create) (mandatory) (2 bytes)

Or:

**Threshold data 1/2 id**: This attribute points to an instance of the threshold data 1 managed entity that contains PM threshold values. Since no threshold value attribute number exceeds 7, a threshold data 2 ME is optional. (R, W, Set-by-create) (mandatory) (2 bytes)

### I.1.9.1 Extended PM

In extended PM, the control block attribute is defined to be (R, W, Set-by-create) (mandatory) (16 bytes). The template for these 16 bytes is as follows:

**Threshold data 1/2 id**: (2 bytes) The definition of this field is the same as that in classical PM. When PM is collected on a continuously-running basis, rather than in 15-minute intervals, counter thresholds should not be established. There is no mechanism to clear a TCA, and any counter parameter may eventually be expected to cross any given threshold value.

**Parent ME class**: (2 bytes) This field contains the enumerated value of the ME class of the PM ME's parent. Together with the parent ME instance field, this permits a given PM ME to be associated with any OMCI ME. The definition of an extended PM ME should list the allowed parent ME classes.

**Parent ME instance**: (2 bytes) This field identifies the specific parent ME instance to which the PM ME is attached.

**Accumulation disable**: (2 bytes) This bit field allows PM accumulation to be disabled; refer to table xx-x. The default value 0 enables PM collection. If bit 15 is set to 1, no PM is collected by this ME instance. If bit 15 = 0 and any of bits 14..1 are set to 1, PM collection is inhibited for the attributes indicated by the 1 bits. Inhibiting PM collection does not change the value of a PM attribute, but if PM is accumulated in 15-minute intervals, the value is lost at the next 15-minute interval boundary.

Bit 16 is an action bit that always reads back as 0. When written to 1, it resets all PM attributes in the ME, and clears any TCAs that may be outstanding.

**Table xx-x – Bit assignments in extended PM control block**

| **Bit** | 16 | 15 | 14 | 13 | … | 3 | 2 | 1 (LSB) |
|---|---|---|---|---|---|---|---|---|
| **Accumulation disable** | Global clear | Global disable | PM14 | … | | … | PM2 | PM1 |
| **TCA disable** | | Global disable | Th14 | … | | … | Th2 | Th1 |

**TCA disable**: (2 bytes) Also clarified in table xx-x, this field permits TCAs to be inhibited, either individually or for the complete managed entity instance. As with the accumulation disable field, the default value 0 enables TCAs, and setting the global disable bit overrides the settings of the individual thresholds. Unlike the accumulation disable field, the bits are mapped to the thresholds defined in the associated threshold data 1 and 2 ME instances. When the global or attribute-specific value changes from 0 to 1, outstanding TCAs are cleared, either for the ME instance globally or for the individual disabled threshold.

These bits affect only notifications, not the underlying parameter accumulation or storage.

If the threshold data 1/2 id attribute does not contain a valid pointer, this field is not meaningful.

Thresholds should be used with caution if PM attributes are accumulated continuously.

**Control fields**: (2 bytes) This field is a bit map whose values govern the behaviour of the PM ME. Bits are assigned as follows:

| | |
|---|---|
| Bit 1 (LSB) | The value 1 specifies continuous accumulation, regardless of 15-minute intervals. There is no concept of current and historic accumulators; get and get current data (if supported) both return current values. The value 0 specifies 15-minute accumulators exactly like those of classical PM. |
| Bit 2 | The value 0 specifies directionality, for example upstream or downstream, or up/down with respect to a bridge port. If this bit is meaningful, the details are part of the definition of the extended PM ME. |
| Bits 3..16 | Reserved. Starting from bit 16, and working downward, these bits may be used in the definition of individual extended PM MEs. Continuing upward from bit 3, these bits may be used for additional purposes that pertain to all extended PM MEs. |

For example, in a VLAN extended PM ME, bits 16-15 could be used to match P-bits, VID or both.

**Reserved**: (4 bytes) These bytes are available for customization in the definition of each extended PM ME.

For example, in a VLAN extended PM ME, two of these bytes could be used to specify TCI.

The other template boiler plate fields are revised to read as follows:

**Managed entity id**: This attribute uniquely identifies each instance of this managed entity. To facilitate discovery, it is encouraged to identify instances sequentially starting with 1. (R, Set-by-create) (mandatory) (2 bytes)

**Interval end time**: This attribute identifies the most recently finished 15-minute interval. If continuous accumulation is enabled in the control block, this attribute is not used and has the fixed value 0. (R) (mandatory) (1 byte)

**Threshold data 1/2 id**: <same textual options as in classical PM>. Thresholding is not advised for counter attributes if PM is accumulated continuously. (R, W, Set-by-create) (mandatory) (2 bytes)

It is not expected that PM accumulation policy will be changed in actual deployment practice, and the behaviour of intervals, TCAs and accumulated history across a transition between continuous and interval accumulation is not specified. It may be desirable to disable and clear PM at such a transition.

The synchronize time action has no observable effect on PM that is accumulated continuously.

Counter PM attributes do not roll over in interval PM mode, but do roll over from maximum to zero in continuous accumulation mode. PM attributes that record averages are undefined in continuous accumulation mode. Both of these behaviours may be overridden by explicit specification in the definition of a given extended PM ME.

### I.1.10 Optional pointers

*No change to this clause*

## I.2 Common services

The common services consist of:

a) Start-up phase of ONT;

b) on demand circuit pack provisioning;

c) on demand circuit pack de-provisioning;

d) plug-and-play circuit pack provisioning;

e) plug-and-play circuit pack de-provisioning;

f) software image download;

g) software image changes;

All the listed services are explained by the use of scenario diagrams.

### I.2.1 Start-up phase of ONT

The start-up phase of an ONT, from the OMCI point of view, belongs to one of two cases:

a) the ONT is "new" to the OLT; or

b) the OLT already "saw" this ONT at this PON.

The details of start-up scenarios also vary for ONTs with different configuration options, e.g.

a) ONT with cardholders at both PON IF and UNI;

b) ONT with integrated interfaces at both PON IF and UNI;

c) ONT with cardholders at PON IF and integrated interfaces at UNI; and

d) ONT with integrated interfaces at PON IF and cardholders at UNI.

Here the following scenarios only show cases a) and b), from which the scenarios for cases c) and d) can be deduced. Additional scenarios can be derived for the cases where an ONT contains common equipment and/or protected equipment.

NOTE – The preferred solution is that cardholder and circuit managed entities should always be modelled, regardless of whether or not the ONT has integrated interfaces. However, the port mapping package provides another way to map heterogeneous ports to a single parent equipment.

Figure I.2.1-1 shows the start-up phase of a "new" ONT with Cardholders on both sides. Figure I.2.1-2 shows the start-up phase of a "new" ONT with integrated interfaces on both sides. Figure I.2.1-3 shows the start-up phase of an "old" ONT.

The behaviour of the ONT with respect to inserted circuit packs during the start-up phase is not shown in the following figures. This behaviour is the subject of clause I.2.2.

Note that if attribute value changes do not arrive at the OLT, the OLT will not know the number of cardholders or integrated ports that reside in the ONT. The OLT can request the information of the newly created managed entity instances by a sequence of “get” requests. If a “get” request is issued

on a non-existing instance, the response message to the OLT indicates the error – Unknown managed entity instance.

In general, the AVCs shown in figures I.2.1-1 and I.2.1-2 should be viewed as a partial method of ONT discovery. The OLT cannot rely on the receipt of AVCs to learn all ONT information, because not all managed entities or attributes issue AVCs, and because AVCs can be lost in transmission without an error being detected. Therefore, the OLT should audit any ONT immediately after a reset is completed.

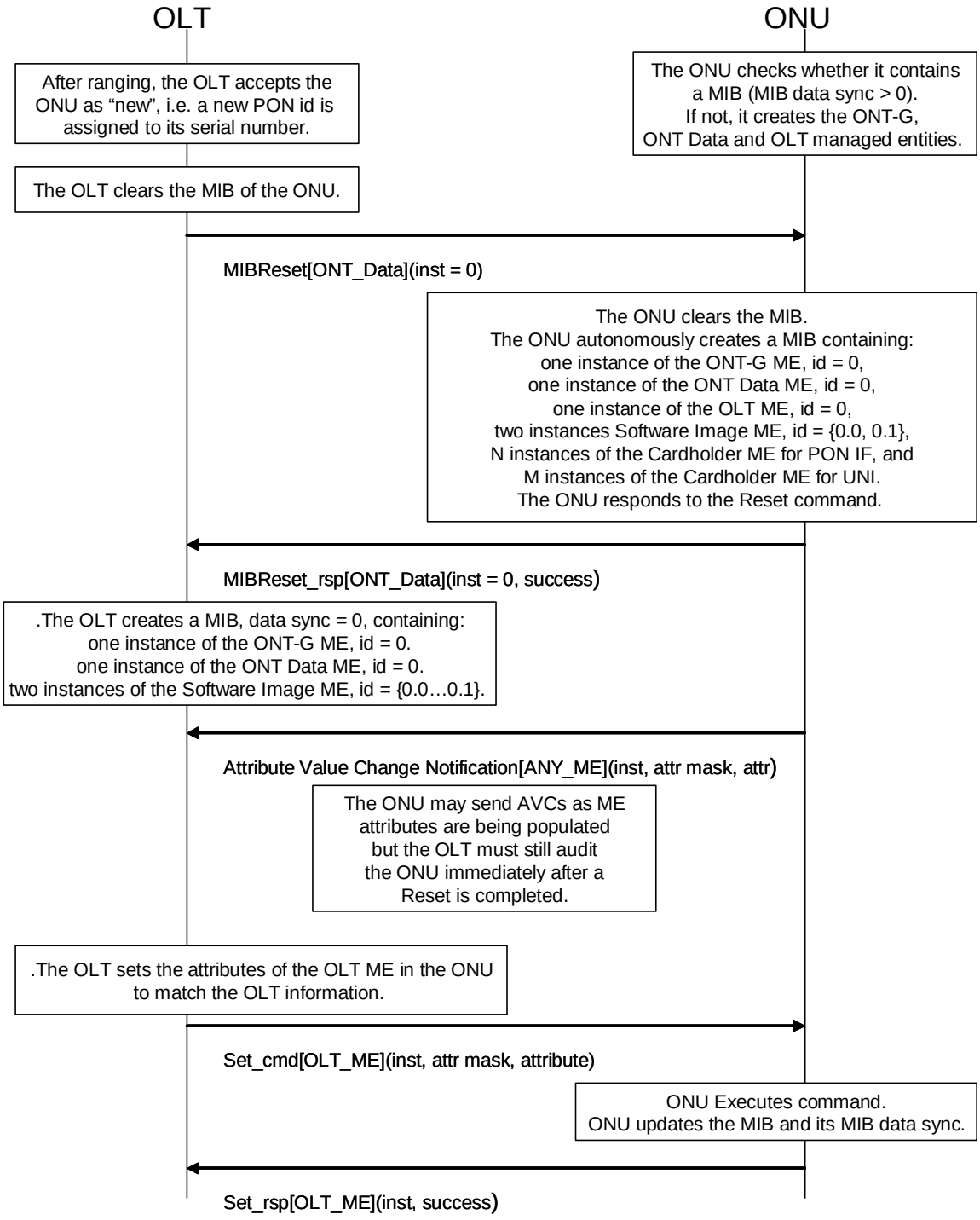



**Figure I.2.1-1 – Start-up of a "new" ONT with cardholders on both sides**

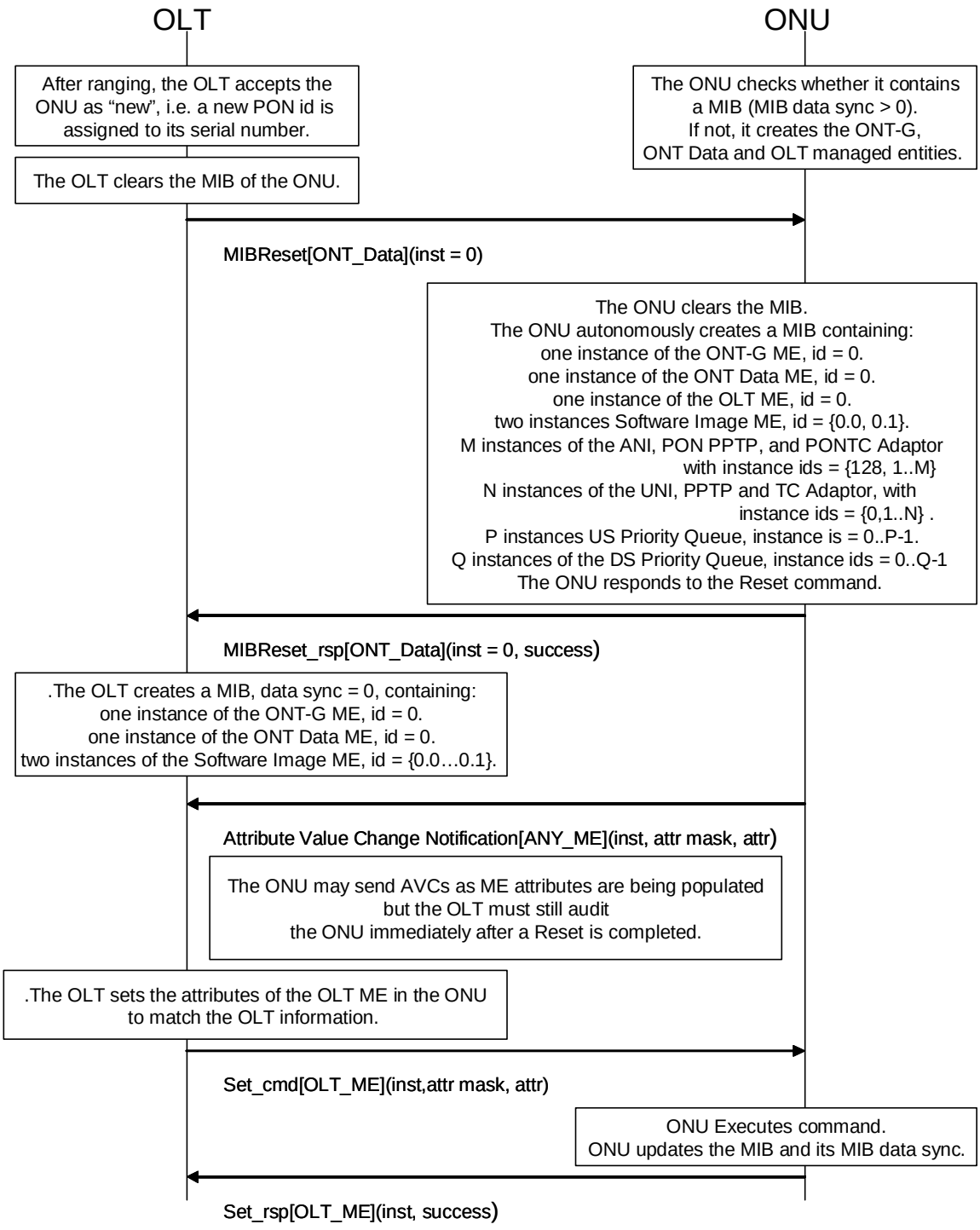



**Figure I.2.1-2 – Start-up phase of a "new" ONT with integrated interfaces on both sides**

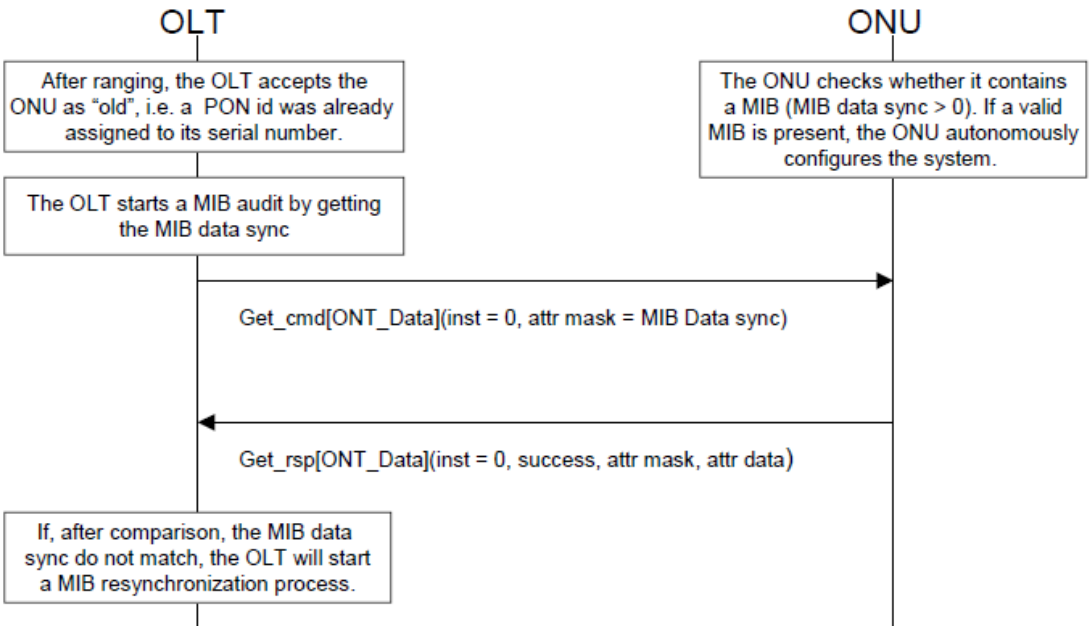



**Figure I.2.1-3 – Start-up phase of an "old" ONT**

### I.2.2 Circuit pack provisioning/de-provisioning

*No change to this clause*

### I.2.3 On-demand circuit pack provisioning

*No change to this clause*

### I.2.4 On-demand circuit pack de-provisioning

*No change to this clause*

### I.2.5 Plug-and-play circuit pack provisioning

*No change to this clause*

### I.2.6 Plug-and-play circuit pack de-provisioning

*No change to this clause*

### I.2.7 Software image download

The software image download operation is a file transfer from the OLT to the ONT. Software download is first described in the context of the baseline OMCI message set; the description is then modified as appropriate for the extended message set.

#### I.2.7.1 Baseline message set download

The atomic unit of file transfer is the **section**, the 31 bytes of data that can be transferred in a single (baseline) download section message. The last section in a software download may be padded with null bytes as needed.

A number of sections comprise a so-called **window**. The size of a window may not exceed 256

sections. During the initial softward download message exchange, the OLT proposes a maximum window size, but a lower value can be stipulated by the ONT, which must be accepted by the OLT. The OLT may send windows with fewer sections than this negotiated maximum, but may not exceed the maximum. Though it is not a preferred choice, the OLT may send all windows at the full negotiated maximum size, with the final window of the download operation padded out with download section messages containing only null pad bytes.

Each download section message contains a sequence number, which begins anew at 0 with each window. By tracking the incrementing sequence numbers, the ONT can confirm that it has in fact received each section of code.

In the message type field of the last download section message of the window, the OLT indicates the end of the window by setting the AR (acknowledgement request) bit – prior download section messages are unacknowledged. If the ONT has not received the entire window correctly, ie if it misses a sequence number or discards a download section because of a CRC error, it acknowledges with a command processing error result, whereupon the OLT falls back to the beginning of the window and tries again. To improve the chance of successful transmission, the OLT may choose to reduce the size of the window on its next attempt.

When the final window has been successfully downloaded, the OLT sends an end software download message whose contents include the size of the downloaded image in bytes, along with a CRC-32 computed according to I.363.5, across the entire image but excluding pad bytes that may have been transmitted. If the ONT agrees with both of these values, it acknowledges successful completion of the download and updates the software image validity attribute to indicate that the newly downloaded image is valid.

The ONT should not positively acknowledge an end download message until it has confirmed image size and CRC, and performed whatever operations may be necessary – such as non-volatile storage – to accept an immediate activate or commit message from the OLT. The ONT should respond with a device busy result code until these operations are complete, and the OLT should periodically re-try the end download command.

The nested state machines in OLT and ONT can conceivably get out of step in a number of unspecified ways, nor is it specified how to escape from a loop of transmission failure and retry. As a recovery mechanism from detectable state errors, it is recommended that the ONT reply with command processing error result codes to both acknowledged download section and end software download commands, and that the OLT send a final end software download command with known bad CRC and image size (eg all 0), whereupon both OLT and ONT should reset to the state in which no download is in progress, that is, state S1/S1' of figure 9.1.6-1. Likewise, the OLT should be able to abort the download operation at any time by sending an end software download message with invalid CRC and image size.

As well as the download of an image to the ONT as a whole, the download messages allow an option to download an image to each of several circuit packs in parallel. The starting assumption is that the OLT knows the set of circuit packs that require the same download file, so that it can specify this set in the download command sequence.

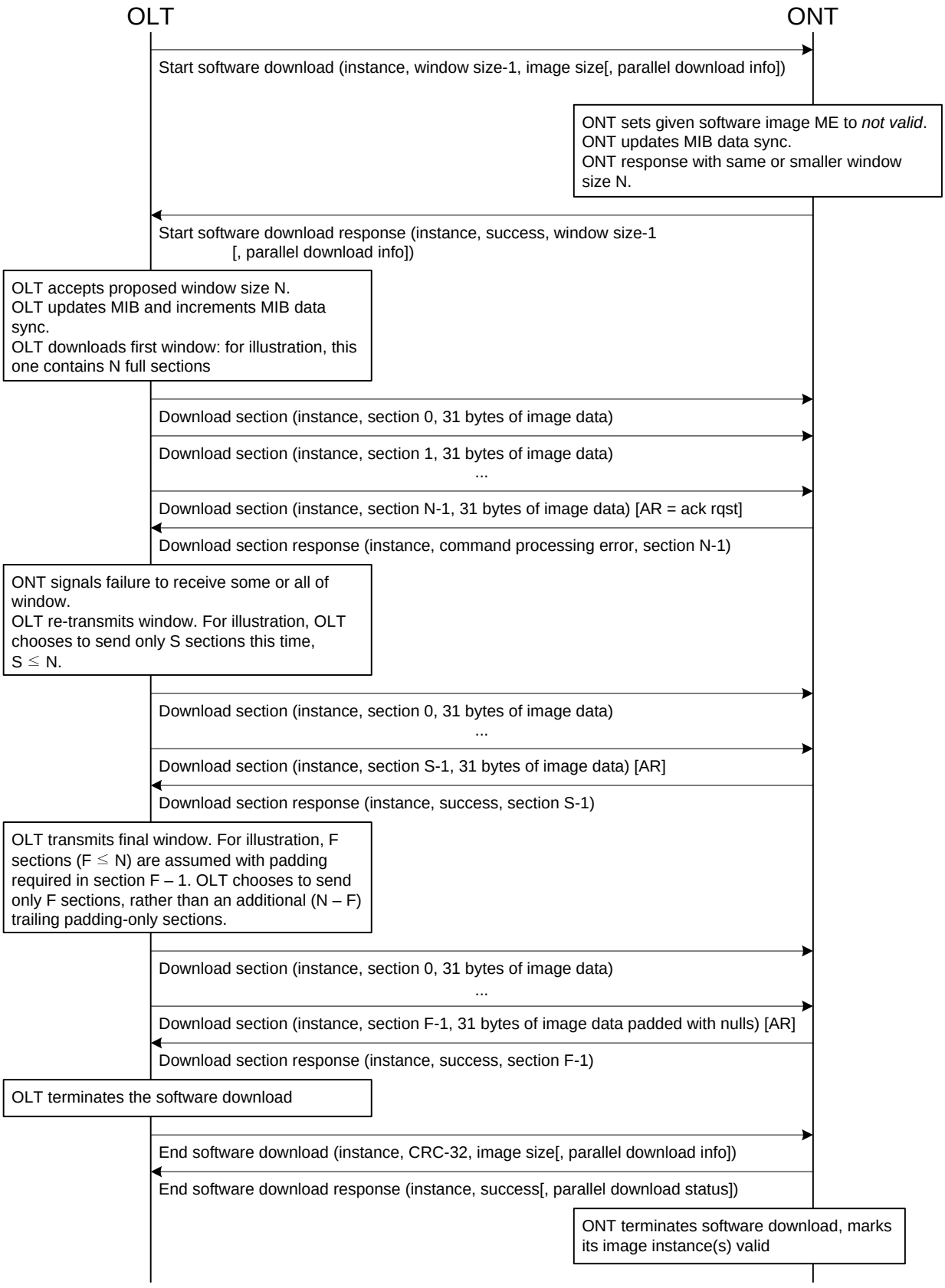



**Figure I.2.7-1 – Software download**

### I.2.7.2 Extended message set download

When the extended message set is used for software download, the maximum size of the section is limited by the extended message format itself, and is potentially as large as 1965 bytes. The OLT may send smaller sections at will, including the final section of a file transfer. Because the extended message format allows for variable length, software image sections are never padded in this message format.

### I.2.8 Software image activate and commit

*No change to this clause*

## Appendix II OMCI message set

*Revise clause Appendix II to read as shown below*

### II.1 General remarks

### II.1.1 Message type identifier

The message types are given in clause 11.1.4.

### II.1.2 Entity class identifier

The entity class identifiers are given in clause 11.1.6.

### II.1.3 Result and reason

Responses to commands can indicate the result of the command. A zero value indicates that the command was processed successfully. Non-zero values indicate the reason of the failure. If the result was failure, the rest of the message contents may provide details of the failure, be filled with all 0, or in the extended message set, simply be omitted. The definition of each result and reason appears in table II.1.3-1:

**Table II.1.3-1 – Result and reason codes**

| Code | Headline | Description |
|---|---|---|
| 0000 | Command processed successfully | There are two functions for command processing: command interpretation and command execution. This result means that the received command, such as get/set/test/reboot, was properly interpreted by the ONT's command interpretation function without errors and that the interpreted command was successfully transferred to the ONT's command execution function. |
| 0001 | Command processing error | This result means the command processing failed at the ONT for some reason not described by one of the more specific error codes. |
| 0010 | Command not supported | This result means that the message type (baseline message set byte 8, extended byte 3) is not supported by the ONT. |

**Table II.1.3-1 – Result and reason codes**

| Code | Headline | Description |
|---|---|---|
| 0011 | Parameter error | This result means that the command message received by the ONT was errored. It would be appropriate if an attribute mask were out of range, for example. In practice, this result code is frequently used interchangeably with code 1001. However, the optional-attribute and attribute execution masks in the reply messages are only defined for code 1001. |
| 0100 | Unknown managed entity | This result means that the managed entity class (baseline bytes 10..11, extended bytes 5..6) is not supported by the ONT. |
| 0101 | Unknown managed entity instance | This result means that the managed entity instance (baseline bytes 12..13, extended bytes 7..8) does not exist in the ONT. |
| 0110 | Device busy | This result means that the command could not be processed due to process-related congestion at the ONT. This result code may also be used as a pause indication to the OLT while the ONT conducts a time-consuming operation such as storage of a software image into non-volatile memory. |
| 0111 | Instance exists | This result means that the ONT already has a managed entity instance that corresponds to the one the OLT is attempting to create. |
| 1001 | Attribute(s) failed or unknown | This result means that an optional attribute is not supported by the ONT or that a mandatory/optional attribute could not be executed by the ONT, even if it is supported, for example because of a range or type violation. In conjunction with this result, attribute masks are used to indicate which attributes failed or were unknown.<br>The following two kinds of attribute masks are used when this result/reason is raised:<br>• *optional attribute mask coding*, which indicates whether or not the optional attribute is supported;<br>• *attribute execution mask coding*, which indicates whether or not the mandatory/optional attribute was executed.<br>See the set response and get response message layouts (clauses II.2.10, II.2.12, II.3.10 and II.3.12) for the placement of these masks.<br>If one or more optional attributes are not supported by the ONT, the optional attribute mask coding for each *unsupported* optional attribute becomes 1 while the corresponding attribute execution mask coding remains 0.<br>If one or more mandatory or optional attributes were not executed by the ONT, the optional attribute mask coding remains 0, while the attribute execution mask coding becomes 1 for each *failed* attribute.<br>If the ONT could not latch copies of all specified table attributes, for example because of insufficient memory, the attribute execution mask is set to 1 for each attribute that does *not* have a latched copy available for get next retrieval. |

### II.1.4 Get, get response, create response, and set messages

For an attribute mask, a bit map is used in the get, get response, create response, and set messages. This bit map indicates which attributes are requested (get) or provided (get response and set). The bit map is composed as follows:

| Byte | Bit | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| 1 | Attribute 1 | Attribute 2 | Attribute 3 | Attribute 4 | Attribute 5 | Attribute 6 | Attribute 7 | Attribute 8 |
| 2 | Attribute 9 | Attribute 10 | Attribute 11 | Attribute 12 | Attribute 13 | Attribute 14 | Attribute 15 | Attribute 16 |

Attribute numbers correspond to the ordering of the attributes in clause 9. Note that the managed entity identifier, which is an attribute of each managed entity, has no corresponding bit in the attribute mask. Thus, attributes are counted starting from the first attribute after the managed entity identifier.

### II.1.5 Alarm notifications

The ONT sends this notification each time an alarm status changes for the entity indicated in the message identifier. The message shows the status of all alarms of this entity. It is up to the OLT to determine which alarm status has changed.

The maximum number of alarms supported by OMCI for a given managed entity instance is 224 because of the available message field of the baseline get all alarm next message. The bit map is composed as follows:

| Byte | Bit | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| 1 | Alarm 0 | Alarm 1 | Alarm 2 | Alarm 3 | Alarm 4 | Alarm 5 | Alarm 6 | Alarm 7 |
| 2 | Alarm 8 | Alarm 9 | Alarm 10 | Alarm 11 | Alarm 12 | Alarm 13 | Alarm 14 | Alarm 15 |
| … | | | | | | | | |
| 28 | Alarm 216 | Alarm 217 | Alarm 218 | Alarm 219 | Alarm 220 | Alarm 221 | Alarm 222 | Alarm 223 |

Alarm numbers correspond to the alarm coding in clause 9. Bits in the alarm bit map that correspond to non-existing alarms are always set to 0. Bits that correspond to defined alarms are set to 0 to indicate that the corresponding alarm is cleared or to 1 to indicate that the alarm is currently active.

Alarm message sequence numbers can take values in the interval 1 to 255. Zero is excluded in order to make this counter similar to the MIB data sync counter.

### II.1.6 Test, test response and test result

*No change to this clause*

## II.2 Baseline OMCI message layout

*In all sub-clauses of this clause, whenever bytes 10-11 read "Message identifier," revise the text to read "Managed entity identifier."*

### II.2.26 Attribute value change

*Revise the notes at the end of the table to read per the following paragraphs.*

Note 1: For table attributes, the AVC message contains no attribute value (only a mask), and no snapshot of the table is created. If the OLT wishes to obtain the new value, it must then do a get operation, followed by the required number of get next operations.

Note 2: If there is insufficient space in the message body for the new values of all changed (non-table) attributes, the ONT should issue multiple AVCs, each with a consistent attribute mask and list of new attribute values, the total to include all changed attributes and their new values.

### II.2.27 Test

*Revise the introductory text to read per the following paragraph, and add the two new text formats at the end of this clause.*

The format of the test message is specific to the target entity class. A number of formats are presently defined. Future test extensions for a given entity class can be supported by adding additional encodings to presently unused bits or bytes. Future specification of tests for other entity classes may use an existing format or may define new formats for the test message. These extension mechanisms allow future tests to be supported without changing the principle of operation.

**Format for dot1ag MEP entity class, loopback test**

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction identifier | 6-7 | | | | | | | | | |
| Message type | 8 | 0 | 1 | 0 | | | | | | DB = 0, AR = 1, AK = 0<br>bits 5-1: action = test |
| Device identifier type | 9 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | OMCI = 0x0A |
| Managed entity identifier | 10-11 | | | | | | | | | Entity class.<br>NOTE – This format applies to the dot1ag MEP entity class |
| | 12 | | | | | | | | | MSB entity instance |
| | 13 | | | | | | | | | LSB entity instance |
| Message contents | 14 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | x | x=select test<br>0: Ethernet loopback test<br>1: 802.1ag link trace test (see separate format description below)<br>Other values reserved |

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| | 15 | 0 | 0 | 0 | c | p | p | p | d | If c = 1, the value of the MEP's CCM and LTM priority attribute is used, with drop eligibility false.<br>If c = 0, pppd represents the priority (p bits) and drop eligibility fields of the transmitted LBM frame. |
| | 16-21 | | | | | | | | | MAC address of target MHF or MEP, or 0 if the destination MEP ID is to be used instead. 802.1ag specifies unicast addresses; Y.1731 also allows for multicast. |
| | 22-23 | | | | | | | | | Destination MEP ID, in the range 1..8191, or 0 if the MAC address in bytes 16-21 is to be used instead. |
| | 24-25 | | | | | | | | | Repetition count, range 1..1024. This governs how many LBMs are generated. The rate at which LBMs are generated is not specified. If 5 seconds elapses with no LBRs received, the test aborts. |
| | 26-27 | | | | | | | | | These four fields are pointers to as many as four octet string MEs, which are concatenated to form an octet string of up to 1500 bytes. The string is packaged into a data TLV and transmitted as part of the LBM. If all four fields are null pointers, no data TLV is sent. If only one octet string is needed, it should be specified in bytes 23..24, etc, with null pointers in the higher-numbered bytes of the test message. |
| | 28-29 | | | | | | | | | |
| | 30-31 | | | | | | | | | |
| | 32-33 | | | | | | | | | |
| | 34-45 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Padding |

**Format for dot1ag MEP entity class, linktrace test**

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction identifier | 6-7 | | | | | | | | | |
| Message type | 8 | 0 | 1 | 0 | | | | | | DB = 0, AR = 1, AK = 0<br>bits 5-1: action = test |
| Device identifier type | 9 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | OMCI = 0x0A |
| Managed entity identifier | 10-11 | | | | | | | | | Entity class.NOTE – This format applies to the dot1ag MEP entity class |
| | 12 | | | | | | | | | MSB entity instance |
| | 13 | | | | | | | | | LSB entity instance |
| Message contents | 14 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | x | x=select test<br>0: Ethernet loopback test (see |

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | separate format description above)<br>1: 802.1ag link trace test<br>Other values reserved |
| | 15 | f | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Flags, a bit mask<br>f: *Use FDB only*. When 1, the bridge uses only its normal MAC forwarding tables for forwarding. When 0, the bridge may also consult its MIP CCM database to determine the forwarding port. |
| | 16-21 | | | | | | | | | Unicast MAC address of target MHF or MEP, or 0 if the destination MEP ID is to be used instead. |
| | 22-23 | | | | | | | | | Destination MEP ID, in the range 1..8191, or 0 if the unicast MAC address in bytes 16-21 is to be used instead. |
| | 24 | | | | | | | | | Max hops count – specifies initial TTL (time to live); limits the number of relay stages through which the LTM is forwarded before being discarded, and the number of LTRs that may be returned. IEEE 802.1ag recommends a default value of 64. |
| | 25-26 | | | | | | | | | Pointer to a general purpose buffer ME, used to return the link trace results. The ONT should deny the test operation command if this field is a null or an invalid pointer. |
| | 27-45 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Padding |

### II.2.44 Get next response

*Replace the message contents part of the table with the following (to correct implication that bytes 17..xx could contain content from more than one attribute):*

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Message contents | 14 | 0 | 0 | 0 | 0 | x | x | x | x | Result, reason<br>0000 = command processed successfully<br>0001 = command processing error<br>0010 = command not supported<br>0011 = parameter error<br>0100 = unknown managed entity<br>0101 = unknown managed entity instance<br>0110 = device busy |
| | 15 | | | | | | | | | MS byte attribute mask |
| | 16 | | | | | | | | | LS byte attribute mask |
| | 17..xx | | | | | | | | | Attribute value (size depending on |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | the type of attribute) |
| | xx-45 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Padding |

### II.2.45 Test result

*Revise the introductory text to read per the following paragraphs, and add two new test result formats at the end of this clause.*

The test result message is used to report the outcome of a test. In the case of a requested test, the transaction identifier of the test result message is identical to the transaction identifier of the test message that initiated the corresponding test. In the case of a self-triggered test result, the transaction identifier is set to 0.

Several formats are currently defined. They are used as follows:

- Self-test results, ONT-G, circuit pack, or any other ME that supports self test
- Vendor-specific test results, generic format, any ME that supports it
- POTS (or BRI) test results, either MLT, dial tone draw-break or vendor-specific POTS tests that use a general purpose buffer
- ICMP tests, either ping or traceroute
- The results of an optical line supervision test on the ANI-G, RE ANI-G, PPTP RE UNI, RE upstream amplifier or RE downstream amplifier.
- IEEE 802.1ag loopback and link trace tests

If a new test for the presently-supported entities is defined in the future, the corresponding test results can be reported by extending the test result message layout. If a new test for other managed entity classes is defined in the future, a new test result message layout may be defined.

**Format for test action invoked against dot1ag MEP entity class, loopback test**

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction identifier | 6-7 | | | | | | | | | |
| Message type | 8 | 0 | 0 | 1 | | | | | | DB = 0, AR = 0, AK =0<br>bits 5-1: action = test result |
| Device identifier type | 9 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | OMCI = 0x0A |
| Managed entity identifier | 10-11 | | | | | | | | | Entity class.<br>NOTE – This message format pertains to the dot1ag MEP entity class. |
| | 12 | | | | | | | | | MSB entity instance |
| | 13 | | | | | | | | | LSB entity instance |
| Message contents | 14 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | x | X = 1: indicates failure to receive any loopback replies (LBRs) within 5 seconds |
| | 15-16 | | | | | | | | | Valid LBRs count: the number of valid, in-order LBRs received. |
| | 17-18 | | | | | | | | | Out of order LBRs count: the number of valid LBRs received that were out of order. |
| | 19-20 | | | | | | | | | Mismatch LBRs count: the number of received LBRs whose MAC SDU did |

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | not match that of the corresponding LBM (except for opcode). Optional feature, set to 0xFF if not supported. |
| | 21-24 | | | | | | | | | Delay from LB message transmission to LB response reception, measured in microseconds. The value 0 indicates no information available. |
| | 25-45 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Padding |

**Format for test action invoked against dot1ag MEP entity class, linktrace test**

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction identifier | 6-7 | | | | | | | | | |
| Message type | 8 | 0 | 0 | 1 | | | | | | DB = 0, AR = 0, AK =0<br>bits 5-1: action = test result |
| Device identifier type | 9 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | OMCI = 0x0A |
| Managed entity identifier | 10-11 | | | | | | | | | Entity class.<br>NOTE – This message format pertains to the dot1ag MEP entity class. |
| | 12 | | | | | | | | | MSB entity instance |
| | 13 | | | | | | | | | LSB entity instance |
| Message contents | 14 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | x | X = 1: indicates failure to receive any linktrace replies (LTRs) within 5 seconds |
| | 15-18 | | | | | | | | | Transaction ID of transmitted LTM |
| | 19-26 | | | | | | | | | Content of egress TLV data field in transmitted LTM (IEEE 802.1ag clause 21.8.8). The LTRs themselves are captured in the general purpose buffer designated by the test command. |
| | 27-45 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Padding |

## II.3 Extended OMCI message layout

Extended OMCI messages may be up to 1980 bytes long, including headers. The previous discussion of oversize get next messages remains applicable, although problems are less likely to occur in practice.

### II.3.1 Create

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |
| Message type | 3 | 0 | 1 | 0 | | | | | | DB = 0, AR = 1, AK = 0<br>bits 5-1: action = create |
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Managed entity identifier | 5-6 | | | | | | | | | Entity class |
| | 7 | | | | | | | | | MS byte entity instance |
| | 8 | | | | | | | | | LS byte entity instance |
| Message contents length | 9-10 | | | | | | | | | Size of message contents field, bytes |
| Message contents | 11-n | | | | | | | | | Value of first set-by-create attribute, NOT the ME ID (size depending on the type of attribute) |
| | | | | | | | | | | … |
| | | | | | | | | | | Value of last set-by-create attribute |

The contents of the create message apply only to attributes that are defined to be set-by-create. Writeable attributes that are not set-by-create are not permitted in a create message. Thus, the first byte of the message contents field begins with the value of the first set-by-create attribute and so forth. Space for each set-by-create attribute must be allocated in the create message, even if the attribute is optional. When an optional attribute is not to be instantiated, the placeholder value to be entered into this space is specific to the definition of each attribute.

### II.3.2 Create response

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |
| Message type | 3 | 0 | 0 | 1 | | | | | | DB = 0, AR = 0, AK = 1<br>bits 5-1: action = create |
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class |
| | 7 | | | | | | | | | MS byte entity instance |
| | 8 | | | | | | | | | LS byte entity instance |
| Message contents length | 9-10 | | | | | | | | | Size of message contents field, bytes |

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Message contents | 11 | 0 | 0 | 0 | 0 | x | x | x | x | Result, reason<br>0000 command processed successfully<br>0001 command processing error<br>0010 command not supported<br>0011 parameter error<br>0100 unknown managed entity<br>0101 unknown managed entity instance<br>0110 device busy<br>0111 instance exists |
| | 12 | | | | | | | | | Attribute execution mask (attributes 1-8), used when result, reason = 0011<br>0 attribute ok<br>1 illegal attribute value |
| | 13 | | | | | | | | | Attribute execution mask (attributes 9-16), used when result, reason = 0011<br>0 attribute ok<br>1 illegal attribute value |

NOTE – If the result, reason code is not 0011, the attribute execution mask in bytes 12-13 is omitted.

### II.3.3 This clause intentionally left blank

### II.3.4 This clause intentionally left blank

### II.3.5 Delete

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |
| Message type | 3 | 0 | 1 | 0 | | | | | | DB = 0, AR = 1, AK = 0<br>bits 5-1: action = delete |
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class |
| | 7 | | | | | | | | | MS byte entity instance |
| | 8 | | | | | | | | | LS byte entity instance |
| Message contents length | 9-10 | | | | | | | | | Size of message contents field = 0 |

### II.3.6 Delete response

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |
| Message type | 3 | 0 | 0 | 1 | | | | | | DB = 0, AR = 0, AK = 1<br>bits 5-1: action = delete |
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class |
| | 7 | | | | | | | | | MS byte entity instance |
| | 8 | | | | | | | | | LS byte entity instance |
| Message contents length | 9-10 | | | | | | | | | Size of message contents field = 1 byte |
| Message contents | 11 | | | | | | | | | Result, reason<br>0000 command processed successfully<br>0001 command processing error<br>0010 command not supported<br>0011 parameter error<br>0100 unknown managed entity<br>0101 unknown managed entity instance<br>0110 device busy |

### II.3.7 This clause intentionally left blank

### II.3.8 This clause intentionally left blank

### II.3.9 Set

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |
| Message type | 3 | 0 | 1 | 0 | | | | | | DB = 0, AR = 1, AK = 0<br>bits 5-1: action = set |
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class |
| | 7 | | | | | | | | | MS byte entity instance |
| | 8 | | | | | | | | | LS byte entity instance |
| Message contents length | 9-10 | | | | | | | | | Size of message contents field, bytes |
| Message contents | 11 | | | | | | | | | Attributes mask (attributes 1-8) |
| | 12 | | | | | | | | | Attributes mask (attributes 9-16) |
| | 13-n | | | | | | | | | Value of first attribute to set (size depending on the type of attribute) |

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | … |
| | | | | | | | | | | Value of last attribute to set |

### II.3.10 Set response

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |
| Message type | 3 | 0 | 0 | 1 | | | | | | DB = 0, AR = 0, AK = 1<br>bits 5-1: action = set |
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class |
| | 7 | | | | | | | | | MS byte entity instance |
| | 8 | | | | | | | | | LS byte entity instance |
| Message contents length | 9-10 | | | | | | | | | Size of message contents field, bytes |
| Message contents | 11 | 0 | 0 | 0 | 0 | x | x | x | x | Result, reason<br>0000 command processed successfully<br>0001 command processing error<br>0010 command not supported<br>0011 parameter error<br>0100 unknown managed entity<br>0101 unknown managed entity instance<br>0110 device busy<br>1001 attribute(s) failed or unknown |
| | 12 | | | | | | | | | Optional-attribute mask (attributes 1-8), used with 1001 encoding:<br>0 default<br>1 unsupported attribute |
| | 13 | | | | | | | | | Optional-attribute mask (attributes 9-16), used with 1001 encoding:<br>0 default<br>1 unsupported attribute |
| | 14 | | | | | | | | | Attribute execution mask (attributes 1-8), used with 1001 encoding:<br>0 default<br>1 failed attribute |
| | 15 | | | | | | | | | Attribute execution mask (attributes 9-16), used with 1001 encoding:<br>0 default<br>1 failed attribute |

NOTE – If the result code is not 1001, the attribute masks in bytes 12-15 are omitted.

### II.3.11 Get

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |
| Message type | 3 | 0 | 1 | 0 | | | | | | DB = 0, AR = 1, AK = 0<br>bits 5-1: action = get |
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class |
| | 7 | | | | | | | | | MS byte entity instance |
| | 8 | | | | | | | | | LS byte entity instance |
| Message contents length | 9-10 | | | | | | | | | Size of message contents field = 2 bytes |
| Message contents | 11 | | | | | | | | | Attributes mask (attributes 1-8) |
| | 12 | | | | | | | | | Attributes mask (attributes 9-16) |

### II.3.12 Get response

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |
| Message type | 3 | 0 | 0 | 1 | | | | | | DB = 0, AR = 0, AK = 1<br>bits 5-1: action = get |
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class |
| | 7 | | | | | | | | | MS byte entity instance |
| | 8 | | | | | | | | | LS byte entity instance |
| Message contents length | 9-10 | | | | | | | | | Size of message contents field, bytes |
| Message contents | 11 | 0 | 0 | 0 | 0 | x | x | x | x | Result, reason<br>0000 command processed successfully<br>0001 command processing error<br>0010 command not supported<br>0011 parameter error<br>0100 unknown managed entity<br>0101 unknown managed entity instance<br>0110 device busy<br>1001 attribute(s) failed or unknown |
| | 12 | | | | | | | | | Attribute mask (attributes 1-8) |
| | 13 | | | | | | | | | Attribute mask (attributes 9-16) |

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| | 14 | | | | | | | | | Optional-attribute mask (attributes 1-8), used with 1001 encoding:<br>0 default<br>1 unsupported attribute |
| | 15 | | | | | | | | | Optional-attribute mask (attributes 9-16), used with 1001 encoding:<br>0 default<br>1 unsupported attribute |
| | 16 | | | | | | | | | Attribute execution mask (attributes 1-8), used with 1001 encoding:<br>0 default<br>1 failed attribute |
| | 17 | | | | | | | | | Attribute execution mask (attributes 9-16), used with 1001 encoding:<br>0 default<br>1 failed attribute |
| | 18-n | | | | | | | | | Value of first attribute included (size depending on the type of attribute) |
| | | | | | | | | | | … |
| | | | | | | | | | | Value of last attribute included |

Bytes 14-17 are reserved for the optional-attribute and attribute execution masks; however, the content of these bytes is only valid in conjunction with result code 1001 used to indicate failed or unknown attributes. When the result code is not 1001, these bytes should be set to 0 by the ONT transmitter and ignored by the OLT receiver.

NOTE – the position of these bytes differs from that in the baseline get response message.

When the OLT wishes to retrieve a table attribute, ie an attribute whose size is, or might be, larger than the space available in one OMCI baseline message, the ONT indicates the size of that attribute in bytes, rather than its value. The size is conveyed as four bytes in the value field for that attribute, with the attribute execution mask set to indicate that the attribute is included. The OLT should then use a sequence of get next messages to retrieve such an attribute. This convention also pertains to extended OMCI messages, even though some table attributes might fit into an extended get response message.

### II.3.13 This clause intentionally left blank

### II.3.14 This clause intentionally left blank

### II.3.15 Get all alarms

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |
| Message type | 3 | 0 | 1 | 0 | | | | | | DB = 0, AR = 1, AK = 0<br>bits 5-1: action = get all alarms |

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class = ONT data |
| | 7 | | | | | | | | | MS byte entity instance |
| | 8 | | | | | | | | | LS byte entity instance |
| Message contents length | 9-10 | | | | | | | | | Size of message contents field = 1 |
| Message contents | 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | x | x = alarm retrieval mode<br>0 Get all alarms regardless of ARC status<br>1 Get all alarms not currently under ARC |

### II.3.16 Get all alarms response

The semantics of the message contents differs from that of the corresponding baseline OMCI message. In the extended OMCI get all alarms next response, a number of managed entity instances can be reported in a single message, so the get all alarms response message contents do *not* indicate the count of managed entity instances, but rather specify the number of get all alarms next messages required.

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |
| Message type | 3 | 0 | 0 | 1 | | | | | | DB = 0, AR = 0, AK = 1<br>bits 5-1: action = get all alarms |
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class = ONT data |
| | 7 | | | | | | | | | MS byte entity instance |
| | 8 | | | | | | | | | LS byte entity instance |
| Message contents length | 9-10 | | | | | | | | | Size of message contents field = 2 |
| Message contents | 11 | | | | | | | | | MS byte of the number of get all alarms next messages required |
| | 12 | | | | | | | | | LS byte of the number of get all alarms next messages required |

### II.3.17 Get all alarms next

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Message type | 3 | 0 | 1 | 0 | | | | | | DB = 0, AR = 1, AK = 0<br>bits 5-1: action = get all alarms next |
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class = ONT data |
| | 7 | | | | | | | | | MS byte entity instance |
| | 8 | | | | | | | | | LS byte entity instance |
| Message contents length | 9-10 | | | | | | | | | Size of message contents field = 2 |
| Message contents | 11 | | | | | | | | | MS byte of command sequence number |
| | 12 | | | | | | | | | LS byte of command sequence number |

Command sequence numbers start from 0.

### II.3.18 Get all alarms next response

This message differs from the equivalent baseline message in that a number of managed entity alarm bit maps may be returned within a single message.

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |
| Message type | 3 | 0 | 0 | 1 | | | | | | DB = 0, AR = 0, AK = 1<br>bits 5-1: action = get all alarms next |
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class = ONT data |
| | 7 | | | | | | | | | MS byte entity instance |
| | 8 | | | | | | | | | LS byte entity instance |
| Message contents length | 9-10 | | | | | | | | | Size of message contents field including all sub-parts, bytes |
| Message contents 1 | 11-12 | | | | | | | | | Entity class whose alarms are reported |
| | 13 | | | | | | | | | MS byte entity instance whose alarms are reported |
| | 14 | | | | | | | | | LS byte entity instance whose alarms are reported |
| | 15-42 | x | x | x | x | x | x | x | x | Bit map alarms |
| Message contents 2 | 43-44 | | | | | | | | | Entity class whose alarms are reported |
| | 45 | | | | | | | | | MS byte entity instance whose alarms are reported |
| | 46 | | | | | | | | | LS byte entity instance whose alarms are reported |
| | 47-74 | x | x | x | x | x | x | x | x | Bit map alarms |
| … | | | | | | | | | | Further managed entity alarm masks as |

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Message contents n | | | | | | | | | | needed |

The bit map used in the get all alarms next response for a given managed entity class is identical to the bit map used in the alarm notifications for that managed entity class.

If the ONT receives a get all alarms next request message whose command sequence number is out of range, the get all alarms next response message should contain a null message contents field.

### II.3.19 MIB upload

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |
| Message type | 3 | 0 | 1 | 0 | | | | | | DB = 0, AR = 1, AK = 0<br>bits 5-1: action = MIB upload |
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class = ONT data |
| | 7 | | | | | | | | | MS byte entity instance |
| | 8 | | | | | | | | | LS byte entity instance |
| Message contents length | 9-10 | | | | | | | | | Size of message contents field = 0 |

### II.3.20 MIB upload response

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |
| Message type | 3 | 0 | 0 | 1 | | | | | | DB = 0, AR = 0, AK = 1<br>bits 5-1: action = MIB upload |
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class = ONT data |
| | 7 | | | | | | | | | MS byte entity instance |
| | 8 | | | | | | | | | LS byte entity instance |
| Message contents length | 9-10 | | | | | | | | | Size of message contents field = 2 |
| Message contents | 14 | | | | | | | | | MS byte of the number of MIB upload next commands required |
| | 15 | | | | | | | | | LS byte of the number of MIB upload next commands required |

### II.3.21 MIB upload next

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |
| Message type | 3 | 0 | 1 | 0 | | | | | | DB = 0, AR = 1, AK = 0<br>bits 5-1: action = MIB upload next |
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class = ONT data |
| | 7 | | | | | | | | | MS byte entity instance |
| | 8 | | | | | | | | | LS byte entity instance |
| Message contents length | 9-10 | | | | | | | | | Size of message contents field = 2 |
| Message contents | 14 | | | | | | | | | MS byte of command sequence number |
| | 15 | | | | | | | | | LS byte of command sequence number |

Command sequence numbers start from 0.

### II.3.22 MIB upload next response

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |
| Message type | 3 | 0 | 0 | 1 | | | | | | DB = 0, AR = 0, AK = 1<br>bits 5-1: action = MIB upload next |
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class = ONT data |
| | 7 | | | | | | | | | MS byte entity instance |
| | 8 | | | | | | | | | LS byte entity instance |
| Message contents length | 9-10 | | | | | | | | | Size of message contents field including all sub-fields, bytes |
| Message contents, ME instance 1 | 11-12 | | | | | | | | | Size of ME instance 1 attribute values included (excluding bytes 11-18), bytes |
| | 13-14 | | | | | | | | | Entity class of ME instance 1 |
| | 15 | | | | | | | | | MS byte entity instance |
| | 16 | | | | | | | | | LS byte entity instance |
| | 17 | | | | | | | | | Attribute mask (attributes 1-8) |
| | 18 | | | | | | | | | Attribute mask (attributes 9-16) |
| | 19-n | | | | | | | | | Value of first attribute (size depending on the type of the attribute) |
| | | | | | | | | | | ... |
| | | | | | | | | | | Value of last attribute |
| Message contents, ME instance 2 | | | | | | | | | | Content of ME instance 2, defined as above. |

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| … | | | | | | | | | | … |
| Message contents, ME instance k | | | | | | | | | | Content of ME instance k, defined as above. |

Note that, if not all attributes of a managed entity fit within one MIB upload next response message, the attributes are split over several messages. The OLT can use the information in the attribute mask to determine which attribute values are reported in which MIB upload next response message.

Thus, a single extended MIB upload next response message must contain an integer number of attribute values. A message may contain leading or trailing fragments of ME instances and any number of complete ME instances.

If the ONT receives a MIB upload next request message whose command sequence number is out of range, it should respond with a message containing no message contents field. This is also the appropriate response if the ONT times out (one minute) from the most recent MIB upload next or MIB upload request from the OLT.

### II.3.23 MIB reset

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |
| Message type | 3 | 0 | 1 | 0 | | | | | | DB = 0, AR = 1, AK = 0<br>bits 5-1: action = MIB reset |
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class = ONT data |
| | 7 | | | | | | | | | MS byte entity instance |
| | 8 | | | | | | | | | LS byte entity instance |
| Message contents length | 9-10 | | | | | | | | | Size of message contents field = 0 |

### II.3.24 MIB reset response

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |
| Message type | 3 | 0 | 0 | 1 | | | | | | DB = 0, AR = 0, AK = 1<br>bits 5-1: action = MIB reset |
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class = ONT data |
| | 7 | | | | | | | | | MS byte entity instance |
| | 8 | | | | | | | | | LS byte entity instance |
| Message contents | 9-10 | | | | | | | | | Size of message contents field = 1 |

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| length | | | | | | | | | | |
| Message contents | 11 | 0 | 0 | 0 | 0 | x | x | x | x | Result, reason<br>0000 command processed successfully<br>0001 command processing error<br>0010 command not supported<br>0011 parameter error<br>0100 unknown managed entity<br>0101 unknown managed entity instance<br>0110 device busy |

### II.3.25 Alarm

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |
| Message type | 3 | 0 | 0 | 0 | | | | | | DB = 0, AR = 0, AK = 0<br>bits 5-1: action = alarm |
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class |
| | 7 | | | | | | | | | MS byte entity instance |
| | 8 | | | | | | | | | LS byte entity instance |
| Message contents length | 9-10 | | | | | | | | | Size of message contents field, bytes |
| Message contents | 11 | | | | | | | | | Alarm mask |
| | | | | | | | | | | … |
| | 38 | | | | | | | | | Alarm mask |
| | 39 | | | | | | | | | Alarm sequence number |

### II.3.26 Attribute value change

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |
| Message type | 3 | 0 | 0 | 0 | | | | | | DB = 0, AR = 0, AK = 0<br>bits 5-1: action = attribute value change |
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class |
| | 7 | | | | | | | | | MS byte entity instance |
| | 8 | | | | | | | | | LS byte entity instance |

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Message contents length | 9-10 | | | | | | | | | Size of message contents field, bytes |
| Message contents | 11 | | | | | | | | | MS byte attribute mask |
| | 12 | | | | | | | | | LS byte attribute mask |
| | 13-n | | | | | | | | | Value of first changed attribute (size depending on the type of attribute) |
| | | | | | | | | | | ... |
| | | | | | | | | | | Value of last changed attribute |

The AVC message for a table attribute contains no attribute value, only a mask, and the ONT creates no snapshot of the table. If the OLT wishes to obtain the new value, it must do a get operation, followed by the required number of get next operations.

### II.3.27 Test

The format of the test message is specific to the target entity class. Several formats are defined. Future test extensions for a given entity class can be supported by adding additional encodings to presently unused bits or bytes. Future specification of tests for other entity classes may use an existing format or may define new formats for the test message. These extension mechanisms allow future tests to be supported without changing the principle of operation.

**Format for ONT-G, ANI-G, RE ANI-G, PPTP RE UNI, RE upstream amplifier, RE downstream amplifier and circuit pack entity classes**

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |
| Message type | 3 | 0 | 1 | 0 | | | | | | DB = 0, AR = 1, AK = 0<br>bits 5-1: action = test |
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class.<br>NOTE – This format applies to entity classes ONT-G, ANI-G, RE ANI-G, PPTP RE UNI, RE upstream amplifier, RE downstream amplifier and circuit pack. |
| | 7 | | | | | | | | | MS byte entity instance |
| | 8 | | | | | | | | | LS byte entity instance |
| Message contents length | 9-10 | | | | | | | | | Size of message contents field = 1 |
| Message contents | 11 | 0 | 0 | 0 | 0 | x | x | x | x | xxxx = select test<br>0000..0110 Reserved for future use<br>0111 Self test<br>1000..1111 Vendor-specific use. See description related to the test result message. |

**Format for IP host config data entity class**

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation indentifier | 1-2 | | | | | | | | | |
| Message type | 3 | 0 | 1 | 0 | | | | | | DB=0, AR=1, AK=0<br>Bits 5-1: action=test |
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class.<br>NOTE – This format applies to entity class IP host config data. |
| | 7 | | | | | | | | | MS byte entity instance |
| | 8 | | | | | | | | | LS byte entity instance |
| Message contents length | 9-10 | | | | | | | | | Size of message contents field, bytes |
| Message contents | 11 | 0 | 0 | 0 | 0 | x | x | x | x | xxxx = select test<br>0001 Ping<br>0010 Traceroute<br>0010..0111 Reserved<br>1000..1111 Vendor-specific use.<br>The ICMP message is intended to be sent from the ONT upstream toward the network. See discussion related to the test result message. |
| | 12-15 | | | | | | | | | IP address of target |

**Format for POTS UNI and PPTP ISDN UNI entity classes**

This message supports two basic categories of test operation, a defined set of tests that look in and out from the POTS port, and a set of codepoints that may be used for vendor-specific tests. The latter category is further subdivided into codepoints that return test results in a general purpose buffer ME, using the test result message primarily as an event trigger to signal test completion, and codepoints that return all test results in an ordinary test result message. If it is needed, the OLT must create the general purpose buffer managed entity before initiating the test action.

Note that a single message can be used to initiate multiple tests on a given ME if desired.

Bytes 12-25 are used by the dial tone make-break test. A zero value for a timer causes the ONT to use its built-in defaults. As many as three dial tone frequencies can be specified, or omitted by setting their values to 0. Other fields are also omitted with the value 0, or controlled by flags. An ONT can support the dial tone test with internal defaults only, and is not required to support any of the attributes of bytes 12-25. Likewise, an ONT can use internal defaults for drop test, rather than the values given in bytes 26-35. The capabilities of an ONT are documented by the vendor and known through administrative practices.

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |
| Message type | 3 | 0 | 1 | 0 | | | | | | DB = 0, AR = 1, AK = 0<br>Bits 5-1: action = test |

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class<br>NOTE – This format applies to entity classes PPTP POTS UNI and PPTP ISDN UNI. |
| | 7 | | | | | | | | | MS byte entity instance |
| | 8 | | | | | | | | | LS byte entity instance |
| Message contents length | 9-10 | | | | | | | | | Size of message contents field, bytes |
| Message contents | 11 | a | 0 | 0 | 0 | x | x | x | x | a – test mode<br>0 normal; deny test if line busy<br>1 forced mode<br>xxxx = select test<br>0000 all MLT tests<br>0001 hazardous potential<br>0010 foreign EMF<br>0011 resistive faults<br>0100 receiver off-hook<br>0101 ringer<br>0110 NT1 dc signature test<br>0111 self test<br>1000 dial tone make-break test<br>1001..1011 vendor-specific test, all results returned in test results message<br>1100..1111 vendor-specific test, test results returned in general purpose buffer ME. The ONT should deny a test operation command in this range if bytes 36..37 do not point to a GP buffer. |
| | 12 | | | | | | | | | DBDT timer T1 (slow dial tone threshold), in units of 0.1 seconds. Range 0.1 to 6.0 seconds. |
| | 13 | | | | | | | | | DBDT timer T2 (no dial tone threshold), in units of 0.1 seconds. Range 1.0 to 10.0 seconds. |
| | 14 | | | | | | | | | DBDT timer T3 (slow break dial tone threshold), in units of 0.1 seconds. Range 0.1 to 3.0 seconds. |
| | 15 | | | | | | | | | DBDT timer T4 (no break dial tone threshold), in units of 0.1 seconds. Range 1.0 to 3.0 seconds. |

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| | 16 | | | | | | | d | p | DBDT control byte<br>d: dialled digit<br>1 dialled digit specified in byte 17<br>0 use default digit<br>p = pulse (1) or tone (0) dialling |
| | 17 | | | | | | | | | Digit to be dialled, ASCII character in range "0"-"9", "*", "#". |
| | 18-19 | | | | | | | | | Dial tone frequency 1, in units of Hz |
| | 20-21 | | | | | | | | | Dial tone frequency 2, in units of Hz. 0 = unused (i.e., if only one tone is specified). |
| | 22-23 | | | | | | | | | Dial tone frequency 3, in units of Hz. 0 = unused (i.e., if only one or two tones are specified). |
| | 24 | | | | | | | | | Dial tone power threshold, absolute value, 0.1 dB resolution, range [–]0.1 to [–]25.3 dBm0. Eg –13 dBm0 = 0x82. 0 = unspecified. |
| | 25 | | | | | | | | | Idle channel power threshold, absolute value, 1 dB resolution, range [–]1 to [–]90 dBm0. 0 = unspecified. |
| | 26 | | | | | | | | | DC hazardous voltage threshold, absolute value, volts. 0 = unspecified. |
| | 27 | | | | | | | | | AC hazardous voltage threshold, volts RMS. 0 = unspecified. |
| | 28 | | | | | | | | | DC foreign voltage threshold, absolute value, volts. 0 = unspecified. |
| | 29 | | | | | | | | | AC foreign voltage threshold, volts RMS. 0 = unspecified. |
| | 30 | | | | | | | | | Tip-ground and ring-ground resistance threshold, kΩ. 0 = unspecified. |
| | 31 | | | | | | | | | Tip-ring resistance threshold, kΩ. 0 = unspecified. |
| | 32-33 | | | | | | | | | Ringer equivalence minimum threshold, in 0.01 REN units. 0 = unspecified. |
| | 34-35 | | | | | | | | | Ringer equivalence maximum threshold, in 0.01 REN units. 0 = unspecified. |
| | 36-37 | | | | | | | | | Pointer to a general purpose buffer ME, used to return vendor-specifc test results. |

**Format for dot1ag MEP entity class, loopback test**

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |
| Message type | 3 | 0 | 1 | 0 | | | | | | DB = 0, AR = 1, AK = 0<br>Bits 5-1: action = test |
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class<br>NOTE – This format applies to the dot1ag MEP entity class |
| | 7 | | | | | | | | | MS byte entity instance |
| | 8 | | | | | | | | | LS byte entity instance |
| Message contents length | 9-10 | | | | | | | | | Size of message contents field, bytes |
| Message contents | 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | x | x=select test<br>0: Ethernet loopback test<br>1: 802.1ag link trace test (see separate format description below)<br>Other values reserved |
| | 12 | 0 | 0 | 0 | c | p | p | p | d | If c = 1, the value of the MEP's CCM and LTM priority attribute is used, with drop eligibility false.<br>If c = 0, pppd represents the priority (p bits) and drop eligibility fields of the transmitted LBM frame. |
| | 13-18 | | | | | | | | | MAC address of target MHF or MEP, or 0 if the destination MEP ID is to be used instead. 802.1ag specifies unicast addresses; Y.1731 also allows for multicast. |
| | 19-20 | | | | | | | | | Destination MEP ID, in the range 1..8191, or 0 if the MAC address in bytes 13..18 is to be used instead. |
| | 21-22 | | | | | | | | | Repetition count, range 1..1024. This governs how many LBMs are generated. The rate at which LBMs are generated is not specified. If 5 seconds elapses with no LBRs received, the test aborts. |
| | 23-24 | | | | | | | | | These four fields are pointers to as many as four octet string MEs, which are concatenated to form an octet string of up to 1500 bytes. The string is packaged into a data TLV and transmitted as part of the LBM. If all four fields are null pointers, no data TLV is sent. If only one octet string is needed, it should be specified in bytes |
| | 25-26 | | | | | | | | | |
| | 27-28 | | | | | | | | | |

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| | 29-30 | | | | | | | | | 23..24, etc, with null pointers in the higher-numbered bytes of the test message. |

**Format for dot1ag MEP entity class, linktrace test**

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |
| Message type | 3 | 0 | 1 | 0 | | | | | | DB = 0, AR = 1, AK = 0<br>Bits 5-1: action = test |
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class<br>NOTE – This format applies to the dot1ag MEP entity class |
| | 7 | | | | | | | | | MS byte entity instance |
| | 8 | | | | | | | | | LS byte entity instance |
| Message contents length | 9-10 | | | | | | | | | Size of message contents field, bytes |
| Message contents | 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | x | x=select test<br>0: Ethernet loopback test (see separate format description above)<br>1: 802.1ag link trace test<br>Other values reserved |
| | 12 | f | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Flags, a bit mask<br>f: *Use FDB only*. When 1, the bridge uses only its normal MAC forwarding tables for forwarding. When 0, the bridge may also consult its MIP CCM database to determine the forwarding port. |
| | 13-18 | | | | | | | | | Unicast MAC address of target MHF or MEP, or 0 if the destination MEP ID is to be used instead. |
| | 19-20 | | | | | | | | | Destination MEP ID, in the range 1..8191, or 0 if the unicast MAC address in bytes 13..18 is to be used instead. |
| | 21 | | | | | | | | | Max hops count – specifies initial TTL (time to live); limits the number of relay stages through which the LTM is forwarded before being discarded, and the number of LTRs that may be returned. IEEE 802.1ag recommends a default value of 64. |

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| | 22-23 | | | | | | | | | Pointer to a general purpose buffer ME, used to return the link trace results. The ONT should deny the test operation command if this field is a null or an invalid pointer. |

### II.3.28 Test response

If an ONT does not support all tests requested in byte 11 of the test request message, it should not execute any test and should respond with result 0010, command not supported. If an ONT supports all of the requested tests but cannot support one or more of the explicitly specified threshold attributes, it should not execute any test and should respond with result 0011, parameter error. The test command could then be re-issued with different thresholds or with default thresholds, and would be expected to succeed.

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |
| Message type | 3 | 0 | 0 | 1 | | | | | | DB = 0, AR = 0, AK = 1<br>bits 5-1: action = test |
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class |
| | 7 | | | | | | | | | MS byte entity instance |
| | 8 | | | | | | | | | LS byte entity instance |
| Message contents length | 9-10 | | | | | | | | | Size of message contents field = 1 |
| Message contents | 11 | 0 | 0 | 0 | 0 | x | x | x | x | Result, reason<br>0000 command processed successfully<br>0001 command processing error<br>0010 command not supported<br>0011 parameter error<br>0100 unknown managed entity<br>0101 unknown managed entity instance<br>0110 device busy |

The test response message is an indication to the OLT that the test request is received and is being processed. Test outcome is reported by a subsequent autonomous test result message.

### II.3.29 Start software download

When a file is to be downloaded to a single instance of the software image managed entity, the target ME id is specified in bytes 7..8. An optional feature permits the same file to be downloaded to a number of circuit packs by setting bytes 7..8 = 0xffff and specifying the software image ME ids in bytes 17-18, 19-20, etc.

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |
| Message type | 3 | 0 | 1 | 0 | | | | | | DB = 0, AR = 1, AK = 0<br>bits 5-1: action = start software download |
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class = software image |
| | 7 | | | | | | | | | MS byte of software image instance<br>0 ONT-G<br>1..254 slot number<br>255 download to multiple software image managed entities |
| | 8 | | | | | | | | | LS byte of software image instance<br>0 instance 0<br>1 instance 1<br>255 multiple download |
| Message contents length | 9-10 | | | | | | | | | Size of message contents field, bytes |
| Message contents | 11 | | | | | | | | | Window size – 1 |
| | 12-15 | | | | | | | | | Image size in bytes |
| | 16 | | | | | | | | | Number of circuit packs to be updated in parallel (value 1...9) |
| | 17 | | | | | | | | | MS byte of software image instance (slot number of circuit pack) |
| | 18 | | | | | | | | | LS byte of software image instance (value 0..1) |
| | 19-xx | | | | | | | | | Software image ME ids (same format as bytes 17..18) for additional simultaneous downloads. |

### II.3.30 Start software download response

When a file is downloaded to a single software image ME, the response contains the target ME id in bytes 7..8, a result code in byte 11, and a window size counter-proposal (which may be the same as that suggested by the OLT in the original request). Bytes 13..N are omitted.

An ONT that supports the optional parallel download feature responds to a multiple download command with the full format shown below, where unused trailing image references may be omitted. If the ONT does not support the parallel download feature, it responds with result code 0b0101, unknown managed entity instance.

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Message type | 3 | 0 | 0 | 1 | | | | | | DB = 0, AR = 0, AK = 1<br>bits 5-1: action = start software download |
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class = software image |
| | 7 | | | | | | | | | MS byte of software image instance<br>0 ONT-G<br>1..254 slot number<br>255 download to multiple software image managed entities |
| | 8 | | | | | | | | | LS byte of software image instance<br>0 instance 0<br>1 instance 1<br>255 multiple download |
| Message contents length | 9-10 | | | | | | | | | Size of message contents field, bytes |
| Message contents | 11 | 0 | 0 | 0 | 0 | x | x | x | x | Result, reason<br>0000 command processed successfully<br>0001 command processing error<br>0010 command not supported<br>0011 parameter error<br>0100 unknown managed entity<br>0101 unknown managed entity instance<br>0110 device busy |
| | 12 | | | | | | | | | Window size – 1 |
| | 13 | | | | | | | | | Number of instances responding (value 0..9) |
| | 14-15 | | | | | | | | | ME id of software image entity instance (slot number plus instance 0..1) |
| | 16 | | | | | | | | | Result, reason for bytes 14..15 – same coding as byte 11 |
| | 17-n | | | | | | | | | Repeat coding of bytes 14..16 for additional requested software image instances. |

### II.3.31 Download section

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Message type | 3 | 0 | x | 0 | | | | | | DB = 0, AR = x, AK = 0<br>x = 0 no response expected (section within a window)<br>x = 1 response expected (last section of a window)<br>bits 5-1: action = sw download section |
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class = software image |
| | 7 | | | | | | | | | MS byte of software image instance<br>0 ONT-G<br>1..254 slot number<br>255 download to multiple software image managed entities |
| | 8 | | | | | | | | | LS byte of software image instance<br>0 instance 0<br>1 instance 1<br>255 multiple download |
| Message contents length | 9-10 | | | | | | | | | Size of message contents field, bytes |
| Message contents | 11 | | | | | | | | | Download section number |
| | 12-n | | | | | | | | | Software image data |

### II.3.32 Download section response

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |
| Message type | 3 | 0 | 0 | 1 | | | | | | DB = 0, AR = 0, AK = 1<br>bits 5-1: action = sw download section |
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class = software image |
| | 7 | | | | | | | | | MS byte of software image instance<br>0 ONT-G<br>1..254 slot number<br>255 download to multiple software image managed entities |
| | 8 | | | | | | | | | LS byte of software image instance<br>0 instance 0<br>1 instance 1<br>255 multiple download |

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Message contents length | 9-10 | | | | | | | | | Size of message contents field = 2 |
| Message contents | 11 | 0 | 0 | 0 | 0 | x | x | x | x | Result, reason<br>0000 command processed successfully<br>0001 command processing error<br>0010 command not supported<br>0011 parameter error<br>0100 unknown managed entity<br>0101 unknown managed entity instance<br>0110 device busy |
| | 12 | | | | | | | | | Download section number |

### II.3.33 End software download

The format of this command is similar to that of the start software download message. Bytes 19..n support the optional parallel download feature, and are omitted for download to a single target.

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |
| Message type | 3 | 0 | 1 | 0 | | | | | | DB = 0, AR = 1, AK = 0<br>bits 5-1: action = end software download |
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class = software image |
| | 7 | | | | | | | | | MS byte of software image instance<br>0 ONT-G<br>1..254 slot number<br>255 download to multiple software image managed entities |
| | 8 | | | | | | | | | LS byte of software image instance<br>0 instance 0<br>1 instance 1<br>255 multiple download |
| Message contents length | 9-10 | | | | | | | | | Size of message contents field, bytes |
| Message contents | 11-14 | | | | | | | | | CRC-32, computed over all bytes of the software image (excluding padding), as specified in ITU-T Rec I.363.5. |
| | 15-18 | | | | | | | | | Image size in bytes |
| | 19 | | | | | | | | | Number of parallel download instances sent in this mesage (value 1..9) |

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| | 20 | | | | | | | | | MS byte of software image instance (slot number of circuit pack) |
| | 21 | | | | | | | | | LS byte of software image instance (value 0..1) |
| | 22-xx | | | | | | | | | Software image ME ids (same format as bytes 20..21) for additional simultaneous downloads. |

### II.3.34 End software download response

The response message informs the OLT whether the download command was successful. If a single software image ME was targeted for download, byte 11 reports the result of the process. If a number of software images were targeted for parallel download, byte 11 reports device busy as long as any of the instances is busy writing the image to non-volatile store. Once the ONT has stored all images successfully, it responds to continued end software download commands with a 0 in byte 11 and a separate result for each software image ME.

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |
| Message type | 3 | 0 | 0 | 1 | | | | | | DB = 0, AR = 0, AK = 1<br>bits 5-1: action = end software download |
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class = software image |
| | 7 | | | | | | | | | MS byte of software image instance<br>0 ONT-G<br>1..254 slot number<br>255 download to multiple software image managed entities |
| | 8 | | | | | | | | | LS byte of software image instance<br>0 instance 0<br>1 instance 1<br>255 multiple download |
| Message contents length | 9-10 | | | | | | | | | Size of message contents field, bytes |
| Message contents | 11 | 0 | 0 | 0 | 0 | x | x | x | x | Result, reason<br>0000 command processed successfully (CRC correct)<br>0001 command processing error (CRC incorrect)<br>0100 unknown managed entity<br>0101 unknown managed entity instance<br>0110 device busy |

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| | 12 | | | | | | | | | Number of instances responding (value 0..9) |
| | 13-14 | | | | | | | | | ME id of software image entity instance (slot number plus instance 0..1) |
| | 15 | | | | | | | | | Result, reason for bytes 13..14 – same coding as byte 11 |
| | 16-n | | | | | | | | | Repeat coding of bytes 13..15 for additional software image instances. |

### II.3.35 Activate image

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |
| Message type | 3 | 0 | 1 | 0 | | | | | | DB = 0, AR = 1, AK = 0<br>bits 5-1: action = activate image |
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class = software image |
| | 7 | | | | | | | | | MS byte entity instance<br>0 ONT-G<br>1..254 slot number |
| | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | x | LS byte entity instance<br>0 first instance<br>1 second instance |
| Message contents length | 9-10 | | | | | | | | | Size of message contents field = 0 |

### II.3.36 Activate image response

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |
| Message type | 3 | 0 | 0 | 1 | | | | | | DB = 0, AR = 0, AK = 1<br>bits 5-1: action = activate image |
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class = software image |
| | 7 | | | | | | | | | MS byte entity instance<br>0 ONT-G<br>1..254 slot number |
| | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | x | LS byte entity instance |

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | 0 first instance<br>1 second instance |
| Message contents length | 9-10 | | | | | | | | | Size of message contents field = 1 |
| Message contents | 11 | 0 | 0 | 0 | 0 | x | x | x | x | Result, reason<br>0000 command processed successfully<br>0001 command processing error<br>0010 command not supported<br>0011 parameter error<br>0100 unknown managed entity<br>0101 unknown managed entity instance<br>0110 device busy |

### II.3.37 Commit image

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |
| Message type | 3 | 0 | 1 | 0 | | | | | | DB = 0, AR = 1, AK = 0<br>bits 5-1: action = commit image |
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class = software image |
| | 7 | | | | | | | | | MS byte entity instance<br>0 ONT-G<br>1..254 slot number |
| | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | x | LS byte entity instance<br>0 first instance<br>1 second instance |
| Message contents length | 9-10 | | | | | | | | | Size of message contents field = 0 |

### II.3.38 Commit image response

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |
| Message type | 3 | 0 | 0 | 1 | | | | | | DB = 0, AR = 0, AK = 1<br>bits 5-1: action = commit image |
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class = software image |

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| | 7 | | | | | | | | | MS byte entity instance<br>0 ONT-G<br>1..254 slot number |
| | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | x | LS byte entity instance<br>0 first instance<br>1 second instance |
| Message contents length | 9-10 | | | | | | | | | Size of message contents field = 1 |
| Message contents | 11 | 0 | 0 | 0 | 0 | x | x | x | x | Result, reason<br>0000 command processed successfully<br>0001 command processing error<br>0010 command not supported<br>0011 parameter error<br>0100 unknown managed entity<br>0101 unknown managed entity instance<br>0110 device busy |

### II.3.39 Synchronize time

The synchronize time command controls the tick boundary for performance monitoring collection, and optionally, a date and time clock. In the usual PM scenario, and using the baseline message set, the synchronize time command must be issued by the OLT at a 15-minute boundary of the OLT's time of day clock.

The extended synchronize time message contains the OLT's current time of day within the message itself, so that the ONT can reset its PM timer to a current value (in the range 0..900 seconds) at any time, rather than just at a 15-minute time of day boundary. Date and time are not explicitly required by any ONT features, but may be useful, for example in logging.

The message contents may be omitted, in which case the message behaves the same as the baseline synchronize time message, only specifying a 15-minute tick boundary by its own arrival time.

If the OLT does not wish to specify a date, it may set year, month and day fields to 0.

There is no intention that this message be used to establish a precise time of day reference.

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |
| Message type | 3 | 0 | 1 | 0 | | | | | | DB = 0, AR = 1, AK = 0<br>bits 5-1: action = synchronize time |
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class = ONT-G |
| | 7 | | | | | | | | | MS byte entity instance = 0 |
| | 8 | | | | | | | | | LS byte entity instance = 0 |

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Message contents length | 9-10 | | | | | | | | | Size of message contents field = 2 |
| Message contents | 11-12 | | | | | | | | | Year, eg 2009 |
| | 13 | | | | | | | | | Month, range 1..12 |
| | 14 | | | | | | | | | Day of month, range 1..31 |
| | 15 | | | | | | | | | Hour of day, range 0..23 |
| | 16 | | | | | | | | | Minute of hour, range 0..59 |
| | 17 | | | | | | | | | Second of minute, range 0..59 |

### II.3.40 Synchronize time response

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |
| Message type | 3 | 0 | 0 | 1 | | | | | | DB = 0, AR = 0, AK = 1<br>Bits 5-1: action = synchronize time |
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class = ONT-G |
| | 7 | | | | | | | | | MS byte entity instance = 0 |
| | 8 | | | | | | | | | LS byte entity instance = 0 |
| Message contents length | 9-10 | | | | | | | | | Size of message contents field = 1 |
| Message contents | 11 | | | | | | | | | Result, reason<br>0000 command processed successfully<br>0001 command processing error<br>0010 command not supported<br>0011 parameter error<br>0100 unknown managed entity<br>0101 unknown managed entity instance<br>0110 device busy |

### II.3.41 Reboot

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |
| Message type | 3 | 0 | 1 | 0 | | | | | | DB = 0, AR = 1, AK = 0<br>bits 5-1: action = reboot |
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Managed entity identifier | 5-6 | | | | | | | | | Entity class |
| | 7 | | | | | | | | | MS byte entity instance |
| | 8 | | | | | | | | | LS byte entity instance |
| Message contents length | 9-10 | | | | | | | | | Size of message contents field = 0 |

### II.3.42 Reboot response

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |
| Message type | 3 | 0 | 0 | 1 | | | | | | DB = 0, AR = 0, AK = 1<br>bits 5-1: action = reboot |
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class |
| | 7 | | | | | | | | | MS byte entity instance |
| | 8 | | | | | | | | | LS byte entity instance |
| Message contents length | 9-10 | | | | | | | | | Size of message contents field = 1 |
| Message contents | 11 | | | | | | | | | Result, reason<br>0000 command processed successfully<br>0001 command processing error<br>0010 command not supported<br>0011 parameter error<br>0100 unknown managed entity<br>0101 unknown managed entity instance<br>0110 device busy |

### II.3.43 Get next

The ONT should reject a get next command:

- If the attribute mask specifies more than one attribute (result code 0011).
- If the attribute mask specifies an attribute that is not a table (result code 0011).
- If the specified attribute has not been prepared for upload with a prior get command (the prior get is subject to one-minute timeout) (result code 0001).

Command sequence numbers start from 0.

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction | 1-2 | | | | | | | | | |

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| correlation identifier | | | | | | | | | | |
| Message type | 3 | 0 | 1 | 0 | | | | | | DB = 0, AR = 1, AK = 0<br>bits 5-1: action = get next |
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class |
| | 7 | | | | | | | | | MS byte entity instance |
| | 8 | | | | | | | | | LS byte entity instance |
| Message contents length | 9-10 | | | | | | | | | Size of message contents field = 4 |
| Message contents | 11 | | | | | | | | | Attribute mask (attributes 1-8) |
| | 12 | | | | | | | | | Attribute mask (attributes 9-16) |
| | 13 | | | | | | | | | MS byte of command sequence number |
| | 14 | | | | | | | | | LS byte of command sequence number |

### II.3.44 Get next response

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |
| Message type | 3 | 0 | 0 | 1 | | | | | | DB = 0, AR = 0, AK =1<br>bits 5-1: action = get next |
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class |
| | 7 | | | | | | | | | MS byte entity instance |
| | 8 | | | | | | | | | LS byte entity instance |
| Message contents length | 9-10 | | | | | | | | | Size of message contents field, bytes |
| Message contents | 11 | 0 | 0 | 0 | 0 | x | x | x | x | Result, reason<br>0000 command processed successfully<br>0001 command processing error<br>0010 command not supported<br>0011 parameter error<br>0100 unknown managed entity<br>0101 unknown managed entity instance<br>0110 device busy |
| | 12 | | | | | | | | | Attribute mask (attributes 1-8) |
| | 13 | | | | | | | | | Attribute mask (attributes 9-16) |

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| | 14-n | | | | | | | | | Value of the specified attribute (size depending on type of attribute, limited by message capacity) |

If the ONT receives a get next request message whose command sequence number is out of range, the ONT responds with parameter error.

### II.3.45 Test result

The test result message is used to report the outcome of a test. In the case of a requested test, the transaction identifier of the test result message is identical to the transaction identifier of the test message that initiated the corresponding test. In the case of a self-triggered test result, the transaction identifier is set to 0.

Several formats are currently defined. They are used as follows:

- Self test results, ONT-G, circuit pack, or any other ME that supports self test
- Vendor-specific test results, generic format, any ME that supports it
- POTS (or BRI) test results, either MLT, dial tone draw-break or vendor-specific POTS tests that use a general purpose buffer
- ICMP tests, either ping or traceroute
- The results of an optical line supervision test on the ANI-G, RE ANI-G, PPTP RE UNI, RE upstream amplifier or RE downstream amplifier.
- IEEE 802.1ag loopback and link trace tests

If a new test for the presently-supported entities is defined in the future, the corresponding test results can be reported by extending the test result message layout. If a new test for other managed entity classes is defined in the future, a new test result message layout may be defined.

**Format for self test action invoked against ONT-G and circuit pack entity classes**

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |
| Message type | 3 | 0 | 0 | 0 | | | | | | DB = 0, AR = 0, AK =0<br>bits 5-1: action = test result |
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class<br>NOTE – This message format pertains to ONT-G and circuit pack entity classes. |
| | 7 | | | | | | | | | MS byte entity instance |
| | 8 | | | | | | | | | LS byte entity instance |
| Message contents length | 9-10 | | | | | | | | | Size of message contents field = 2 |
| Message contents | 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Reserved |

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| | 12 | 0 | 0 | 0 | 0 | 0 | 0 | x | x | xx: self test result<br>00 failed<br>01 passed<br>10 not completed |

**Format for vendor-specific test actions invoked against ONT-G and circuit pack entity classes**

This format is also used for vendor-specific test actions invoked against the PPTP POTS UNI entity class when no general purpose buffer is needed.

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |
| Message type | 3 | 0 | 0 | 0 | | | | | | DB = 0, AR = 0, AK =0<br>bits 5-1: action = test result |
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class<br>NOTE – This message format pertains to ONT-G, circuit pack and PPTP POTS UNI entity classes. |
| | 7 | | | | | | | | | MS byte entity instance |
| | 8 | | | | | | | | | LS byte entity instance |
| Message contents length | 9-10 | | | | | | | | | Size of message contents field, bytes |
| Message contents | 11 | | | | | | | | | Type 1 (note) |
| | 12-13 | | | | | | | | | Value 1 |
| | 14 | | | | | | | | | Type 2 |
| | 15-16 | | | | | | | | | Value 2 |
| | 17 | | | | | | | | | Type 3 |
| | 18-19 | | | | | | | | | Value 3 |
| | 20 | | | | | | | | | Type 4 |
| | 21-22 | | | | | | | | | Value 4 |
| | 23 | | | | | | | | | Type 5 |
| | 24-25 | | | | | | | | | Value 5 |
| | 26 | | | | | | | | | Type 6 |
| | 27-28 | | | | | | | | | Value 6 |
| | 29 | | | | | | | | | Type 7 |
| | 30-31 | | | | | | | | | Value 7 |
| | 32 | | | | | | | | | Type 8 |
| | 33-34 | | | | | | | | | Value 8 |
| | 35 | | | | | | | | | Type 9 |
| | 36-37 | | | | | | | | | Value 9 |
| | 38 | | | | | | | | | Type 10 |

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| | 39-40 | | | | | | | | | Value 10 |
| NOTE – Test result types are specified in clause 11.2.10. Type-value fields are packed in the lowest byte positions. Unused trailing byte positions may be omitted. If more than 10 type-value pairs are to be returned, an additional test type should be defined in the test message. At the vendor's discretion, a test result may include an ordered sequence of repeated type-value pairs to represent, for example, port ordering, or first/second power input. In this case, missing values can be flagged with type = 255. | | | | | | | | | | |

**Format for POTS UNI and PPTP ISDN UNI entity classes**

Byte 11 reports a summary MLT test result. The result for each test category is limited to the two values *pass test or test not run* or *failed test*. Byte 13 reports the results of a dial tone test.

Byte 12 reports the result of a self test or a vendor-specific test that returns results in a general purpose buffer. At present, self test is not supported for the POTS UNI and PPTP ISDN UNI entity classes, and this byte should be set to 0.

There are four possible outcomes for a given test: it can pass, fail, not be run, or not be recognized by the ONT. If an ONT does not support or recognize a given test, it is expected to deny the test request message. To avoid physical damage, an ONT may cease testing if a test fails – usually hazardous potential – and thus some subsequent tests will not be run. In addition, the ONT may support some but not all tests of a given suite, such as power measurements in the dial tone test sequence. The category summary in byte 11 includes two values. The value 1 indicates either that all tests in a category passed, or that nothing in the category was tested, while 0 indicates that at least one test in the category failed. Further information appears in flags specific to each test results attribute to indicate whether each detailed test was run or not, whether it passed or failed and whether a measured result is reported or not.

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |
| Message type | 3 | 0 | 0 | 0 | | | | | | DB = 0, AR = 0, AK = 0<br>bits 5-1: action = test result |
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class<br>NOTE – This message format pertains to PPTP POTS UNI and PPTP ISDN UNI entity classes. |
| | 7 | | | | | | | | | MS byte entity instance |
| | 8 | | | | | | | | | LS byte entity instance |
| Message contents length | 9-10 | | | | | | | | | Size of message contents field, bytes |

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Message contents | 11 | 0 | 0 | a | b | c | d | e | f | MLT drop test result:<br>0 fail test a/b/c/d/e/f<br>1 pass test, or test not run<br>a hazardous potential<br>b foreign EMF<br>c resistive faults<br>d receiver off-hook<br>e ringer<br>f NT1 dc signature test |
| | 12 | 0 | 0 | 0 | 0 | 0 | 0 | x | x | xx: Result of self test or vendor-specific test<br>00 failed<br>01 passed<br>10 not completed |
| | 13 | | | b | b | b | d | d | d | Dial tone make-break flags:<br>ddd – Dial tone draw<br>000 test not run<br>01m failed, could not draw<br>10m slow draw<br>11m passed<br>bbb – Dial tone break<br>000 test not run<br>01m failed, could not break<br>10m slow break<br>11m passed<br>m – measured value flag<br>0 measurement not reported<br>1 measurement reported |
| | 14 | | | a | a | a | b | b | b | Dial tone power flags (note)<br>aaa – Quiet channel power<br>bbb – Dial tone power |
| | 15 | | | a | a | a | b | b | b | Loop test DC voltage flags (note)<br>aaa – VDC, tip-ground<br>bbb – VDC, ring-ground |
| | 16 | | | a | a | a | b | b | b | Loop test AC voltage flags (note)<br>aaa – VAC, tip-ground<br>bbb – VAC, ring-ground |
| | 17 | | | a | a | a | b | b | b | Loop test resistance flags 1 (note)<br>aaa – Resistance, tip-ground<br>bbb – Resistance, ring-ground |
| | 18 | | | a | a | a | b | b | b | Loop test resistance flags 2 (note)<br>aaa – Resistance, tip-ring<br>bbb – Ringer load test |
| | 19 | | | | | | | | | Time to draw dial tone, in 0.1 second units. Valid only if byte 13 ddd = xx1. |

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| | 20 | | | | | | | | | Time to break dial tone, in 0.1 second units. Valid only if byte 13 bbb = xx1. |
| | 21 | | | | | | | | | Total dial tone power measurement, unsigned absolute value, 0.1 dB resolution, range 0 to [–]25.5 dBm0. Values above 0 dBm0 are reported as 0. Valid only if byte 14 bbb = xx1. |
| | 22 | | | | | | | | | Quiet channel power measurement, unsigned absolute value, 1 dB resolution, range 0 to [–]90 dBm0. Valid only if byte 14 aaa = xx1. |
| | 23-24 | | | | | | | | | Tip-ground DC voltage, 2 s complement, resolution 1V. Valid only if byte 15 aaa = xx1. |
| | 25-26 | | | | | | | | | Ring-ground DC voltage, 2 s complement, resolution 1V. Valid only if byte 15 bbb = xx1. |
| | 27 | | | | | | | | | Tip-ground AC voltage, Vrms. Valid only if byte 16 aaa = xx1. |
| | 28 | | | | | | | | | Ring-ground AC voltage, Vrms. Valid only if byte 16 bbb = xx1. |
| | 29-30 | | | | | | | | | Tip-ground DC resistance, kΩ. Infinite resistance: 0xffff. Valid only if byte 17 aaa = xx1. |
| | 31-32 | | | | | | | | | Ring-ground DC resistance, kΩ. Infinite resistance: 0xffff. Valid only if byte 17 bbb = xx1. |
| | 33-34 | | | | | | | | | Tip-ring DC resistance, kΩ. Infinite resistance: 0xffff. Valid only if byte 18 aaa = xx1. |
| | 35 | | | | | | | | | Ringer equivalence, in 0.1 REN units. Valid only if byte 18 bbb = xx1. |
| | 36-37 | | | | | | | | | Pointer to a general purpose buffer ME. Valid only for vendor-specific tests that require a GP buffer. |

NOTE – Coding for 3 bit flag sets is as follows:

000 test not run;

010 fail, measurement not reported;

011 fail, measurement reported;

110 pass, measurement not reported;

111 pass, measurement reported.

**Format for test action invoked against IP host config data entity class**

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |
| Message type | 3 | 0 | 0 | 0 | | | | | | DB = 0, AR = 0, AK =0<br>bits 5-1: action = test result |
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class<br>NOTE – This format applies to entity class IP host config data. |
| | 7 | | | | | | | | | MS byte entity instance |
| | 8 | | | | | | | | | LS byte entity instance |
| Message contents length | 9-10 | | | | | | | | | Size of message contents field, bytes |
| Message contents | 11 | 0 | 0 | 0 | 0 | 0 | x | x | x | xxx: Test result<br>000 timed out, no response<br>001 ICMP echo responses attached<br>010 ICMP time exceeded responses attached<br>011 Unexpected ICMP response<br>100..111 Reserved |
| | 12..n | | | | | | | | | See following descriptions for the content of these bytes |

If xxx = 001 (echo response – ping), the remainder of the message contains the following content. How many echo requests are sent and the resolution of the delay measurement are specific to a vendor's implementation. The special value 0xFFFF indicates a lost response.

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 12-13 | | | | | | | | | 16-bit measurement of response delay 1, expressed in ms. |
| | 14-15 | | | | | | | | | 16-bit measurement of response delay 2, expressed in ms. |
| | … | | | | | | | | | Etc. |

If xxx = 010 (time exceeded – traceroute), the remainder of the message contains the following content. In PON applications, it is not expected that a route trace will exceed the available space in the message, but if it does, the more distant responses should be dropped.

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 12-15 | | | | | | | | | IP address of nearest neighbour |
| | 18-21 | | | | | | | | | IP address of second nearest neighbour |
| | … | | | | | | | | | Etc. |

If xxx = 011 (unexpected ICMP response), the remainder of the message contains the following content:

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 12 | | | | | | | | | Type |
| | 13 | | | | | | | | | Code |
| | 14-15 | | | | | | | | | Checksum |
| | 18-21 | | | | | | | | | Bytes 5-8 of ICMP message (meaning depends on type/code) |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 22-n | | | | | | | | | Internet header + original datagram (truncated if necessary by extended OMCI message size limit) |

**Format for optical line supervision test action invoked against ANI-G, RE ANI-G, PPTP RE UNI, RE upstream amplifier or RE downstream amplifier entity class**

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |
| Message type | 3 | 0 | 0 | 0 | | | | | | DB = 0, AR = 0, AK =0<br>bits 5-1: action = test result |
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class<br>NOTE – This message format pertains to ANI-G, RE ANI-G, PPTP RE UNI, RE upstream amplifier or RE downstream amplifier entity class |
| | 7 | | | | | | | | | MS byte entity instance |
| | 8 | | | | | | | | | LS byte entity instance |
| Message contents length | 9-10 | | | | | | | | | Size of message contents field, bytes |
| Message contents | 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | Type = 1, Power feed voltage |
| | 12-13 | | | | | | | | | V, 2s complement, 20 mV resolution |
| | 14 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | Type = 3, Received optical power |
| | 15-16 | | | | | | | | | dBµW, 2s complement, 0.002 dB resolution |
| | 17 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | Type = 5, Transmitted optical power |
| | 18-19 | | | | | | | | | dBµW, 2s complement, 0.002 dB resolution |
| | 20 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | Type = 9, Laser bias current |
| | 21-22 | | | | | | | | | Unsigned integer, 2 µA resolution |
| | 23 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | Type 12, Temperature, degrees |
| | 24-25 | | | | | | | | | 2s complement, 1/256 degree C resolution |

NOTE – Unsupported tests are indicated with test type indicator 0 and 2 bytes of 0 data.

**Format for test action invoked against dot1ag MEP entity class, loopback test**

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |
| Message type | 3 | 0 | 0 | 0 | | | | | | DB = 0, AR = 0, AK =0<br>bits 5-1: action = test result |

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class<br>NOTE – This message format pertains to ONT-G and circuit pack entity classes. |
| | 7 | | | | | | | | | MS byte entity instance |
| | 8 | | | | | | | | | LS byte entity instance |
| Message contents length | 9-10 | | | | | | | | | Size of message contents field = 2 |
| Message contents | 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | x | X = 1: indicates failure to receive any loopback replies (LBRs) within 5 seconds |
| | 12-13 | | | | | | | | | Valid LBRs count: the number of valid, in-order LBRs received. |
| | 14-15 | | | | | | | | | Out of order LBRs count: the number of valid LBRs received that were out of order. |
| | 16-17 | | | | | | | | | Mismatch LBRs count: the number of received LBRs whose MAC SDU did not match that of the corresponding LBM (except for opcode). Optional feature, set to 0xFF if not supported. |
| | 18-19 | | | | | | | | | Delay from LB message transmission to LB response reception, measured in microseconds. The value 0 indicates no information available. |

**Format for test action invoked against dot1ag MEP entity class, linktrace test**

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |
| Message type | 3 | 0 | 0 | 0 | | | | | | DB = 0, AR = 0, AK =0<br>bits 5-1: action = test result |
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class<br>NOTE – This message format pertains to ONT-G and circuit pack entity classes. |
| | 7 | | | | | | | | | MS byte entity instance |
| | 8 | | | | | | | | | LS byte entity instance |
| Message contents length | 9-10 | | | | | | | | | Size of message contents field |

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Message contents | 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | x | X = 1: indicates failure to receive any linktrace replies (LTRs) within 5 seconds |
| | 12-15 | | | | | | | | | Transaction ID of transmitted LTM |
| | 16-23 | | | | | | | | | Content of egress TLV data field in transmitted LTM (IEEE 802.1ag clause 21.8.8). The LTRs themselves are captured in the general purpose buffer designated by the test command. |

### II.3.46 Get current data

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |
| Message type | 3 | 0 | 1 | 0 | | | | | | DB = 0, AR = 1, AK = 0<br>bits 5-1: action = get current data |
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class |
| | 7 | | | | | | | | | MS byte entity instance |
| | 8 | | | | | | | | | LS byte entity instance |
| Message contents length | 9-10 | | | | | | | | | Size of message contents field = 2 |
| Message contents | 11 | | | | | | | | | Attribute mask (attributes 1-8) |
| | 12 | | | | | | | | | Attribute mask (attributes 9-16) |

### II.3.47 Get current data response

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |
| Message type | 3 | 0 | 0 | 1 | | | | | | DB = 0, AR = 0, AK = 1<br>bits 5-1: action = get current data |
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class |
| | 7 | | | | | | | | | MS byte entity instance |
| | 8 | | | | | | | | | LS byte entity instance |
| Message contents length | 9-10 | | | | | | | | | Size of message contents field, bytes |

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Message contents | 11 | 0 | 0 | 0 | 0 | x | x | x | x | Result, reason<br>0000 command processed successfully<br>0001 command processing error<br>0010 command not supported<br>0011 parameter error<br>0100 unknown managed entity<br>0101 unknown managed entity instance<br>0110 device busy<br>1001 attribute(s) failed or unknown |
| | 12 | | | | | | | | | Attribute mask (attributes 1-8) |
| | 13 | | | | | | | | | Attribute mask (attributes 9-16) |
| | 14 | | | | | | | | | Optional-attribute mask (attributes 1-8), used with 1001 encoding:<br>0 default<br>1 unsupported attribute |
| | 15 | | | | | | | | | Optional-attribute mask (attributes 9-16), used with 1001 encoding:<br>0 default<br>1 unsupported attribute |
| | 16 | | | | | | | | | Attribute execution mask (attributes 1-8), used with 1001 encoding:<br>0 default<br>1 failed attribute |
| | 17 | | | | | | | | | Attribute execution mask (attributes 9-16), used with 1001 encoding:<br>0 default<br>1 failed attribute |
| | 18-n | | | | | | | | | Value of first attribute included (size depending on the type of attribute) |
| | | | | | | | | | | … |
| | | | | | | | | | | Value of last attribute included |

Bytes 14..17 are reserved for the optional-attribute and attribute execution masks; however, the content of these bytes is only valid in conjunction with the 1001 encoding used to indicate failed or unknown attributes. If the result code is not 1001, these bytes are still present, but should be set to 0 by the ONT and ignored by the OLT.

### II.3.48 Set table

The set table command provides a way in which a number of rows may be written into a table with a single command. The same function can be achieved with individual set commands, with each command instance directed to a single row of the table.

Writeable tables in OMCI have various mechanisms to control whether a given set operation causes a new row to be added to the table, an existing row to be overwritten or deleted, or the entire table cleared. All such mechanisms are embedded within the definition of the table row itself. Conflicting

control semantics are therefore possible. The set table command executes each table row sequentially, in list order.

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |
| Message type | 3 | 0 | 1 | 0 | | | | | | DB = 0, AR = 1, AK = 0<br>bits 5-1: action = set table |
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class |
| | 7 | | | | | | | | | MS byte entity instance |
| | 8 | | | | | | | | | LS byte entity instance |
| Message contents length | 9-10 | | | | | | | | | Size of message contents field, bytes |
| Message contents | 11 | | | | | | | | | Attribute mask (note) (attributes 1-8) |
| | 12 | | | | | | | | | Attribute mask (attributes 9-16) |
| | 13-n | | | | | | | | | Value of first table row (size depending on table definition) |
| | | | | | | | | | | … |
| | | | | | | | | | | Value of last table row |

NOTE – exactly one bit of the attribute mask must be set, and that bit must correspond to a read-write table attribute in the definition of the parent managed entity.

### II.3.10 Set table response

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Transaction correlation identifier | 1-2 | | | | | | | | | |
| Message type | 3 | 0 | 0 | 1 | | | | | | DB = 0, AR = 0, AK = 1<br>bits 5-1: action = set table |
| Device identifier | 4 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | Extended OMCI = 0x0B |
| Managed entity identifier | 5-6 | | | | | | | | | Entity class |
| | 7 | | | | | | | | | MS byte entity instance |
| | 8 | | | | | | | | | LS byte entity instance |
| Message contents length | 9-10 | | | | | | | | | Size of message contents field = 1 |

| Field | Byte | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Comments |
|---|---|---|---|---|---|---|---|---|---|---|
| Message contents | 11 | 0 | 0 | 0 | 0 | x | x | x | x | Result, reason<br>0000 command processed successfully<br>0001 command processing error<br>0010 command not supported<br>0011 parameter error<br>0100 unknown managed entity<br>0101 unknown managed entity instance<br>0110 device busy<br>1001 attribute(s) failed or unknown |

## APPENDIX III Traffic management options

### III.2 Explicit traffic scheduler configuration

*Revise this clause to read as follows:*

In slightly more complex implementations, ONTs may implement some level of traffic scheduling within each T-CONT. These are described using priority queues and one or more levels of traffic scheduler MEs. The arrangement of priority queues and traffic schedulers is determined by the ONT, and is by default not controllable by the OLT. An example of the configuration of the traffic scheduler appears in figure III.2/G.984.4. This model consists of three stages, such as two delay control and one guaranteed rate control stages. A delay control stage can be worked by HOL (head of line) scheduling. A guaranteed rate control stage can be worked by WRR. This configuration may also be used when the traffic management option attribute in the ONT-G ME is 0 (priority controlled).

*Revise the following two clauses to read as shown (change* GEM traffic descriptor *to* traffic descriptor*).*

### III.3 Traffic descriptor configuration

An alternative method of controlling traffic in ONTs is to provide traffic descriptors to the ONT, and leave the details of honouring and enforcing these contracts to the ONT implementation. This is controlled using traffic descriptor MEs. This method makes the theoretical assumption that a work-conserving scheduling methodology will be used. In this configuration, traffic is shaped to conform to PIR and PBS in the traffic descriptor ME. This configuration is used when the traffic management option attribute in the ONT-G ME is 1 (rate controlled).

### III.4 Priority and rate controlled configuration

Another method of controlling traffic in ONTs is to provide not only priority control with traffic scheduling, but also traffic descriptors. This is controlled using traffic descriptor, priority queue-G and traffic scheduler-G MEs. This method makes the theoretical assumption that a work-conserving scheduling methodology will be used. In this configuration, traffic is policed to conform to PIR and PBS, and may be marked green or yellow according to CIR / CBS / PIR / PBS in the traffic descriptor ME. This configuration is used when the traffic management option attribute in the ONT-G ME is 2 (priority and rate controlled).

*Add the following new clause:*

### III.5 Flexible assignment

By default, priority queues and traffic schedulers are assigned to T-CONTs by the ONT architecture in a fixed configuration, which may not be altered. It is also possible that the ONT implements its QoS components in such a way that they may be flexibly reassigned (note). ONT flexibility is signalled to the OLT by means of the QoS configuration flexibility bit map attribute of the ONT2-G managed entity.

NOTE – Given the slot-port model of ONT equipment, which appears among other places in the managed entity identifiers of T-CONT, physical path termination points, the traffic scheduler, and the related port attribute of the priority queue managed entity, it is not anticipated that implementation flexibility would extend across slots. Accordingly, OMCI restricts flexibility to be only within a slot, but does not permit flexible assignment across slots.

## Bibliography

*Add the reference:*

[b-ISO/IEC 9798-4] ISO/IEC 9798-4:1999 Information technology – Security techniques – Entity authentication – Part 4: Mechanisms using a cryptographic check function