IEEE P1904.2™/D0.4 Draft Standard for Universal Management Tunnel for Ethernet-based Subscriber Access Networks

Sponsor

Standards Development Board
of the
IEEE Communications Society

Approved <XX MONTH 20XX>

IEEE-SA Standards Board

Abstract: This standard TBD

Keywords: TBD

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

PDF: ISBN 978-0-XXXX-XXXX-X STDXXXXX
Print: ISBN 978-0-XXXX-XXXX-X STDPDXXXXX

Introduction

This introduction is not part of IEEE P1904.2/D0.4

This standard TBD …

Notice to users

Laws and regulations

Users of IEEE Standards documents should consult all applicable laws and regulations. Compliance with the provisions of any IEEE Standards document does not imply compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Copyrights

This document is copyrighted by the IEEE. It is made available for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making this document available for use and adoption by public authorities and private users, the IEEE does not waive any rights in copyright to this document.

Updating of IEEE documents

Users of IEEE Standards documents should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect. In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit the IEEE-SA Website at http://standards.ieee.org/index.html or contact the IEEE at the address listed previously. For more information about the IEEE Standards Association or the IEEE standards development process, visit the IEEE-SA Website at http://standards.ieee.org/index.html.

Errata

Errata, if any, for this and all other standards can be accessed at the following URL: http://standards.ieee.org/findstds/errata/index.html. Users are encouraged to check this URL for errata periodically.

Interpretations

Current interpretations can be accessed at the following URL:
http://standards.ieee.org/findstds/interps/index.html.

Patents

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken by the IEEE with respect to the existence or validity of any patent rights in connection therewith. If a patent holder or patent applicant has filed a statement of assurance via an Accepted Letter of Assurance, then the statement is listed on the IEEE-SA website http://standards.ieee.org/about/sasb/patcom/patents.html. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

Essential Patent Claims may exist for which a Letter of Assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims, or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or nondiscriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

Participants

At the time this draft standard was submitted to the IEEE-SA Standards Board for approval, the following is a place holder:

, Working Group Chair
, Editor

The following individuals submitted technical contributions or commented on the draft standard at various stages of the project development.

Name

The following members of the <individual/entity> balloting committee voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

(to be supplied by IEEE)

Balloter1	Balloter4	Balloter7
Balloter2	Balloter5	Balloter8
Balloter3	Balloter6	Balloter9

When the IEEE-SA Standards Board approved this standard on <XX MONTH 20XX>, it had the following membership:

(to be supplied by IEEE)

<Name>, *Chair*
<Name>, *Vice Chair*
<Name>, *Past President*
<Name>, *Secretary*

SBMember1
SBMember2
SBMember3
SBMember4
SBMember5
SBMember6
SBMember7
SBMember8
SBMember9

*Member Emeritus

Also included are the following nonvoting IEEE-SA Standards Board liaisons:

<Name>, *NRC Representative*
<Name>, *DOE Representative*
<Name>, *NIST Representative*

<Name>
IEEE Standards Program Manager, Document Development

<Name>
IEEE Standards Program Manager, Technical Program Development

Contents

1 Overview

1.1 Scope

This standard describes a Universal Management Tunnel (UMT) for devices used in Ethernet-based subscriber access networks. The key characteristics of the specified management mechanism are:

- The ability to transit Layer 2 bridges in a single IEEE 802 Media Access Control (MAC) domain to allow remote device management;
- Extensibility to accommodate new management protocols and new types of devices;
- The ability to simultaneously send messages to multiple UMT stations using broadcast or multicast addressing.

The standard describes the message format as well as processing operations at the stations participating in the UMT protocol.

1.2 Coverage

In their quest to find the optimal balance between the performance of subscriber access networks and their cost, the network operators increasingly combine optical distribution section with a copper-based drop section, which typically includes a twisted pair, a Category-5 cable, or a coaxial cable. Network operators require a management system that would allow them to efficiently access and manage the subscriber demarcation device as well as the various devices that interconnect their optical and copper sections of the network.
In addition, to achieve the best-possible service quality, the access network operators find it necessary to extend their management domains past the typical subscriber demarcation device, such as an Optical Network Unit (ONU), a Coaxial Network Unit (CNU), Cable or DSL modem, or a Residential Gateway (RGW).
As Ethernet-based networks (switched Ethernet, point-to-point Ethernet, or Ethernet Passive Optical Network) are becoming technologies of choice for public subscriber access network, there is a pressing need to provide a universal management channel compatible with Ethernet and that would allow network operators to manage a variety of devices in access network or in subscriber premises in a uniform and consistent way.

1.3 Overview of clauses

This subclause provides an overview of the scope of individual clauses included in this specification, namely:

- Clause 1 provides an overview of the IEEE 1904.2 specifications, including the scope and purpose of the specification and the scope of individual clauses.
- Clause 2 lists normative references used within this specification.
- Clause 3 presents definitions of specific terms as used in this standard. Terms may be introduced in this specification or may exist with multiple industry definitions. Additionally, a list of acronyms used in this standard is included.
- Clause 4 defines individual … <TBD>

2 Normative references

The following referenced documents are indispensable for the application of this document (i.e., they must be understood and used, so each referenced document is cited in text and its relationship to this document is explained). For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

IEEE Std 802.1Q™-2018, IEEE Standard for Information technology—Telecommunications and information systems—Local and metropolitan area networks—Bridges and Bridged Networks.

IEEE Std 802.3™-2018, IEEE Standard for Ethernet.

3 Definitions, acronyms, and abbreviations

3.1 Definitions

For the purposes of this document, the following terms and definitions apply. The IEEE Standards Dictionary Online should be consulted for terms not defined in this clause.[1]

Network management system (NMS): In the scope of IEEE Std 1904.2, any network management, control, information storage, and other type of entities, located in the same or different geographical locations, functionally combined to a single point of reference. This entity is responsible for controlling, managing, and supervising the operation of a UMT-aware L2 network. NMS combines, terminates, proxies, or snoops a number of different control and management protocols (outside the scope of this standard), used to drive the operation of the Optical Line Terminal (OLT) and its functions, providing Faults, Accounting, Configuration, Performance, and Security (FCAPS) functionality for a network operator.

3.2 Acronyms and abbreviations

UMT	Universal Management Tunnel
PDU	Protocol Data Unit
CTE	Classification and Translation Engine
OAM	Operations, Administration, and Management
OMCI	
MAC	Media Access Control
OLT	Optical Line Terminal
ONU	Optical Network Unit
NMS	Network Management System
FCAPS	Faults, Accounting, Configuration, Performance, and Security

3.3 Special Terms

Term: Definition

3.4 Notation for state diagrams

All the state diagrams used in this standard meet the set of requirements included in the following subclauses.

[1] IEEE Standards Dictionary Online subscription is available at http://www.ieee.org/portal/innovate/products/standard/standards_dictionary.html.

3.4.1 General conventions

The operation of any protocol defined in this standard can be described by subdividing the protocol into a number of interrelated functions. The operation of the functions can be described by state diagrams. Each diagram represents the domain of a function and consists of a group of connected, mutually exclusive states. Only one state of a function is active at any given time (see Figure 3-1).

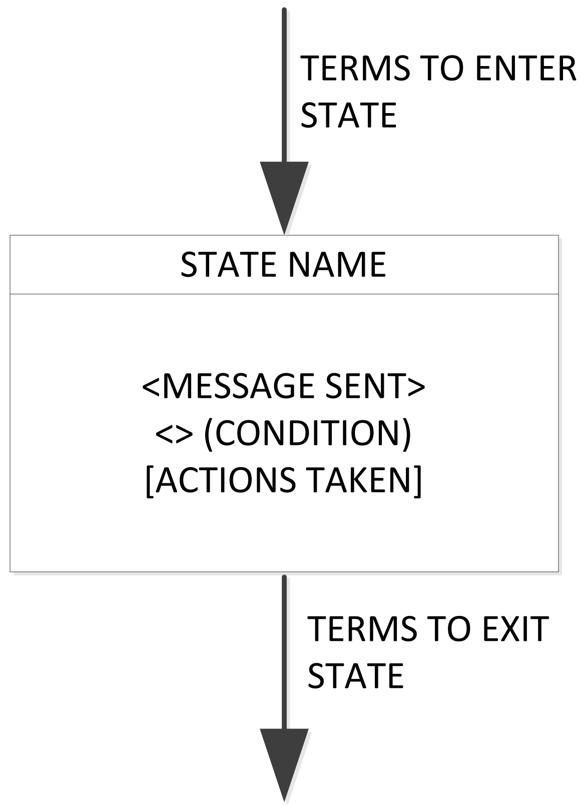


Figure 3-1—State diagram notation example

3.4.1.1 Representation of states

Each state that the function can assume is represented by a rectangle. These are divided into two parts by a horizontal line. In the upper part the state is identified by a name in capital letters. The lower part contains the body of the given state, containing description of the actions taken in this state, as defined in 3.4.3.

3.4.1.2 Transitions

All permissible transitions between the states of a function are represented graphically by arrows between them. A transition that is global in nature (for example, an exit condition from all states to the IDLE or RESET state) is indicated by an open arrow (an arrow with no source block). Global transitions are evaluated continuously whenever any state is evaluating its exit conditions. When the condition for a global transition becomes true, it supersedes all other transitions, including Unconditional Transition (UCT), returning control to the block pointed to by the open arrow.

Labels on transitions are qualifiers that are required to be fulfilled before the transition is taken. The label UCT designates an unconditional transition. Qualifiers described by short phrases are enclosed in parentheses.

The following terms are valid transition qualifiers:

- Boolean expressions
- An event such as the expiration of a timer: timer_done
- An event such as the reception of a message: MAC_DATA.indication
- An unconditional transition: UCT
- A branch taken when other exit conditions are not satisfied: ELSE

State transitions occur instantaneously. No transition in the state diagram can cross another transition. When possible, any two transitions with different logical conditions are not joined together into a single transition line.

3.4.2 State diagrams and accompanying text

State diagrams take precedence over text.

3.4.3 Actions inside state blocks

The actions inside a state block execute instantaneously. Actions inside state blocks are atomic (i.e., uninterruptible).

After performing all the actions listed in a state block one time, the state diagram then continuously evaluates exit conditions for the given state block until one is satisfied, at which point control passes through a transition arrow to the next block. While the state awaits fulfillment of one of its exit conditions, the actions inside do not implicitly repeat.

Valid state actions may include generation of *indication* and *request* primitives.

No actions are taken outside of any blocks of the state diagram.

3.4.4 State diagram variables

Once set, variables retain their values as long as succeeding blocks contain no references to them.

Setting the parameter of a formal interface message assures that, on the next transmission of that message, the last parameter value set is transmitted.

Testing the parameter of a formal interface message tests the value of that message parameter that was received on the last transmission of said message. Message parameters may be assigned default values that persist until the first reception of the relevant message.

3.4.5 Operators

The state diagram operators are shown in Table 3-1.

Table 3-1—State diagram operators

Character	Meaning
AND	Boolean AND
OR	Boolean OR
XOR	Boolean XOR
!	Boolean NOT

Character	Meaning
<	Less than
>	More than
≤	Less than or equal to
≥	More than or equal to
==	Equals (a test of equality)
!=	Not equals
()	Indicates precedence
=	Assignment operator
|	Concatenation operation that combines several sub-fields or parameters into a single aggregated field or parameter
else	No other state condition is satisfied
true	Designation of a Boolean value of TRUE
false	Designation of a Boolean value of FALSE

3.4.6 Timers

Some of the state diagrams use timers for various purposes, e.g., measurement of time, and confirmation of activity. All timers operate in the same fashion.

A timer is reset and starts counting upon entering a state where [start x_timer, x_timer_value] is asserted. Time "x" after the timer has been started, "x_timer_done" is asserted and remains asserted until the timer is reset. At all other times, "x_timer_not_done" is asserted.

When entering a state where [start x_timer, x_timer_value] is asserted, the timer is reset and restarted even if the entered state is the same as the exited state.

Any timer can be stopped at any time upon entering a state where [stop x_timer] is asserted, which aborts the operation of the "x_timer" asserting "x_timer_not_done" indication until the timer is restarted again.

3.4.7 Hexadecimal notation

Numerical values designated by the 0x prefix indicate a hexadecimal notation of the corresponding number, with the least significant bit shown on the right. For example: 0x0F represents an 8-bit hexadecimal value of the decimal number 15; 0x00-00-00-00 represents a 32-bit hexadecimal value of the decimal number 0; 0x11-AB-11-AB represents a 32-bit hexadecimal value of the decimal number 296423851.

3.4.8 Binary notation

Numerical values designated by the 0b prefix indicate a binary notation of the corresponding number, with the least significant bit shown on the right. For example: 0b0001000 represents an 8-bit binary value of the decimal number 8.

3.5 Notation for PICS

The supplier of a device implementation that is claimed to conform to this standard is required to complete a protocol implementation conformance statement (PICS) proforma.

A completed PICS proforma is the PICS for the implementation in question. The PICS is a statement of which capabilities and options of this standard have been implemented. The PICS can be used for a variety of purposes by various parties, including the following:

a) As a checklist by the protocol implementer, to reduce the risk of failure to conform to the standard through oversight;

b) As a detailed indication of the capabilities of the implementation, stated relative to the common basis for understanding provided by the standard PICS proforma, by the supplier and acquirer, or potential acquirer, of the implementation;

c) As a basis for initially checking the possibility of interworking with another implementation by the user, or potential user, of the implementation (note that, while interworking can never be guaranteed, failure to interwork can often be predicted from incompatible PICS);

d) As the basis for selecting appropriate tests against which to assess the claim for conformance of the implementation, by a protocol tester.

Each PICS entry is uniquely identified by an item number, with the following form: [Package][Device]-[Feature][Number], where:

— [Package] is the designation of the given Package,

— [Device] identifies whether the given PICS item describes the ONU (U) or OLT (T) requirements,

— [Feature] is the identification of individual features, and finally,

— [Number] is a number allocated to each subsequent PICS entry. This item may have one of two possible formats: a decimal number or a decimal number followed by a lower-case letter. The first format is used to designate PICS with functionally distinct requirements. The latter format is used to designate PICS with functionally similar requirements.

For example, CU-LPTK3a represents a PICS entry for an ONU compliant with Package C for the “optical link protection, trunk type” feature, item 3, subitem a.

3.5.1 Abbreviations and special symbols

The following symbols are used in the PICS proforma:

M	mandatory field/function
!	negation
O	optional field/function
O.<n>	optional field/function, but at least one of the group of options labeled by the same numeral <n> is required
O/<n>	optional field/function, but one and only one of the group of options labeled by the same numeral <n> is required
X	prohibited field/function
<item>:	simple-predicate condition, dependent on the support marked for <item>
<item1>*<item2>:	AND-predicate condition, the requirement needs to be met if both optional items are implemented

3.5.2 Instructions for completing the PICS proforma

The first part of the PICS proforma, Implementation Identification and Protocol Summary, is to be completed as indicated with the information necessary to identify fully both the supplier and the implementation.

The main part of the PICS proforma is a fixed-format questionnaire divided into subclauses, each containing a group of items. Answers to the questionnaire items are to be provided in the right-most column, either by simply marking an answer to indicate a restricted choice (usually Yes, No, or Not Applicable), or by entering a value or a set or range of values. (Note that there are some items where two or more choices from a set of possible answers can apply; all relevant choices are to be marked.)

Each item is identified by an item reference in the first column; the second column contains the question to be answered; the third column contains the reference or references to the material that specifies the item in the main body of the standard; the fourth column contains values and/or comments pertaining to the question to be answered. The remaining columns record the status of the items—whether the support is mandatory, optional or conditional—and provide the space for the answers.

The supplier may also provide, or be required to provide, further information, categorized as either Additional Information or Exception Information. When present, each kind of further information is to be provided in a further subclause of items labeled A<i> or X<i>, respectively, for cross-referencing purposes, where <i> is any unambiguous identification for the item (e.g., simply a numeral); there are no other restrictions on its format or presentation.

A completed PICS proforma, including any Additional Information and Exception Information, is the protocol implementation conformance statement for the implementation in question.

Note that where an implementation is capable of being configured in more than one way, according to the items listed under Major Capabilities/Options, single PICS may be able to describe all such configurations. However, the supplier has the choice of providing more than one PICS, each covering some subset of the implementation's configuration capabilities, if that would make presentation of the information easier and clearer.

3.5.3 Additional information

Items of Additional Information allow a supplier to provide further information intended to assist the interpretation of the PICS. It is not intended or expected that a large quantity be supplied, and the PICS can be considered complete without any such information. Examples might be an outline of the ways in which a (single) implementation can be set up to operate in a variety of environments and configurations; or a brief rationale, based perhaps upon specific application needs, for the exclusion of features that, although optional, are nonetheless commonly present in implementations.

References to items of Additional Information may be entered next to any answer in the questionnaire, and may be included in items of Exception Information.

3.5.4 Exception information

It may occasionally happen that a supplier wishes to answer an item with mandatory or prohibited status (after any conditions have been applied) in a way that conflicts with the indicated requirement. No pre-printed answer is found in the Support column for this; instead, the supplier is required to write into the Support column an X<i> reference to an item of Exception Information, and to provide the appropriate rationale in the Exception item itself.

An implementation for which an Exception item is required in this way does not conform to this standard. Note that a possible reason for the situation described above is that a defect in the standard has been reported, a correction for which is expected to change the requirement not met by the implementation.

3.5.5 Conditional items

The PICS proforma may contain conditional items. These are items for which both the applicability of the item itself, and its status if it does apply—mandatory, optional, or prohibited—are dependent upon whether or not certain other items are supported.

Individual conditional items are indicated by a conditional symbol of the form “<item>:<s>” in the Status column, where “<item>” is an item reference that appears in the first column of the table for some other item, and “<s>” is a status symbol, M (Mandatory), O (Optional), or X (Not Applicable).

If the item referred to by the conditional symbol is marked as supported, then:

a) the conditional item is applicable,

b) its status is given by “<s>”, and

c) the support column is to be completed in the usual way.

Each item whose reference is used in a conditional symbol is indicated by an asterisk in the Item column.

4 Universal Management Tunnel (UMT) Overview and Architecture

4.1 Principles of operation

Universal Management Tunnel (UMT) defines the method of encapsulating various protocol data units (xPDUs) in Ethernet frames with UMT Ethertype (0xA8-C8). An Ethernet frame with UMT Ethertype is called an UMTPDU. That portion of the network path that xPDUs traverse while they are encapsulated as UMTPDUs is referred to as a *tunnel*.

The xPDU-to-UMTPDU and UMTPDU-to-xPDU conversions take place within the UMT Sublayer (see 4.2). The UMT sublayer is optional, i.e., in any multi-port device, the UMT sublayer may be implemented in only some ports and not the other. Devices that implement the UMT Sublayer in at least one of the ports are said to be UMT-aware.

Devices that don't implement UMT sublayer in any of the ports are called UMT-unaware. UMT-unaware devices are able to relay UMTPDUs as generic Ethernet frames using existing L2 forwarding mechanisms, but are unable to consume or generate UMTPDUs.

The UMT Sublayer includes the Classification and Translation Engine (CTE) that converts xPDUs into UMTPDUs and vice versa. The CTE behavior is governed by a set of rules that are either statically configured or dynamically provisioned by the NMS (see 6.1).

The UMT Sublayer provides a service interface to OAM sublayer, UMT Client, and may provide service interface to other L2 protocol-specific clients. The only messages that are passed to and received from the UMT Client are the UMT configuration messages (see *UMT_CONFIG* UMTPDU in 7.1).

All UMTPDUs except the *UMT_CONFIG* UPTPDUs carry tunneling payloads associated with specific protocols (xPDU). Any payload-carrying UMTPDU that is consumed by a device is first converted into its native xPDU format and then passed to a specific client associated with that xPDU protocol type. Correspondingly, any payload-carrying UMTPDU that is generated by a device originates in a protocol-specific client as xPDU and is then converted into UMTPDU within the UMT sublayer.

A device port where xPDUs are converted into UMTPDUs (within the UMT sublayer) is referred to as *tunnel entrance point* and a port where the opposite conversion takes place is referred to as *tunnel exit point*.

4.1.1 UMT discovery protocol

The tunnel entrance and exit points may be pre-configured or provisioned via *UMT_CONFIG* UMTPDUs based on known network topology and L2 device addresses. An automatic UMT discovery protocol is out-of-scope for this revision of the standard.

4.2 UMT sublayer

UMT functionality is confined to the UMT sublayer. Figure 4-1 depicts architectural positioning of the UMT sublayer, which is a client of the MAC Control sublayer (see IEEE Std 802.3, Clause 31).

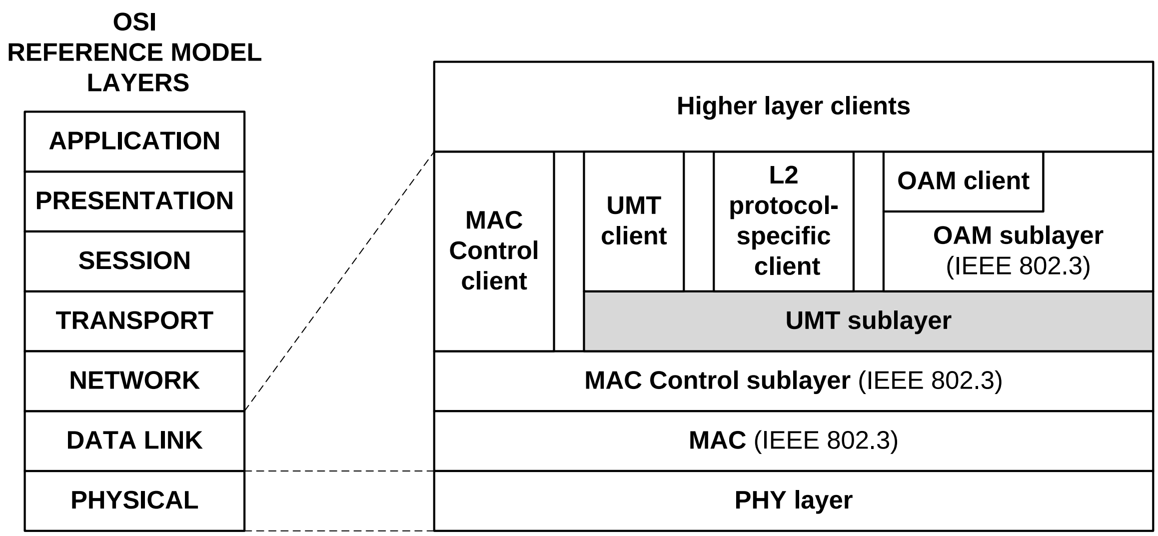


Figure 4-1— UMT sublayer relationship to the ISO/IEC Open Systems Interconnection (OSI) reference model and the IEEE Std 802.3 Ethernet model

4.3 UMT service interfaces

The UMT sublayer is a client of MAC Control sublayer and implements a standard IEEE Std 802.3 MAC service interface (see IEEE Std 802.3, Clause 2).

The UMT Sublayer provides UMT service interface (UMTSI) to OAM sublayer, UMT Client, and to other L2 protocol-specific clients (see Figure 4-2). To the OAM sublayer, the UMT sublayer presents a standard IEEE Std 802.3 MAC service interface (*UMTSI:MA_DATA*). To the UMT Client, the UMT sublayer presents UMT-specific service interface (*UMTSI:UMTPDU*). To the L2 protocol-specific clients, the UMT sublayer presents a protocol-specific service interface. The only protocol-specific client defined in this standard is the OMCI Client (see 5.2.3).

Inter-layer interfaces are depicted in Figure 4-2.

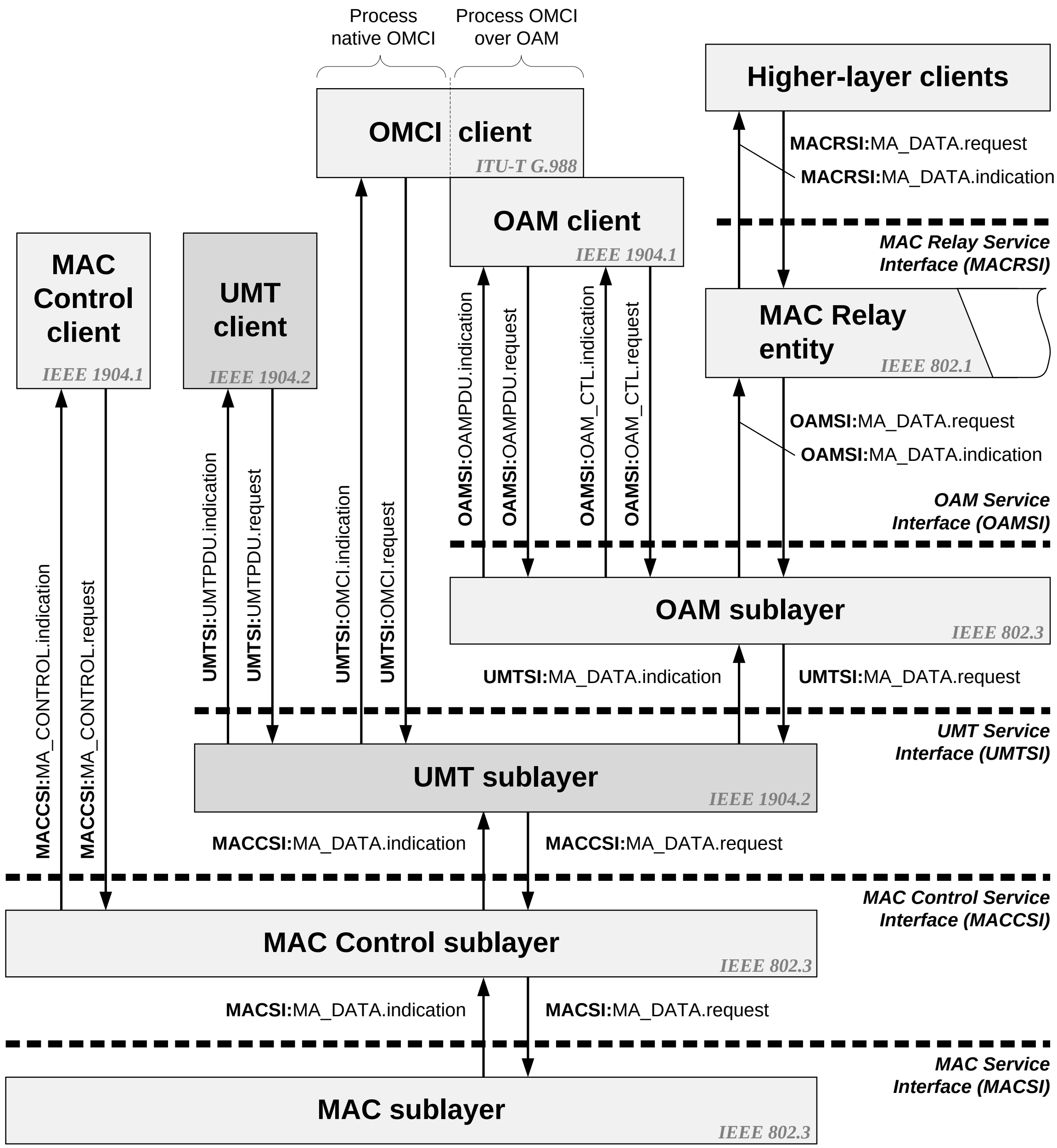
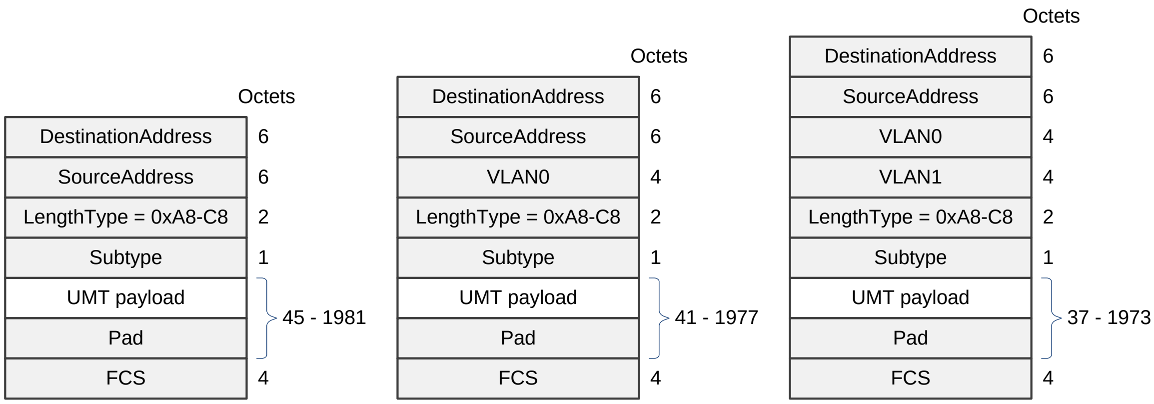


Figure 4-2—Positioning of UMT sublayer and service interfaces

5 Universal Management Tunnel Protocol Data Units (UMTPDU)

5.1 UMTPDU Structure

A Universal Management Tunnel Protocol Data Unit (UMTPDU) is an Ethernet MAC frame with the value of Ethertype field equal to the UMT Ethertype (0xA8-C8). The UMTPDU format is shown in IEEE Std 802.3, Clause 3.



a) Untagged UMTPDU **b) Single-tagged UMTPDU** **c) Double-tagged UMTPDU**

Figure 5-1—UMTPDU format

The UMTPDU structure is shown in Figure 5-1 and it includes the following fields:

—*DestinationAddress*:

In a UMTPDU, the *DestinationAddress* is the individual (unicast) MAC address associated with the device consuming xPDU carried within the UMTPDU. Note that the destination device may not be UMT-aware and the UMT tunnel may be terminated before the frame reaches that device.

—*SourceAddress*:

In UMTPDUs, the *SourceAddress* is the individual MAC address associated with the device that generated xPDU.

—*LengthType*:

The *LengthType* field in a UMTPDU carries the UMT Ethertype value 0xA8-C8.

—*Subtype*:

The *Subtype* field identifies the type of xPDU being encapsulated in the UMTPDU. *Subtype* field values are defined in Table 5-1.

—*UMT payload*:

The *UMT payload* field represents a set of fields associated with the *Subtype*-specific protocols, as defined in 5.2.

—*Pad*:

This field is present only when the total length of the *UMT payload* is below 45 octets. The *Pad* field is added to bring the UMTPDU length up to the minimum frame size (see IEEE Std 802.3, 4A.2.3.2.4). This field is filled with zeros on transmission, and is ignored on reception.

—*FCS*:

This field contains the Frame Check Sequence, typically generated by the MAC.

Fields within a frame are transmitted from top to bottom. When consecutive octets are used to represent a single numerical value, the most significant octet is transmitted first, followed by successively less significant octets. Bits within each octet are transmitted from LSB to MSB.

5.2 UMTPDU *Subtype* encoding

The value encoding of the *Subtype* field shall be as defined in Table 5-1.

Table 5-1—*Subtype* field encoding

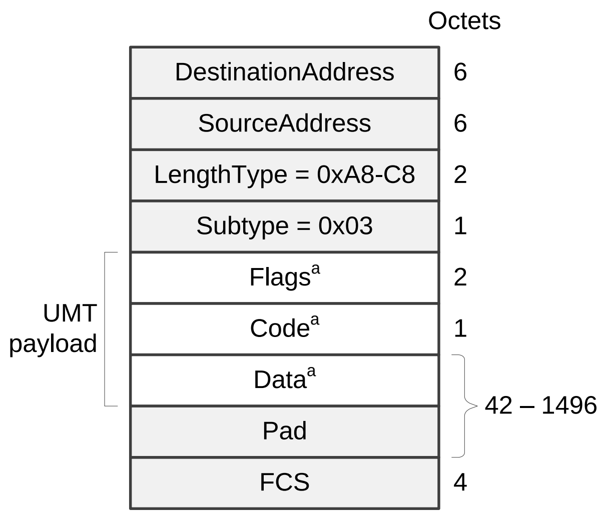
Value	Designation	Description
0x00	*UMT_config*	*UMT_config* subtype identifies *UMT_Request* and *UMT_Response* UMTPDUs used for configuring the UMT Classification and Translation Engine (see 6.1).
0x01, 0x02	n/a	Reserved for UMT Discovery protocol; ignored on reception.
0x03	*OAM_Subtype*	*OAM_Subtype* represents the OAMPDU payload carried within the UMTPDU (see 5.2.1).
0x04	*OMCI_Subtype*	*OMCI_Subtype* represents the OMCI payload carried within the UMTPDU (see 5.2.2).
0x05	*L2_subtype*	*L2_Subtype* represents a generic Ethernet frame carried within the UMTPDU (e.g., MAC-in-MAC) (see 5.2.3).
0x06	*L3_Subtype*	*L3_Subtype* represents a generic L3 packet (plus TPID) carried within the UMTPDU (see 5.2.4).
0x05 to 0xFD	n/a	Reserved; ignored on reception.
0xFE, 0xFF	*Org_Subtype*	*Org_Subtype* represents an organization-specific payload carried within the UMTPDU (see 5.2.5).

5.2.1 UMT configuration subtype

A UMTPDU with UMT configuration subtype (*Subtype* field = 0x00) identifies *UMT_CONFIG* UMTPDU used for configuring the UMT Classification and Translation Engine (see 6.1). This UMTPDU is defined in 7.1.

5.2.2 OAM subtype

A UMTPDU with OAM subtype (*Subtype* field = 0x03) is an instantiation of a generic UMTPDU, as defined in 5.1, that carries an Operations, Administration, and Maintenance (OAM) payload (see IEEE Std 802.3, 57.4). The frame structure of UMTPDU with OAM subtype shall be as depicted in Figure 5-2.



a – This field is defined in IEEE 802.3, 57.4

Figure 5-2—Format of UMTPDU with OAM subtype

The structure of the *UMT payload* in the UMTPDU with OAM subtype is defined as follows:

—*Flags*:
This field carries the value of the *Flags* field as defined in IEEE Std 802.3, 57.4.

—*Code*:
This field carries the value of the *Code* field as defined in IEEE Std 802.3, 57.4.

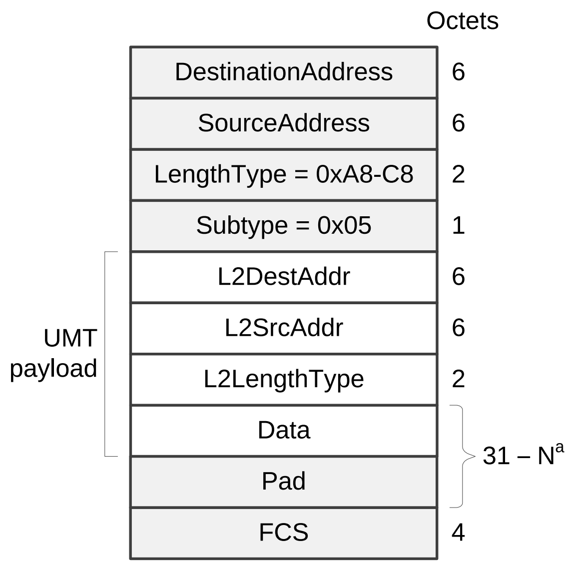
—*Data*:
This field carries the payload portion of the OAMPDU as defined IEEE Std 802.3, 57.4.

5.2.3 OMCI Subtype

<TBD>

5.2.4 L2 Subtype

A UMTPDU with L2 subtype (*Subtype* field = 0x05) is an instantiation of a generic UMTPDU, as defined in 5.1, that carries a complete L2 frame as its payload. The frame structure of UMTPDU with L2 subtype shall be as depicted in Figure 5-3.



a – Maximum field length depends on frame type (see Figure 5-1).

Figure 5-3—Format of UMTPDU with L2 subtype

The structure of the *UMT payload* in the UMTPDU with L2 subtype is defined as follows:

—*L2DestAddr*:
This field carries the L2 destination address of the original L2 frame being tunneled using UMT.

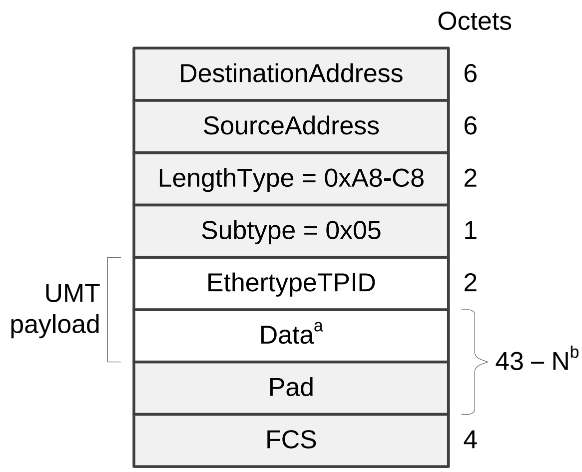
—*L2SrcAddr*:
This field carries the L2 source address of the original L2 frame being tunneled using UMT.

—*L2LenthType*:
This field carries the Length/Type value of the original L2 frame being tunneled using UMT.

—*Data*:
This field carries the L2 payload of the original L2 frame being tunneled using UMT. The combined size of the *Data* and *Pad* fields ranges between 31 and *N*, where *N* is defined in Figure 5-1.

5.2.5 L3 Subtype

A UMTPDU with L3 subtype (*Subtype* field = 0x06) is an instantiation of a generic UMTPDU, as defined in 5.1, that carries an L3 packet as its payload. The frame structure of UMTPDU with L3 subtype shall be as depicted in Figure 5-4. The format of the Data/Pad field is dependent on the value of the Ethertype/TPID field and is beyond the scope of this standard.



a – Field format depends on the value of *EthertypeTPID* field.
b – Maximum field length depends on frame type (see Figure 5-1).

Figure 5-4—Format of UMTPDU with L3 subtype

The structure of the *UMT payload* in the UMTPDU with L3 subtype is defined as follows:

—*EthertypeTPID*:

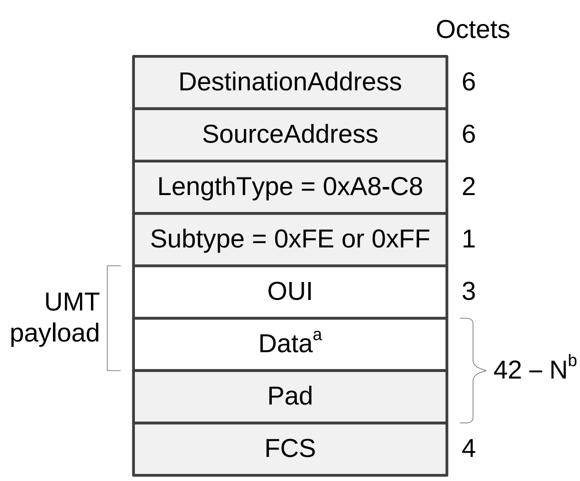
This field carries the L2 Ethertype/TPID value of the original L3 packet being tunneled using UMT.

—*Data*:

This field carries the L3 packet being tunneled using UMT. The combined size of the *Data* and *Pad* fields ranges between 43 and *N*, where *N* is defined in Figure 5-1.

5.2.6 Organization-specific extension subtype

The Organization-specific UMTPDU is an instantiation of a generic UMTPDU as defined in 5.1. It is identified with the *Subtype* field value of 0xFE or 0xFF and it is used for organization specific extensions. The Organization Specific UMTPDU frame structure shall be as depicted in Figure 5-5. The field *OUI* immediately following the *Subtype* field shall contain the Organizationally Unique Identifier (OUI) or Company ID (CID).



a – Field format depends on the value of *OUI* field.
b – Maximum field length depends on frame type (see Figure 5-1).

Figure 5-5—Format of UMTPDU with organization-specific extension subtype

The structure of the *UMT payload* in the UMTPDU with organization-specific extension subtype is defined as follows:

—*OUI*:

This field carries the Organizationally Unique Identifier (OUI) or Company ID (CID).

—*Data*:

This field carries the OUI/CID-specific data payload. The internal format of the *Data* field is dependent on *OUI* field value and is beyond the scope of this standard. The combined size of the *Data* and *Pad* fields ranges between 42 and *N*, where *N* is defined in Figure 5-1.

5.3 VLAN-Tagged UMTPDU

Editor's Note: Need to decide whether
(a) VLAN tags are allowed
(b) whether VLAN tag goes before or after UMT Ethertype.

1) How to convert VLANed xPDU into VLANed UMTPDU?
   (i) VLAN gets buried in payload
   (ii) VLAN is placed after SA

6 UMT sublayer

6.1 UMT Classification and Translation Engine

The function of the UMT Classification and Translation Engine (CTE) is to classify frames by certain criteria and to perform specific modification on the frames that match the criteria. The classification criteria together with the associated modification action comprise an entity called a *rule*. The concept of a rule is similar to that defined in IEEE 1904.1, 6.5.2.1.

By matching frames to specific rules, the CTE is able to translate UMTPDUs into xPDUs (i.e., into frames with different Ethertype values) and vice versa. A frame that does not match any CTE rules traverses the UMT sublayer without any modifications.

There are separate CTE instances in the transmit path and in the receive path of each physical or virtual port. The CTE located in the receive path is called *Ingress CTE* and the CTE located in the transmit path is called *Egress CTE* (see Figure 6-1). Fundamentally, a CTE instance is simply a table that stores multiple rules. Some of the rules are statically pre-configured (i.e., available and active at all times); other rules are dynamically added/deleted by NMS when tunnels are established or destroyed.

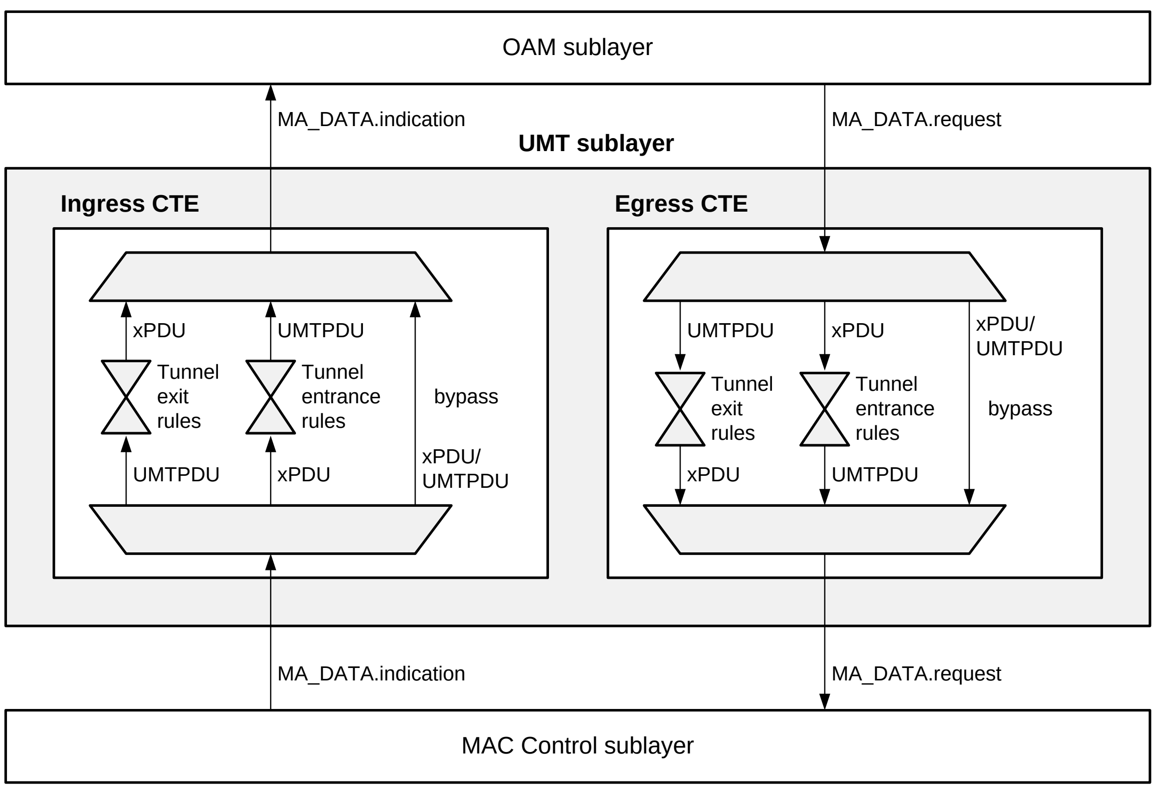


Figure 6-1—UMT sublayer functional block diagram

6.1.1 CTE rule structure

A CTE rule consists of a set of classification conditions {C_1, C_2, … C_N} and a set of modification actions {A_1, A_2, … A_M}. A rule is represented by the following notation:

IF (C_1 AND C_2 AND … C_N) THEN (A_1 AND A_2 AND … A_M)

6.1.1.1 CTE rule conditions

A condition may compare a particular header field in a frame against a provisioned value, test for existence of a field, or unconditionally return "true" or "false". A condition consists of a comparison operator and one or two operands. Supported comparison operators are listed in 6.1.1.1.1. An operand may be a numeric value or a code representing a specific field in the frame's header. Supported field codes are listed in 6.1.1.1.2. The same field may be used in multiple comparisons (either in different rules or in different conditions of the same rule). The results of all conditions provisioned for a given rule are logically ANDed together to determine whether the rule is a match. If all conditions in a rule evaluate to "true", the rule is considered to match the frame. A rule match causes all the actions associated with the rule to be applied to the frame.

6.1.1.1.1 Comparison operators

The comparison operators are used when comparing fields to the value argument of a given condition element of a CTE rule. The supported comparison operators are provided in Table 6-1.

Table 6-1—Comparison operators for the CTE rules

Symbol	Numeric Code	Meaning
nop	0x00	No operation. This operation is eqivalent to the operation '*true*'
exists	0xE1	True if field exists (value is ignored)
!exist	0xE0	True if field does not exist
==	0x11	Field equal to value
!=	0x10	Field not equal to value
true	0xA1	Always a match, i.e., the condition always evaluates to true

6.1.1.1.2 Classification fields

The CTE comparison operation elements recognize the fields shown in Table 6-2. Note that field codes listed below represent unique identifiers of various fields accessible to the CTE rules. The field codes are shown in all capital letters as opposed to the field names, which are shown as a mixture of capital and lowercase letters.

Table 6-2—L2 classification fields

FIELD_CODE	Numeric Code	Field size (bits)	Description
DST_ADDR	0x01	48	Outermost MAC Destination Address.
SRC_ADDR	0x02	48	Outermost MAC Source Address.
ETH_TYPE_LEN	0x03	16	Outermost Ethernet Type/Length field, per IEEE Std 802.3, 3.1.1
VLAN0	0x04	32	*Outermost VLAN tag*. This parameter corresponds to the first VLAN tag following the SRC_ADDR field. If no VLAN tags follow the SRC_ADDR field, then the VLAN0 field does not exist.
VLAN0_TPID	0x05	16	*Tag Protocol Identifier* of the VLAN0.
VLAN0_VID	0x06	12	*VLAN Identifier* of the VLAN0.
VLAN1	0x07	32	*Innermost VLAN tag*. This parameter corresponds to the VLAN tag that follows the outermost tag VLAN0. If no VLAN tags follow the VLAN0 field, then the VLAN1 field does not exist.

FIELD_CODE	Numeric Code	Field size (bits)	Description
VLAN1_TPID	0x08	16	*Tag Protocol Identifier* of the VLAN1.
VLAN1_VID	0x09	12	*VLAN Identifier* of the VLAN1.
UMT_SUBTYPE	0x0A	8	*UMT Subtype field*. This field exists in UMTPDUs only, where it is located immediately after the ETH_TYPE_LEN field.

6.1.1.2 CTE rule actions

An action represents a specific modification of a single header field. A field may be modified using any of the atomic operations defined in Table 6-3.

Table 6-3—Actions used in CTE rules

Action	Numeric Code	Mnemonic / Description
Add a field	0xAD	`ADD(FIELD_CODE, field_value)` This operation adds a field of the type indicated by the `FIELD_CODE` and having the value of `field_value`.
Delete (remove) a field	0xDE	`DELETE(FIELD_CODE)` This operation removes a field of the type indicated by the `FIELD_CODE`. The result of the `DELETE` operation is undefined if the field indicated by the `FIELD_CODE` is not present in the frame.
Change (replace) a field	0xCE	`CHANGE(FIELD_CODE, field_value)` This operation replaces the value of the field indicated by the `FIELD_CODE` with the value of `field_value`. The result of the `CHANGE` operation is undefined if the field indicated by the `FIELD_CODE` is not present in the frame.

The actions are applied in the order they are listed in the rule. The list of modifiable fields is shown is Table 6-2, with the following exceptions:

— No modification actions shall be applied to the `SRC_ADDR` field;

— Only CHANGE action may be applied to the `DST_ADDR` and `ETH_TYPE_LEN` fields.

Note that in a double-tagged frame, deleting an outermost VLAN tag produces a frame with an outermost VLAN tag only. Therefore, applying the following two commands results in an error:

`DELETE(VLAN0)`
`DELETE(VLAN1)` – error: `VLAN1` field does not exists

However, any of the following two sequences of actions achieve the desired result of removing both VLAN tags:

`DELETE(VLAN0)` – delete outermost tag first
`DELETE(VLAN0)` – delete the remaining tag

`DELETE(VLAN1)` – delete innermost tag first
`DELETE(VLAN0)` – delete the remaining tag

6.1.2 CTE rule categories

CTE rules are distinguished by whether they are provisioned for the receive path or the transmit path of the UMT sublayer. The rules provisioned for the receive path are called *ingress* rules and the rules provisioned for the transmit path are called *egress* rules.

Rules are also distinguished by the outcome of their actions. A rule that converts an UMTPDU into any other PDU (xPDU) is called a *tunnel exit rule* and a rule that converts xPDU into an UMTPDU is called a *tunnel entrance rule* (see Figure 6-1).

Therefore, there exist four broad categories of rules:

— Ingress tunnel exit rules;

— Ingress tunnel entrance rules;

— Egress tunnel exit rules;

— Egress tunnel entrance rules.

Figure 6-2 illustrates a network segment where the network manager (Manager) and the managed station A are both UMT-aware and where the bidirectional UMT tunnel is extended all the way from the manager to Station A. In this scenario, the intermediate switch (L2 Switch) is not required to be UMT-aware. The L2 Switch treats UMTPDUs as generic L2 frames, i.e., it forwards them based on learned or statically-provisioned MAC address tables. This scenario uses the ingress tunnel exit and egress tunnel entrance rules only.

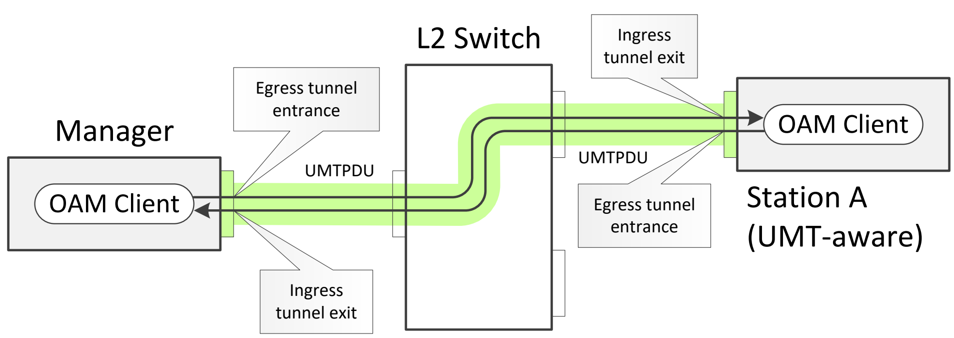


Figure 6-2—Network segment with UMT-aware station A

Figure 6-3 illustrates a network segment where the Manager is UMT-aware, but the managed station B is not. In this scenario, the intermediate switch (L2 Switch) is required to be UMT-aware in order to convert UMTPDUs into xPDUs. This scenario uses the ingress tunnel exit and egress tunnel entrance rules in the Manager port, and it uses egress tunnel exit and ingress tunnel entrance rules in the Switch port connected to the Station B.

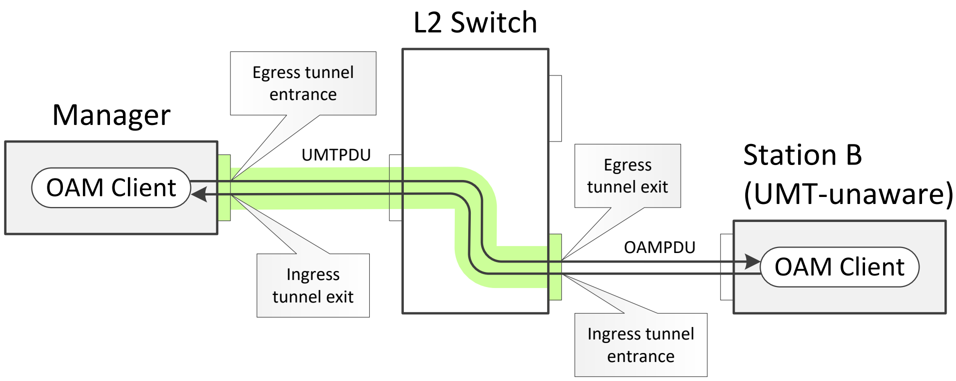


Figure 6-3—Network segment with UMT-unaware station B

6.2 Receive path specification

6.2.1 Ingress tunnel exit rules

The ingress tunnel exit rules are provisioned in the receive path of the UMT sublayer. These rules specify the conditions and the associated actions required for an UMT frame to exit the tunnel. A frame that exits a tunnel is converted from UMTPDU into a specific xPDU associated with the given UMT subtype. Generally, only a single ingress UMT exit rule is required per each protocol type and all such rules are statically pre-configured into a UMT-aware device. Different ingress ports may be pre-configured to accept different protocol types.

The ingress exit rules do not depend on any network-specific parameters. Therefore, these rules may be statically pre-configured for the UMT-aware devices.

6.2.1.1 Ingress tunnel exit rule for OAM subtype

The tunnel exit rule for the OAM subtype is shown in Table 6-4. This rule converts an UMTPDU into an OAMPDU. The conversion is straight-forward and involves only a replacement of the destination MAC address value and the Ethertype value.

Table 6-4—Ingress tunnel exit rule for OAM subtype

<table>
<tr><th>Conditions</th><th>Actions</th></tr>
<tr><td>1. ETYPE_LEN == UMT_TYPE
2. DA == <local_MAC_addr>
3. UMT_SUBTYPE == OAM_subtype</td><td>1. CHANGE(DA, SP_DA)
2. CHANGE(ETYPE_LEN, SP_TYPE)</td></tr>
<tr><td colspan="2">NOTE:
UMT_TYPE – Ethertype value identifying UMTPDUs (see 5.1)
<local_MAC_addr> - MAC address associated with the given port
OAM_SUBTYPE – UMT subtype value identifying OAMPDU payload (see 5.2)
SP_DA – Destination MAC address associated with Slow Protocols (see IEEE Std 802.3, 57A.3)
SP_TYPE – Slow Protocol Ethertype value (see IEEE Std 802.3, 57A.4)</td></tr>
</table>

6.2.1.2 Ingress tunnel exit rule for L2 subtype

<TBD>

6.2.1.3 Ingress tunnel exit rule for L3 subtype

<TBD>

6.2.2 Ingress tunnel entrance rules

The ingress tunnel exit rules are provisioned in the receive path of the UMT sublayer. These rules specify the conditions and the associated actions required for an UMT frame to exit the tunnel. A frame that exits a tunnel is converted from UMTPDU into a specific xPDU associated with the given UMT subtype. Generally, only a single ingress UMT exit rule is required per each protocol type and all such rules are statically pre-configured into a UMT-aware device. Different ingress ports may be pre-configured to accept different protocol types.

6.2.2.1 Ingress tunnel entrance rule for OAM subtype

The ingress tunnel entrance rule for the OAM subtype is shown in Table 6-5. This rule converts an OAMPDU into an UMTPDU. The conversion involves only a replacement of the destination MAC address value with the value provisioned for this port. The OAM defined in IEEE Std 802.3 is a link-based protocol (i.e., a protocol operating between two peer connected by a single point-to-point link). Therefore, there could only be a single OAM client connected to the given port.

Table 6-5—Ingress tunnel entrance rule for OAM subtype

Conditions	Actions
1. DA == SP_DA 2. ETH_TYPE_LEN == SP_type 3. SUBTYPE == OAM_subtype	1. CHANGE(DA, <MAC$_i$>) 2. CHANGE(ETH_TYPE_LEN, UMT_type)
NOTE: SP_TYPE – Slow Protocol Ethertype value (see IEEE Std 802.3, 57A.4) UMT_TYPE – Ethertype value identifying UMTPDUs (see 5.1) OAM_SUBTYPE – Subtype value identifying OAMPDUs (see IEEE Std 802.3, 57.3.1.1) SP_DA – Destination MAC address associated with Slow Protocols (see IEEE Std 802.3, 57A.3) <MAC$_i$> – Tunnel destination MAC address provisioned for this rule.	

6.2.2.2 Ingress tunnel entrance rule for L2 subtype

<TBD>

6.2.2.3 Ingress tunnel entrance rule for L3 subtype

<TBD>

6.3 Transmit path specification

6.3.1 Egress tunnel exit rules

The egress tunnel exit rules are provisioned in the transmit path of the UMT sublayer. These rules specify the conditions and the associated actions required for an UMT frame to exit the tunnel. A frame that exits a tunnel is converted from UMTPDU into a specific xPDU associated with the given UMT subtype. The egress tunnel exit rules are employed when it is necessary for a UMTPDU to be relayed by a switch (i.e., an egress port needs to be selected based on UMTPDU destination MAC address). Once the egress port is selected, the frame is converted into xPDU, possibly losing the unicast destination MAC address value. Different egress ports may be pre-configured to terminate tunnels for different protocol types.

6.3.1.1 Egress tunnel exit rule for OAM subtype

The egress tunnel exit rule for the OAM subtype is shown in Table 6-6. This rule converts an UMTPDU into an OAMPDU. The conversion is straight-forward and involves only a replacement of the destination MAC address value. The rule does not need to check for the UMT destination MAC address since that MAC address has been used by forwarding engine to select the given egress port.

Table 6-6—Egress tunnel exit rule for OAM subtype

Conditions	Actions
`1. ETH_TYPE_LEN == UMT_type` `2. UMT_SUBTYPE == OAM_subtype`	`1.CHANGE( DA, SP_DA )` `2.CHANGE( ETH_TYPE_LEN, SP_type )`

NOTE:
`UMT_TYPE` – Ethertype value identifying UMTPDUs (see 5.1)
`SP_TYPE` – Slow Protocol Ethertype value (see IEEE Std 802.3, 57A.4)
`OAM_SUBTYPE` – Subtype value identifying OAMPDUs (see IEEE Std 802.3, 57.3.1.1)
`SP_DA` – Destination MAC address associated with Slow Protocols (see IEEE Std 802.3, 57A.3)

6.3.1.2 Egress tunnel exit rule for L2 subtype

<TBD>

6.3.1.3 Egress tunnel exit rule for L3 subtype

<TBD>

6.3.2 Egress tunnel entrance rules

The egress tunnel entrance rules are provisioned in the transmit path of the UMT sublayer. These rules specify the conditions and the associated actions required for an xPDU to enter the tunnel. A frame that enters a tunnel is converted from an xPDU into an UMTPDU with a specific UMT_SUBTYPE value associated with the given xPDU type.

6.3.2.1 Egress tunnel entrance rule for OAM subtype

The egress tunnel entrance rule for the OAM subtype is shown in Table 6-7. This rule converts a locally-generated OAMPDU into an UMTPDU at the egress of an UMT-aware device. The conversion involves the replacement of the destination MAC address value with the value provisioned for this port and the replacement the Slow Protocol Ethertype with the UMT Ethertype.

Table 6-7—Egress tunnel entrance rule for OAM subtype

Conditions	Actions

`1. DA == SP_DA`
`2. ETH_TYPE_LEN == SP_type`
`3. SUBTYPE == OAM_subtype`
	1.CHANGE(DA, <MAC$_i$>)
	`2.CHANGE( ETH_TYPE_LEN, UMT_type )`

NOTE:

SP_TYPE – Slow Protocol Ethertype value (see IEEE Std 802.3, 57A.4)
UMT_TYPE – Ethertype value identifying UMTPDUs (see 5.1)
OAM_SUBTYPE – Subtype value identifying OAMPDUs (see IEEE Std 802.3, 57.3.1.1)
SP_DA – Destination MAC address associated with Slow Protocols (see IEEE Std 802.3, 57A.3)
<MAC$_i$> – Tunnel destination MAC address provisioned for this rule.

6.3.2.2 Egress tunnel entrance rule for L2 subtype

<TBD>

6.3.2.3 Egress tunnel entrance rule for L3 subtype

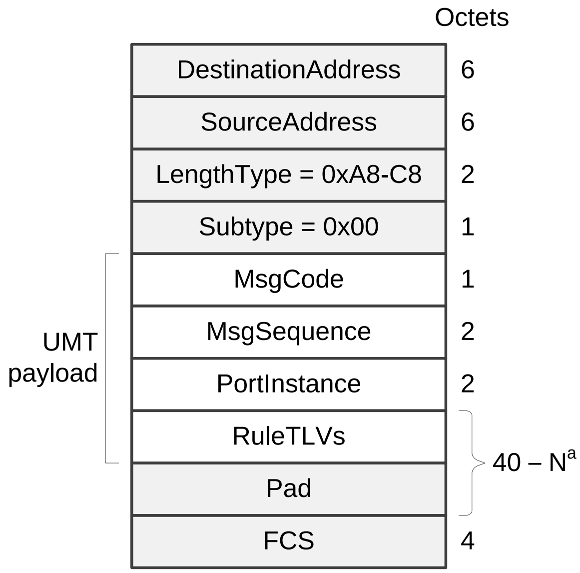
<TBD>

7 UMT configuration

The tunnels originate and terminate in the UMT-aware devices. The tunnels are configured by means of provisioning specific CTE rules for the tunnel entry and exit points. These rules are provisioned by the operator using the *UMT_CONFIG* UMTPDUs, which carry a set of *condition-encoding* TLVs and a set of *action-encoding* TLVs.

7.1 Configuration UMTPDU

The *UMT_CONFIG* UMPTPDU format shall be as depicted in Figure 7-1. The *UMT_CONFIG* UMTPDU is used as both a request to configure a CTE rule as well as a response containing the result of the configuration request.



a – Maximum field length depends on frame type (see Figure 5-1).

Figure 7-1—*UMT_CONFIG* UMTPDU format

The *UMT_CONFIG* UMTPDU is an instantiation of the generic UMTPDU (see Figure 5-1). It is identified by the *Subtype* field value of 0x00. The structure of the *UMT payload* is defined as follows:

—*MsgCode*:

The *MsgCode* field identifies whether the UMT_CONFIG message is a request message or a response. If the UMTPDU is a request, this field encodes the requested action. If the UMTPDU is a response, this field echoes the requested action and encodes the result code for this action. The format of the *MsgCode* field is shown in Table 7-1.

Table 7-1—Format of the *MsgCode* field

Bits	Field name	Value	Description
3:0	*MsgType*	0x0	The message is a request
		0x1	The message is a response indicating successful action
		0x2	The message is a response indicating failed action
		0x3	The message is a response indicating that no action was necessary

		0x4	The message is a response indicating invalid request
		0x5 to 0xF	Reserved, ignored on reception
7:4	*RequestCode*	0x0	Query all rules
		0x1	Add a rule
		0x2	Remove a rule
		0x4 to 0xF	Reserved, ignored on reception

—*MsgSequence*:

In situations when a UMT configuration request or a response consists of multiple messages, this field identifies the message sequence number. The field is represented by a decrementing counter, with the last message in a sequence having the *MsgSequence* value of zero. When a request or a response consists of a single UMTPDU, this field has the value of zero.

—*PortInstance*:

This field identifies a port instance in the UMT-aware device to which the given *UMT_CONFIG* UMTPDU applies. The format of the *PortInstance* field is shown in Table 7-2.

Table 7-2—Format of the *PortInstance* field

Bits	Field name	Value	Description
14:0	*PortIndex*	0x00-00 to 0x7F-FF	Index of a port (UMT sublayer) to which the requested action is to be applied.
15	*Direction*	0	The rule is to be applied to the transmit path of UMT sublayer (i.e., an ingress rule)
		1	The rule is to be applied to the receive path of UMT sublayer (i.e., an egress rule)

In the UMT response message, this field reflects the *PortInstance* field value from the corresponding UMT request message.

— *RuleTLVs*:

This field includes one or more CTE rule TLV(s) as defined in 7.2. The combined size of the *RuleTLV* and *Pad* fields ranges between 40 and *N*, where *N* is defined in Figure 5-1.

7.2 CTE rule TLV structure

The structure of a CTE rule TLV is shown in Table 7-3. Each *UMT_CONFIG* UMTPDU shall contain at least one CTE rule TLV.

Table 7-3—CTE rule TLV structure

Field Size (octets)	Field Name	Value	Description
1	*Type*	0xC0	Type code identifying the condition-encoding TLV
		0xAC	Type code identifying the action-encoding TLV
		0x00	Type code indicating that there are no more TLVs to process. The Length field and other fields (if present) are ignored. The TLV with Type = 0x00 shall be the last TLV in every *UMT_CONFIG* UMTPDU and it may be the only TLV in the *UMT_CONFIG* UMTPDU.
1	*Length*	N+4	The *Length* field encompasses the entire TLV, including the *Type* and *Length* fields. A TLV with length of 0x00 or 0x01 is invalid, and on reception, should be treated as TLV with Type 0x00.
1	*Operation*	per Table 6-1	Comparison operator code, if the TLV *Type* = 0xC0
		per Table 6-3	Action code, if the TLV *Type* = 0xAC
1	*FieldCode*	per Table 6-2	Identifies a field to be used in a comparison, or to be modified by an action.
N	*Value*	various	The value to be used in a comparison or by an Add/Change action. Some TLVs may omit this field.

8 Protocol implementation conformance statement (PICS) proforma for Universal management Tunnel (UMT) specification

8.1 Introduction

This subclause specifies the PICS proforma for Universal management Tunnel (UMT).

The supplier of an UMT implementation that is claimed to conform to this standard shall complete the following PICS proforma.[11]

A detailed description of the symbols used in the PICS proforma, along with instructions for completing the PICS proforma, can be found in 3.5.

8.2 Implementation identification

UMT Supplier[1]	
Contact point for enquiries about the PICS[1]	
Implementation Name(s) and Version(s)[1,3]	
Other information necessary for full identification, e.g., name(s) and version(s) for machines and/or operating systems; System Name(s)[2]	
NOTE 1—Required for all implementations. NOTE 2—May be completed as appropriate in meeting the requirements for the identification. NOTE 3—The terms *Name* and *Version* should be interpreted appropriately to correspond with a supplier's terminology (e.g., Type, Series, Model).	

8.3 Protocol summary

Identification of the UMT implementation	IEEE Std 1904.2-202x
Identification of amendments and corrigenda to this PICS proforma that have been completed as part of this PICS	
Have any Exception items been required?	[ ] [ ] No
	[ ] [ ] Yes
(See 3.6; the answer Yes means that the implementation of the given UMT implementation does not conform to IEEE Std 1904.2)	

[11] *Copyright release for PICS proformas*: Users of this standard may freely reproduce the PICS proforma in this subclause so that it can be used for its intended purpose and may further publish the completed PICS.

Date of Statement	

8.4 UMT Capabilities

Item	Description	Subclause	Value/Comment	Status	Support

Annex 7A
(informative)
UMT configuration examples (informative)

7A.1 IEEE Std 802.3 OAM over UMT Use case

<TBD>