IEEE P1904.2™/D0.9 Draft Standard for Control and Management of Virtual Links in 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

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. The IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and serve without compensation. While the IEEE administers the process and establishes rules to promote fairness in the consensus development process, the IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

Use of an IEEE Standard is wholly voluntary. The IEEE disclaims liability for any personal injury, property or other damage, of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, or reliance upon this, or any other IEEE Standard document.

The IEEE does not warrant or represent the accuracy or content of the material contained herein, and expressly disclaims any express or implied warranty, including any implied warranty of merchantability or fitness for a specific purpose, or that the use of the material contained herein is free from patent infringement. IEEE Standards documents are supplied "**AS IS**."

The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE Standard is subjected to review at least every five years for revision or reaffirmation, or every ten years for stabilization. When a document is more than five years old and has not been reaffirmed, or more than ten years old and has not been stabilized, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE Standard.

In publishing and making this document available, the IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity. Nor is the IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing this, and any other IEEE Standards document, should rely upon his or her independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration. A statement, written or oral, that is not processed in accordance with the IEEE-SA Standards Board Operations Manual shall not be considered the official position of IEEE or any of its committees and shall not be considered to be, nor be relied upon as, a formal interpretation of the IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position, explanation, or interpretation of the IEEE.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Recommendations to change the status of a stabilized standard should include a rationale as to why a revision or withdrawal is required. Comments and recommendations on standards, and requests for interpretations should be addressed to:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
Piscataway, NJ 08854
USA

Authorization to photocopy portions of any individual standard for internal or personal use is granted by The Institute of Electrical and Electronics Engineers, Inc., provided that the appropriate fee is paid to Copyright Clearance Center. To arrange for payment of licensing fee, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Introduction

This introduction is not part of IEEE P1904.2/D0.9

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 Virtual Link Control (VLC) 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 VLC stations using broadcast or multicast addressing.

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

ITU-T Recommendation G.988, ONU management and control interface (OMCI) specification

ITU-T Recommendation G.984.3, Gigabit-capable Passive Optical Networks (G-PON): Transmission convergence layer specification

ITU-T Recommendation G.987.3, 10-Gigabit-capable passive optical networks (XG-PON): Transmission convergence (TC) layer specification

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 VLC-aware L2 network. NMS combines, terminates, proxies, or snoops a number of different control and management protocols (outside the scope of this standard), providing Fault, Configuration, Accounting, Performance, Security (FCAPS) management functionality for a network operator.

3.2 Acronyms and abbreviations

VLC	Virtual Link Control
PDU	Protocol Data Unit
CTE	Classification and Translation Engine
OAM	Operations, Administration, and Management
OMCI	ONU Management Control Interface
MAC	Media Access Control
OLT	Optical Line Terminal
ONU	Optical Network Unit
NMS	Network Management System
FCAPS	Fault, Configuration, Accounting, Performance, 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).

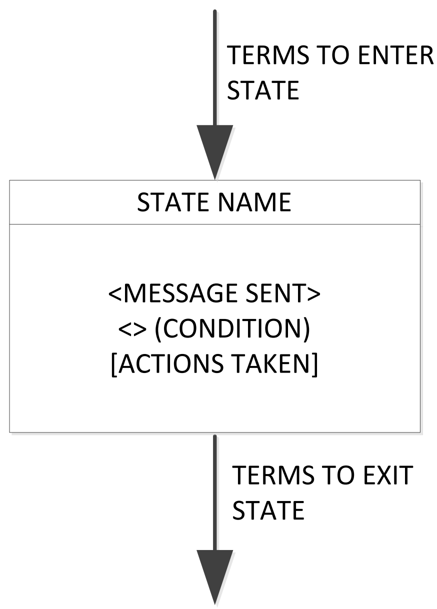


Figure 3-1—State diagram notation example

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:

Deleted: Figure 3-1

Formatted: Font:

Deleted: 3.4.3

- — 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.

Deleted: Table 3-1

Table 3-1—State diagram operators

Character	Meaning
AND	Boolean AND
OR	Boolean OR
XOR	Boolean XOR
!	Boolean NOT
<	Less than
>	More than

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

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: 0b00001000 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.

Editorial Note (to be removed prior to publication): The following text in yellow needs to be replaced with a valid example of PICS, once PICS become available.

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 Virtual Link Control (VLC) Overview and Architecture

4.1 Principles of operation

Virtual Link Control (VLC) defines the method of encapsulating various protocol data units (xPDUs) in Ethernet frames with VLC Ethertype (0xA8-C8). An Ethernet frame with VLC Ethertype is called a Virtual Link Control Protocol Data Unit (VLCPDU). That portion of the network path that xPDUs traverse while they are encapsulated as VLCPDUs is referred to as a *tunnel*.

The xPDU-to-VLCPDU and VLCPDU-to-xPDU conversions take place within the VLC sublayer (see 4.2). Both VLC client and VLC sublayer are optional, i.e., in any multi-port device, the VLC sublayer may be implemented in only some ports. Devices that implement the VLC sublayer in at least one of the ports are said to be VLC-aware.

Devices that do not implement the VLC sublayer in any of the ports are called VLC-unaware. VLC-unaware devices are able to relay VLCPDUs as generic Ethernet frames using existing L2 forwarding mechanisms but are unable to consume or generate VLCPDUs.

The VLC sublayer includes the Classification and Translation Engine (CTE) that converts xPDUs into VLCPDUs 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 VLC sublayer provides a service interface to the OAM sublayer and the VLC client, and may provide a service interface to other L2 protocol-specific clients. The only messages that are passed to and received from the VLC client are VLC configuration messages (see *VLC_CONFIG* VLCPDU in 8.1.1).

Deleted: ,

Deleted: the

All VLCPDUs, except *VLC_CONFIG* VLCPDUs, carry tunneling payloads associated with specific protocols (xPDU). Any payload-carrying VLCPDU 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 VLCPDU that is generated by a device originates in a protocol-specific client as an xPDU and is then converted into a VLCPDU within the VLC sublayer.

Deleted: the

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

4.1.1 VLC discovery protocol

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

4.2 VLC sublayer

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

Deleted: Figure 4-1

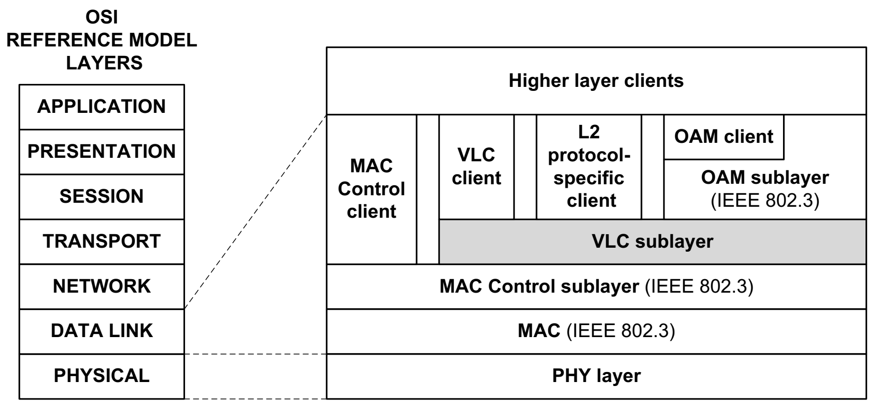


Figure 4-1—VLC sublayer relationship to the ISO/IEC Open Systems Interconnection (OSI) reference model and the IEEE Std 802.3 Ethernet model

4.3 VLC service interfaces

4.3.1 Definitions of VLC primitives

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

The VLC sublayer provides a VLC service interface (VLCSI) to the OAM sublayer, the VLC client, and to other L2 protocol-specific clients (see Figure 4-2). To the OAM sublayer, the VLC sublayer presents a standard IEEE Std 802.3 MAC service interface (*VLCSI:MA_DATA*). To the VLC client, the VLC sublayer presents a VLC-specific service interface (*VLCSI:VLCPDU*). To the L2 protocol-specific clients, the VLC 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.

Deleted: Figure 4-2

Deleted: Figure 4-2

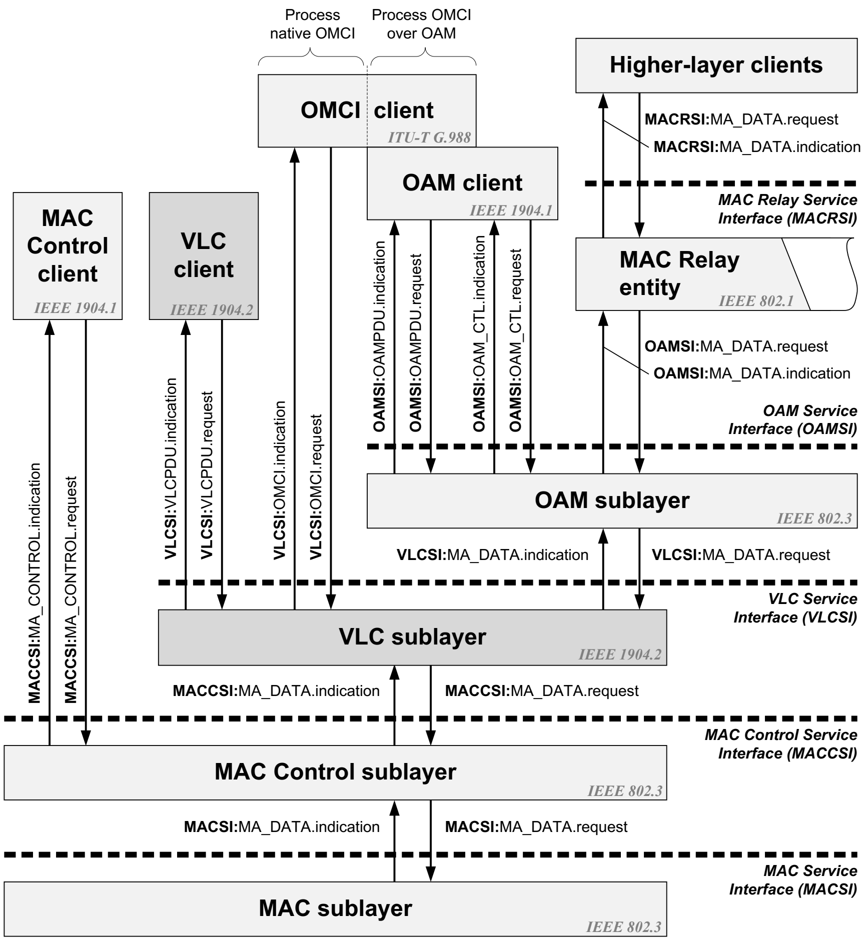


Figure 4-2—Positioning of VLC sublayer and service interfaces

4.3.1.1 VLCSI:MA_DATA primitives

4.3.1.1.1 VLCSI:MA_DATA.request

4.3.1.1.2 VLCSI:MA_DATA.indication

4.3.1.2 VLCSI:VLCPDU primitives

4.3.1.2.1 VLCSI:VLCPDU.request

4.3.1.2.2 VLCSI:VLCPDU.indication

4.3.1.3 VLCSI:OAMPDU primitives

4.3.1.3.1 VLCSI:OAMPDU.request

4.3.1.3.2 VLCSI:OAMPDU.indication

4.3.1.4 VLCSI:OMCI primitives

The OMCI client communicates with the VLC CTE using the following service primitives:

Deleted: C

— *VLCSI:OMCI.request*
— *VLCSI:OMCI.indication*

The VLCSI:OMCI interface (see Figure 4-2) is optional, but if it is implemented, the *VLCSI:OMCI.request* and *VLCSI:OMCI.indication* service primitives described in this subclause shall be supported..

Deleted: Figure 4-2

4.3.1.4.1 VLCSI:OMCI.request

4.3.1.4.1.1 Function

This primitive defines the transfer of data from the OMCI client entity to the VLC CTE. This primitive is only relevant in the egress direction.

Deleted: C

4.3.1.4.1.1.1.1.1 Semantics of the service primitive

Formatted: Heading 9

The semantics of the primitive are as follows:

VLCSI:OMCI.request (
omci_vendor_id,
omci_serial_number,
omci_frame_sdu
)

The *omci_vendor_id* parameter specifies the 4-octet Vendor ID assigned to the ONU that is the intended destination of this OMCI frame. Note that the ONU may not be the same device where the *VLCSI:OMCI.request* primitive was generated.

The *omci_serial_number* parameter specifies the 4-octet Vendor-Specific Serial Number assigned to the ONU that is the intended destination of this OMCI frame.

The *omci_frame_sdu* parameter contains the pre-formed OMCI frame (according to ITU-T Rec G.988) that is related to the ONU identified by the unique combination of the *omci_vendor_id* and *omci_serial_number*.

4.3.1.4.1.2 When Generated

This primitive is generated by the OMCI client entity whenever an OMCI frame is to be transferred to a peer entity.

Deleted: C

4.3.1.4.1.3 Effect of Receipt

The receipt of this primitive will cause the VLC CTE to apply the rules installed in the egress CTE instance to perform any required parsing and transformations of the request parameters necessary to encapsulate and transmit the OMCI frame as a VLCPDU. After performing these actions, the VLC CTE entity asserts the *MACCSI:MA_DATA.request* primitive according to the procedures described in 4.3.1.x.

4.3.1.4.2 VLCSI:OMCI.indication

4.3.1.4.2.1 Function

This primitive defines the transfer of data from the VLC sublayer to the OMCI client entity. This primitive is only relevant in the ingress direction.

Deleted: C

4.3.1.4.2.2 Semantics of the service primitive

The semantics of the primitive are as follows:

VLCSI:OMCI.indication (
omci_vendor_id,
omci_serial_number,
omci_frame_sdu
)

The *omci_vendor_id, omci_serial_number, and omci_frame_sdu* parameters are as defined in 4.3.1.4.1.1.1.1.

Deleted: 4.3.1.4.1.2

5 Virtual Link Control Protocol Data Units (VLCPDU)

5.1 VLCPDU Structure

A Virtual Link Control Protocol Data Unit (VLCPDU) is an Ethernet MAC frame with the value of *Ethertype* field equal to the VLC Ethertype (0xA8-C8). The Ethernet MAC frame format is shown in IEEE Std 802.3, Clause 3.

	Octets
DestinationAddress	6
SourceAddress	6
LengthType = 0xA8-C8	2
Subtype	1
VLC payload	45 - 1981
Pad	
FCS	4

Figure 5-1—VLCPDU format

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

Deleted: Figure 5-1

—*DestinationAddress*:

In a VLCPDU, the *DestinationAddress* is the MAC address associated with the device consuming the xPDU carried within the VLCPDU.

NOTE – The station identified by *DestinationAddress* might not be VLC-aware, in which case the VLC tunnel is terminated before the VLCPDU reaches that station.

—*SourceAddress*:

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

NOTE – The station identified by *SourceAddress* might not be VLC-aware, in which case the VLC tunnel is initiated after the xPDU leaves that station.

—*LengthType*:

The *LengthType* field in a VLCPDU carries the VLC Ethertype value 0xA8-C8.

—*Subtype*:

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

Deleted: Table 5-1

—*VLC payload*:

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

—*Pad*:

The *Pad* field is added to bring the VLCPDU 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 VLCPDU *Subtype* encoding

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

Deleted: Table 5-1

Table 5-1—*Subtype* field encoding

Value	Designation	Description
0x00	*VLC_config*	*VLC_config* subtype identifies *VLC_Request* and *VLC_Response* VLCPDUs used for configuring the VLC Classification and Translation Engine (see 6.1).
0x01, 0x02	n/a	Reserved for VLC Discovery protocol; ignored on reception.
0x03	*OAM_Subtype*	*OAM_Subtype* represents the OAMPDU payload carried within the VLCPDU (see 5.2.2).
0x04	n/a	Reserved; ignored on reception
0x05	*L2_subtype*	*L2_Subtype* represents a generic Ethernet frame carried within the VLCPDU (e.g., MAC-in-MAC) (see 5.2.4).
0x06	*L3_Subtype*	*L3_Subtype* represents a generic L3 packet (plus TPID) carried within the VLCPDU (see 5.2.5).
0x07 to 0x0B	n/a	Reserved; ignored on reception.
0x0C	*OMCI_Subtype*	*OMCI_Subtype* represents the OMCI payload carried within the VLCPDU (see 5.2.3).
0x0D to 0xFD	n/a	Reserved; ignored on reception.
0xFE	*OUI24_Subtype*	*OUI24_Subtype* represents an organization-specific payload carried within the VLCPDU. The organization is identified by a unique OUI/CID value (see 5.2.6).
0xFF	*OUI36_Subtype*	*OUI36_Subtype* represents an organization-specific payload carried within the VLCPDU. The organization is identified by a unique OUI-36 value (see 5.2.6).

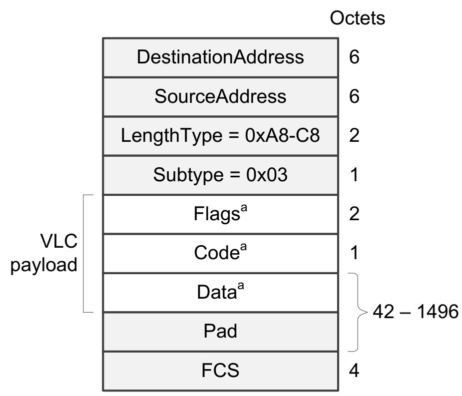
5.2.1 VLC configuration subtype

A VLCPDU with VLC configuration subtype (*Subtype* field = 0x00) identifies a *VLC_CONFIG* VLCPDU used for configuring the VLC Classification and Translation Engine (see 6.1). This VLCPDU is defined in 8.1.1.

5.2.2 OAM subtype

A VLCPDU with OAM subtype (*Subtype* field = 0x03) is an instantiation of a generic VLCPDU, 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 a VLCPDU with OAM subtype shall be as depicted in Figure 5-2.

Deleted: Figure 5-2



a – This field is defined in IEEE 802.3, 57.4

Figure 5-2—Format of VLCPDU with OAM subtype

The structure of the *VLC payload* in the VLCPDU 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

A VLCPDU with OMCI subtype (*Subtype* field = 0x0C) is an instantiation of a generic VLCPDU, as defined in 5.1, that carries an ONU Management and Control Interface (OMCI) payload (see ITU-T Rec G.988). The frame structure of a VLCPDU with OMCI subtype shall be as depicted in Figure 5-3.

Deleted: Figure 5-3

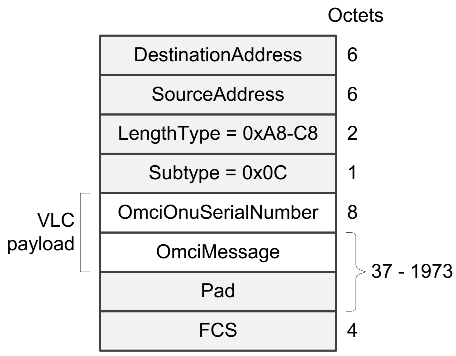


Figure 5-3—Format of VLCPDU with OMCI Subtype

The structure of the *VLC payload* in the VLCPDU with OMCI subtype is defined as follows:

—*OmciOnuSerialNumber*:

This field carries the serial number of the ONU associated with the OMCI message. The serial number is defined in ITU-T Rec G.988, 9.1.1.

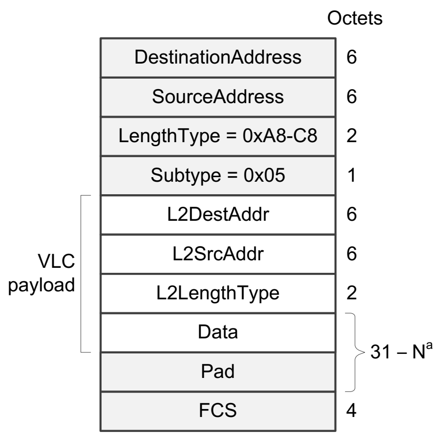
—*OmciMessage*:

This field carries one OMCI message in baseline or extended format. The OMCI baseline and extended message formats are defined in ITU-T Rec G.988, Clause 11.

5.2.4 L2 Subtype

A VLCPDU with L2 subtype (*Subtype* field = 0x05) is an instantiation of a generic VLCPDU, as defined in 5.1, that carries a complete L2 frame as its payload. The frame structure of a VLCPDU with L2 subtype shall be as depicted in Figure 5-4.

Deleted: Figure 5-4



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

Figure 5-4—Format of VLCPDU with L2 subtype

The structure of the *VLC payload* in the VLCPDU with L2 subtype is defined as follows:

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

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

—*L2LengthType*:
This field carries the Length/Type value of the original L2 frame being tunneled using VLC.

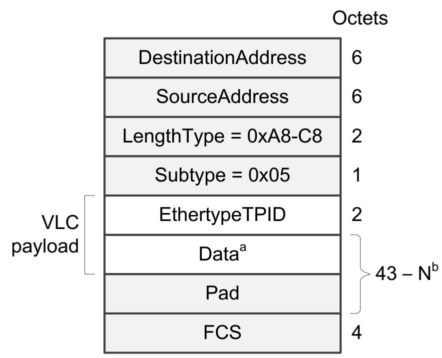
—*Data*:
This field carries the L2 payload of the original L2 frame being tunneled using VLC. The combined size of the *Data* and *Pad* fields ranges between 31 and *N*, where *N* is defined in Figure 5-1.

Deleted: Figure 5-1

5.2.5 L3 Subtype

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

Deleted: Figure 5-5



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

Figure 5-5—Format of VLCPDU with L3 subtype

The structure of the *VLC payload* in the VLCPDU with L3 subtype is defined as follows:

—*EthertypeTPID*:
This field carries the L2 Ethertype/TPID value of the original L3 packet being tunneled using VLC.

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

Deleted: Figure 5-1

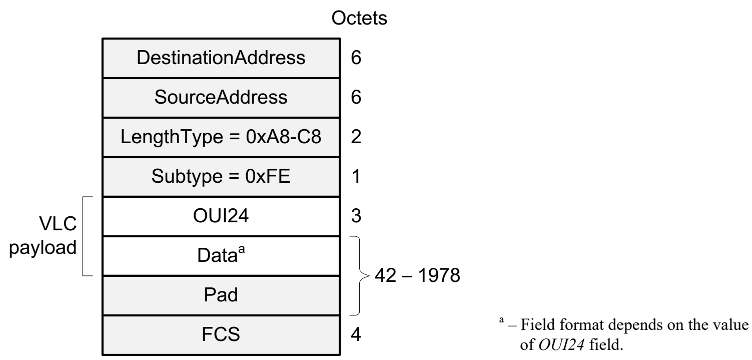
5.2.6 Organization-specific extension subtypes

The Organization-specific VLCPDU is an instantiation of a generic VLCPDU as defined in 5.1. It is identified by the *Subtype* field value of *OUI24_Subtype* or *OUI36_Subtype* and it is used for organization specific extensions.

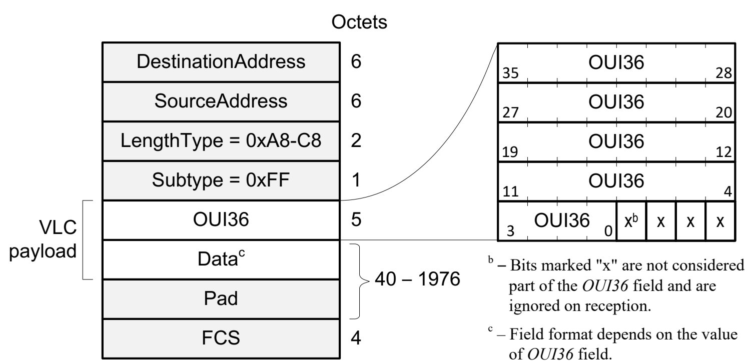
The format and frame structure of the Organization-Specific VLCPDU with *OUI24_Subtype* shall be as depicted in Figure 5-6(a) and the format and frame structure of the VLCPDU with *OUI36_Subtype* shall be as depicted in Figure 5-6(b).

Deleted: Figure 5-6

Deleted: Figure 5-6



a) VLCPDU format with *OUI24_Subtype* (0xFE)



b) VLCPDU format with *OUI36_Subtype* (0xFF)

Figure 5-6—Format of VLCPDU with organization-specific extension subtype

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

—*OUI24*:

This field carries the Organizationally Unique Identifier (OUI) or Company ID (CID) value assigned to an organization by the IEEE Registration Authority (IEEE RA)[2].

---

[2] Refer to Guidelines for Use of Extended Unique Identifier (EUI), Organizationally Unique Identifier (OUI), and Company ID (CID) at https://standards.ieee.org/content/dam/ieee-standards/standards/web/documents/tutorials/eui.pdf.

—*OUI36*:

This field carries the Organizationally Unique 36-bit Identifier (OUI-36) value assigned to an organization by the IEEE RA.

—*Data*:

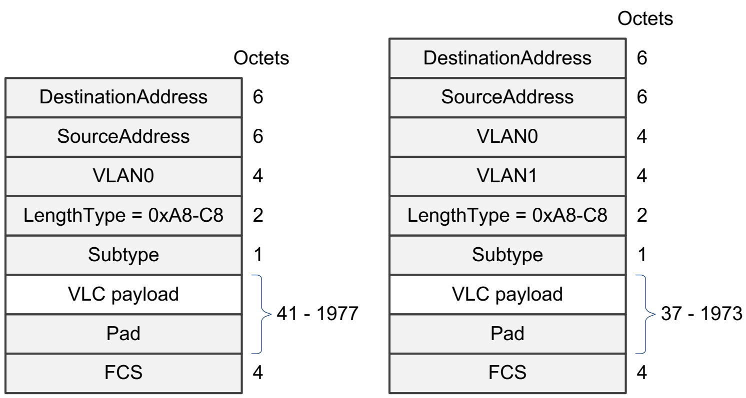
This field carries the OUI/CID-specific data payload. The internal format of the *Data* field is dependent on *OUI24* or *OUI36* field value and is beyond the scope of this standard. The combined size of the *Data* and *Pad* fields ranges between 42 and *N*, where *N* is defined in Figure 5-1.

Deleted: Figure 5-1

5.3 VLAN-Tagged VLCPDU

All VLCPDU subtypes defined in 5.2.1 through 5.2.6 may include one or two VLAN tags. If a single VLAN tag is used as part of VLCPDU header, the maximum allowed VLC payload size is reduced by 4 octets. If two VLAN tags are used, the maximum VLC payload size is reduced by 8 octets. The format of single-tagged and double-tagged VLCPDUs is shown in Figure 5-7.

Deleted: Figure 5-7



a) Single-tagged VLCPDU **b) Double-tagged VLCPDU**

Figure 5-7—Single-tagged and double-tagged VLCPDU format

Operations on VLAN-tagged VLCPDUs are described in 6.1.3.

6 VLC sublayer

6.1 VLC Classification and Translation Engine

The function of the VLC Classification and Translation Engine (CTE) is to classify frames by certain criteria and to perform specific modifications on the frames that match the criteria. The classification criteria together with the associated modification actions 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 VLCPDUs into xPDUs (i.e., into frames with different Ethertype values) and vice versa.

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 the *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.

Deleted: Figure 6-1

6.1.1 CTE rule structure

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

$$\text{IF } (C_1 \text{ AND } C_2 \text{ AND } \ldots C_N) \text{ THEN } (A_1 \text{ AND } A_2 \text{ AND } \ldots A_M)$$

6.1.1.1 CTE rule classification 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.

Deleted: Table 6-1

Table 6-1—Comparison operators for the CTE rules

Symbol	Numeric Code	Meaning
nop	0x00	No operation. This operation is equivalent 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

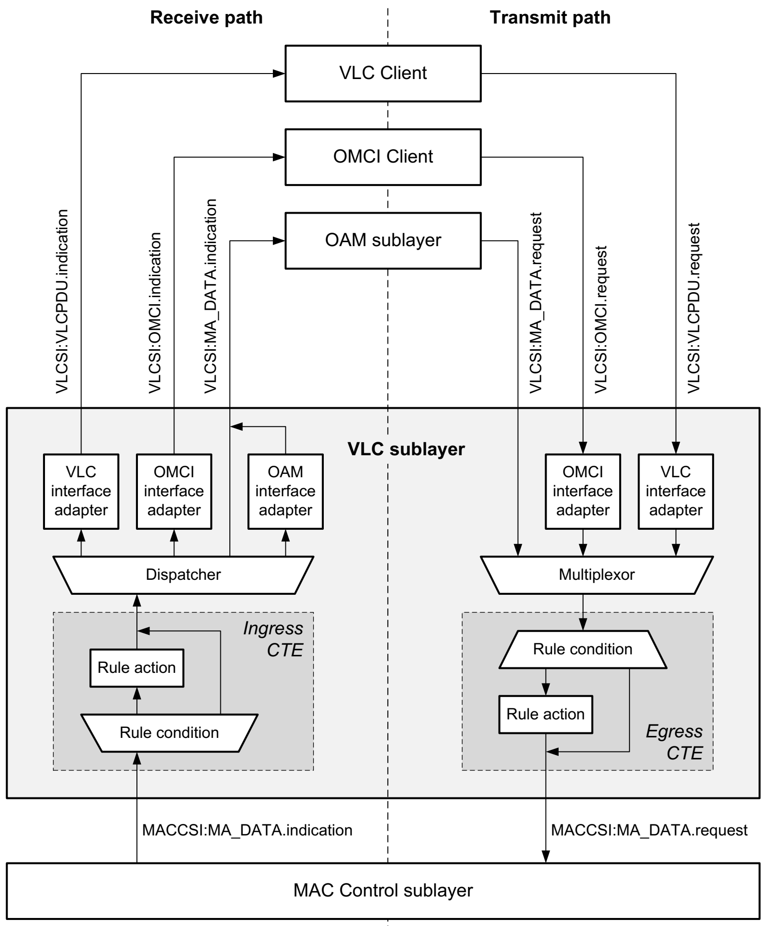


Figure 6-1—VLC sublayer functional block diagram

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.

Deleted: Table 6-2

Table 6-2—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.
VLAN1	0x05	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.
VLC_DST_ADDR	0x11	48	*VLCPDU MAC Destination Address*. In VLCPDUs, this field code is equivalent to DST_ADDR. In other (non-VLC) PDU types, this field does not exist.
VLC_SRC_ADDR	0x12	48	*VLCPDU MAC Source Address*. In VLCPDUs, this field code is eqivalent to SRC_ADDR. In other (non-VLC) PDU types, this field does not exist.
VLC_ETH_TYPE	0x13	16	*VLC Ethernet Type*. In VLCPDUs, this field code is eqivalent to ETH_TYPE_LENGTH. In other (non-VLC) PDU types, this field does not exist.
VLC_VLAN0	0x14	32	*VLCPDU Outermost VLAN tag*. In VLCPDUs, this field code is eqivalent to VLAN0. In other (non-VLC) PDU types, this field does not exist.
VLC_VLAN1	0x15	32	*VLCPDU Innermost VLAN tag*. In VLCPDUs, this field code is eqivalent to VLAN1. In other (non-VLC) PDU types, this field does not exist.
VLC_SUBTYPE	0x16	8	*VLC Subtype field*. This field exists in VLCPDUs only, where it is located immediately after the *VLC_ETH_TYPE* field.
XPDU_DST_ADDR	0x21	48	*xPDU MAC Destination Address*. In xPDUs (non-VLC types), this field code is eqivalent to DST_ADDR. In VLCPDUs, this field does not exist.
XPDU_SRC_ADDR	0x22	48	*xPDU MAC Source Address*. In xPDUs (non-VLC types), this field code is eqivalent to SRC_ADDR. In VLCPDUs, this field does not exist.
XPDU_ETH_TYPE	0x23	16	*xPDU Ethernet Type*. In xPDUs (non-VLC types), this field code is eqivalent to ETH_TYPE_LENGTH. In VLCPDUs, this field does not exist.

FIELD_CODE	Numeric Code	Field size (bits)	Description
XPDU_VLAN0	0x24	32	*xPDU Outermost VLAN tag*. In xPDUs (non-VLC types), this field code is eqivalent to VLAN0. In VLCPDUs, this field does not exist.
XPDU_VLAN1	0x25	32	*xPDU Innermost VLAN tag*. In xPDUs (non-VLC types), this field code is eqivalent to VLAN1. In VLCPDUs, this field does not exist.
XPDU_SUBTYPE	0x26	8	*XPDU Subtype field*. This field may not exist in all xPDU types. Where it exists, it is located immediately after the *XPDU_ETH_TYPE* field. An example of this field, is the *Subtype* field in OAMPDU (see IEEE Std 802.3, 57.4.2).

6.1.1.2 CTE rule modification 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.

Deleted: Table 6-3

Table 6-3—Actions used in CTE rules

Action	Numeric Code	Mnemonic / Description
Add a field	0xAD	`ADD(TARGET_FIELD_CODE, field_value)` This operation adds a field of the type indicated by the `TARGET_FIELD_CODE` and having the value of `field_value`.
Remove (delete) a field	0xDE	`REMOVE(TARGET_FIELD_CODE)` This operation removes a field of the type indicated by the `TARGET_FIELD_CODE`. The result of the `REMOVE` operation is undefined if the field indicated by the `TARGET_FIELD_CODE` is not present in the frame.
Replace (change) a field	0xCE	`REPLACE(TARGET_FIELD_CODE, field_value)` This operation replaces the value of the field indicated by the `TARGET_FIELD_CODE` with the value of `field_value`. The result of the `REPLACE` operation is undefined if the field indicated by the `TARGET_FIELD_CODE` is not present in the frame.
Copy (duplicate) a field	0xD8	`COPY(TARGET_FIELD_CODE, SOURCE_FIELD_CODE)` This operation adds a field of the type indicated by the `TARGET_FIELD_CODE` with the value of the field indicated by the `SOURCE_FIELD_CODE`. The result of the `COPY` operation is undefined if the field indicated by the `TARGET_FIELD_CODE` is already present in the frame or if the field indicated by the `SOURCE_FIELD_CODE` is not present in the frame. The result is also undefined if the fields identified by the `TARGET_FIELD_CODE` and `SOURCE_FIELD_CODE` are not of the same size.

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

Deleted: s

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

Only REPLACE 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:

`REMOVE(VLAN0)`
`REMOVE(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:

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

`REMOVE(VLAN1)` – delete innermost tag first
`REMOVE(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 VLC 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 a VLCPDU into any other PDU (xPDU) is called a *tunnel exit rule* and a rule that converts xPDU into a VLCPDU 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 VLC-aware and where the bidirectional VLC 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 VLC-aware. The L2 Switch treats VLCPDUs 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.

Deleted: Figure 6-2

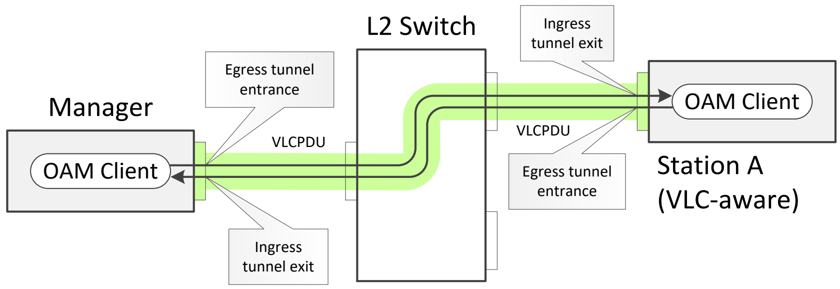


Figure 6-2—Network segment with VLC-aware station A

Figure 6-3 illustrates a network segment where the Manager is VLC-aware, but the managed station B is not. In this scenario, the intermediate switch (L2 Switch) is required to be VLC-aware in order to convert VLCPDUs 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.

Deleted: Figure 6-3

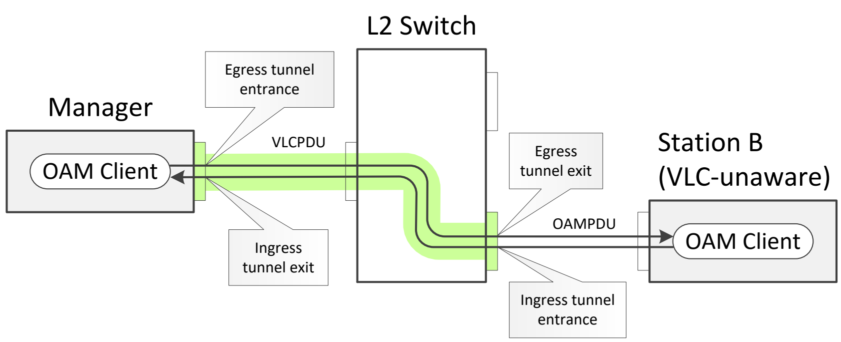


Figure 6-3—Network segment with VLC-unaware station B

6.1.3 CTE rules involving operations on the VLAN tags

The classification clauses in the CTE rules may classify the incoming xPDUs and VLCPDUs based on *VLAN0* or *VLAN1* fields or based on some sub-fields of these fields (see Table 6-2).

Deleted: fields, or

Deleted: Table 6-2

The action clauses in the CTE rules may add VLAN0 and VLAN1 tags to VLCPDUs or delete these tags from VLCPDUs. When performing a translation of an xPDU into a VLCPDU, and if the original xPDU includes any VLAN tags, the action clauses may also copy these tags from xPDU into VLCPDU. The COPY operation leaves the VLAN tags in the original xPDU intact.

Even though the VLC sublayer may be configured to manipulate VLAN tags in VLCPDUs, it does not imply that a given VLC-aware device is also VLAN-aware and that it is a participant in Multiple VLAN Registration Protocol (MVRP). The VLAN manipulation applied by the VLC sublayer is entirely based on the provisioned CTE rules and not on any higher-layer protocol behavior or device configuration. In a VLAN-enabled L2 network, the management entity responsible for VLC port configuration and provisioning is expected to be aware of VLAN topology and to participate in MVRP if necessary.

6.2 Receive path specification

6.2.1 Principles of operation

The receive path of the VLC sublayer includes the Receive process. The Receive process waits for a frame to be received on MACCSI:MA_DATA interface (via `MACCSI:MA_DATA.indication()` primitive, as defined in 4.3.1.x). When a frame is received, it is processed by the ingress Classification and Translation Engine (CTE) and if a match is found, the frame is modified according to the matched rule's action. If the frame does not match any rules, it is passed through the CTE block unmodified.

After traversing the ingress CTE block (highlighted in Figure 6-4), the frame is dispatched to one of the VLCSI interfaces: (VLCSI:VLCPDU, VLCSI:OMCI, or VLCSI:MA_DATA). The dispatching decision is based on the values of the MAC destination address, Ethertype, and VLC subtype.

Deleted: Figure 6-4

VLCPDUs with the destination address matching the local MAC address and the VLC subtype equal to `VLC_SUBTYPE` (see Table 5-1) are modified to match the parameters expected by the `VLCSI:VLCPDU.indication()` primitive (see 4.3.1.x) and are passed to the VLCSI:VLCPDU interface.

Deleted: The

Deleted: Table 5-1

VLCPDUs with the destination address matching the local MAC address and the VLC subtype equal to `OAM_SUBTYPE` (see Table 5-1) are converted into OAMPDUs and are passed to the VLCSI:MA_DATA interface.

Deleted: The

Deleted: Table 5-1

VLCPDUs with the destination address matching the local MAC address and the VLC subtype equal to `OMCI_SUBTYPE` (see Table 5-1) are modified to match the parameters expected by the `VLCSI:OMCI.indication()` primitive (see 4.3.1.4.2) and are passed to the VLCSI:OMCI interface.

Deleted: The

Deleted: Table 5-1

All other xPDUs are passed unmodified to the VLCSI:MA_DATA interface. Note that there still may be other local clients that will intercept/consume these xPDUs at a higher layer.

The Receive process does not discard any frames, i.e., every `MACCSI:MA_DATA.indication()` primitive results in a generation of a single indication primitive on either VLCSI:VLCPDU, VLCSI:OMCI, or VLCSI:MA_DATA interface.

Note that no provisioning of the ingress tunnel exit rules is required in situations where the tunnel is terminated at the same port where the xPDUs are to be consumed by their respective clients. The functionality to convert VLCPDUs into xPDUs is built-in into the Receive process.

6.2.2 Constants

`DST_ADDR`

This constant identifies a field in a frame, as defined in Table 6-2.

Deleted: Table 6-2

`ETH_TYPE_LEN`

This constant identifies a field in a frame, as defined in Table 6-2.

Deleted: Table 6-2

`LOCAL_MAC_ADDR`

TYPE: 48-bit MAC address

This constant holds the value of the MAC address associated with the port where the Receive process state diagram is instantiated. Some devices may associate the same MAC address value with multiple ports. The format of the MAC address is defined in IEEE Std 802.3, 3.2.3.

VALUE: device-specific

`OMCI_SUBTYPE`

This constant represents a VLCPDU subtype as defined in Table 5-1.

Deleted: Table 5-1

`SP_ADDR`

This constant holds the value of the destination MAC address associated with Slow Protocols (see IEEE Std 802.3, 57A.3).

`SP_TYPE`

This constant holds the value of the Ethertype identifying the Slow Protocol (see IEEE Std 802.3, 57A.4).

`SRC_ADDR`

This constant identifies a field in a frame, as defined in Table 6-2.

Deleted: Table 6-2

`SUBTYPE`

This constant identifies a field in a frame, as defined in Table 6-2.

Deleted: Table 6-2

`VLC_ETHERTYPE`

TYPE: 16-bit Ethertype

This constant holds the Ethertype value identifying the VLCPDUs.

VALUE: 0xA8-C8

`VLC_SUBTYPE`

This constant represents a VLCPDU subtype as defined in Table 5-1.

Deleted: Table 5-1

6.2.3 Variables

`IngressRuleId`

TYPE: 16-bit unsigned integer

This variable identifies one of the provisioned CTE ingress rules. It also may have a special value, `none`, that does not identify any of the provisioned rules.

`RxInputPdu`

TYPE: structure containing an Ethernet frame

This variable holds an Ethernet frame received from the MACCSI:MA_DATA interface. The fields of this structure correspond to the parameters of the `MA_DATA.indication()` primitive, as defined in IEEE Std 802.3, 2.3.2.

`RxOutputPdu`

TYPE: structure containing an Ethernet frame

This variable holds an Ethernet frame to be passed to one of the the VLCSI interfaces (VLCSI:VLCPDU, VLCSI:OMCI, or VLCSI:MA_DATA). The fields of this structure correspond to the parameters of the `MA_DATA.indication()` primitive, as defined in IEEE Std 802.3, 2.3.2.

Additionally, the `RxOutputPdu` structure supports the `RemoveField(field_code)` method, which removes a field identified by the `field_code` from the structure. Thus, unlike the `RxInputPdu` structure, the `RxOutputPdu` may contain only a partial Ethernet frame. The `field_code` parameter takes values as defined in Table 6-2.

Deleted: Table 6-2

6.2.4 Functions

`CheckIngressRules(input_pdu)`

This function returns the identification of an ingress rule that matched the frame contained in the `RxInputPdu` structure. If multiple rules match the frame, the function returns an identification of any of these rules. If none of the rules match the frame, a special value, `none`, is returned.

Deleted: ct
Deleted: ed
Deleted: ed

`Modify(rule_id, input_pdu)`

This function returns a frame that is a result of applying the modification action(s) of the rule identified by the `rule_id` parameter to the frame contained in the `input_pdu` parameter.

Deleted: s

6.2.5 Primitives

The primitives referenced in this state diagram are defined in 4.3.1.

6.2.6 State Diagram

The VLC sublayer shall implement the Receive process as defined in the state diagram in Figure 6-4.

Deleted: Figure 6-4

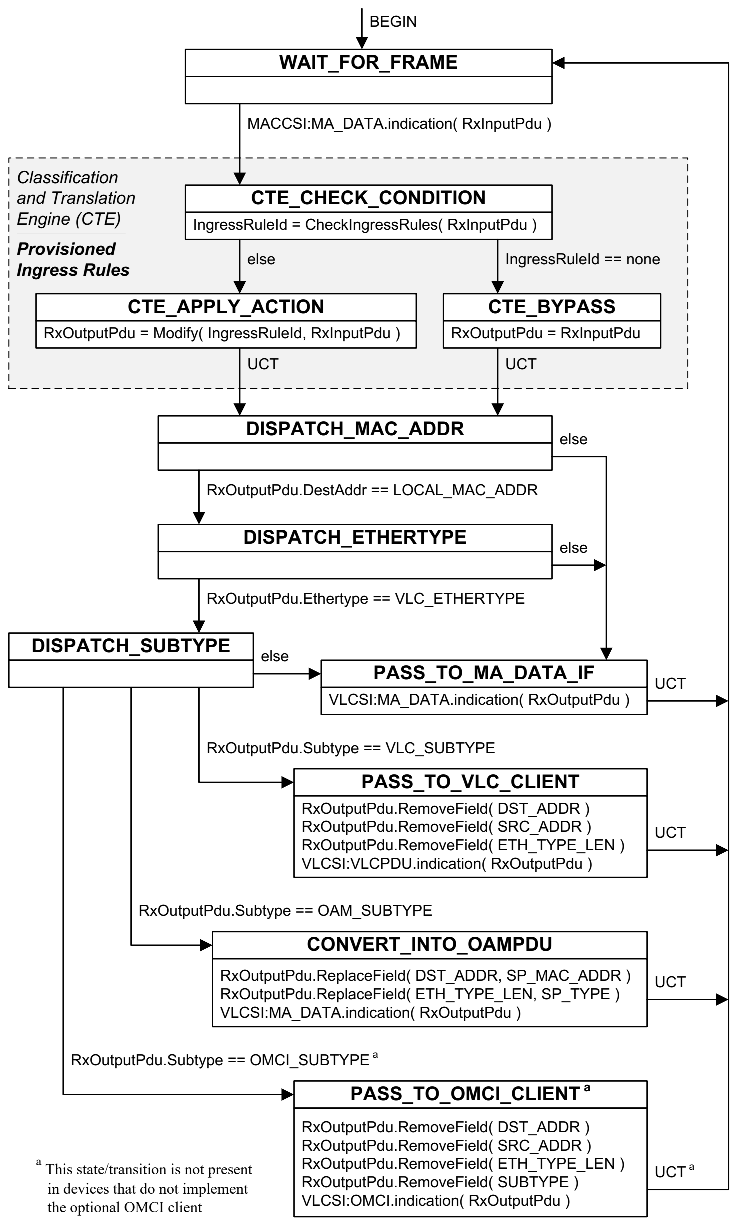


Figure 6-4—Receive process state diagram

6.3 Transmit path specification

6.3.1 Principles of operation

The transmit path of the VLC sublayer includes the Transmit process. The Transmit process waits for an xPDU to be received from one of the VLCSI interfaces: (VLCSI:MA_DATA, VLCSI:VLCPDU, or VLCSI:OMCI).

If a VLC xPDU is received from the VLCSI:VLCPDU interface, it is converted into a VLCPDU with subtype `VLC_CONFIG` (see Table 5-1) by prepeding a VLCPDU header to the VLC xPDU payload. The header cosnsists of the destination address, source address, and Ethertype fields. Note that both the destination and the source addresses are equal to the local MAC address assigned to the given port.

Deleted: n

Deleted: Table 5-1

If an OMCI xPDU is received from the VLCSI:OMCI interface, it is converted into VLCPDU with subtype `OMCI_SUBTYPE` (see Table 5-1) by prepeding a VLCPDU header to the VLC xPDU payload. The header cosnsists of the destination address, source address, Ethertype, and subtype fields. Note that both the destination and the source addresses are equal to the local MAC address assigned to the given port.

Deleted: Table 5-1

After the above modifications, the VLC or OMCI xPDU is formed into a complete frame, which is then processed by the Egress Classification and Translation Engine (CTE). If a match is found, the frame is modified according to the matched rule action. If the frame does not match any rules, it is passed through the CTE block unmodified.

Note that to enter a tunnel, the VLC xPDU or the OMCI xPDU require a matching egress CTE rule that, at a minimum, overwrites the local MAC address value in the VLCPDU destination address field with the MAC address associated with the xPDU destination for the given tunnel.

Deleted: s

6.3.2 Constants

The constants referenced in this state diagram are defined in 6.2.2.

6.3.3 Variables

`EgressRuleId`

TYPE: 16-bit unsigned integer

This variable identifies one of the provisioned CTE egress rules. It also may have a special value `none` that does not identify any of the provisioned rules.

`TxInputPdu`

TYPE: structure containing an Ethernet frame

This variable holds a PDU received from one of the the VLCSI interfaces (VLCSI:VLCPDU, VLCSI:OMCI, or VLCSI:MA_DATA). When received from the VLCSI:MA_DATA interface, the `TxInputPdu` structure contains a complete and properly-formed Ethernet frame. When received from VLCSI:VLCPDU or VLCSI:OMCI interfaces, the `TxInputPdu` structure contains a partial frame, that only includes the parameters defined for the respective `request()` primitive (see 4.4).

Additionally, the `TxInputPdu` structure supports the `AddField(field_code, field_value)` method, which adds a field identified by the `field_code` and having the value `field_value` to the structure. The `field_code` parameter takes values as defined in Table 6-2.

Deleted: Table 6-2

`TxOutputPdu`

TYPE: structure containing an Ethernet frame

This variable holds an Ethernet frame to be passed to the MACCSI:MA_DATA interface. The fields of this structure correspond to the parameters of the `MA_DATA.request()` primitive, as defined in IEEE Std 802.3, 2.3.1. A CTE egress rule is considered misconfigured if applying this rule to the `TxInputPdu` results in a malformed Ethernet frame being stored in the `TxOutputPdu` structure.

6.3.4 Functions

`CheckEgressRules(input_pdu)`

This function returns the identification of an egress rule that matched the the frame contained in `TxInputPdu` structure. If multiple rules match the frame, the function returns an identification of any of these rules. If none of the rules matched the frame, a special value `none` is returned.

`Modify(rule_id, input_pdu)`

This functions is defined in 6.2.4.

6.3.5 Primitives

The primitives referenced in this state diagram are defined in 4.3.1.

6.3.6 State Diagram

The VLC sublayer shall implement the Transmit process as defined in the state diagram in Figure 6-5.

Deleted: c

Deleted: ed

Deleted: Figure 6-5

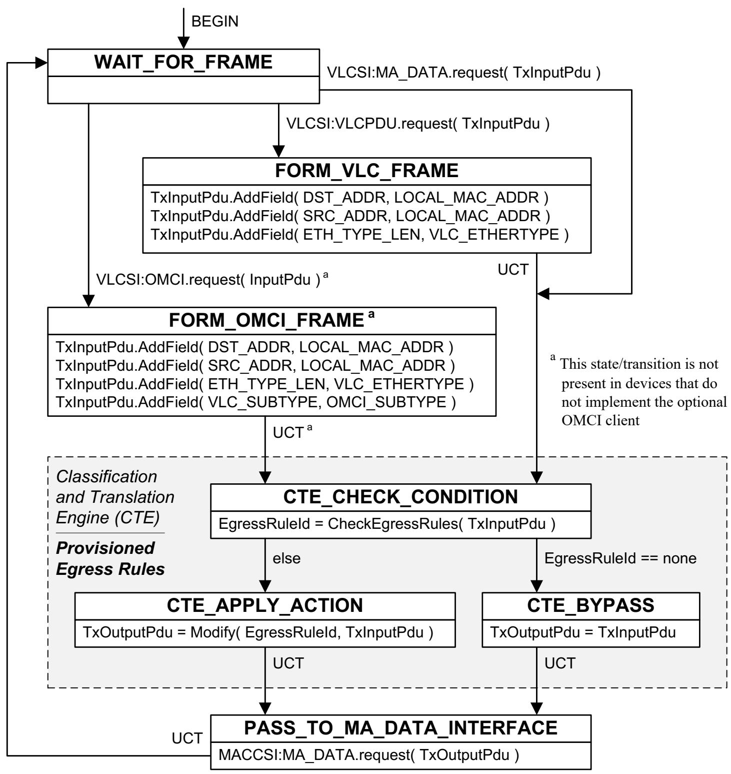


Figure 6-5—Transmit process state diagram

7 Protocol-Specific behavior

7.1 <TBD>

7.2 Support for OAM remote loopback

7.2.1 Overview

OAM defined in IEEE Std 802.3, 57.2.11 provides an optional data link layer frame-level loopback mode, which can be used for fault localization and link performance testing.

The OAM entity that initiates the loopback mode is called the *local* OAM entity. The OAM entity on the opposite end of a link is called the *remote* OAM entity. In the OAM remote loopback mode, the local and remote OAM entities operate as follows:

a) The local OAM entity transmits frames from the MAC client and OAMPDUs from the local OAM client or OAM sublayer.
b) Within the OAM sublayer of the remote OAM entity, every received OAMPDU is passed to the OAM client, while non-OAMPDUs, including other Slow Protocol frames, are looped back without altering any field of the frame.
c) Frames received by the local OAM entity are parsed by the OAM sublayer. OAMPDUs are passed to the OAM client and all other frames are discarded.

Both OAM entities continue exchanging OAMPDUs in order to keep the OAM discovery process from restarting and to perform other management tasks.

7.2.2 OAM loopback over VLC tunnel

When the OAM loopback is initiated over a VLC tunnel, the behavior of the local and remote OAM entities remains as it is described in 7.2.1. Specifically, the remote OAM sublayer loops back all non-OAMPDUs (i.e., generates an *MA_DATA.request()* primitive in response to every *MA_DATA.indication()* primitive that does not contain an OAMPDU). The local OAM sublayer discards all received non-OAMPDU frames.

However, to ensure that the non-OAMPDUs transmitted by the local MAC client are delivered to the remote OAM sublayer, an additional VLC tunnel needs to be established from the local DTE to the remote DTE. Similarly, to deliver the looped-back frames from the remote DTE back to the local DTE, a VLC tunnel operating in the opposite direction also needs to be established.

Since the OAM is a link-level protocol (i.e., operates over a single-span link), either a DTE itself or a bridge immediately adjacent to that DTE must be VLC-aware. A network configuration with both the local and the remote DTE being VLC-unaware is illustrated in Figure 7-1.

Deleted: Figure 7-1

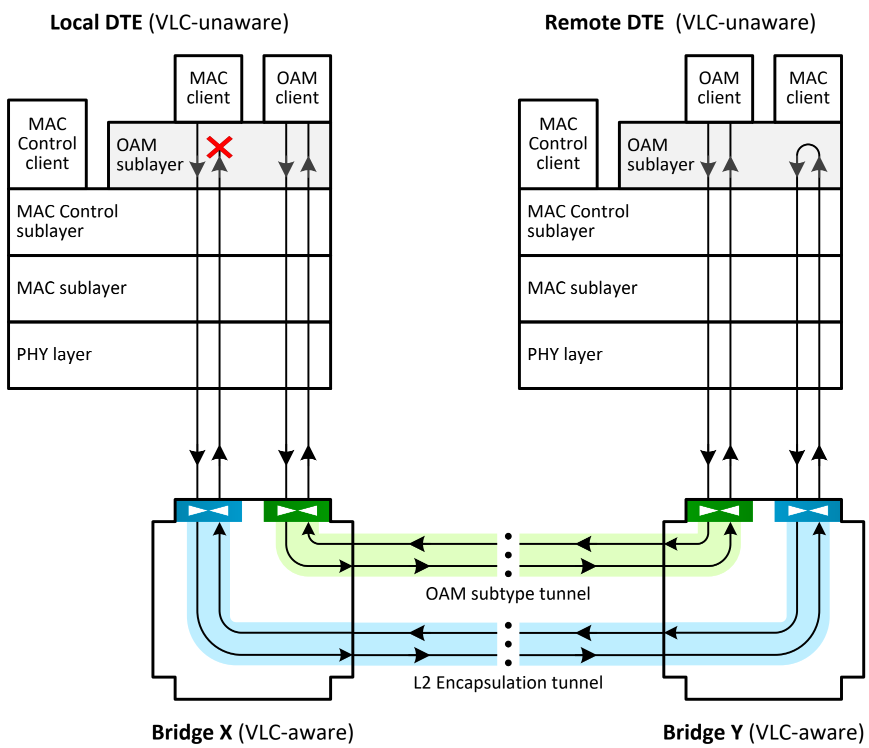


Figure 7-1—Remote OAM loopback over VLC tunnel with VLC-unaware local DTE and VLC-unaware remote DTE.

The remote OAM loopback can also be established when one of the DTEs is VLC-aware and the other is not. Figure 7-2 illustrates a network configuration with the local DTE being VLC-aware and the remote DTE being VLC-unaware.

Deleted: Figure 7-2

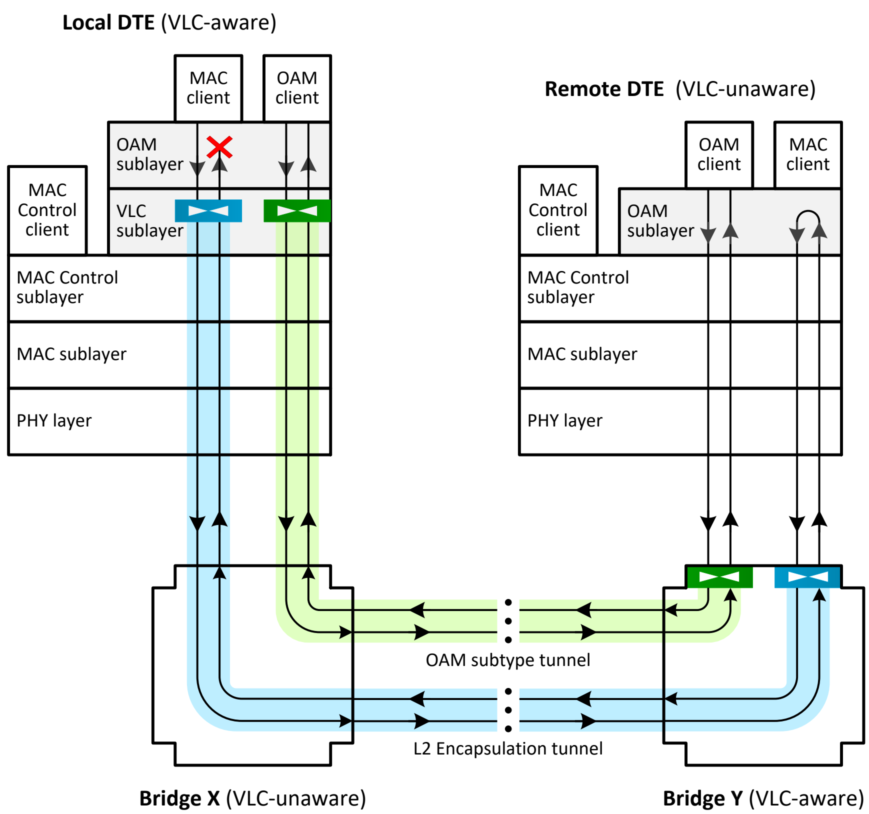


Figure 7-2—Remote OAM loopback over VLC tunnel with VLC-aware local DTE and VLC-unaware remote DTE.

Figure 7-3 represents a similar network configuration, but with both the local and the remote DTEs being VLC-aware.

Deleted: Figure 7-3

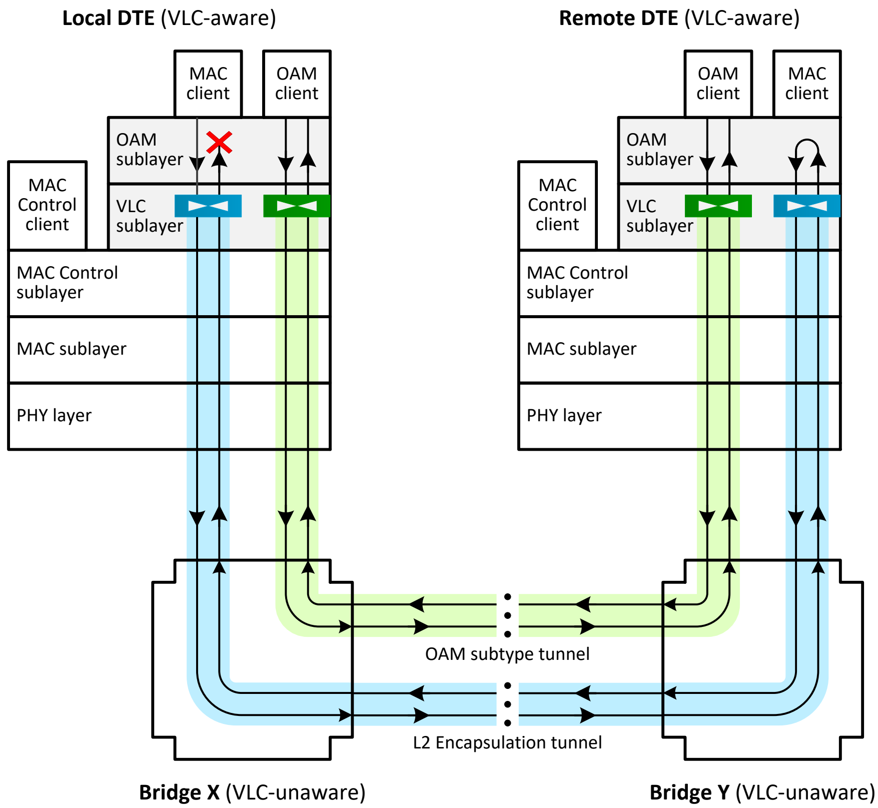


Figure 7-3—Remote OAM loopback over VLC tunnel with VLC-aware local DTE and VLC-aware remote DTE.

While the OAM subtype tunnel between the local and remote DTEs persists permanently to ensure that the two OAM entities are able to exchange OAMPDUs, the bidirectional tunnel for the looped-back data only needs to be established for the duration of the loopback mode. This tunnel has L2 encapsulation subtype in order to deliver any non-OAMPDU (regardless of their Source and destination MAC addresses) to from the local DTE to the remote DTE and in the reverse direction, from the remote DTE to the local DTE.

Table 7-1 illustrates the tunnel entrance rules for the VLC L2 encapsulation tunnel from the local DTE to the remote DTE. The table shows two rules that have different conditions, but identical actions. If these rules are provisioned in the bridge adjacent to the local DTE, as illustrated in Figure 7-1, these are ingress tunnel entrance rules. If the rules are provisioned in the local DTE itself, as illustrated in Figure 7-2 and Figure 7-3, these are egress tunnel entrance rules.

Deleted: Table 7-1

Deleted: Figure 7-1

Deleted: Figure 7-2

Deleted: Figure 7-3

Table 7-1—Tunnel entrance rule for non-OAMPDU traffic from local DTE to remote DTE

<table>
<tr><th>Conditions</th><th>Actions</th></tr>
<tr><td>1. ETYPE_LEN != SP_TYPE</td><td rowspan="2">1. ADD(VLC_DST_ADD, <remote_MAC>)
2. ADD(VLC_SRC_ADD, <local_MAC>)
3. ADD(VLC_ETH_TYPE, VLC_TYPE)
4. ADD(VLC_SUBTYPE, L2_subtype)</td></tr>
<tr><td>1. ETYPE_LEN == SP_TYPE
2. XPDU_SUBTYPE != OAM_subtype</td></tr>
<tr><td colspan="2">NOTE:
<local_MAC > - MAC address associated with the loopback port in the local DTE
<remote_MAC > - MAC address associated with the loopback port in the remote DTE

SP_TYPE – Slow Protocol Ethertype value (see IEEE Std 802.3, 57A.4)
VLC_TYPE – Ethertype value identifying VLCPDUs (see 5.1)

OAM_subtype – VLC subtype value identifying OAMPDU payload (see 5.2)
L2_subtype – VLC subtype value identifying L2 encapsulation payload (see 5.2)</td></tr>
</table>

Table 7-2 illustrates the tunnel exit rule for the VLC L2 encapsulation tunnel from the local DTE to the remote DTE. If this rule is provisioned in the bridge adjacent to the remote DTE, as illustrated in Figure 7-1 and Figure 7-2, this rule is an egress tunnel exit rule. If the rule is provisioned in the remote DTE itself, as illustrated in Figure 7-3, this rule is an ingress tunnel exit rule.

Table 7-2—Tunnel exit rule for non-OAMPDU traffic from local DTE to remote DTE

<table>
<tr><th>Conditions</th><th>Actions</th></tr>
<tr><td>1. DST_ADDR == <remote_MAC>
2. SRC_ADDR == <local_MAC>
3. ETH_TYPE == VLC_TYPE
4. VLC_SUBTYPE == L2_subtype</td><td>1. REMOVE(VLC_DST_ADDR)
2. REMOVE(VLC_SRC_ADDR)
3. REMOVE(VLC_ETH_TYPE)
4. REMOVE(VLC_SUBTYPE)</td></tr>
<tr><td colspan="2">NOTE:
<local_MAC > - MAC address associated with the loopback port in the local DTE
<remote_MAC > - MAC address associated with the loopback port in the remote DTE

VLC_TYPE – Ethertype value identifying VLCPDUs (see 5.1)
L2_subtype – VLC subtype value identifying L2 encapsulation payload (see 5.2)</td></tr>
</table>

The entrance rules for the return tunnel (from the remote DTE back to the local DTE), the rules are similar to the rules shown in Table 6-8, but with *<local_MAC>* and *<remote_MAC>* values swapped. Similarly, the tunnel exit rule is as shown in Table 6-9, but also with *<local_MAC>* and *<remote_MAC>* values swapped.

Deleted: Table 7-2
Deleted: Figure 7-1
Deleted: Figure 7-2
Deleted: Figure 7-3

8 VLC Management

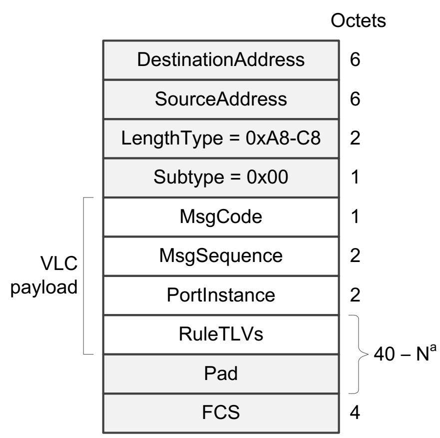
8.1 VLC Configuration

The tunnels originate and terminate in the VLC-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 *VLC_CONFIG* VLCPDUs, which carry a set of *condition-encoding* TLVs and a set of *action-encoding* TLVs.

8.1.1 Configuration VLCPDU

The *VLC_CONFIG* UMPTPDU format shall be as depicted in Figure 8-1. The *VLC_CONFIG* VLCPDU is used as both a request to configure a CTE rule as well as a response containing the result of the configuration request.

Deleted: Figure 8-1



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

Figure 8-1—*VLC_CONFIG* VLCPDU format

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

Deleted: Figure 5-1

—*MsgCode*:

The *MsgCode* field identifies whether the VLC_CONFIG message is a request message or a response. If the VLCPDU is a request, this field encodes the requested action. If the VLCPDU 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 8-1.

Deleted: Table 8-1

Table 8-1—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 VLC configuration request or a response consists of multiple messages, this field identifies the message sequence number. The format of the *MsgSequence* field is shown in Table 8-2.

Deleted: Table 8-2

Table 8-2—Format of the *MsgSequence* field

Bits	Field name	Value	Description
14:0	*MsgCounter*	0x00-01 to 0x7F-FF	A counter that increments by one for each message in a sequence. In the first message in a sequence, the *MsgCounter* is equal to 1.
15	*EndOfSequence*	0	This message is not the last message in a sequence
		1	This message is the last message in a sequence

When a request or a response consists of a single VLCPDU, the *MsgCounter* subfield is equal to 0x00-01 and the *EndOfSequence* flag is equal to 1.

Note that even when a VLC configuration request or a response consists of multiple messages, a single rule is not split across multiple messages and as such – no reassembly mechanism is necessary to reconstruct any rule. An example scenario where the response consists of multiple messages would be a VLC configuration response to a 'Query all rules' request, where multiple rules are being reported.

—*PortInstance*:

This field identifies a port instance in the VLC-aware device to which the given *VLC_CONFIG* VLCPDU applies. The format of the *PortInstance* field is shown in Table 8-3.

Deleted: Table 8-3

Table 8-3—Format of the *PortInstance* field

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

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

—*RuleTLVs*:

This field includes one or more CTE rule TLV(s) as defined in 8.1.2. The combined size of the *RuleTLV* and *Pad* fields ranges between 40 and *N*, where *N* is defined in Figure 5-1.

Deleted: Figure 5-1

8.1.2 CTE rule TLV structure

The structure of a CTE rule TLV is shown in Table 8-4. Each *VLC_CONFIG* VLCPDU shall contain at least one CTE rule TLV.

Deleted: Table 8-4

Table 8-4—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 *VLC_CONFIG* VLCPDU and it may be the only TLV in the *VLC_CONFIG* VLCPDU.
1	*Length*	*V*+*M*+4	The *Length* field encompasses the entire TLV, including the *Type* and *Length* fields. A TLV with length of 0x00 through 0x03 is invalid.
1	*Operation*[a]	per Table 6-1	Comparison operator code, if the TLV *Type* = 0xC0
		per Table 6-3	Action code, if the TLV *Type* = 0xAC
V	*FieldCode*[a]	per Table 6-2	Identifies a field to be used in a comparison, or to be modified by an action.
L	*Value*	Various	The value to be used in a comparison or by an Add/Change action. Some TLVs may omit this field.
M[b]	*Mask*	various	The mask pattern to be used in a comparison condition. The mask pattern is applied as a bitwise-AND operation to both the value to be used in a comparison (see the *Value* field above) as well the value of the field identified by the *FieldCode* parameter of this TLV. Some TLVs may omit this field[c]. When *Mask* is omitted, the comparison applies to the entire field.

Deleted: Table 6-1

Deleted: Table 6-3

Deleted: Table 6-2

[a] Fields *Operation* and *FieldCode* shall be present in all TLVs, even if they are not used. When these fields are not used, they should be set to the value of zero.

[b] The length *M* of *Mask* field shall be the same as the length of *Value* field, if mask field is present. Otherwise, the length *M* is considered to be equal to zero.

[c] If a CTE rule TLV omits the *Value* field, the *Mask* field shall also be omitted.

8.2 Management Attributes

<TBD>

9 Protocol implementation conformance statement (PICS) proforma for Virtual Link Control (VLC) specification

9.1 Introduction

This subclause specifies the PICS proforma for Virtual Link Control (VLC).

The supplier of an VLC 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.

9.2 Implementation identification

VLC 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]	
1. NOTE 1—Required for all implementations. 2. 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).	

9.3 Protocol summary

Identification of the VLC 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 VLC 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	

9.4 VLCPDU encoding

Item	Description	Subclause	Value/Comment	Status	Support
PDU01	*Subtype* field encoding	5.2	Per Table 5-1	M	
PDU02	VLCPDU with OAM subtype	5.2.2	Structure per Figure 5-2	M	
PDU03	VLCPDU with L2 subtype	5.2.4	Structure per Figure 5-4	M	
PDU04	VLCPDU with L3 subtype	5.2.5	Structure per Figure 5-5	M	
PDU05	VLCPDU with organization-specific extension subtype	5.2.6	Structure per Figure 5-6(a) for Organization-Specific VLCPDU with *OUI24_Subtype* and Figure 5-6(b) for Organization-Specific VLCPDU with *OUI36_Subtype*	M	
PDU06	*VLC_CONFIG* VLCPDU structure	8.1.1	Structure per Figure 8-1	M	
PDU07a	*VLC_CONFIG* VLCPDU TLV content	8.1.2	Each *VLC_CONFIG* VLCPDU contains at least one CTE rule TLV	M	
PDU07b	TLV with Type = 0x00 positioning	8.1.2	The TLV with Type = 0x00 is the last TLV in every *VLC_CONFIG* VLCPDU	M	
PDU07c	Presence of Fields *Operation* and *FieldCode*	8.1.2	Present in all TLVs, even if they are not used	M	
PDU07d	Value of Fields *Operation* and *FieldCode*	8.1.2	When not used, these fields are set to zero	O	
PDU07e	The length *M* of *Mask* field	8.1.2	The same as the length of *Value* field, if mask field is present	M	
PDU07f	Presence of the *Mask* field	8.1.2	If a CTE rule TLV omits the *Value* field, the *Mask* field is omitted	M	

Deleted: Table 5-1
Deleted: Figure 5-2
Deleted: Figure 5-4
Deleted: Figure 5-5
Deleted: Figure 5-6
Deleted: Figure 5-6
Deleted: Figure 8-1

9.5 CTE

Item	Description	Subclause	Value/Comment	Status	Support
CTE01	Actions on `SRC_ADDR` field	6.1.1.2	No modification to `SRC_ADDR` field is allowed	M	

Annex 8A
(informative)
VLC configuration examples

8A.1 OAM over VLC use case, VLC-unaware end points

8A.1.1 Introduction

This example illustrates OAM communication between a Manager M and a Station S carried over VLC that traverses multiple L2 bridges (see Figure 8A-1). Both the Manager and the Station are VLC-unaware. The bridge X nearest to the Manager M is VLC-aware, and so is the bridge Y nearest to the Station S. There can be numerous other bridges between the bridges X and Y; those bridges may or may be not VLC-aware.

Deleted: Figure 8A-1

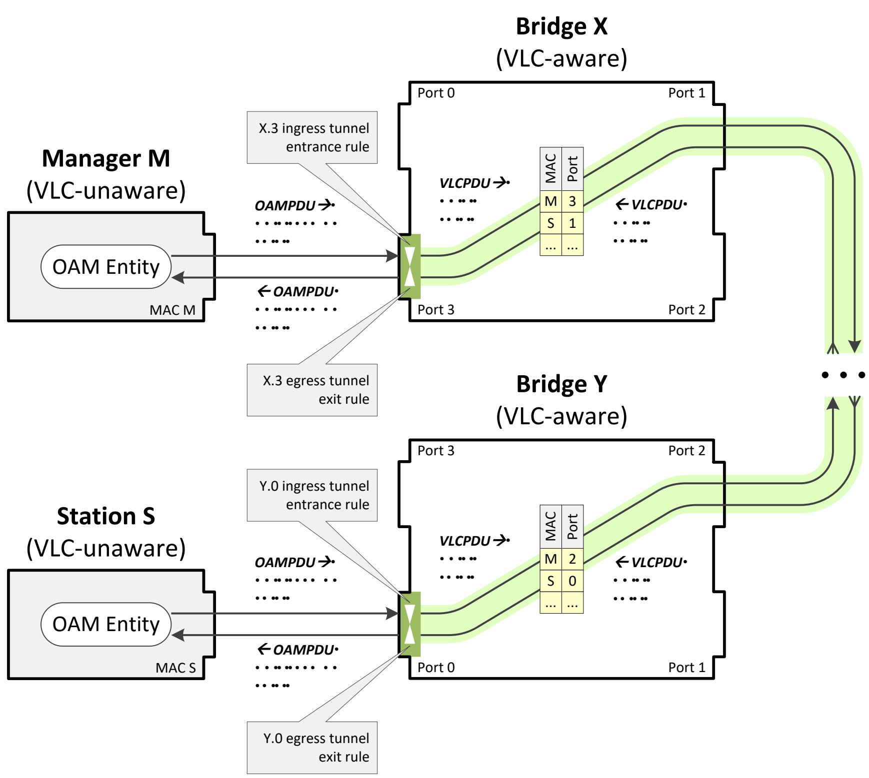


Figure 8A-1—OAM over VLC use case, VLC-unaware end points

In Figure 8A-1, the Manager M, station S, Bridges X and Y have MAC addresses M, S, X, and Y respectively. For simplicity, it is assumed that all ports in a given device use the same MAC address, but this is not a requirement.

Deleted: Figure 8A-1

Furthermore, it is assumed that Bridges X and Y, as well as all intermediate bridges, have already populated their forwarding tables with entries for MAC addresses M and S. These entries may be created dynamically by a MAC learning function or be provisioned statically by the NMS.

8A.1.2 VLC provisioning to establish tunnels

Since the Manager M is not directly connected to the managed Station S, the OAM messages need to be carried over VLCPDUs. Therefore, before the Manager M and the Station S are able to exchange OAM messages, two VLC tunnels need to be provisioned:

— A forward VLC tunnel from bridge X, port 3 to bridge Y, port 0.

— A reverse VLC tunnel from bridge Y, port 0 to bridge X, port 3.

The establishement of each VLC tunnel involves provisioning of two rules - one to configure the VLC tunnel entrance point and one to configure the VLC tunnel exit point.

To establish a VLC tunnel from Manager M to Station S, the following rules are provisioned:

— A VLC tunnel entrance rule at the ingress of Bridge X, port 3

— A VLC tunnel exit rule at the egress of Bridge Y, port 0

To establish a VLC tunnel from Station S to Manager M, the following rules are provisioned:

— A VLC tunnel entrance rule at the ingress of Bridge Y, port 0

— A VLC tunnel exit rule at the egress of Bridge X, port 3

Each rule is provisioned using a separate *VLC_CONFIG* message. The contents of all four messages required to establish two VLC tunnles for bidirectional communication for the network segment illustrated in Figure 8A-1 are shown below.

Deleted: Figure 8A-1

8A.1.2.1 Addition of tunnel entrance rule at the ingress of Bridge X, port 3

The VLC tunnel entrance rule at the ingress of Bridge X, port 3 is shown in Table 8A-1. This rule converts an OAMPDU into a VLCPDU in the receive path of port 3. The conversion replaces the destination MAC address value (*SP_DA*) with the MAC address of Station S and replaces the Slow Protocol Ethertype (`SP_type`) with the VLC Ethertype (`VLC_type`).

Table 8A-1—Tunnel entrance rule at the ingress of Bridge X, port 3

Conditions	Actions
`1. DA == SP_DA` `2. ETH_TYPE_LEN == SP_type` `3. SP_SUBTYPE == OAM_subtype`	`1. REPLACE( DA, S )` `2. REPLACE( ETH_TYPE_LEN, VLC_type )`

NOTE:

`SP_type` – Slow Protocol Ethertype value (see IEEE Std 802.3, 57A.4)

`VLC_type` – Ethertype value identifying VLCPDUs (see 5.1)

`OAM_subtype` – Subtype value identifying OAMPDUs (see IEEE Std 802.3, 57A.4)

`SP_DA` – Destination MAC address associated with Slow Protocols (see IEEE Std 802.3, 57A.3)

`S` – MAC address of Station S.

Table 8A-2 provides the contents of a *VLC_CONFIG* VLCPDU that provisions the rule shown in Table 8A-1.

Table 8A-2—Contents of *VLC_CONFIG* message

Field	Subfield	Value	Description
DestinationAddress	n/a	X	*VLC_CONFIG* VLCPDU directed to bridge X
SourceAddress	n/a	any	Source address of a device that issued the *VLC_CONFIG* VLCPDU
LengthType	n/a	0xA8-C8	Ethertype value identifying VLCPDUs (see 5.1)
Subtype	n/a	0x00	VLCPDU carrying *VLC_CONFIG* message
MsgCode	*MsgType*	0x0	This message is a Request (see Table 8-1)
	RequestCode	0x1	Request to add a rule (see Table 8-1)
MsgSequence	*MsgCounter*	0x00-01	This request consists of a single message
	EndOfSequence	1	
PortInstance	*PortIndex*	3	The rule is to be provisioned for port #3
	Direction	1	The rule is to be provisioned for the receive path (i.e., an ingress rule)
RuleTLV (condition)	*Type*	0xCO	This is a condition TLV (see Table 8-4)
	Length	0x0A	TLV length is 10 octets
	Operation	0x11	Comparison for equality (see Table 6-1)
	FieldCode	0x01	Compare *DST_ADDR* field (see Table 6-2)
	Value	0x01-80-C2-00-00-02	IEEE 802.3 Slow_Protocols_Multicast address (see IEEE Std 802.3, 57A.3)
RuleTLV (condition)	*Type*	0xCO	This is a condition TLV (see Table 8-4)
	Length	0x06	TLV length is 6 octets
	Operation	0x11	Comparison for equality (see Table 6-1)
	FieldCode	0x03	Compare *ETH_TYPE_LEN* field (see Table 6-2)
	Value	0x88-09	Slow Protocol Ethertype value (see IEEE Std 802.3, 57A.4)
RuleTLV (condition)	*Type*	0xCO	This is a condition TLV (see Table 8-4)
	Length	0x05	TLV length is 5 octets
	Operation	0x11	Comparison for equality (see Table 6-1)
	FieldCode	0x26	Compare *XPDU_SUBTYPE* field (see Table 6-2)
	Value	0x03	Slow Protocol Subtype value for OAM (see IEEE Std 802.3, 57A.4)
RuleTLV (action)	*Type*	0xAC	This is an action TLV (see Table 8-4)
	Length	0x0A	TLV length is 10 octets
	Operation	0xCE	Change (replacement) of a field (see Table 6-3)

Deleted: Table 8-1
Deleted: Table 8-1
Deleted: Table 8-4
Deleted: Table 6-1
Deleted: Table 6-2
Deleted: Table 8-4
Deleted: Table 6-1
Deleted: Table 6-2
Deleted: Table 8-4
Deleted: Table 6-1
Deleted: Table 6-2
Deleted: Table 8-4
Deleted: Table 6-3

Field	Subfield	Value	Description
	FieldCode	0x01	Modify *DST_ADDR* field (see Table 6-2)
	Value	S	Set Station S MAC address as the destination for resulting VLCPDUs.
RuleTLV (action)	*Type*	0xAC	This is an action TLV (see Table 8-4)
	Length	0x06	TLV length is 6 octets
	Operation	0xCE	Change (replacement) of a field (see Table 6-3)
	FieldCode	0x03	Modify *ETH_TYPE_LEN* field (see Table 6-2)
	Value	0xA8-C8	Set Ethertype to be equal to VLC_Ethertype in the resulting VLCPDUs.
RuleTLV (termination)	*Type*	0x00	This is a termination (end-of-rule) TLV (see Table 8-4)
	Length	0x04	TLV length is 4 octets
	Operation	0x00	Filled with zeros when not used (see Table 8-4, note)
	FieldCode	0x00	

Deleted: Table 6-2
Deleted: Table 8-4
Deleted: Table 6-3
Deleted: Table 6-2
Deleted: Table 8-4
Deleted: Table 8-4

8A.1.2.2 Addition of tunnel exit rule at the egress of Bridge Y, port 0

The VLC tunnel exit rule at the ingress of Bridge Y, port 0 is shown in Table 8A-3. This rule converts a VLCPDU into an OAMPDU in the transmit path of port 0. The conversion replaces the destination MAC address of Station S with the MAC address used for Slow Protocol xPDUs (`SP_DA`) and replaces the VLC Ethertype (`VLC_type`) with the Slow Protocol Ethertype (`SP_type`).

Table 8A-3—Tunnel exit rule at the egress of Bridge Y, port 0

Conditions	Actions
`1. DA == S` `2. ETH_TYPE_LEN == VLC_type` `3. VLC_SUBTYPE == OAM_Subtype`	`1. REPLACE( DA, SP_DA )` `2. REPLACE( ETH_TYPE_LEN, SP_type )`
NOTE: `SP_type` – Slow Protocol Ethertype value (see IEEE Std 802.3, 57A.4) `VLC_type` – Ethertype value identifying VLCPDUs (see 5.1) `OAM_Subtype` – Subtype value identifying OAM payload (see Table 5-1) `SP_DA` – Destination MAC address associated with Slow Protocols (see IEEE Std 802.3, 57A.3) `S` – MAC address of Station S.	

Deleted: Table 5-1

Table 8A-4 provides the contents of a *VLC_CONFIG* VLCPDU that provisions the rule shown in Table 8A-3.

Table 8A-4—Contents of *VLC_CONFIG* message

Field	Subfield	Value	Description
DestinationAddress	n/a	Y	*VLC_CONFIG* VLCPDU directed to bridge Y
SourceAddress	n/a	any	Source address of a device that issued the *VLC_CONFIG* VLCPDU
LengthType	n/a	0xA8-C8	Ethertype value identifying VLCPDUs (see 5.1)
Subtype	n/a	0x00	VLCPDU carrying *VLC_CONFIG* message
MsgCode	*MsgType*	0x0	This message is a Request (see Table 8-1)
	RequestCode	0x1	Request to add a rule (see Table 8-1)
MsgSequence	*MsgCounter*	0x00-01	This request consists of a single message
	EndOfSequence	1	
PortInstance	*PortIndex*	0	The rule is to be provisioned for port #0
	Direction	0	The rule is to be provisioned for the transmit path (i.e., an egress rule)
RuleTLV (condition)	*Type*	0xCO	This is a condition TLV (see Table 8-4)
	Length	0x0A	TLV length is 10 octets
	Operation	0x11	Comparison for equality (see Table 6-1)
	FieldCode	0x01	Compare *DST_ADDR* field (see Table 6-2)
	Value	S	The dstination address is equal to MAC address of Station S.
RuleTLV (condition)	*Type*	0xCO	This is a condition TLV (see Table 8-4)
	Length	0x06	TLV length is 6 octets
	Operation	0x11	Comparison for equality (see Table 6-1)
	FieldCode	0x03	Compare *ETH_TYPE_LEN* field (see Table 6-2)
	Value	0xA8-C8	VLC Ethertype value (see 5.1)
RuleTLV (condition)	*Type*	0xCO	This is a condition TLV (see Table 8-4)
	Length	0x05	TLV length is 5 octets
	Operation	0x11	Comparison for equality (see Table 6-1)
	FieldCode	0x1A	Compare *VLC_SUBTYPE* field (see Table 6-2)
	Value	0x03	VLC Subtype identifying OAM payload (see Table 5-1)
RuleTLV (action)	*Type*	0xAC	This is an action TLV (see Table 8-4)
	Length	0x0A	TLV length is 10 octets
	Operation	0xCE	Change (replacement) of a field (see Table 6-3)
	FieldCode	0x01	Modify *DST_ADDR* field (see Table 6-2)
	Value	0x01-80-C2-00-00-02	IEEE 802.3 Slow_Protocols_Multicast address (see IEEE Std 802.3, 57A.3)

Deleted: Table 8-1
Deleted: Table 8-1
Deleted: Table 8-4
Deleted: Table 6-1
Deleted: Table 6-2
Deleted: Table 8-4
Deleted: Table 6-1
Deleted: Table 6-2
Deleted: Table 8-4
Deleted: Table 6-1
Deleted: Table 6-2
Deleted: Table 5-1
Deleted: Table 8-4
Deleted: Table 6-3
Deleted: Table 6-2

Field	Subfield	Value	Description
RuleTLV (action)	*Type*	0xAC	This is an action TLV (see Table 8-4)
	Length	0x06	TLV length is 6 octets
	Operation	0xCE	Change (replacement) of a field (see Table 6-3)
	FieldCode	0x03	Modify *ETH_TYPE_LEN* field (see Table 6-2)
	Value	0x88-09	Slow Protocol Ethertype value (see IEEE Std 802.3, 57A.4)
RuleTLV (termination)	*Type*	0x00	This is a termination (end-of-rule) TLV (see Table 8-4)
	Length	0x04	TLV length is 4 octets
	Operation	0x00	Filled with zerous when not used (see Table 8-4 note)
	FieldCode	0x00	

Deleted: Table 8-4

Deleted: Table 6-3

Deleted: Table 6-2

Deleted: Table 8-4

Deleted: Table 8-4

8A.1.2.3 Addition of VLC tunnel entrance rule at the ingress of Bridge Y, port 0

The VLC tunnel entrance rule at the ingress of Bridge Y, port 0 is shown in Table 8A-5. This rule converts an OAMPDU into a VLCPDU in the receive path of port 0. The conversion replaces the destination MAC address value (*SP_DA*) with the MAC address of Manager M and replaces the Slow Protocol Ethertype (`SP_type`) with the VLC Ethertype (`VLC_type`).

Table 8A-5—VLC tunnel entrance rule at the ingress of Bridge Y, port 0

Conditions	Actions
1. `DA == SP_DA` 2. `ETH_TYPE_LEN == SP_type` 3. `SP_SUBTYPE == OAM_subtype`	1. `REPLACE( DA, M )` 2. `REPLACE( ETH_TYPE_LEN, VLC_type )`
NOTE: `SP_type` – Slow Protocol Ethertype value (see IEEE Std 802.3, 57A.4) `VLC_type` – Ethertype value identifying VLCPDUs (see 5.1) `OAM_subtype` – Subtype value identifying OAMPDUs (see IEEE Std 802.3, 57A.4) `SP_DA` – Destination MAC address associated with Slow Protocols (see IEEE Std 802.3, 57A.3) `M` – MAC address of Manager M.	

Table 8A-6 provides the contents of a *VLC_CONFIG* VLCPDU that provisions the rule shown in Table 8A-5.

Table 8A-6—Contents of *VLC_CONFIG* message

Field	Subfield	Value	Description
DestinationAddress	n/a	Y	*VLC_CONFIG* VLCPDU directed to bridge Y
SourceAddress	n/a	any	Source address of a device that issued the *VLC_CONFIG* VLCPDU
LengthType	n/a	0xA8-C8	Ethertype value identifying VLCPDUs (see 5.1)

Field	Subfield	Value	Description
Subtype	n/a	0x00	VLCPDU carrying *VLC_CONFIG* message
MsgCode	*MsgType*	0x0	This message is a Request (see Table 8-1)
	RequestCode	0x1	Request to add a rule (see Table 8-1)
MsgSequence	*MsgCounter*	0x00-01	This request consists of a single message
	EndOfSequence	1	
PortInstance	*PortIndex*	3	The rule is to be provisioned for port #3
	Direction	1	The rule is to be provisioned for the receive path (i.e., an ingress rule)
RuleTLV (condition)	*Type*	0xCO	This is a condition TLV (see Table 8-4)
	Length	0x0A	TLV length is 10 octets
	Operation	0x11	Comparison for equality (see Table 6-1)
	FieldCode	0x01	Compare *DST_ADDR* field (see Table 6-2)
	Value	0x01-80-C2-00-00-02	IEEE 802.3 Slow_Protocols_Multicast address (see IEEE Std 802.3, 57A.3)
RuleTLV (condition)	*Type*	0xCO	This is a condition TLV (see Table 8-4)
	Length	0x06	TLV length is 6 octets
	Operation	0x11	Comparison for equality (see Table 6-1)
	FieldCode	0x03	Compare *ETH_TYPE_LEN* field (see Table 6-2)
	Value	0x88-09	Slow Protocol Ethertype value (see IEEE Std 802.3, 57A.4)
RuleTLV (condition)	*Type*	0xCO	This is a condition TLV (see Table 8-4)
	Length	0x05	TLV length is 5 octets
	Operation	0x11	Comparison for equality (see Table 6-1)
	FieldCode	0x26	Compare *XPDU_SUBTYPE* field (see Table 6-2)
	Value	0x03	Slow Protocol Subtype value for OAM (see IEEE Std 802.3, 57A.4)
RuleTLV (action)	*Type*	0xAC	This is an action TLV (see Table 8-4)
	Length	0x0A	TLV length is 10 octets
	Operation	0xCE	Change (replacement) of a field (see Table 6-3)
	FieldCode	0x01	Modify *DST_ADDR* field (see Table 6-2)
	Value	M	Set manager M MAC address as the destination for resulting VLCPDUs.
RuleTLV (action)	*Type*	0xAC	This is an action TLV (see Table 8-4)
	Length	0x06	TLV length is 6 octets
	Operation	0xCE	Change (replacement) of a field (see Table 6-3)
	FieldCode	0x03	Modify *ETH_TYPE_LEN* field (see Table 6-2)

Deleted: Table 8-1
Deleted: Table 8-1
Deleted: Table 8-4
Deleted: Table 6-1
Deleted: Table 6-2
Deleted: Table 8-4
Deleted: Table 6-1
Deleted: Table 6-2
Deleted: Table 8-4
Deleted: Table 6-1
Deleted: Table 6-2
Deleted: Table 8-4
Deleted: Table 6-3
Deleted: Table 6-2
Deleted: Table 8-4
Deleted: Table 6-3
Deleted: Table 6-2

Field	Subfield	Value	Description
	Value	0xA8-C8	Set Ethertype to be equal to VLC_Ethertype in the resulting VLCPDUs.
RuleTLV (termination)	*Type*	0x00	This is a termination (end-of-rule) TLV (see Table 8-4)
	Length	0x04	TLV length is 4 octets
	Operation	0x00	Filled with zerous when not used (see Table 8-4 note)
	FieldCode	0x00	

Deleted: Table 8-4

Deleted: Table 8-4

8A.1.2.4 Addition of VLC tunnel exit rule at the egress of Bridge X, port 3

The VLC tunnel exit rule at the ingress of Bridge X, port 3 is shown in Table 8A-7. This rule converts a VLCPDU into an OAMPDU in the transmit path of port 3. The conversion replaces the destination MAC address of Manager M with the MAC address used for Slow Protocol xPDUs (`SP_DA`) and replaces the VLC Ethertype (`VLC_type`) with the Slow Protocol Ethertype (`SP_type`).

Table 8A-7—VLC tunnel exit rule at the egress of Bridge X, port 3

Conditions	Actions
`1. DA == M` `2. ETH_TYPE_LEN == VLC_type` `3. VLC_SUBTYPE == OAM_Subtype`	`1. REPLACE( DA, SP_DA )` `2. REPLACE( ETH_TYPE_LEN, SP_type )`
NOTE: `SP_type` – Slow Protocol Ethertype value (see IEEE Std 802.3, 57A.4) `VLC_type` – Ethertype value identifying VLCPDUs (see 5.1) `OAM_Subtype` – Subtype value identifying OAM payload (see Table 5-1) `SP_DA` – Destination MAC address associated with Slow Protocols (see IEEE Std 802.3, 57A.3) `M` – MAC address of Manager M.	

Deleted: Table 5-1

Table 8A-8 provides the contents of a *VLC_CONFIG* VLCPDU that provisions the rule shown in Table 8A-7.

Table 8A-8—Contents of *VLC_CONFIG* message

Field	Subfield	Value	Description
DestinationAddress	n/a	X	*VLC_CONFIG* VLCPDU directed to bridge X
SourceAddress	n/a	any	Source address of a device that issued the *VLC_CONFIG* VLCPDU
LengthType	n/a	0xA8-C8	Ethertype value identifying VLCPDUs (see 5.1)
Subtype	n/a	0x00	VLCPDU carrying *VLC_CONFIG* message
MsgCode	*MsgType*	0x0	This message is a Request (see Table 8-1)
	RequestCode	0x1	Request to add a rule (see Table 8-1)
MsgSequence	*MsgCounter*	0x00-01	This request consists of a single message

Deleted: Table 8-1

Deleted: Table 8-1

Field	Subfield	Value	Description
	EndOfSequence	1	
PortInstance	*PortIndex*	3	The rule is to be provisioned for port #3
	Direction	0	The rule is to be provisioned for the transmit path (i.e., an egress rule)
RuleTLV (condition)	*Type*	0xCO	This is a condition TLV (see Table 8-4)
	Length	0x0A	TLV length is 10 octets
	Operation	0x11	Comparison for equality (see Table 6-1)
	FieldCode	0x01	Compare *DST_ADDR* field (see Table 6-2)
	Value	M	The dstination address is equal to MAC address of Manager M.
RuleTLV (condition)	*Type*	0xCO	This is a condition TLV (see Table 8-4)
	Length	0x06	TLV length is 6 octets
	Operation	0x11	Comparison for equality (see Table 6-1)
	FieldCode	0x03	Compare *ETH_TYPE_LEN* field (see Table 6-2)
	Value	0xA8-C8	VLC Ethertype value (see 5.1)
RuleTLV (condition)	*Type*	0xCO	This is a condition TLV (see Table 8-4)
	Length	0x05	TLV length is 5 octets
	Operation	0x11	Comparison for equality (see Table 6-1)
	FieldCode	0x16	Compare *VLC_SUBTYPE* field (see Table 6-2)
	Value	0x03	VLC Subtype identifying OAM payload (see Table 5-1)
RuleTLV (action)	*Type*	0xAC	This is an action TLV (see Table 8-4)
	Length	0x0A	TLV length is 10 octets
	Operation	0xCE	Change (replacement) of a field (see Table 6-3)
	FieldCode	0x01	Modify *DST_ADDR* field (see Table 6-2)
	Value	0x01-80-C2-00-00-02	IEEE 802.3 Slow_Protocols_Multicast address (see IEEE Std 802.3, 57A.3)
RuleTLV (action)	*Type*	0xAC	This is an action TLV (see Table 8-4)
	Length	0x06	TLV length is 6 octets
	Operation	0xCE	Change (replacement) of a field (see Table 6-3)
	FieldCode	0x03	Modify *ETH_TYPE_LEN* field (see Table 6-2)
	Value	0x88-09	Slow Protocol Ethertype value (see IEEE Std 802.3, 57A.4)
RuleTLV (termination)	*Type*	0x00	This is a termination (end-of-rule) TLV (see Table 8-4)
	Length	0x04	TLV length is 4 octets

Deleted: Table 8-4
Deleted: Table 6-1
Deleted: Table 6-2
Deleted: Table 8-4
Deleted: Table 6-1
Deleted: Table 6-2
Deleted: Table 8-4
Deleted: Table 6-1
Deleted: Table 6-2
Deleted: Table 5-1
Deleted: Table 8-4
Deleted: Table 6-3
Deleted: Table 6-2
Deleted: Table 8-4
Deleted: Table 6-3
Deleted: Table 6-2
Deleted: Table 8-4

Field	Subfield	Value	Description
	Operation	0x00	Filled with zerous when not used (see Table 8-4 note)
	FieldCode	0x00	

Deleted: Table 8-4

8A.1.3 VLC provisioning to delete tunnels

The deletion of a VLC tunnel involves the deletion of rules that control VLC tunnel entrance and VLC tunnel exit. Therefore, to delete a tunnel from Manager M to Station S, the following rules are removed:

- VLC tunnel entrance rule at the ingress of Bridge X, port 3
- VLC tunnel exit rule at the egress of Bridge Y, port 0

To delete a VLC tunnel from Station S to Manager M, the following rules are removed:

- VLC tunnel entrance rule at the ingress of Bridge Y, port 0
- VLC tunnel exit rule at the egress of Bridge X, port 3

Each rule deletion is provsioned using a separate *VLC_CONFIG* VLCPDU. The contents of all four messages required to delete two tunnels for bidirectional communication for the network segment illustrated in Figure 8A-1 are shown below.

Deleted: Figure 8A-1

8A.1.3.1 Deletion of VLC tunnel entrance rule at the ingress of Bridge X, port 3

The *VLC_CONFIG* VLCPDU that deletes the VLC tunnel entrance rule at the ingress of Bridge X, port 3 is identical to the *VLC_CONFIG* VLCPDU shown in Table 8A-2, with the exception of the value of the field *MsgCode*, subfield *RequestCode*, which in case of rule deletion has the value of 0x2 (see Table 8-1).

Deleted: Table 8-1

8A.1.3.2 Deletion of VLC tunnel exit rule at the egress of Bridge Y, port 0

The *VLC_CONFIG* VLCPDU that deletes the VLC tunnel exit rule at the egress of Bridge Y, port 0 is identical to the *VLC_CONFIG* VLCPDU shown in Table 8A-4, with the exception of the value of the field *MsgCode*, subfield *RequestCode*, which in case of rule deletion has the value of 0x2 (see Table 8-1).

Deleted: Table 8-1

8A.1.3.3 Deletion of VLC tunnel entrance rule at the ingress of Bridge Y, port 0

The *VLC_CONFIG* VLCPDU that deletes the VLC tunnel entrance rule at the ingress of Bridge Y, port 0 is identical to the *VLC_CONFIG* VLCPDU shown in Table 8A-6, with the exception of the value of the field *MsgCode*, subfield *RequestCode*, which in case of rule deletion has the value of 0x2 (see Table 8-1).

Deleted: Table 8-1

8A.1.3.4 Deletion of VLC tunnel exit rule at the egress of Bridge X, port 3

The *VLC_CONFIG* VLCPDU that deletes the VLC tunnel exit rule at the egress of Bridge X, port 3 is identical to the *VLC_CONFIG* VLCPDU shown in Table 8A-8, with the exception of the value of the field *MsgCode*, subfield *RequestCode*, which in case of rule deletion has the value of 0x2 (see Table 8-1).

Deleted: Table 8-1

8A.2 OAM over VLC use case, VLC-aware end points

8A.2.1 Introduction

This example illustrates OAM communication between a Manager M and a Station S carried over VLC that traverses multiple L2 bridges (see Figure 8A-2). Both the Manager and the Station are VLC-aware. The VLC awarness is not required in the intermediate Bridges X and Y, as well as any possible other bridges between them.

Deleted: Figure 8A-2

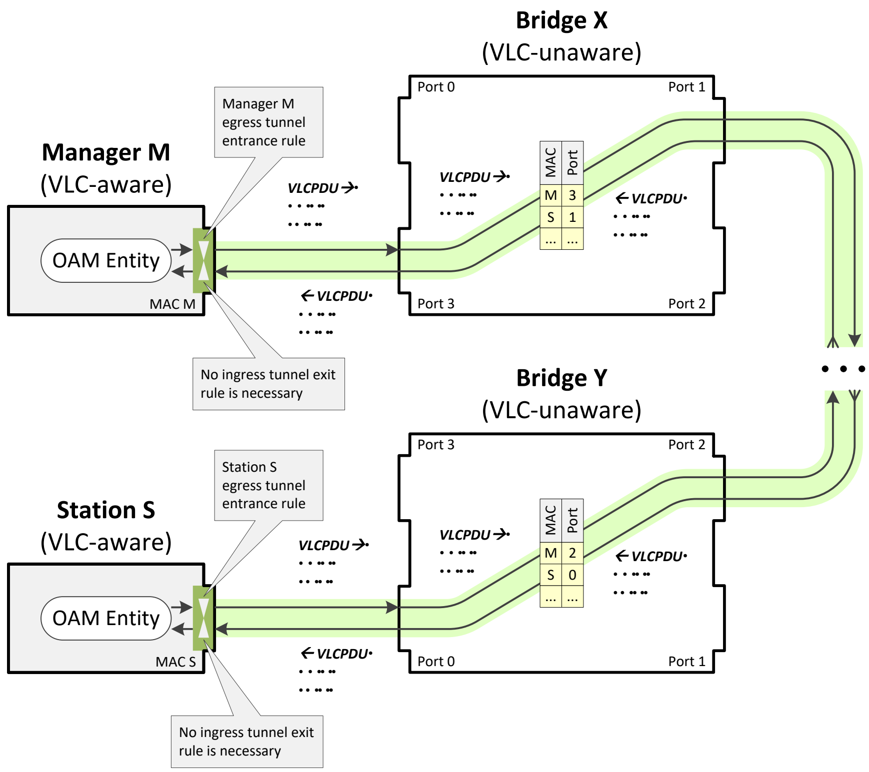


Figure 8A-2—OAM over VLC use case, VLC-aware end points

In Figure 8A-2, the Manager M, station S, Bridges X and Y have MAC addresses M, S, X, and Y respectively. For simplicity, it is assumed that all ports in a given device use the same MAC address, but this is not a requirement.

Deleted: Figure 8A-2

Furthermore, it is assumed that Bridges X and Y, as well as all intermediate bridges, have already populated their forwarding tables with entries for MAC addresses M and S. These entries may be created dynamically by a MAC learning function or be provisioned statically by the NMS.

8A.2.2 VLC provisioning to establish tunnels

Since the Manager M is not directly linked with the managed Station S, the OAM messages need to be carried over VLCPDUs. Therefore, before the Manager M and the Station S are able to exchange OAM messages, two VLC tunnels need to be provisioned:

— A forward VLC tunnel from Manager M to Station S.
— A reverse VLC tunnel from Station S to Manager M.

To establish a VLC tunnel from Manager M to Station S, a tunnel entrance rule is provisioned at the egress of Manager M. No tunnel exit rule is necessary at the ingress of Station S, since the VLC sublayer provides a built-in translation of VLCPDUs with subtype `OAM_subtype` into OAMPDUs (see Receive Path Specification in 6.2).

Similarly, to establish a VLC tunnel from Station S to Manager M, a tunnel entrance rule is provisioned at the egress of Station S. No tunnel exit rule is necessary at the ingress of Manager M, since the VLC sublayer provides a built-in translation of VLCPDUs with subtype `OAM_subtype` into OAMPDUs.

Each rule is provisioned using a separate `VLC_CONFIG` message. The contents of two messages required to establish two VLC tunnles for bidirectional communication for the network segment illustrated in Figure 8A-2 are shown below.

Deleted: Figure 8A-2

8A.2.2.1 Addition of tunnel entrance rule at the egress of Manager M

The VLC tunnel entrance rule at the egress of Manager M is shown in Table 8A-9. This rule converts an OAMPDU into a VLCPDU in the transmit path of a given port of Manager M. The conversion replaces the destination MAC address value (`SP_DA`) with the MAC address of Station S and replaces the Slow Protocol Ethertype (`SP_TYPE`) with the VLC Ethertype (`VLC_TYPE`).

Table 8A-9—Tunnel entrance rule at the egress of Manager M

Conditions	Actions
1. DA == SP_DA 2. ETH_TYPE_LEN == SP_TYPE 3. SUBTYPE == OAM_SUBTYPE	1.REPLACE(DA, S) 2.REPLACE(ETH_TYPE_LEN, VLC_TYPE)
NOTE: `SP_TYPE` – Slow Protocol Ethertype value (see IEEE Std 802.3, 57A.4) `VLC_TYPE` – Ethertype value identifying VLCPDUs (see 5.1) `OAM_SUBTYPE` – Subtype value identifying OAMPDUs (see IEEE Std 802.3, 57A.4) `SP_DA` – Destination MAC address associated with Slow Protocols (see IEEE Std 802.3, 57A.3) `S` – MAC address of Station S.	

Table 8A-10 provides the contents of a *VLC_CONFIG* VLCPDU that provisions the rule shown in Table 8A-9.

Table 8A-10—Contents of VLC_CONFIG message

Field	Subfield	Value	Description
DestinationAddress	n/a	M	`VLC_CONFIG` VLCPDU directed to Manager M
SourceAddress	n/a	any	Source address of the device that issued the `VLC_CONFIG` VLCPDU
LengthType	n/a	0xA8-C8	Ethertype value identifying VLCPDUs (see 5.1)
Subtype	n/a	0x00	VLCPDU carrying `VLC_CONFIG` message
MsgCode	*MsgType*	0x0	This message is a Request (see Table 8-1)
	RequestCode	0x1	Request to add a rule (see Table 8-1)

Deleted: Table 8-1

Deleted: Table 8-1

Field	Subfield	Value	Description
MsgSequence	*MsgCounter*	0x00-01	This request consists of a single message
	EndOfSequence	1	
PortInstance	*PortIndex*	1	The rule is to be provisioned for port #1
	Direction	0	The rule is to be provisioned for the transmit path (i.e., an egress rule)
RuleTLV (condition)	*Type*	0xCO	This is a condition TLV (see Table 8-4)
	Length	0x0A	TLV length is 10 octets
	Operation	0x11	Comparison for equality (see Table 6-1)
	FieldCode	0x01	Compare `DST_ADDR` field (see Table 6-2)
	Value	0x01-80-C2-00-00-02	IEEE 802.3 Slow_Protocols_Multicast address (see IEEE Std 802.3, 57A.3)
RuleTLV (condition)	*Type*	0xCO	This is a condition TLV (see Table 8-4)
	Length	0x06	TLV length is 6 octets
	Operation	0x11	Comparison for equality (see Table 6-1)
	FieldCode	0x03	Compare `ETH_TYPE_LEN` field (see Table 6-2)
	Value	0x88-09	Slow Protocol Ethertype value (see IEEE Std 802.3, 57A.4)
RuleTLV (condition)	*Type*	0xCO	This is a condition TLV (see Table 8-4)
	Length	0x05	TLV length is 5 octets
	Operation	0x11	Comparison for equality (see Table 6-1)
	FieldCode	0x26	Compare `SUBTYPE` field (see Table 6-2)
	Value	0x03	Slow Protocol Subtype value for OAM (see IEEE Std 802.3, 57A.4)
RuleTLV (action)	*Type*	0xAC	This is an action TLV (see Table 8-4)
	Length	0x0A	TLV length is 10 octets
	Operation	0xCE	Change (replacement) of a field (see Table 6-3)
	FieldCode	0x01	Modify `DST_ADDR` field (see Table 6-2)
	Value	S	Set Station S MAC address as the destination for resulting VLCPDUs.
RuleTLV (action)	*Type*	0xAC	This is an action TLV (see Table 8-4)
	Length	0x06	TLV length is 6 octets
	Operation	0xCE	Change (replacement) of a field (see Table 6-3)
	FieldCode	0x03	Modify `ETH_TYPE_LEN` field (see Table 6-2)
	Value	0xA8-C8	Set Ethertype to be equal to VLC Ethertype (`VLC_TYPE`) in the resulting VLCPDUs.

Deleted: Table 8-4
Deleted: Table 6-1
Deleted: Table 6-2
Deleted: Table 8-4
Deleted: Table 6-1
Deleted: Table 6-2
Deleted: Table 8-4
Deleted: Table 6-1
Deleted: Table 6-2
Deleted: Table 8-4
Deleted: Table 6-3
Deleted: Table 6-2
Deleted: Table 8-4
Deleted: Table 6-3
Deleted: Table 6-2

Field	Subfield	Value	Description
RuleTLV (termination)	*Type*	0x00	This is a termination (end-of-rule) TLV (see Table 8-4)
	Length	0x04	TLV length is 4 octets
	Operation	0x00	Filled with zeros when not used (see Table 8-4 note)
	FieldCode	0x00	

Deleted: Table 8-4

Deleted: Table 8-4

8A.2.2.2 Addition of VLC tunnel entrance rule at the egress of Station S

The VLC tunnel entrance rule at the egress of Station S is shown in Table 8A-11. This rule converts an OAMPDU into a VLCPDU in the transmit path of port 0. The conversion replaces the destination MAC address value (`SP_DA`) with the MAC address of Manager M and replaces the Slow Protocol Ethertype (`SP_TYPE`) with the VLC Ethertype (`VLC_TYPE`).

Table 8A-11—VLC tunnel entrance rule at the ingress of Station S

Conditions	Actions
`1. DA == SP_DA` `2. ETH_TYPE_LEN == SP_TYPE` `3. SUBTYPE == OAM_SUBTYPE`	`1.REPLACE( DA, M )` `2.CHANGE( ETH_TYPE_LEN, VLC_TYPE )`
NOTE: `SP_TYPE` – Slow Protocol Ethertype value (see IEEE Std 802.3, 57A.4) `VLC_TYPE` – Ethertype value identifying VLCPDUs (see 5.1) `OAM_SUBTYPE` – Subtype value identifying OAMPDUs (see IEEE Std 802.3, 57A.4) `SP_DA` – Destination MAC address associated with Slow Protocols (see IEEE Std 802.3, 57A.3) `M` – MAC address of Manager M.	

Table 8A-12 provides the contents of a *VLC_CONFIG* VLCPDU that provisions the rule shown in Table 8A-11.

Table 8A-12—Contents of VLC_CONFIG message

Field	Subfield	Value	Description
DestinationAddress	n/a	S	`VLC_CONFIG` VLCPDU directed to Station S
SourceAddress	n/a	any	Source address of the device that issued the `VLC_CONFIG` VLCPDU
LengthType	n/a	0xA8-C8	Ethertype value identifying VLCPDUs (see 5.1)
Subtype	n/a	0x00	VLCPDU carrying `VLC_CONFIG` message
MsgCode	*MsgType*	0x0	This message is a Request (see Table 8-1)
	RequestCode	0x1	Request to add a rule (see Table 8-1)
MsgSequence	*MsgCounter*	0x00-01	This request consists of a single message

Deleted: Table 8-1

Deleted: Table 8-1

Field	Subfield	Value	Description
	EndOfSequence	1	
PortInstance	*PortIndex*	0	The rule is to be provisioned for port #0
	Direction	0	The rule is to be provisioned for the transmit path (i.e., an egress rule)
RuleTLV (condition)	*Type*	0xCO	This is a condition TLV (see Table 8-4)
	Length	0x0A	TLV length is 10 octets
	Operation	0x11	Comparison for equality (see Table 6-1)
	FieldCode	0x01	Compare DST_ADDR field (see Table 6-2)
	Value	0x01-80-C2-00-00-02	IEEE 802.3 Slow_Protocols_Multicast address (see IEEE Std 802.3, 57A.3)
RuleTLV (condition)	*Type*	0xCO	This is a condition TLV (see Table 8-4)
	Length	0x06	TLV length is 6 octets
	Operation	0x11	Comparison for equality (see Table 6-1)
	FieldCode	0x03	Compare ETH_TYPE_LEN field (see Table 6-2)
	Value	0x88-09	Slow Protocol Ethertype value (see IEEE Std 802.3, 57A.4)
RuleTLV (condition)	*Type*	0xCO	This is a condition TLV (see Table 8-4)
	Length	0x05	TLV length is 5 octets
	Operation	0x11	Comparison for equality (see Table 6-1)
	FieldCode	0x26	Compare SUBTYPE field (see Table 6-2)
	Value	0x03	Slow Protocol Subtype value for OAM (see IEEE Std 802.3, 57A.4)
RuleTLV (action)	*Type*	0xAC	This is an action TLV (see Table 8-4)
	Length	0x0A	TLV length is 10 octets
	Operation	0xCE	Change (replacement) of a field (see Table 6-3)
	FieldCode	0x01	Modify DST_ADDR field (see Table 6-2)
	Value	M	Set Manager M MAC address as the destination for resulting VLCPDUs.
RuleTLV (action)	*Type*	0xAC	This is an action TLV (see Table 8-4)
	Length	0x06	TLV length is 6 octets
	Operation	0xCE	Change (replacement) of a field (see Table 6-3)
	FieldCode	0x03	Modify ETH_TYPE_LEN field (see Table 6-2)
	Value	0xA8-C8	Set Ethertype to be equal to VLC Ethertype (VLC_TYPE) in the resulting VLCPDUs.

Deleted: Table 8-4
Deleted: Table 6-1
Deleted: Table 6-2
Deleted: Table 8-4
Deleted: Table 6-1
Deleted: Table 6-2
Deleted: Table 8-4
Deleted: Table 6-1
Deleted: Table 6-2
Deleted: Table 8-4
Deleted: Table 6-3
Deleted: Table 6-2
Deleted: Table 8-4
Deleted: Table 6-3
Deleted: Table 6-2

Field	Subfield	Value	Description
RuleTLV (termination)	*Type*	0x00	This is a termination (end-of-rule) TLV (see Table 8-4)
	Length	0x04	TLV length is 4 octets
	Operation	0x00	Filled with zeros when not used (see Table 8-4 note)
	FieldCode	0x00	

Deleted: Table 8-4

Deleted: Table 8-4

8A.2.3 VLC provisioning to delete tunnels

The deletion of a VLC tunnel involves the deletion of a rule that controls VLC tunnel entrance. Therefore, to delete a tunnel from Manager M to Station S, the VLC tunnel entrance rule at the egress of Manager M is deleted. And to delete a VLC tunnel from Station S to Manager M, the VLC tunnel entrance rule at the egress of Station S is deleted.

Each rule deletion is provsioned using a separate `VLC_CONFIG` VLCPDU. The contents of two messages required to delete two tunnels illustrated in Figure 8A-2 are described below.

Deleted: Figure 8A-2

8A.2.3.1 Deletion of VLC tunnel entrance rule at the egress of Manager M

The contents of a `VLC_CONFIG` VLCPDU that deletes the VLC tunnel entrance rule at the egress of Manager M are identical to the `VLC_CONFIG` VLCPDU shown in Table 8A-10, with the exception of the value of the field `MsgCode`, subfield `RequestCode`, which in case of rule deletion has the value of 0x2 (see Table 8-1).

Deleted: Table 8-1

8A.2.3.2 Deletion of VLC tunnel entrance rule at the egress of Station S

The contents of a `VLC_CONFIG` VLCPDU that deletes the VLC tunnel entrance rule at the ingress of Bridge Y, port 0 is identical to the `VLC_CONFIG` VLCPDU shown in Table 8A-12, with the exception of the value of the field `MsgCode`, subfield `RequestCode`, which in case of rule deletion has the value of 0x2 (see Table 8-1).

Deleted: Table 8-1

8A.3 OAM over VLC use case, VLC-aware end point and VLC-unaware end point

8A.3.1 Introduction

This example illustrates OAM communication between a Manager M and a Station S carried over VLC that traverses multiple L2 bridges (see Figure 8A-3). The Manager M is VLC-aware, while the Station S is VLC-unaware. The Bridge X nearest to the Manager M may or may be not VLC-aware. The Bridge Y nearest to the Station S is VLC-aware and is responsible for converting OAMPDUs into VLCPDUs and vise versa. There can be numerous other bridges between the Bridges X and Y; those bridges may or may be not VLC-aware.

Deleted: Figure 8A-3

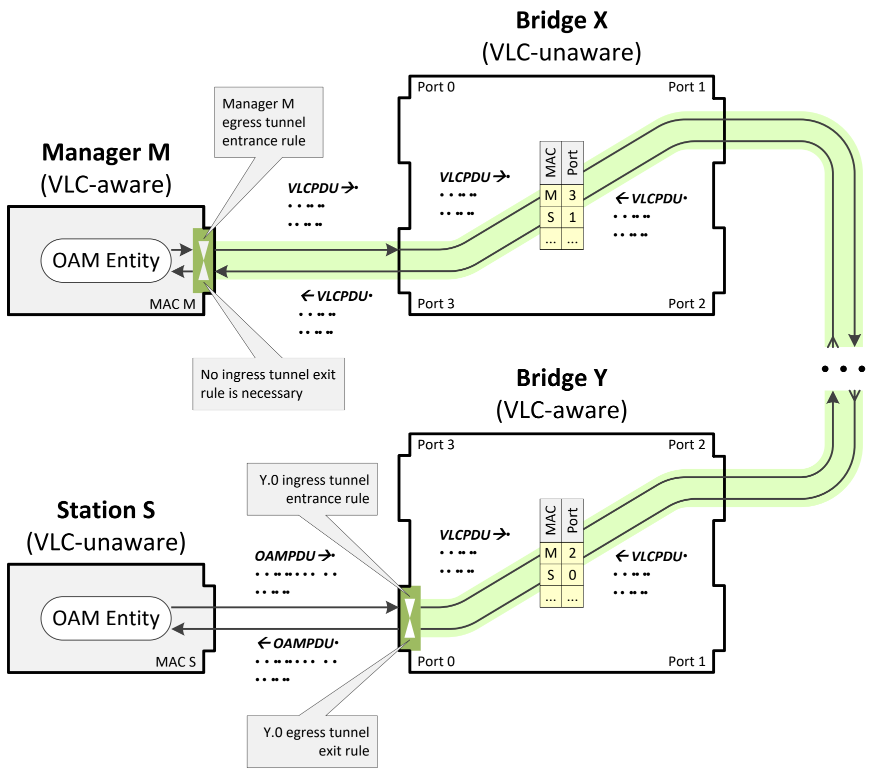


Figure 8A-3—OAM over VLC use case, one VLC-unaware end point and one VLC-aware end-point

In Figure 8A-3, the Manager M, station S, Bridges X and Y have MAC addresses M, S, X, and Y respectively. For simplicity, it is assumed that all ports in a given device use the same MAC address, but this is not a requirement.

Deleted: Figure 8A-3

Furthermore, it is assumed that Bridges X and Y, as well as all intermediate bridges, have already populated their forwarding tables with entries for MAC addresses M and S. These entries may be created dynamically by a MAC learning function or be provisioned statically by the NMS.

8A.3.2 VLC provisioning to establish tunnels

Since the Manager M is not directly connected to the managed Station S, the OAM messages need to be carried over VLCPDUs. Therefore, before the Manager M and the Station S are able to exchange OAM messages, two VLC tunnels need to be provisioned:

— A forward VLC tunnel from Manager M to Bridge Y, port 0.
— A reverse VLC tunnel from Bridge Y, port 0 to Manager M.

To establish a VLC tunnel from Manager M to Bridge Y, port 0, the following rules are provisioned:

- A VLC tunnel entrance rule at the egress of Manager M
- A VLC tunnel exit rule at the egress of Bridge Y, port 0

To establish a VLC tunnel from Bridge Y, port 0 to Manager M, only one rule is provisioned:

- A VLC tunnel entrance rule at the ingress of Bridge Y, port 0

No tunnel exit rule is necessary at the ingress of Manager M, since the VLC sublayer provides a built-in translation of VLCPDUs with subtype `OAM_subtype` into OAMPDUs.

Each rule is provisioned using a separate `VLC_CONFIG` message. The contents of all three messages required to establish two VLC tunnles for bidirectional communication for the network segment illustrated in Figure 8A-3 are described below.

Deleted: Figure 8A-3

8A.3.2.1 Addition of tunnel entrance rule at the egress of Manager M

The CTE rule and the content of the `VLC_CONFIG` VLCPDU are identical to those described in 8A.2.2.1.

8A.3.2.2 Addition of tunnel exit rule at the egress of Bridge Y, port 0

The CTE rule and the content of the `VLC_CONFIG` VLCPDU are identical to those described in 8A.2.2.1.

8A.3.2.3 Addition of VLC tunnel entrance rule at the ingress of Bridge Y, port 0

The CTE rule and the content of the `VLC_CONFIG` VLCPDU are identical to those described in 8A.1.2.3.

8A.3.3 VLC provisioning to delete tunnels

The deletion of a VLC tunnel involves the deletion of rules that control VLC tunnel entrance and VLC tunnel exit Therefore, to delete a tunnel from Manager M to Station S, the following rules are removed:

- VLC tunnel entrance rule at the egress of Manager M
- VLC tunnel exit rule at the egress of Bridge Y, port 0

To delete a VLC tunnel from Station S to Manager M, the following rule is removed:

- VLC tunnel entrance rule at the ingress of Bridge Y, port 0

Each rule deletion is provsioned using a separate `VLC_CONFIG` VLCPDU. The contents of all three messages required to delete two tunnels for bidirectional communication for the network segment illustrated in Figure 8A-3 are described below.

Deleted: Figure 8A-3

8A.3.3.1 Deletion of VLC tunnel entrance rule at the egress of Manager M

The contents of a `VLC_CONFIG` VLCPDU that deletes the VLC tunnel entrance rule at the egress of Manager M are identical to the `VLC_CONFIG` VLCPDU described in 8A.2.3.1.

8A.3.3.2 Deletion of VLC tunnel exit rule at the egress of Bridge Y, port 0

The contents of a `VLC_CONFIG` VLCPDU that deletes the VLC tunnel entrance rule at the egress of Bridge Y, port 0 are identical to the `VLC_CONFIG` VLCPDU described in 8A.1.2.2.

8A.3.3.3 Deletion of VLC tunnel entrance rule at the ingress of Bridge Y, port 0

The contents of a `VLC_CONFIG` VLCPDU that deletes the VLC tunnel entrance rule at the egress of Bridge Y, port 0 are identical to the `VLC_CONFIG` VLCPDU described in 8A.1.2.3.

8A.4 Remote PON Management over VLC use case

8A.4.1 Introduction

This example illustrates a use case in which mulitple protocols are configured together to enable remote management of an OLT and its subtended ONUs. In this example, the OLT is managed using an extension of IEEE 802.3 Clause 57 OAM. Traditionally, ONUs would be managed by an entity that resides inside the OLT[4] and the GPON ONUs[5] are managed using OMCI. The "manager" entity for both protocols is located in a station (referred to simply as the manager) that is separate from the OLT (the management function is disaggregated from the phyiscal OLT).

In the most general sense, the manager is separated from the OLT by one or more MAC bridge entities (see Figure 8A-4). This use case assumes that the manager and the OLT are VLC aware, but the intermediate network elements and the ONUs are VLC unaware.

Deleted: Figure 8A-4

In Figure 8A-1, the Manager and OLT have MAC addresses M and L respectively. For simplicity, it is assumed that the Manager and OLT are single Ethernet port devices, but this is not a requirement.

Deleted: Figure 8A-1

Furthermore, it is assumed that Bridges X and Y, as well as all intermediate bridges, have already populated their forwarding tables with entries for MAC addresses M and L. These entries may be created dynamically by a MAC learning function or be provisioned statically by the NMS.

Note that this example assumes ITU-T PON and hence the reference to OMCI.

8A.4.2 VLC provisioning to establish tunnels

Since the Manager is not directly connected to the managed OLT and ONUs, the OAM and OMCI messages need to be carried over VLCPDUs. Therefore, before the Manager and the OLT are able to exchange OAM messages and the manager and ONUs are able to exchange OMCI messages, two VLC tunnels need to be provisioned:

— A forward VLC tunnel from Manager to OLT.
— A reverse VLC tunnel from OLT to Manager.

The establishement of each VLC tunnel involves provisioning of multiple rules to configure the VLC tunnel entrance and exit points.

To establish a VLC tunnel from PON contoller to OLT, the following rules are provisioned:

— A VLC tunnel entrance rule at the egress of Manager for OLT OAM messages
— A VLC tunnel entrance rule at the egress of Manager for ONU OMCI messages

To establish a VLC tunnel from OLT to Manager, the following rules are provisioned:

— A VLC tunnel entrance rule at the egress of OLT for OLT OAM messages
— A VLC tunnel entrance rule at the egress of OLT for ONU OMCI messages

[4] In this use case, OLT is used generically to refer to an L-OLT, S-OLT or C-OLT as defined by IEEE Srd 1904.1. If the distinction is important, the specific element name will be used.

[5] In this use case, ONU is used generically to refer to an L-ONU, S-ONU or C-ONU as defined by IEEE Std 1904.1. If the distinction is important, the specific element name will be used.

No tunnel exit rule is necessary at the ingress of Manager M or at the ingress of OLT, since the VLC sublayer provides a built-in translation of VLCPDUs with subtype `OAM_subtype` into OAMPDUs and a built-in translation of VLCPDUs with subtype `OMCI_subtype` into OMCI frames (see Receive Path Specification in 6.2)

Each rule is provisioned using a separate *VLC_CONFIG* message.

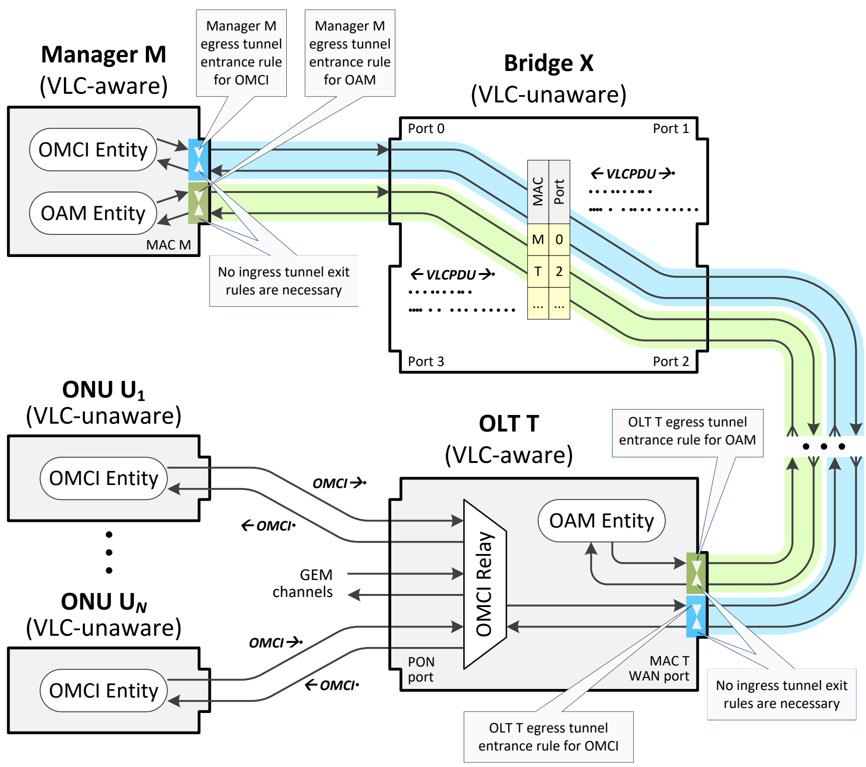


Figure 8A-4—Remote PON Management over VLC

8A.4.2.1 Addition of tunnel entrance rule at the egress of Manager for OLT OAM messages

The entrance rule for the VLC tunnel carrying the OAM messages is shown in Table 8A-13. The rule is provisioned at the egress of the Manager and its action is to replace the Slow Protocol destination address value (SP_DA) with the MAC address of OLT L and to replace the Slow Protocol Ethertype (SP_type) with the VLC Ethertype (VLC_type).

Table 8A-13—Tunnel entrance rule at the egress of Manager for OLT OAM messages

Conditions	Actions

1. DA == SP_DA 2. ETH_TYPE_LEN == SP_type 3. SP_SUBTYPE == OAM_subtype	1.REPLACE(DA, L) 2.REPLACE(ETH_TYPE_LEN, VLC_type)
NOTE: SP_type – Slow Protocol Ethertype value (see IEEE Std 802.3, 57A.4) VLC_type – Ethertype value identifying VLCPDUs OAM_subtype – Subtype value identifying OAMPDUs (see IEEE Std 802.3, 57A.4) SP_DA – Destination MAC address associated with Slow Protocols (see IEEE Std 802.3, 57A.3) L – MAC address of OLT	

8A.4.2.2 Addition of tunnel entrance rule at the egress of Manager for ONU OMCI messages

The OMCI frames generated by the OMCI entity (OMCI client) in the Manager are encapsulated as a payload of VLCPDUs within the Transmit process (see Figure 6-3). The entrance rule for the VLC tunnel carrying the OMCI messages is shown in Table 8A-14. The rule is provisioned at the egress of the Manager and its only action is to replace the VLCPDU's placeholder destination address (LOCAL_MAC_ADDR) with the MAC address of the OLT L.

Deleted: Figure 6-3

Table 8A-14—Tunnel entrance rule at the egress of Manager for ONU OMCI messages

Conditions	Actions
1. SA == LOCAL_MAC_ADDR 2. ETH_TYPE_LEN == VLC_type 3. SP_SUBTYPE == OMCI_subtype	1. REPLACE(DA, L)
NOTE: VLC_type – Ethertype value identifying VLCPDUs OMCI_subtype – Subtype value identifying OMCI frames LOCAL_MAC_ADDR – MAC address associated with the port where the Receive process state diagram is instantiated L – MAC address of OLT	

8A.4.2.3 Addition of tunnel entrance rule at the egress of OLT for OLT OAM messages

The entrance rule for the VLC tunnel carrying the OAM messages is shown in Table 8A-15. The rule is provisioned at the egress of the OLT and its action is to replace the Slow Protocol destination address value (SP_DA) with the MAC address of Manager M and to replace the Slow Protocol Ethertype (SP_type) with the VLC Ethertype (VLC_type).

Table 8A-15—Tunnel entrance rule at the egress of OLT for OLT OAM messages

Conditions	Actions
1. DA == SP_DA 2. ETH_TYPE_LEN == SP_type 3. SP_SUBTYPE == OAM_subtype	1.REPLACE(DA, M) 2.REPLACE(ETH_TYPE_LEN, VLC_type)

NOTE:
`SP_type` – Slow Protocol Ethertype value (see IEEE Std 802.3, 57A.4)
`VLC_type` – Ethertype value identifying VLCPDUs
`OAM_subtype` – Subtype value identifying OAMPDUs (see IEEE Std 802.3, 57A.4)
`SP_DA` – Destination MAC address associated with Slow Protocols (see IEEE Std 802.3, 57A.3)
`M` – MAC address of Manager.

8A.4.2.4 Addition of tunnel entrance rule at the egress of OLT for ONU OMCI messages

The OMCI frames generated by the OMCI entity (OMCI client) in the OLT are encapsulated as a payload of VLCPDUs within the Transmit process (see Figure 6-3). The entrance rule for the VLC tunnel carrying the OMCI messages is shown in Table 8A-16. The rule is provisioned at the egress of the OLT and its only action is to replace the VLCPDU's placeholder destination address (LOCAL_MAC_ADDR) with the MAC address of the Manager M.

Deleted: Figure 6-3

Table 8A-16—Tunnel entrance rule at the egress of OLT for ONU OMCI messages

Conditions	Actions
`1. SA == LOCAL_MAC_ADDR` `2. ETH_TYPE_LEN == VLC_type` `3. SP_SUBTYPE == OMCI_subtype`	`1. REPLACE( DA, M )`

NOTE:
`VLC_type` – Ethertype value identifying VLCPDUs
`OMCI_subtype` – Subtype value identifying OMCI frames
`LOCAL_MAC_ADDR` – MAC address associated with the port where the Receive process state diagram is instantiated
`M` – MAC address of Manager.

8A.4.3 VLC provisioning to delete tunnels

The deletion of a VLC tunnel involves the deletion of rules that control VLC tunnel entrance and VLC tunnel exit. Therefore, to delete a tunnel from Manager to OLT, the following rules are removed:

— A VLC tunnel entrance rule at the egress of Manager for OLT OAM messages
— A VLC tunnel entrance rule at the egress of Manager for ONU OMCI messages

To delete a VLC tunnel from OLT to Manager, the following rules are removed:

— A VLC tunnel entrance rule at the egress of OLT for OLT OAM messages
— A VLC tunnel entrance rule at the egress of OLT for ONU OMCI messages

Each rule deletion is provsioned using a separate *VLC_CONFIG* VLCPDU. The contents of all messages required to delete two tunnels for bidirectional communication are not shown here. The *VLC_CONFIG* VLCPDUs for deleting the rules are same as the corresponding *VLC_CONFIG* VLCPDUs for establishing the rules with the exception of the value of the field *MsgCode*, subfield *RequestCode*, which in case of rule deletion has the value of 0x2.