IEEE STANDARDS ASSOCIATION

IEEE Standard for

Local and metropolitan area networks—

Virtual Bridged Local Area Networks— Bridge Port Extension

IEEE Computer Society

Sponsored by the
LAN/MAN Standards Committee

IEEE
3 Park Avenue
New York, NY 10016-5997
USA

IEEE Std 802.1BR™-2012

16 July 2012

IEEE Std 802.1BR™-2012

IEEE Standard for
Local and metropolitan area networks—

Virtual Bridged Local Area Networks— Bridge Port Extension

Sponsor

LAN/MAN Standards Committee
of the
IEEE Computer Society

Approved 14 May 2012

IEEE-SA Standards Board

Abstract: This standard specifies the operation of Bridge Port Extenders, including management, protocols, and algorithms. Bridge Port Extenders operate in support of the MAC Service by Extended Bridges.

Keywords: Bridged Local Area Networks, Data Center Bridging, DCB, Edge Virtual Bridging, EVB, IEEE 802.1BR, LANs, local area networks, MAC Bridges, MANs, metropolitan area networks, Virtual Bridged Local Area Networks

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

Print: ISBN 978-0-7381-7281-1 STD97256
PDF: ISBN 978-0-7381-7387-0 STDPD97256

Notice and Disclaimer of Liability Concerning the Use of IEEE Documents: IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. 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 IEEE administers the process and establishes rules to promote fairness in the consensus development process, 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. 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 any IEEE Standard document.

IEEE does not warrant or represent the accuracy or content of the material contained in its standards, 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 in its standards 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 ten years. When a document is more than ten years old and has not undergone a revision process, 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 its standards available, IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity. Nor is IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing any IEEE Standards document, should rely upon his or her own 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.

Translations: The IEEE consensus development process involves the review of documents in English only. In the event that an IEEE standard is translated, only the English version published by IEEE should be considered the approved IEEE standard.

Official Statements: 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 position of 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 of IEEE.

Comments on Standards: Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE. However, IEEE does not provide consulting information or advice pertaining to IEEE Standards documents. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Since IEEE standards represent a consensus of concerned interests, it is important to ensure that any responses to comments and questions also receive 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 comments or questions except in those cases where the matter has previously been addressed. Any person who would like to participate in evaluating comments or revisions to an IEEE standard is welcome to join the relevant IEEE working group at http://standards.ieee.org/develop/wg/.

Comments on standards should be submitted to the following address:

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

Photocopies: 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.

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 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.

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 at 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 non-discriminatory. 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 standard was approved, the IEEE 802.1 Working Group had the following voting members:

Tony Jeffree, *Chair*
Glenn Parsons, *Vice Chair*
Patricia Thaler, *Data Center Bridging Task Group Chair*
Joe Pelissier, *Editor*

Zehavit Alon
Yafan An
Ting Ao
Peter Ashwood-Smith
Christian Boiger
Paul Bottorff
Rudolf Brandner
Craig Carlson
Xin Chang
Weiying Cheng
Paul Congdon
Rodney Cummings
Claudio DeSanti
Zhemin Ding
Donald Eastlake, 3rd
Janos Farkas
Donald Fedyk
Norman Finn
Geoffrey Garner
Anoop Ghanwani
Franz Goetz
Mark Gravel
Eric Gray
Yingjie Gu
Craig Gunther
Stephen Haddock
Hitoshi Hayakawa
Girault Jones
Daya Kamath
Hal Keen
Srikanth Keesara
Yongbum Kim
Philippe Klein
Oliver Kleineberg
Michael Krause
Lin Li
Jeff Lynch
Ben Mack-Crane
David Martin
John Messenger
John Morris
Eric Multanen
Yukihiro Nakagawa
David Olsen
Donald Pannell
Mark Pearson
Rene Raeber
Karen Randall
Josef Roese
Dan Romascanu
Jessy Rouyer
Ali Sajassi
Panagiotis Saltsidis
Michael Seaman
Koichiro Seto
Rakesh Sharma
Takeshi Shimizu
Kevin Stanton
Robert Sultan
Michael Johas Teener
Jeremy Touve
Maarten Vissers
Yuehua Wei
Min Xiao

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

Thomas Alexander
Hugh Barrass
Nancy Bravin
William Byrd
Craig Carlson
Keith Chow
Charles Cook
Rodney Cummings
Claudio DeSanti
Patrick Diamond
Thomas Dineen
Sourav Dutta
Richard Edgar
Yukihiro Fujimoto
Ilango Ganga
Evan Gilman
Randall C. Groves
Stephen Haddock
Noriyuki Ikeuchi
Atsushi Ito
Raj Jain
Tony Jeffree
Peter Jones
Shinkyo Kaku
Piotr Karocki
Stuart Kerry
Yongbum Kim
Paul Lambert
Brian L'Ecuyer
Greg Luri
Michael Lynch
Elvis Maculuba
Arthur Marris
David Martin
Gary Michel
Jose Morales
Yukihiro Nakagawa
Michael S. Newman
Nick S.A. Nikjoo
Satoshi Obara
Maximilian Riegel
Benjamin Rolfe
Jessy Rouyer
Herbert Ruck
Panagiotis Saltsidis
Bartien Sayogo
Rich Seifert
Gil Shultz
Kapil Sood
Matthew Squire
Manikantan Srinivasan
Thomas Starai
Walter Struppler
Joseph Tardo
William Taylor
Michael Johas Teener
Patricia Thaler
Dmitri Varsanofiev
Prabodh Varshney
John Vergis
Karl Weber
Yuehua Wei
Ludwig Winkel
Oren Yuen

When the IEEE-SA Standards Board approved this standard on 14 May 2012, it had the following membership:

Richard H. Hulett, *Chair*
John Kulick, *Vice Chair*
Robert M. Grow, *Past Chair*

Satish Aggarwal
Masayuki Ariyoshi
Peter Balma
William Bartley
Ted Burse
Clint Chaplin
Wael Diab
Jean-Philippe Faure
Alexander Gelman
Paul Houzé
Jim Hughes
Young Kyun Kim
Joseph L. Koepfinger*
David J. Law
Thomas Lee
Hung Ling
Oleg Logvinov
Ted Olsen
Gary Robinson
Jon Walter Rosdahl
Mike Seavey
Yatin Trivedi
Phil Winston
Yu Yuan

*Member Emeritus

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

Richard DeBlasio, *DOE Representative*
Michael Janezic, *NIST Representative*

Michelle Turner
IEEE Standards Program Manager, Document Development

Kathryn Bennett
IEEE Standards Program Manager, Technical Program Development

Introduction

This introduction is not part of IEEE Std 802.1BR-2012, IEEE Standard for Local and metropolitan area networks—Virtual Bridged Local Area Networks—Bridge Port Extension.

This standard specifies the devices, protocols, procedures, and managed objects necessary to extend a bridge and its management beyond its physical enclosure using IEEE 802® LAN technologies.

To this end, it:

a) Identifies and isolates traffic between ports within an Extended Bridge;
b) Specifies a tag format for this identification;
c) Establishes an Extended Bridge consisting of a Controlling Bridge and one or more Bridge Port Extenders;
d) Specifies the functionality and the specific requirements of a Bridge Port Extender;
e) Extends the MAC service of a Bridge Port across the interconnected Bridge Port Extenders, including support of Customer Virtual Local Area Networks (C-VLANs).
f) Establishes the requirements of bridge components and systems for the attachment of Bridge Port Extenders;
g) Specifies a protocol to provide for the configuration and monitoring of Bridge Port Extenders by a Controlling Bridge; and
h) Establishes the requirements for Bridge Management to support Port Extension, identifying the managed objects and defining the management operations.

This standard contains state-of-the-art material. The area covered by this standard is undergoing evolution. Revisions are anticipated within the next few years to clarify existing material, to correct possible errors, and to incorporate new related material. Information on the current revision state of this and other IEEE 802 standards may be obtained from:

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

Contents

Figures

Tables

IEEE Standard for
Local and metropolitan area networks—

Virtual Bridged Local Area Networks—
Bridge Port Extension

IMPORTANT NOTICE: Implementers of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, and interference protection practices and all applicable laws and regulations.

This IEEE document is made available for use subject to important notices and legal disclaimers. These notices and disclaimers appear in all publications containing this document and may be found under the heading "Important Notice" or "Important Notices and Disclaimers Concerning IEEE Documents." They can also be obtained on request from IEEE or viewed at http://standards.ieee.org/IPR/disclaimers.html.

1. Overview

This standard specifies a Bridge Port Extender that provides the capability to extend MAC service over an Extended Bridge. This capability may be used, for example, to extend a bridge over multiple physical devices or to extend the MAC service of a bridge to a virtual end station.

1.1 Scope

This standard specifies the devices, protocols, procedures, and managed objects necessary to extend a bridge and its management beyond its physical enclosure using IEEE 802® LAN technologies.

To this end, it:

a) Identifies and isolates traffic between ports within an Extended Bridge;
b) Specifies a tag format for this identification;
c) Establishes an Extended Bridge consisting of a Controlling Bridge and one or more Bridge Port Extenders;
d) Specifies the functionality and the specific requirements of a Bridge Port Extender;
e) Extends the MAC service of a Bridge Port across the interconnected Bridge Port Extenders, including support of Customer Virtual Local Area Networks (C-VLANs).

f) Establishes the requirements of bridge components and systems for the attachment of Bridge Port Extenders;

g) Specifies a protocol to provide for the configuration and monitoring of Bridge Port Extenders by a Controlling Bridge; and

h) Establishes the requirements for Bridge Management to support Port Extension, identifying the managed objects and defining the management operations.

1.2 Purpose

The purpose of this standard is to extend a bridge, and the management of its objects, beyond its physical enclosure using IEEE 802 LAN technologies and interoperable interfaces.

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 the 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 802.1AB™, IEEE Standard for Local and metropolitan area networks—Station and Media Access Control—Connectivity Discovery.[1, 2]

IEEE Std 802.1Q™-2011, IEEE Standard for Local and Metropolitan Area Networks—Media Access Control (MAC) and Virtual Bridged Local Area Networks (as amended).

IEEE Std 802.1Qaz™-2011, IEEE Standard for Local and metropolitan area networks—Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks—Amendment 18: Enhanced Transmission Selection for Bandwidth Sharing Between Traffic Classes.

IEEE Std 802.1Qbb™-2011, IEEE Standard for Local and metropolitan area networks—Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks—Amendment 17: Priority-based Flow Control.

IEEE Std 802.1Qbc™-2011, IEEE Standard for Local and metropolitan area networks—Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks—Amendment 16: Provider Bridging—Remote Customer Service Interfaces.

IEEE Std 802.1Qbe™-2011, IEEE Standard for Local and metropolitan area networks—Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks—Amendment 15: Multiple I-SID Registration Protocol.

IEEE Std 802.1Qbg™-2012, IEEE Standard for Local and metropolitan area networks—Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks—Admendment 21: Edge Virtual Bridging.

IEEE Std 802.3.1™-2011, IEEE Standard for Management Information Base (MIB) Definitions for Ethernet.

IETF RFC 1042, A Standard for the Transmission of IP Datagrams over IEEE 802 Networks, Postel, J., and Reynolds, J., February 1988.[3]

IETF RFC 1390, STD 36, Transmission of IP and ARP over FDDI Networks, Katz, D., January 1993.

IETF RFC 2578, STD 58, Structure of Management Information Version 2 (SMIv2), McCloghrie, K., et al., April 1999.

ISO/IEC TR 11802-5:1997, Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Technical reports and guidelines—Part 5: Media Access Control (MAC) Bridging of Ethernet V2.0 in Local Area Networks.[4]

[1] IEEE publications are available from The Institute of Electrical and Electronics Engineers (http://standards.ieee.org/).

[2] The IEEE standards or products referred to in this clause are trademarks of The Institute of Electrical and Electronics Engineers, Inc.

[3] IETF documents (i.e., RFCs) are available for download at http://www.rfc-archive.org/.

[4] ISO/IEC publications are available from the ISO Central Secretariat (http://www.iso.org/). ISO publications are also available in the United States from the American National Standards Institute (http://www.ansi.org/).

3. Definitions

For the purposes of this document, the definitions in Clause 3 of IEEE Std 802.1Q and the following terms and definitions apply.[5] The *IEEE Standards Dictionary Online* should be consulted for terms not defined in this clause.[6]

Aggregating Port Extender: A Bridge Port Extender that supports the full E-channel Identifier (E-CID) space and is capable of aggregating base Port Extenders.

Base Port Extender: A Bridge Port Extender that supports a subset of the E-channel Identifier (E-CID) space.

Bridge Port Extender: A device used to extend the MAC service of a C-VLAN component to form a Controlling Bridge and to extend the MAC service of a Controlling Bridge to form an Extended Bridge.

Cascade Port: A Port of a Controlling Bridge or Bridge Port Extender that connects to an Upstream Port. In the case of the connection between two Bridge Port Extenders, the Cascade Port is the Port closest to the Controlling Bridge.

Controlling Bridge: A Bridge that supports one or more Bridge Port Extenders.

E-channel: An instance of the MAC service supported by a set of two E-paths forming a bidirectional service. An E-channel is point-to-point or point-to-multipoint.

E-channel Identifier (E-CID): A value conveyed in a E-TAG that identifies an E-channel.

E-path: A configured unidirectional connectivity path between an internal Extended Port and one or more external Extended Ports and/or Upstream Ports. E-paths originating from the Internal Bridge Port Extender can be point-to-point or point-to-multipoint. E-paths originating from an External Bridge Port Extender can be point-to-point or multipoint-to-point.

E-TAG: A tag header with a Tag Protocol Identification value allocated for “IEEE 802.1BR E-Tag Type.”

Extended Bridge: A Controlling Bridge and at least one Bridge Port Extender under the Controlling Bridge’s control.

Extended Port: A Port of a Bridge Port Extender that is not operating as a Cascade Port or Upstream Port. This includes the Ports of a Bridge Port Extender connected via internal LANs to the Port of a C-VLAN component within a Controlling Bridge.

External Bridge Port Extender: A Bridge Port Extender that is not physically part of a Controlling Bridge but is controlled by the Controlling Bridge.

external Extended Port: An Extended Port that is part of an External Bridge Port Extender.

Internal Bridge Port Extender: A Bridge Port Extender that is physically part of a Controlling Bridge.

internal Extended Port: An Extended Port that is part of an Internal Bridge Port Extender.

[5] Information on references can be found in Clause 2.

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

Port Extender Control and Status Agent: The entity within a Bridge Port Extender that implements the Port Extender Control and Status Protocol (PE CSP).

Port Extender Control and Status Protocol (PE CSP): A protocol used between a Controlling Bridge and Bridge Port Extenders that provides the ability of the Controlling Bridge to assert control over and retrieve status information from its associated Bridge Port Extenders.

Replication Group: Within a Controlling Bridge, the set of C-VLAN component Ports connected to a single Bridge Port Extender.

Upstream Port: A Port on a Bridge Port Extender that connects to a Cascade Port. In the case of the connection between two Bridge Port Extenders, the Upstream Port is the Port furthest from the Controlling Bridge.

4. Acronyms and abbreviations

The acronyms and abbreviations in Clause 4 of IEEE Std 802.1Q and the following abbreviations are used in this standard:

E-CID	E-channel Identifier
PCID	Port E-CID
PE CSP	Port Extender Control and Status Protocol
PEISS	Port Extender Internal Sublayer Service

5. Conformance

This clause specifies the mandatory and optional capabilities provided by conformant implementations of this standard.

5.1 Terminology

For consistency with IEEE and existing IEEE 802.1™ standards terminology, requirements placed upon conformant implementations of this standard are expressed using the following terminology:

a) ***shall*** is used for mandatory requirements;

b) ***may*** is used to describe implementation or administrative choices ("may" means "is permitted to," and hence, "may" and "may not" mean precisely the same thing);

c) ***should*** is used for recommended choices (the behaviors described by "should" and "should not" are both permissible but not equally desirable choices).

The PICS proforma (Annex A) reflects the occurrences of the words *shall, may,* and *should* within the standard.

The standard avoids needless repetition and apparent duplication of its formal requirements by using ***is***, ***is not***, ***are***, and ***are not*** for definitions and the logical consequences of conformant behavior. Behavior that is permitted but is neither always required nor directly controlled by an implementor or administrator, or whose conformance requirement is detailed elsewhere, is described by ***can***. Behavior that never occurs in a conformant implementation or system of conformant implementations is described by ***can not***. The word ***allow*** is used as a replacement for the phrase "support the ability for," and the word ***capability*** means "is able to or can be configured to."

5.2 Protocol Implementation Conformance Statement (PICS)

The supplier of an implementation that is claimed to conform to this standard shall complete a copy of the PICS proforma provided in Annex A and shall provide the information necessary to identify both the supplier and the implementation.

5.3 Bridge Port Extender Conformance

A conformant implementation of a Bridge Port Extender shall:

a) Conform to the relevant standard for Media Access Control (MAC) technology implemented at each Port in support of the MAC Internal Sublayer Service (ISS), as specified in 6.6 and 6.7 of IEEE Std 802.1Q;

b) Support the Port Extender Internal Sublayer Service (PEISS) at each Port, as specified in 6.10;

c) Relay and filter frames as specified in 6.11;

d) For each Port, support a Port E-channel Identifier (PCID) value (6.10.4);

e) Allow E-TAG headers to be inserted and removed from relayed frames, as specified in 6.10;

f) Support determination of and encoding of priority and drop eligibility as specified in 6.9; and

g) Support the removal of Customer VLAN tags (C-TAGs) as specified in 6.9.

A conformant implementation of an External Bridge Port Extender shall:

h) Implement the Port Extender Control and Status Protocol (PE CSP) (Clause 9);

i) Implement Link Layer Discovery Protocol (LLDP) (IEEE Std 802.1AB);

j) Implement the LLDP Port Extension Type, Length, Value (TLV) (B.2);

k) Set the destination addresses in all LLDP protocol data units (PDUs) carrying the Port Extension TLV to the Nearest non-TPMR Bridge group address;

l) Process received LLDP PDUs with the destination address set to the Nearest non-TPMR Bridge group address; and

m) Implement the LLDP Port Extension MIB Module (B.3).

A conformant implementation of an External Bridge Port Extender may:

n) Support Priority-based Flow Control (PFC) (5.11 of IEEE Std 802.1Q);

A conformant implementation of an Internal Bridge Port Extender shall:

o) Support PFC (5.11 of IEEE Std 802.1Q) if the C-VLAN component to which it is attached supports PFC.

A conformant implementation of a Bridge Port Extender may:

p) Support congestion notification (CN).

A conformant implementation of a Bridge Port Extender that supports congestion notification shall:

q) Support, on one or more Ports, the creation of at least one Congestion Point (CP) (31.1.1 of IEEE Std 802.1Q);

r) Support, at each Congestion Point, the generation of Congestion Notification Messages (CNMs) (32.7 of IEEE Std 802.1Q) and the removal of Congestion Notification Tags (31.1.1 of IEEE Std 802.1Q);

s) Support the CN Parameters Set and CN Parameters Get messages and associated TLVs within PE CSP (9.8.13, 9.8.14); and

t) Support the Bridge Port Extender specific requirements in 6.16.

NOTE—A Bridge Port Extender operates in conjunction with the Controlling Bridge to implement congestion notification. Specifically, the Bridge Port Extender provides the functionality of the congestion point. The C-VLAN component performs the remaining functionality as specified in IEEE Std 802.1Q.[7]

5.4 Controlling Bridge Conformance

A conformant implementation of a Controlling Bridge shall comprise a conformant C-VLAN component as specified in IEEE Std 802.1Q, and support the instantiation of one or more Internal Bridge Port Extenders (5.3) connected as specified in Clause 8. In addition, the C-VLAN component shall:

a) Implement the PE CSP (Clause 9);

b) Implement LLDP (IEEE Std. 802.1AB);

c) Implement the LLDP Port Extension TLV (B.2);

d) Implement the LLDP Port Extension MIB Module (B.3); and

e) Support the Bridge Port Extension requirements specified in Clause 8.

[7] Notes in text, tables, and figures of a standard are given for information only and do not contain requirements needed to implement this standard.

A conformant implementation of a C-VLAN component that utilizes Bridge Port Extension may:

f) Support the Bridge Port Extension Management Objects (10.2);

g) Support the IEEE-PE MIB module (11.4); and

h) Support Congestion Points within a Bridge Port Extender (9.8.13, 9.8.14).

6. Principles of Bridge Port Extension

This clause:

a) Explains the principal elements of the operation of Bridge Port Extension and the operation of components within supporting systems, and lists the supporting functions;

b) Establishes a Bridge Port Extension architecture that governs the provision of these functions; and

c) Provides a model of Bridge Port Extension operation in terms of processes and entities that support the functions.

The principal elements of Bridge Port Extension are as follows:

d) Bridge Port Extender

e) Controlling Bridge

These elements combine to form an Extended Bridge.

6.1 Bridge Port Extension Overview

Bridge Port Extension introduces a new component referred to as a Bridge Port Extender that operates in conjunction with a bridge comprised of a C-VLAN component (IEEE Std 802.1Q). An Internal Bridge Port Extender is instantiated internal within a bridge system and attaches via internal LANs to Ports that would otherwise be exposed externally. This is illustrated in Figure 6-1.

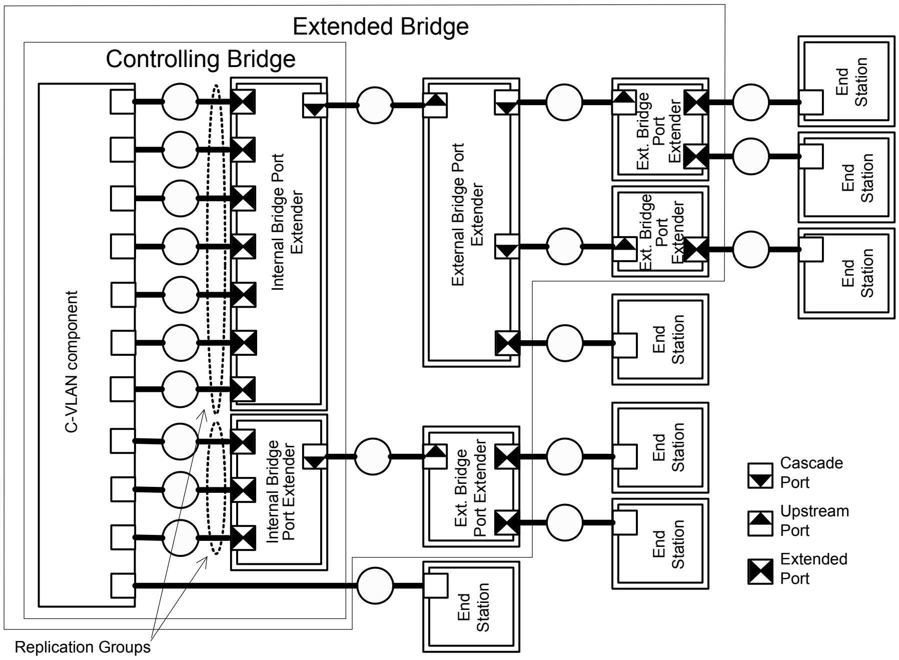


Figure 6-1—Extended Bridge

The Internal Bridge Port Extender aggregates these Ports into a single external Port. An E-TAG is inserted into each frame to provide identification of the port to which each frame belongs.

The aggregated port, referred to as a *Cascade Port,* is attached to an External Bridge Port Extender, or to a tree of Bridge Port Extenders, over a point-to-point LAN. These External Bridge Port Extenders provide a

Port, referred to as an *external Extended Port,* corresponding to a C-VLAN component Port. To the user, this provides the ability to extend the Ports of a physical bridge across point-to-point LANs to remote Bridge Port Extenders, including physical Bridge Port Extenders and Bridge Port Extenders instantiated as part of an end station such as a server.

The bidirectional path between the external Extended Port and the corresponding internal Extended Port is referred to as an *E-channel.* A unidirectional path between an internal Extended Port and one or more external Extended Ports is referred to as an *E-path.* An E-channel is comprised of a pair of congruent E-paths, one in each direction. E-paths are also used in multicast. Therefore, E-paths can be point-to-point, point-to-multipoint (i.e., have a single ingress point with multiple egress points), or multipoint-to-point (i.e., have multiple ingress points with a single egress point). The multipoint-to-point E-path is used to return frames in support of congestion notification. E-channels are identified by an E-channel Identifier (E-CID) that is contained in the E-TAG.

The External Bridge Port Extenders utilize the E-TAG to identify a path to the external port on egress, and to the internal port on ingress. Additionally, the External Bridge Port Extenders perform frame replication in support of multicast traffic (e.g., frames that are to be flooded or frames that are addressed to a group address).

The C-VLAN component and the set of attached Internal Bridge Port Extenders is referred to as a *Controlling Bridge.* The set of the Controlling Bridge and the attached External Bridge Port Extenders is referred to as an *Extended Bridge.* The Controlling Bridge provides the single point of management for the extended bridge. The Port Extender Control and Status Protocol (PE CSP) provides the ability for the Controlling Bridge to manage the Bridge Port Extenders under its control.

6.2 Extended Bridge

An example of an Extended Bridge is illustrated in Figure 6-1.

An Extended Bridge is constructed utilizing a single Controlling Bridge and one or more External Bridge Port Extenders. A Controlling Bridge is constructed from a C-VLAN component that supports Port Extension options (Clause 8) and one or more Bridge Port Extenders connected by internal LANs (6.14 of IEEE Std 802.1Q). Bridge Port Extenders enable the extension of the Controlling Bridge by supplying external Ports that correspond to the internal Ports of the C-VLAN component and by performing replication of frames as necessary. The entire Extended Bridge operates as a single bridge and is managed as a single entity.

Bridge Port Extenders may be attached to other Bridge Port Extenders forming a cascade. In this case, each Bridge Port Extender instantiates a single Upstream Port that provides connectivity towards the Controlling Bridge. This topologically forms a loop-free pruned tree. Therefore, within the Extended Bridge (i.e., between the Controlling Bridge and the Bridge Port Extenders), it is not necessary to run spanning tree. The spanning tree protocol operates on the C-VLAN component and the Bridge Protocol Data Units (BPDUs) transit through the intervening Bridge Port Extenders between the C-VLAN component and the external Extended Ports. The same applies to other protocols such as the Multiple Registration Protocol (MRP) family.

Each link between Bridge Port Extenders or between a Bridge Port Extender and its Controlling Bridge contains one Cascade Port (the Port closest to the Controlling Bridge) and one Upstream Port (the Port furthest from the controlling bridge). The Bridge Port Extender Ports that connect to other LAN components (including those that connect to the internal C-VLAN component, end stations, bridges, routers, and other Extended bridges) are referred to as *Extended Ports.*

Each internal Extended Port is paired with one Cascade or one external Extended Port. An independent E-channel is formed between the internal Extended Port and its paired Cascade or external Extended Port to enable segregation of the traffic for each Port pair as it traverses the common link.

The external Extended Port provides external connectivity to the Extended Bridge. The Upstream Port provides the termination of the E-channel over which frames between the Controlling Bridge and the Control and Status Agent of the attached Bridge Port Extender flow.

Within the Upstream Port, received frames whose E-CID matches the Port's PCID are forwarded by the PEISS to the MAC Service interface and higher layer entities. Conversely, frames initiated by the higher layer entities via the MAC Service are forwarded through the Upstream Port.

For each internal Extended Port, a corresponding Port is instantiated on the C-VLAN component within the Controlling Bridge. This Port on the C-VLAN component is attached to the internal Extended Port via an internal LAN. As a result, all unicast frames emitted from the Port on the C-VLAN component travel through the E-channel to the corresponding external Extended or Cascade Port. Likewise, every frame received on an external Extended Port, or initiated by a Bridge Port Extender on the Upstream Port, travels through the E-channel to the corresponding Port on the C-VLAN component.

A frame entering an external Extended Port has an E-TAG inserted by the ingress Bridge Port Extender. The same is true for a frame entering a Cascade Port initiated by the attached Bridge Port Extender. The E-CID encoded in the E-TAG identifies the E-channel and is maintained with the frame through any intervening Bridge Port Extenders (the E-TAG may be modified at the boundary between a base and aggregating Bridge Port Extender to support both types in a single cascade). Upon reaching the Controlling Bridge, the Internal Bridge Port Extender removes the E-TAG and forwards the frame to the corresponding Port on the C-VLAN component.

E-channels are configured within Bridge Port Extenders using member sets (similar to the method used to configure VLANs in C-VLAN components). A member set is the set of Bridge Port Extender Ports that are part of the E-channel. The Upstream Port of a Bridge Port Extender is implicitly in the member set of all E-channels. Within a given Bridge Port Extender, an E-channel that is used for unicast has exactly one other Port in its member set. Likewise, within a given Bridge Port Extender, an E-channel that is used for multicast has one or more additional Ports in its member set.

To provide efficient support of Multicast, Bridge Port Extenders provide frame replication. The set of C-VLAN bridge component Ports used for remote replication that attach to a single Internal Bridge Port Extender is referred to as a *Replication Group.* For each combination of Ports within the Replication Group to which a frame may need to be forwarded (based on the current state of the filtering database within the C-VLAN component), an E-channel is allocated. In this case, the E-channel is configured as a point-to-multipoint channel. The E-channel terminates within the Internal Bridge Port Extender at exactly one of the internal Extended Ports that share an internal LAN with one of the Ports in the combination. The selection of which Port within the Internal Bridge Port Extender terminates the E-channel is at the discretion of the implementation. The other end terminates at a set of external Extended Ports.

An E-channel member set is used to identify the set of Extended Ports that ultimately transmit a given multicast frame. The relay function within the C-VLAN component of a Controlling Bridge provides a port map in the connection_identifier parameter of the EM_UNITDATA.request. This port map indicates to which external Extended Ports the frame is to be forwarded. This port map travels with the frame through the ISS and across the internal LAN to the Internal Bridge Port Extender. In addition, the C-VLAN component maintains a Remote Replication Registration Table in each Internal Bridge Port Extender. Each entry in this table contains a port map and E-CID. Upon receipt of a frame with a port map included in the connection_identifier, the Internal Bridge Port Extender finds a matching port map in the Remote Replication Registration Table. The PEISS of the Internal Bridge Port Extender uses the E-CID from the matching entry to construct the E-TAG.

The relay function within the C-VLAN component will transmit a multicast frame to all ports of the Internal Bridge Port Extender corresponding to the port map. However, only one internal Extended Port is part of the member set of any given E-channel. Therefore, all but one copy of the frame is discarded.

In addition, if one of the external Extended Ports to which a frame is being forwarded is the Port on which the frame was received and the Port to which the frame is being forwarded is configured to provide reflective relay, ingress_echannel_identifier is set to zero in the connection_identifier parameter. The PEISS within the Internal Bridge Port Extender encodes this value in the E-TAG. As a result, the ingress external Extended Port does not filter the frame.

E-channels do not have an untagged set. Instead the Port type and the value of the per Port parameter PCID determines whether or not the Bridge Port Extender inserts an E-TAG. An Extended Port never inserts an E-TAG. In the case of a Cascade Port or an Upstream Port, an E-TAG is inserted if the E-CID associated with a frame does not match the PCID of the Cascade or Upstream Port.

Internal Bridge Port Extenders are configured directly. External Bridge Port Extenders are configured by the Controlling Bridge using the Bridge PE CSP (Clause 9). The entity within a Bridge Port Extender that processes the PE CSP is the Control and Status Agent.

A message is provided in the PE CSP that allows an External Bridge Port Extender to communicate its MAC_Operational status to the Controlling Bridge. The Controlling Bridge reflects this status in the MAC_Enabled parameter of the corresponding internal Extended Port. The setting of the MAC_Enabled parameter on the internal Extended Port is reflected across the internal LAN to the MAC_Operational parameter of the Port in the C-VLAN component. This provides MAC_Operational propagation from the external Extended Port to the C-VLAN component.

An External Bridge Port Extender sets the MAC-Enabled parameter of all Ports not capable of operating as an Upstream Port to FALSE whenever it is not in contact with the Controlling Bridge. MAC-Enabled is set TRUE upon successful completion of the PE CSP Extended Port Create command (9.8.2).

6.3 Base and aggregating Bridge Port Extenders

Two slightly different versions of Bridge Port Extenders are specified. A base Bridge Port Extender supports up to 4095 point-to-point E-channels and up to 12 287 point-to-multipoint E-channels. A Controlling Bridge assigns E-CIDs within a restricted range for base Bridge Port Extenders. For point-to-point E-channels, the range starts at one and ends at the number of point-to-point E-channels supported by the Bridge Port Extender. For point-to-multipoint E-channels, the range starts at 4096 and ends at 4095 plus the number of point-to-multipoint E-channels supported by the Bridge Port Extender (8.11). As a result, the E-CID can simply be used as a pointer into the Bridge Port Extender’s Filtering Database enabling a simplified implementation.

An aggregating Bridge Port Extender can support up to 1 048 575 point-to-point E-channels and 3 145 727 point-to-multipoint E-channels. A Controlling Bridge can assign E-CIDs sparsely and the Bridge Port Extender forwarding process (6.11) performs a look-up operation on the E-CID to find the corresponding filtering entry (6.12). In addition, an aggregating Bridge Port Extender can provide additional E-CID bits to the E-Tag produced by a downstream base Bridge Port Extender enabling the aggregating Bridge Port Extender to aggregate multiple base Bridge Port Extenders within a single cascade providing expanded scalability.

An aggregating Bridge Port Extender may be placed downstream of a base Bridge Port Extender; however, the E-CID range that may be used by the aggregating Bridge Port Extender is limited to that of the upstream base Bridge Port Extender.

Figure 6-2 illustrates an example Extended Bridge composed of aggregating and base Bridge Port Extenders. In this example, the PCIDs for each port are shown. The PCID is broken down into a triple of GRP bits, extension bits, and base bits separated by periods (see 7.5 and 8.11). The extension bits in a base Bridge Port Extender are always set to zero and are ignored in the E-TAG. Frames sent from the Controlling Bridge to the top End Station are tagged with an E-CID of 0.2.2. The aggregating Bridge Port Extender is configured to forward frames with this E-CID to the top Cascade Port. When the frame arrives at the top base Bridge Port Extender, the extension bits are ignored, and the frame is forwarded to the top Extended Port.

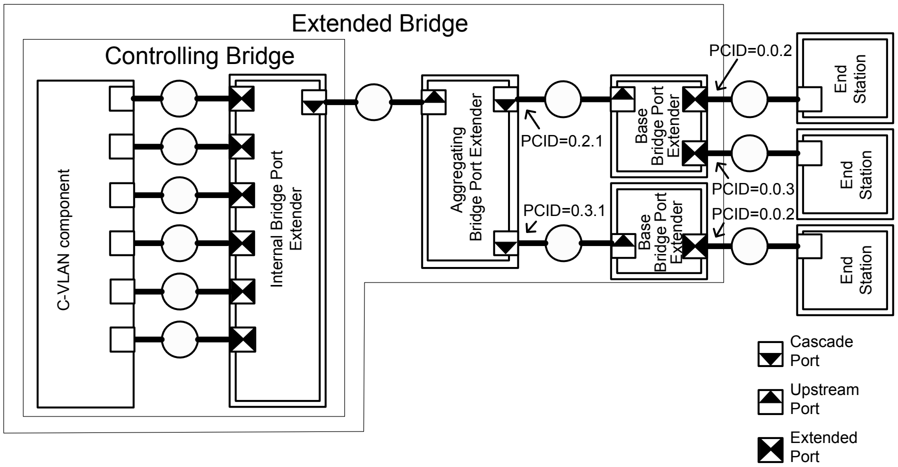


Figure 6-2—Aggregating and base Bridge Port Extenders

Frames sent from the top End Station enter the top base Bridge Port Extender and are tagged with an E-CID of 0.0.2, and forwarded to the aggregating Bridge Port Extender. The aggregating Bridge Port Extender replaces the extension bits with that of the receiving Ports PCID, thus forming an E-CID of 0.2.2. The frame is then forwarded to the Controlling Bridge with an E-CID of 0.2.2.

6.4 Bridge Port Extender operation

The principal elements of Bridge Port Extender operation are as follows:

a) Relay and filtering of frames (6.4.1);

b) Maintenance of the information required to make frame filtering and relaying decisions (6.4.2); and

c) Control of and status collection from items a) and b) (Clause 9).

6.4.1 Relay

A Bridge Port Extender relays individual MAC user data frames between the Upstream Port MACs and the Extended and Cascade Port MACs. The functions that support relaying of frames and maintaining the Quality of Service are as follows:

a) Frame reception (6.7)

b) Discard on received frame in error (6.7.1)

c) Discard of frames that do not carry user data (6.7 of IEEE Std 802.1Q)

d) Priority and drop eligibility decoding from a VLAN TAG, if present (6.9)

e) Classification of each received frame to a particular E-channel (6.10)

f) Frame discard to support segregation of traffic on independent E-paths (6.11.1)

g) Frame discard following the application of filtering information (6.12.1)

h) Forwarding of received frames to other Bridge Port Extender Ports (6.11.4)

i) Selection of traffic class and queuing of frames by traffic class (6.11.5)

j) Frame discard to ensure that a maximum Bridge Port Extender transit delay is not exceeded (6.11.5)

k) Preferential discard of drop eligible frames to preserve quality of service (QoS) for other frames (6.11.5)

l) Selection of queued frames for transmission (6.11.6)

m) Mapping of service data units and Frame Check Sequence recalculation, if required (6.10.6)

n) Frame transmission (6.8)

6.4.2 Filtering and relaying information

A Bridge Port Extender maintains filtering and relaying information for the following purposes:

a) Traffic segregation: to separate communication operating over different E-channels;

b) Traffic reduction: to enable replication of frames within Bridge Port Extenders;

c) Traffic expediting: to classify frames in order to expedite time critical traffic; and

d) Frame format conversion: to insert or remove tag headers as appropriate for the destination LAN and stations.

6.4.2.1 Traffic segregation

A Bridge Port Extender filters frames to restrict them to LANs that belong to the E-channels to which they are assigned and, thus, define the E-channel's extent. The functions that support the use and maintenance of information for this purpose are as follows:

a) Configuration of a PCID for each Port, to associate an E-channel with received frames that do not contain an E-TAG and to identify the E-channel for which frames transmitted by the Port are not to carry an E-TAG; and

b) Configuration of E-channel Registration Entries (6.12).

6.4.2.2 Traffic reduction

A Bridge Port Extender performs replication of multicast frames reducing the traffic on links between Bridge Port Extenders (6.10.7).

6.4.2.3 Traffic expediting

A Bridge Port Extender classifies frames into traffic classes in order to expedite transmission of frames generated by critical or time-sensitive services. The function that supports the use and maintenance of information for this purpose is as follows:

a) Explicit configuration of traffic class information associated with the Ports of the Bridge Port Extender.

6.4.2.4 Conversion of frame formats

A Bridge Port Extender adds and removes VLAN tag headers (6.9) and E-TAGs (6.10) from frames and performs the associated frame translations that may be required. The functions that support the use and maintenance of information for this purpose are as follows:

a) The Bridge Port Extender Tag Handler (6.9);

b) The Bridge Port Extender ISS (6.10); and

c) The Bridge Port Extender filtering database (6.12).

6.5 Bridge Port Extender architecture

A Bridge Port Extender comprises the following:

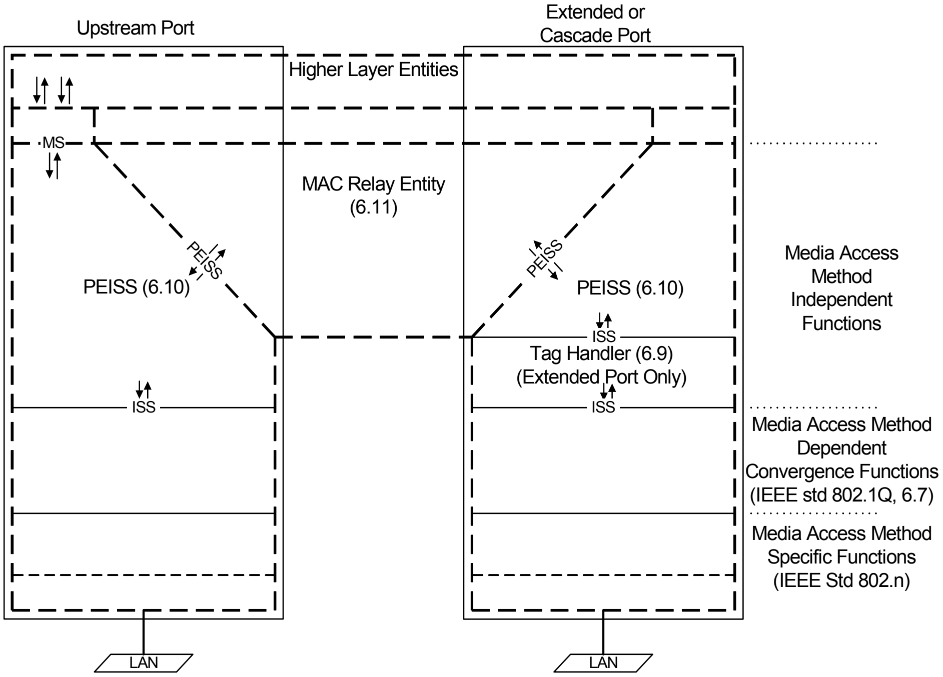
a) A MAC Relay Entity that interconnects the Bridge Port Extender’s Ports;

b) One Upstream Port (External Bridge Port Extender) or one Cascade Port (Internal Bridge Port Extender);

c) At least one additional Port; and

d) Higher layer entities, including at least a Port Extender Control and Status Protocol Entity (External Bridge Port Extender).

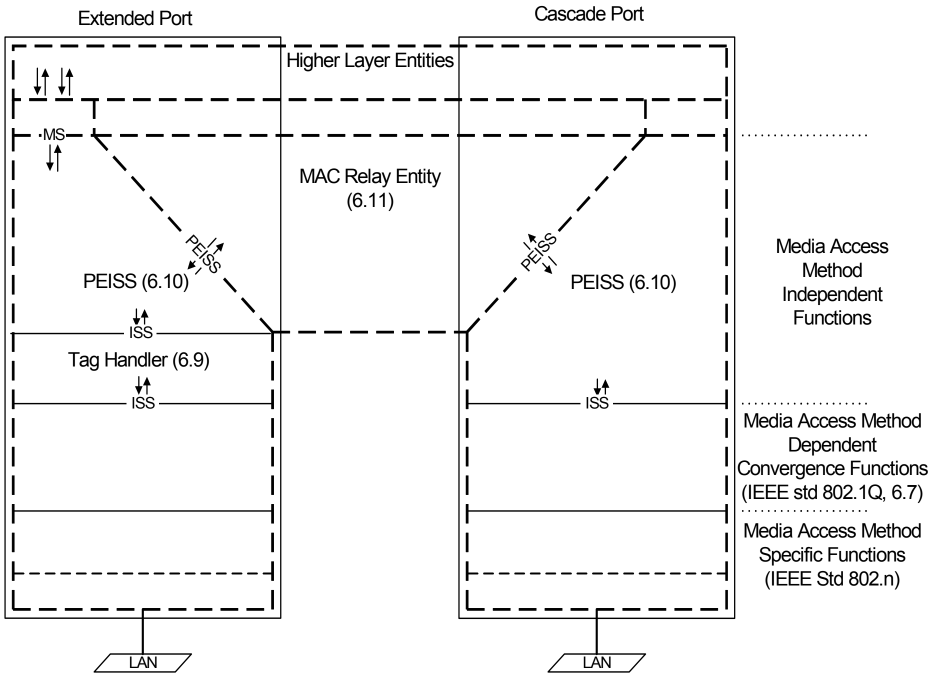
The external and Internal Bridge Port Extender architecture is illustrated in Figure 6-3 and Figure 6-4.[8] The MAC Relay Entity handles the media access method independent functions of filtering frames and relaying frames among Bridge Port Extender Ports. It uses the PEISS (6.10) provided by each Bridge Port Extender Port. Each Bridge Port Extender Upstream Port of External Bridge Port Extenders also functions as an end station providing one instance of the MAC Service. This instance of the MAC Service is provided to a distinct LLC Entity that supports protocol identification, multiplexing, and demultiplexing, for PDU transmission and reception by one or more higher layer entities, including the Port Extender Control and Status Agent. A tag handler function (6.9) on each Extended Port processes C-TAGs.


Upstream Port
Extended or Cascade Port
Higher Layer Entities
MS
MAC Relay Entity (6.11)
PEISS
PEISS
PEISS (6.10)
PEISS (6.10)
ISS
Tag Handler (6.9) (Extended Port Only)
ISS
ISS
Media Access Method Independent Functions
Media Access Method Dependent Convergence Functions (IEEE std 802.1Q, 6.7)
Media Access Method Specific Functions (IEEE Std 802.n)
LAN
LAN

NOTE—The notation “IEEE Std 802.n” in this figure indicates that the specifications for these functions can be found in the relevant standard for the media access method concerned; for example, n would be 3 (IEEE Std 802.3™ [B2]) in the case of Ethernet.

Figure 6-3—External Bridge Port Extender architecture

[8] The numbers in brackets correspond to those of the Bibliography in Annex E.



NOTE—The notation "IEEE Std 802.n" in this figure indicates that the specifications for these functions can be found in the relevant standard for the media access method concerned; for example, n would be 3 (IEEE Std 802.3™ [B2]) in the case of Ethernet.

Figure 6-4—Internal Bridge Port Extender architecture

6.6 Bridge Port Extender Model of operation

The model of operation is simply a basis for describing the functionality of the Bridge Port Extender. It is in no way intended to constrain real implementations of a Bridge Port Extender; these may adopt any internal model of operation compatible with the externally visible behavior that this standard specifies. Conformance of equipment to this standard is purely in respect of observable protocol.

The processes and entities that model the operation of a Bridge Port Extender Port include the following:

a) A Bridge Port Extender Port Transmit and Receive Process (6.8) that:
   1) Receives and transmits frames from and to the attached LAN (6.8);
   2) Determines the format, VLAN-tagged or untagged, of transmitted frames; and
   3) Delivers and accepts frames to and from the MAC Relay Entity and LLC Entities.
b) The LLC Entity that supports Higher Layer Entities such as:
   1) Port Extender Control and Status Protocol (PE CSP) (Clause 9); and
   2) Link Layer Discovery Protocol (LLDP) (IEEE Std 802.1AB).

The processes and entities that model the operation of the MAC Relay Entity are the following:

c) The Forwarding Process (6.11), that:
   1) Filters frames based on their E-CID (6.11.1); and
   2) Forwards received frames that are to be relayed to other Bridge Port Extender Ports.
d) The Filtering Database (6.12), that holds filtering information and supports queries by the Forwarding Process as to whether frames with a given value of E-CID can be forwarded to a given Port.

Figure 6-5 illustrates a single instance of frame relay between the Ports of a Bridge Port Extender with two Ports. Within an Internal Bridge Port Extender, frame relay takes place between a Cascade Port and the Extended Ports. Within an External Bridge Port Extender, frame relay takes place between the Upstream Port and the set of Cascade and Extended Ports.

Figure 6-6 illustrates the operation of the Port Extender Control and Status Protocol Agent.

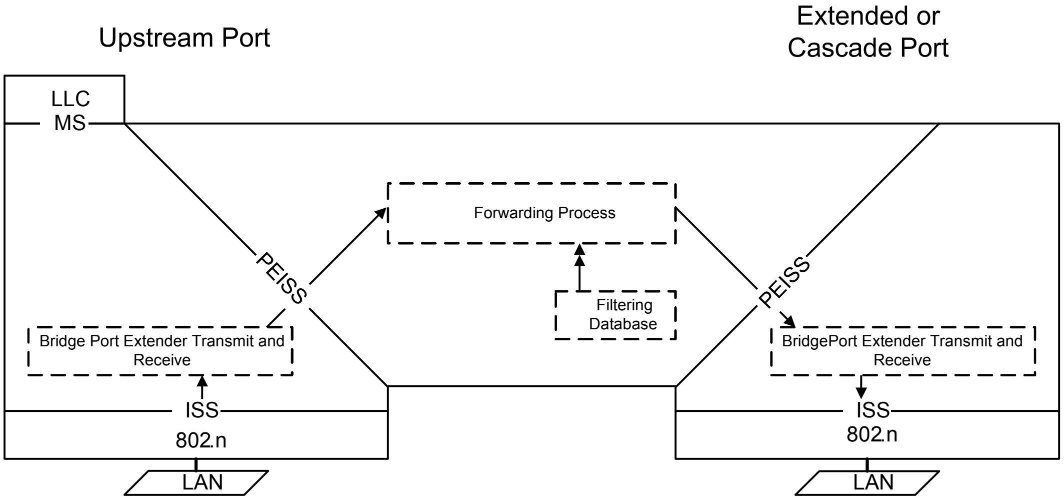


Figure 6-5—Relaying MAC frames in an External Bridge Port Extender

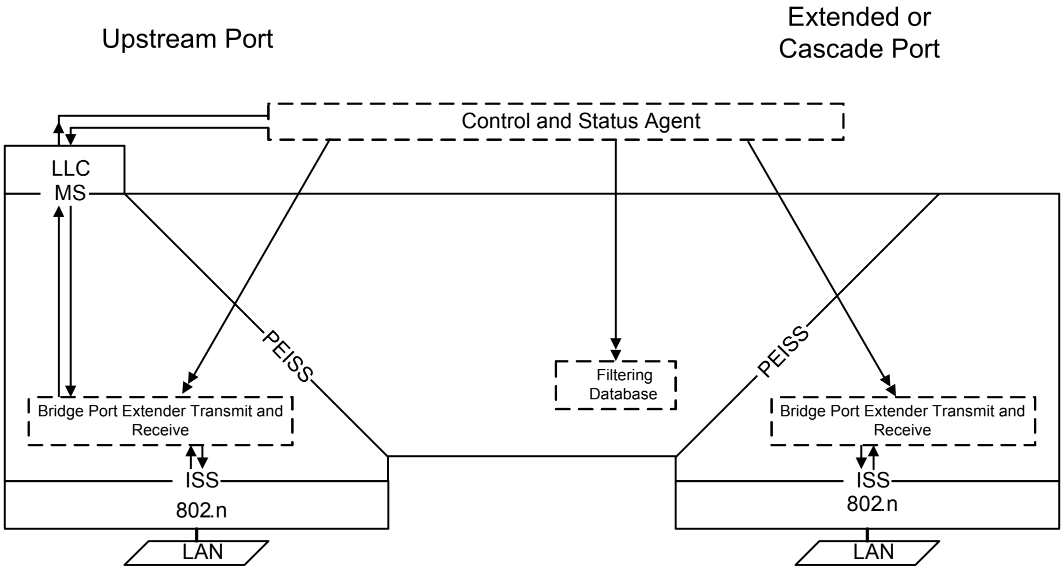


Figure 6-6—Operation of Bridge Port Extender Control and Status Protocol Agent

6.7 Bridge Port Extender Frame Reception

Frame reception comprises the following functions:

a) Discard of frames received in error (6.7.1); and
b) Discard of frames that do not contain user data (6.7 of IEEE Std 802.1Q).

6.7.1 Frame Loss

The MAC Service does not guarantee the delivery of Service Data Units. Frames transmitted by a source station arrive, uncorrupted, at the destination station with high probability. The operation of a Bridge Port Extender introduces minimal additional frame loss.

A frame transmitted by a source station can fail to reach its destination station as a result of the following:

a) Frame corruption during physical layer transmission or reception;
b) Frame discard by a Bridge Port Extender because
   1) It is unable to transmit the frame within some maximum period of time and, hence, must discard the frame to prevent the Bridge Port Extender transit delay (6.11.5) from being exceeded;
   2) It is unable to continue to store the frame due to exhaustion of internal buffering capacity as frames continue to arrive at a rate in excess of that at which they can be transmitted;
   3) The size of the service data unit carried by the frame exceeds the maximum supported by the MAC procedures employed on the LAN to which the frame is to be relayed;
   4) The frame was received on a port that is not in the frame's E-channel member set; and
   5) No potential transmission Port is in the member set of the E-channel identified by the frame's E-CID.

6.8 Bridge Port Extender Transmit and Receive

The Port Transmit and Receive process supports the attachment of the Bridge Port Extender Port to a network. It comprises the following functions:

a) Mapping between the PEISS (6.10) provided to the MAC Relay Entity, and the ISS (6.6 of IEEE Std 802.1Q), adding, recognizing, interpreting, and removing E-TAGs;
b) A shim that uses and provides the ISS (6.6 of IEEE Std 802.1Q), recognizing and interpreting C-TAGs as specified in 6.9; and
c) Connectivity between the following ISS access points:
   1) That provided by the MAC Entity for the LAN attached to the Port, as specified in 6.7 of IEEE Std 802.1Q; and
   2) That supporting the MAC Relay Entity; and
   3) On an Upstream Port, an LLC entity.

6.8.1 Port connectivity

Figure 6-7 illustrates Port connectivity. Each M_UNITDATA.indication provided by the ISS access point for an attached LAN shall result in a corresponding M_UNITDATA.indication with identical parameters at each of the access points supporting the MAC Relay and Higher Layer Entities. Each M_UNITDATA.request from the ISS access point supporting the MAC Relay Entity shall result in a corresponding M_UNITDATA.indication with identical parameters at the access point for the LAN. Each M_UNITDATA.request from an ISS access point supporting a Higher Layer Entity shall result in a corresponding M_UNITDATA.indication with identical parameters at the access point for the LAN.

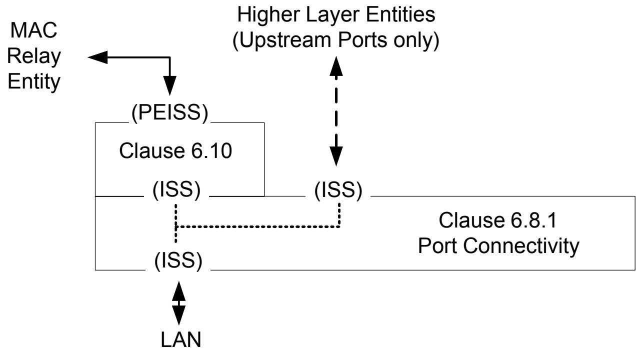


Figure 6-7—Port connectivity

6.8.2 Support of Higher Layer Entities

The MAC Service is provided to a Higher Layer Entity using one of the ISS access points provided for that purpose by the Port connectivity function (6.8.1).

Each ISS M_UNITDATA.indication with a destination MAC address that is either the individual address of a MAC service access point (MSAP) provided by the Port or a group address used by the attached LLC Entity shall cause an MA_UNITDATA.indication at that MSAP with destination address, source address, MSDU, and priority parameters identical to those in the M_UNITDATA.indication. No other indications or frames give rise to indications to the MAC Service user.

Each MA_UNITDATA.request at the MSAP shall result in an M_UNITDATA.request at the ISS access point with identical destination address, source address, MSDU, and priority parameters. Each MA_UNITDATA.request at the MSAP with a destination MAC address that is either the individual address of the MSAP or a group address used by the attached LLC Entity shall also result in an MA_UNITDATA.indication at the MSAP with identical destination address, source address, MSDU, and priority parameters.

NOTE—The requirement to reflect self-addressed MA_UNITDATA.request primitives back as MA_UNITDATA.indications to the MSAP at which they were received exists because the MAC Service expects this behavior (ISO/IEC 15802-1 [B5]) but the ISS does not expect it or provide it.

6.9 Bridge Port Extender tag handler

The functions specified in this subclause comprise a shim that uses an ISS SAP supported by a MAC entity (6.7 of IEEE Std 802.1Q) to provide functions to handle C-TAGs within Bridge Port Extenders. This shim is utilized and instantiated only on Extended Ports, see Figure 6-3 and Figure 6-4.

This shim provides the following functions:

a) Upon a M_UNITDATA.indication within an external Extended Port, determination of a frame’s Priority and Drop Eligibility; and

b) Upon a M_UNITDATA.request within an external Extended Port, deletion of the C-TAG if the corresponding VLAN is in the shim’s untagged set.

NOTE—On egress from the C-VLAN component of a Controlling Bridge, all frames on point-to-multipoint E-channels contain a C-TAG. Deletion of the C-TAG for Extended Ports that are in the untagged set of the corresponding VLAN is handled by the Extended Port of the External Bridge Port Extender. This is done to facilitate remote replication in which a subset of the Ports to which the frame is replicated may be in the untagged set.

To provide this capability, the following parameters are configured for each Extended Port within Bridge Port Extenders:

c) use_dei: Use_DEI parameter as specified in 6.9.3 of IEEE Std 802.1Q; and
d) untagged_vlan_list: Applies only to External Bridge Port Extenders. A list of customer VLANs for which this Port is in the VLAN's untagged set. An External Bridge Port Extender shall support at least one entry in this list.

The setting of these parameters by the Controlling Bridge is specified in Clause 8.

6.9.1 Data Indications

On receipt of an M_UNITDATA.indication primitive from the lower ISS SAP, an M_UNITDATA.indication primitive shall be invoked to the upper ISS SAP with parameter values determined as follows:

a) The priority and drop_eligible parameters of the M_UNITDATA.indication are set as follows:
   1) If the mac_service_data_unit contains a C-TAG, the drop_eligible parameter and received priority are derived from the VLAN tag as specified in 6.9.3 of IEEE Std 802.1Q, and then the priority parameter is regenerated using the received priority as specified in 6.9.4 of IEEE Std 802.1Q; otherwise
   2) The priority and drop_eligible indication are set to the corresponding parameter in the M_UNITDATA.indication from the lower ISS SAP.
b) All other parameters are set to the value in the corresponding parameter in the M_UNITDATA.indication from the lower ISS SAP.

6.9.2 Data Requests

On receipt of an M_UNITDATA.request primitive from the upper ISS SAP destined to an external Extended Port, the following operations shall be performed:

a) Extract the VLAN Identifier (VID) from the C-TAG in the M_UNITDATA.request mac_service_data_unit parameter;
b) If the VID is in the untagged_vlan_list, then remove the C-TAG from the M_UNITDATA.request mac_service_data_unit parameter;
c) Invoke a M_UNITDATA.request to the lower ISS SAP with the following parameters:
   1) The mac_service_data_unit is set to the corresponding parameter in the received data as updated by this process;
   2) The frame_check_sequence parameter is set to the value in the corresponding parameter in the receive data if the mac_service_data_unit was not modified; otherwise the frame_check_sequence parameter is set to indicate invalid;
   3) All other parameters are set to the corresponding parameter in the M_UNITDATA.request from the upper ISS SAP.

On receipt of an M_UNITDATA.request primitive from the upper ISS SAP destined to an internal Extended Port, an M_UNITDATA.request to the lower ISS SAP shall be invoked with all parameters copied from that of the upper ISS SAP.

6.10 Bridge Port Extender Internal Sublayer Service

The Bridge Port Extender Internal Sublayer Service (PEISS) provides for the insertion and removal of E-TAGs.

The PEISS provides the same service status and point-to-point parameters as the ISS (6.6.2 and 6.6.3 of IEEE Std 802.1Q).

6.10.1 Service primitives

The unit-data primitives that define this service are as follows:

```
PEM_UNITDATA.indication      (
                             destination_address,
                             source_address,
                             mac_service_data_unit,
                             priority,
                             drop_eligible,
                             ingress_echannel_identifier,
                             echannel_identifier,
                             frame_check_sequence,
                             service_access_point_identifier,
                             connection_identifier
                             )

PEM_UNITDATA.request         (
                             destination_address,
                             source_address,
                             mac_service_data_unit,
                             priority,
                             drop_eligible,
                             ingress_echannel_identifier,
                             echannel_identifier,
                             frame_check_sequence,
                             service_access_point_identifier,
                             connection_identifier
                             )
```

The destination_address, source_address, mac_service_data_unit, priority, drop_eligible, service_access_point_identifier, connection_identifier, and frame_check_sequence parameters are as defined for the ISS.

NOTE—Some of the functions supporting the PEISS may result in changes to the mac_service_data_unit or other parameters used to construct a frame. The original FCS associated with a frame is invalidated if there are changes to any fields of the frame, if fields are added or removed, or if bit ordering or other aspects of the frame encoding have changed. An invalid FCS is signaled in the PEISS by an unspecified value in the frame_check_sequence parameter. This signals the need for the FCS to be regenerated according to the normal procedures for the transmitting MAC. The options for regenerating the FCS under these circumstances are discussed in Annex F of IEEE Std 802.1D [B1].

The echannel_identifier parameter carries the E-CID.

The ingress_echannel_identifier carries the E-CID of the E-channel over which a frame was received.

6.10.2 Status Parameters

The PEISS also makes available the MAC_Enabled and MAC_Operational status parameters that reflect the operational state and administrative controls over each instance of the service provided. The values of these parameters are mapped directly from the values made available by the ISS (6.6.2 of IEEE Std 802.1Q).

6.10.3 Point-to-point parameters

The PEISS also makes available the operPointToPointMAC and adminPointToPointMAC status parameters that reflect the point-to-point status of each instance of the service provided and provide administrative control over the use of that information. The values of these parameters are mapped directly from the values made available by the ISS (6.6.3 of IEEE Std 802.1Q).

6.10.4 Support of the PEISS

The PEISS is supported by E-TAG (7.5) insertion and removal functions that in turn use the ISS (6.6 and 6.7 of IEEE Std 802.1Q).

Each Extended and Cascade Port shall support a PCID value that provides the default E-CID value for that Port.

Each Extended and Cascade Port of an aggregating External Bridge Port Extender shall support a useDefault Boolean. This value indicates whether or not the aggregating Bridge Port Extender is to provide the extension bits of the E-CID.

6.10.5 Data indications

On receipt of an M_UNITDATA.indication primitive from the ISS, the indication is discarded if all of the following are true:

a) The Port is a Cascade Port;

b) The initial octets of the mac_service_data_unit contain a valid E-TAG header (7.5); and

c) The E-CID value within the header is zero or 0x3F FFFF.

Otherwise a PEM_UNITDATA.indication primitive is invoked, with parameter values determined as follows:

The destination_address, source_address, and frame_check_sequence parameters carry values equal to the corresponding parameters in the received data indication. The values of the remaining parameters are affected by the contents of the E-TAG if present.

The value of the mac_service_data_unit parameter is as follows:

d) If the frame contains an E-TAG then the value used is equal to the value of the received mac_service_data_unit following removal of the E-TAG; otherwise,

e) The value used is equal to the value of the received mac_service_data_unit.

The value of the echannel_identifier parameter is as follows:

f) If the mac_service_data_unit contains an E-TAG and the Port is a Cascade Port of an external aggregating Bridge Port Extender and useDefault is FALSE, the value decoded as specified in 7.5.2; otherwise

g) If the mac_service_data_unit contains an E-TAG and the Port is a Cascade Port of an external aggregating Bridge Port Extender and useDefault is TRUE, the value decoded as specified in 7.5.2 except that extension bits of the echannel_identifier shall be set to the extension bits of the PCID; otherwise

h) If the mac_service_data_unit contains an E-TAG, and the Port is a Cascade Port or and Upstream Port of an external base Bridge Port Extender, the value decoded as specified in 7.5.2 except that extension bits of the echannel_identifier shall be set to zero; otherwise

i) If the connection_identifier contains a port map, the E-CID value extracted from the Remote Replication Registration Table as specified in 6.10.7; otherwise,

j) The value of the PCID for the Port.

NOTE—The connection_identifier is used to pass a Port Map across an internal LAN (6.14 of IEEE Std 802.1Q) from a C-VLAN component to a Bridge Port Extender within a Controlling Bridge for use with remote replication. Otherwise, the port map is not present in the connection_identifier.

The value of the ingress_echannel_identifier parameter is as follows:

k) If the frame contains an E-TAG, the value decoded as specified in 7.5.1; otherwise

l) If the connection_identifier contains a ingress E-CID, then that E-CID, otherwise

m) Zero.

The value of the drop_eligible and priority parameters are determined as follows:

n) If the frame contains an E-TAG, the value of the drop_eligible parameter and the received priority value are set to the DEI and Priority Code Point (PCP) values contained in the E-TAG, respectively; otherwise,

o) The received priority value and the drop_eligible parameter value are the values in the M_UNITDATA.indication.

6.10.6 Data requests

On invocation of a PEM_UNITDATA.request primitive by a user of the PEISS, if the ingress_echannel_identifier matches the PCID, the PEM_UNITDATA.request is discarded. Otherwise M_UNITDATA.request primitive is invoked, with parameter values as follows:

The destination_address, source_address, drop_eligible, and priority parameters carry values equal to the corresponding parameters in the received data request.

An E-TAG is inserted as the initial octets of the mac_service_data_unit parameter if the Port is an Upstream Port and the echannel_identifier does not match the Port's PCID. An E-TAG is inserted as the initial octets of the mac_service_data_unit parameter if the Port is a Cascade Port and if any of the following conditions are true:

a) The echannel_identifier is in the range of 0x00 0001 to 0x0F FFFF and does not match the PCID; or

b) The echannel_identifier is greater than 0x0F FFFF and the Port is in the member set of more than one E-channel whose E-CID is in the range of 0x00 0001 to 0x0F FFFF.

If an E-TAG is inserted, it shall be formatted as described in Clause 7.

NOTE—A Bridge Port Extender does not have direct knowledge if one of its Ports is operating as a Cascade Port versus an Extended Port, nor does it need to. A sufficient but not necessary condition to infer that a Port is operating as a Cascade Port is if the Port is in the member set of more than one unicast E-channel. However, a Port operating as a Cascade Port connected to a Bridge Port Extender that does not have any Extended Ports active would reside in only one such member set. An E-TAG should never be inserted on an Extended Port. Furthermore, an E-TAG is inserted on Cascade Ports except for the frames destined to the attached Bridge Port Extender (i.e., the frames on the E-channel identified by the Cascade Port PCID). Condition a) covers the Cascade Port case for unicast. Condition b) covers the multicast case for most Cascade Ports. Neither case covers a Cascade Port that is attached to a Bridge Port Extender with no extended Ports. As a result, no frame will be sent to the attached Bridge Port Extender with an E-TAG, which is the desired behavior.

The ingress_echannel_identifier is set to zero if all of the following conditions are true:

c) The port is a Cascade or Extended Port of an external aggregating Bridge Port Extender;

d) useDefault is TRUE;

e) The value of echannel_identifier is greater than or equal to 0x10 0000 (i.e., a point-to-multipoint E-channel); and

f) The extension bits of the ingress_echannel_identifier do not match those of the PCID.

The values of the echannel_identifier, ingress_echannel_identifier, priority, and drop_eligible parameters are used to determine the contents of the E-TAG, in accordance with 7.5.

The remaining octets of the mac_service_data_unit parameter are those accompanying the PEM_UNITDATA.request.

If the data request is a consequence of relaying a frame and the MAC type of the Port differs from that used to receive the frame, they are modified, if necessary, in accordance with the procedures described in ISO/IEC TR 11802-5, IETF RFC 1042, and IETF RFC 1390.

The value of the frame_check_sequence parameter is determined as follows:

g) If the frame_check_sequence parameter received in the data request is either unspecified or still carries a valid value, then that value is used; otherwise,

h) The value used is either derived from the received FCS information by modification to take account of the conditions that have caused it to become invalid, or the unspecified value is used.

6.10.7 Support of remote replication by the PEISS

A remote replication registration table shall be provided within each Internal Bridge Port Extender. This table is configured by the C-VLAN component to which it is attached (8.10.1). It is used by the PEISS of the Internal Bridge Port Extender to determine the E-CID to be used for remote replication.

Each entry in the Remote Replication Registration Table shall comprise the following:

a) The E-CID of the E-channel to which the filtering information applies; and

b) A Port Map, with a control element for each outbound Port in the Replication Group. This Port Map operates as the key to identify the Remote Replication Registration Entry. Each control element may be set to *filter* or *forward*.

The PEISS uses this table to determine the E-CID and Ingress_E-CID when processing data indications (6.10.5).

6.11 Bridge Port Extender Forwarding Process

Each frame submitted to the MAC Relay Entity shall be forwarded subject to the constituent functions of the Forwarding Process (Figure 6-8). Each function is described in terms of the action taken for a given frame received on a given Port (termed “the reception Port”). The frame can be forwarded for transmission on some Ports (termed “transmission Ports”) and discarded without being transmitted at the other Ports.

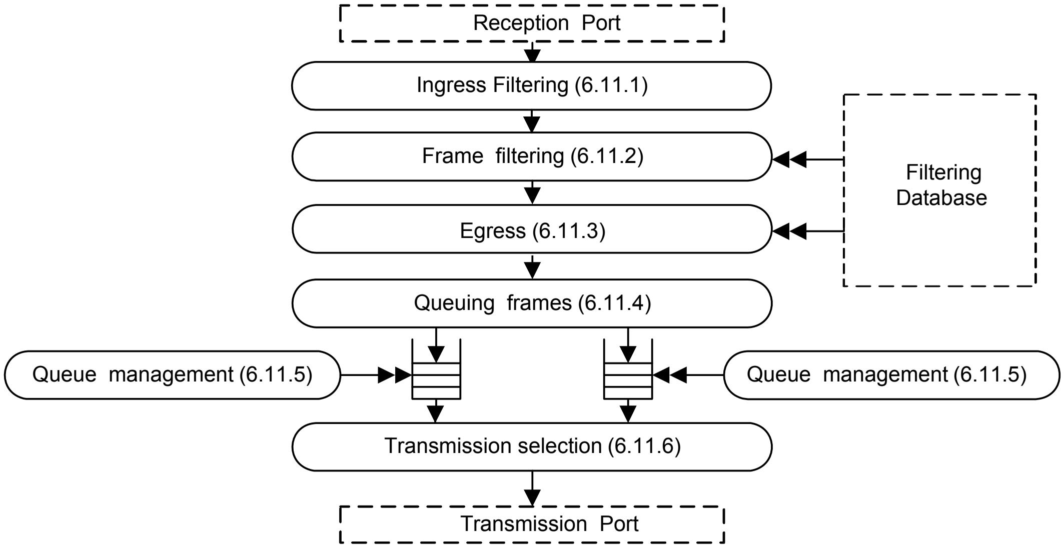


Figure 6-8—Bridge Port Extender Forwarding Process functions

6.11.1 Ingress filtering

Each internal Extended Port supports ingress filtering. A frame received on an internal Extended Port that is not in the member set associated with the frame's echannel_identifier shall be discarded.

6.11.2 Frame filtering

The Forwarding Process takes filtering decisions, i.e., reduces the set of potential transmission Ports (6.12.1), for each received frame on the basis of

a) The E-CID; and
b) The information contained in the Filtering Database for that E-CID

in accordance with the definition of the Filtering Database entry (6.12). The required behavior is specified in 6.12.1.

6.11.3 Egress

The Forwarding Process shall queue each received frame to each of the potential transmission Ports that is present in the member set (6.12.1) for the frame's E-CID.

NOTE—The Forwarding Process is modeled as receiving a frame as the parameters of a data indication and transmitting through supplying the parameters of a data request. Queueing a frame awaiting transmission amounts to placing the parameters of a data request on an outbound queue.

6.11.4 Queuing frames

The Forwarding Process provides storage for queued frames, awaiting an opportunity to submit these for transmission. The order of frames received on the same Bridge Port Extender Port shall be preserved for frames with a given E-CID and priority combination.

The Forwarding Process provides one or more queues for a given Bridge Port Extender Port, each corresponding to a distinct traffic class. Each frame is mapped to a traffic class using the priority to traffic class mapping table for the Port and the frame's priority. Up to eight traffic classes may be supported, allowing separate queues for each priority.

NOTE 1—Different numbers of traffic classes may be implemented for different Ports. Ports with media access methods that support a single transmission priority, such as CSMA/CD, can support more than one traffic class.

NOTE 2 —A queue in this context is not necessarily a single, first in, first out (FIFO) data structure. A queue is a record of all frames of a given traffic class awaiting transmission on a given Bridge Port. The structure of this record is not specified. The transmission selection algorithm (6.11.6) determines which traffic class, among those classes with frames available for transmission, provides the next frame for transmission. The method of determining which frame within a traffic class is the next available frame is not specified beyond conforming to the frame ordering requirements of this subclause. This allows a variety of queue structures such as a single FIFO, or a set of FIFOs with one for each pairing of ingress and egress Ports (i.e., Virtual Output Queuing), or a set of FIFOs with one for each VLAN or priority, or hierarchical structures.

In a congestion aware Bridge Port Extender (Clause 30 of IEEE Std 802.1Q), the act of queuing a frame for transmission on a Port can result in the Forwarding Process generating a Congestion Notification Message (CNM). The CNM is injected back into the Forwarding Process as if it had been received on that Port.

A frame shall be filtered if the Ingress_E-CID parameter matches that of the PCID of the port to which it is to be queued.

6.11.5 Queue management

A Bridge Port Extender shall provide a global Bridge Port Extender transit delay parameter.

A frame queued for transmission on a Port shall be removed from that queue:

a) Following a transmit data request. No further attempt is made to transmit the frame on that Port even if the transmission is known to have failed; or
b) If it is necessary to ensure that the maximum Bridge Port Extender transit delay will not be exceeded at the time at which the frame would subsequently be transmitted.

The frame may be removed from the queue, and not subsequently transmitted,

c) By a queue management algorithm that attempts to improve the QoS provided by deterministically or probabilistically managing the queue depth based on the current and past queue depths.

Removal of a frame from a queue for any particular Port does not affect queuing of that frame for transmission on any other Port.

NOTE—Applicable queue management algorithms include RED (random early detection), and WRED (weighted random early detection) (IETF RFC 2309). If the Bridge Port Extender supports drop precedence, i.e., is capable of decoding or encoding the drop_eligible parameter from or to the PCP field of a VLAN tag (6.9.3 of IEEE Std 802.1Q) or the E-TAG, the algorithm should exhibit a higher probability of dropping frames with drop_eligible True.

The probability of removing a frame with drop_eligible True shall not be less than that of removing a frame with drop_eligible False, all other conditions being equal. If a queue management algorithm is implemented, it should preferentially discard frames with drop_eligible True.

6.11.6 Transmission selection

For each Port, frames are selected for transmission on the basis of the traffic classes that the Port supports and the operation of the transmission selection algorithms supported by the corresponding queues on that Port. For a given Port and supported value of traffic class, frames are selected from the corresponding queue for transmission if and only if:

a) The operation of the transmission selection algorithm supported by that queue determines that there is a frame available for transmission; and

b) For each queue corresponding to a numerically higher value of traffic class supported by the Port, the operation of the transmission selection algorithm supported by that queue determines that there is no frame available for transmission.

In a Port that supports PFC, a frame of priority n is not available for transmission if that priority is paused {i.e., if Priority_Paused[n] is TRUE (36.1.3.2 of IEEE Std 802.1Q)} on that port. When Transmission Selection is running above Link Aggregation, a frame of priority n is not available for transmission if that priority is paused on the physical port to which the frame is to be distributed.

NOTE 1—Two or more priorities can be combined in a single queue. In this case if one or more of the priorities in the queue are paused, it is possible for frames in that queue not belonging to the paused priority to not be scheduled for transmission.

NOTE 2—Mixing PFC and non-PFC priorities in the same queue results in non-PFC traffic being paused causing congestion spreading, and therefore is not recommended.

The strict priority transmission selection algorithm defined in 6.11.6.1 shall be supported as the default algorithm for selecting frames for transmission. The Enhanced Transmission Selection algorithm defined in 6.11.6.3 may be supported in addition to the strict priority algorithm.

The Transmission Selection Algorithm Table for a given Port assigns, for each traffic class that the Port supports, the transmission selection algorithm that is to be used to select frames for transmission from the corresponding queue. Transmission Selection Algorithm Tables may be managed, and allow the identification of vendor-specific transmission selection algorithms. The transmission selection algorithms are identified in the Transmission Selection Algorithm Table by means of integer identifiers, as defined in Table 8-5 of IEEE Std 802.1Q.

6.11.6.1 Strict priority algorithm

For a given queue that supports strict priority transmission selection, the algorithm determines that there is a frame available for transmission if the queue contains one or more frames.

6.11.6.2 Credit-based shaper algorithm

Bridge Port Extenders do not support the Credit-based shaper algorithm. Queues assigned the Credit-based shaper algorithm use the Strict priority algorithm (6.11.6.1) instead.

This does not imply that an Extended Bridge cannot support the Credit-based shaper algorithm. However, if supported, the algorithm itself is implemented in the C-VLAN component. Treating the queues as strict priority in the Bridge Port Extenders gives equivalent behavior.

6.11.6.3 Enhanced Transmission Selection (ETS) algorithm

If ETS is enabled for a traffic class, transmission selection is performed based on the allocation of bandwidth to that traffic class. Bandwidth is distributed amongst ETS traffic classes that support enhanced transmission selection algorithm such that each traffic class is allocated available bandwidth in proportion to its TCBandwidth (Clause 37 of IEEE Std 802.1Q).

For a given queue that supports enhanced transmission selection, the algorithm determines that there is a frame available for transmission if the following conditions are all true:

a) The queue contains one or more frames;

b) The ETS algorithm (38.3 of IEEE Std 802.1Q) determines that a frame should be transmitted from the queue; and

c) There are no frames available for transmission for any queues running the Strict priority algorithm.

NOTE—Support of ETS is optional for C-VLAN components. An Internal Bridge Port Extender attached to a C-VLAN component that does not support ETS would never have ETS enabled for a traffic class and therefore the ETS algorithm need not be supported in the Internal Bridge Port Extender.

6.12 Bridge Port Extender Filtering Database

The Filtering Database supports queries by the Forwarding Process to determine whether received frames, with a given value of E-CID, are to be forwarded through a given potential transmission Port (6.12.1). The Filtering Database contains filtering information in the form of filtering entries that are configured by the Controlling Bridge (Clause 44 of IEEE Std 802.1Q).

Each entry in the Filtering Database is a E-CID Registration Entry and specifies the forwarding of frames with a particular E-CID. Each entry comprises:

a) The E-CID to which the filtering information applies;

b) A Port Map, consisting of a control element for each Port, specifying

    1) Forward, if the Port is in the member set of the E-channel; or

    2) Filter, if the Port is not in the member set of the E-channel.

The Upstream Port is a member of all E-channels. Therefore, the Port Map element corresponding to the Upstream Port is permanently configured to Forward.

6.12.1 Querying the Filtering Database

Potential transmission ports are identified as follows:

a) For a frame received on an Upstream Port, each Port, other than the reception Port, is identified as a potential transmission Port; otherwise

b) Within an External Bridge Port Extender, only the Upstream Port is identified as a potential transmission Port; otherwise

c) Within an Internal Bridge Port Extender, only the Cascade Port is identified as a potential transmission Port.

The E-CID assigned to a relayed frame identifies the applicable entry in the filtering database. The Port Map indicates the action to be taken (Filter or Forward) at each potential transmission Port.

NOTE—Implementations of base Bridge Port Extenders can assume, by virtue of the requirements placed on Controlling Bridges, that unicast E-CIDs will be assigned by the Controlling Bridge in the range of one up through the number of E-CIDs supported. Furthermore, a base Bridge Port Extender may assume that multicast E-CIDs will be assigned starting at 0x10 0000 continuing up to 0x10 0FFF, then from 0x20 0000 through 0x20 0FFF, and finally from 0x30 0000 through 0x30 0FFE. As a result, an implementation can choose to simply ignore the extension bits in the E-Tag and use the remaining bits as a pointer into the filtering database. Aggregating Bridge Port Extenders cannot make such assumptions and therefore the implementations accept arbitrary E-CID assignments by the Controlling Bridge.

6.13 Determination of the Upstream Port

An External Bridge Port Extender shall have exactly one Port acting as the Upstream Port at any given time. However, a Bridge Port Extender may provide more than one Port that is capable of acting as the Upstream Port.

If more than one Port is capable of acting as the Upstream Port, the Bridge Port Extender shall determine the Port to act as the Upstream Port as follows:

a) Determine the subset of the Ports capable of acting as Upstream Ports that are attached to peer Ports capable of acting as a Cascade Port;
b) Select the Peer Port with the numerically smallest cascade_port_priority (B.2.1);
c) If multiple Ports have the numerically smallest cascade_port_priority, select the Peer Port with the numerically lowest PE CSP MAC address of those Ports (B.2.3).

If a Port that is acting as an Upstream Port becomes unavailable (e.g., due to a link failure), the process is repeated to select a new Upstream Port.

Additional methods, such as manual configuration, may be provided.

An Internal Bridge Port Extender shall have no Ports operating as an Upstream Port.

Ports that are not selected by this method are available for use as Extended or Cascade Ports.

6.14 Upstream Port Addressing

A separate individual MAC Address is associated with each instance of the MAC Service provided to the LLC Entity of the Upstream Port. That MAC Address is used as the source address of frames transmitted by the LLC Entity, including the Port Extender Control and Status Agent. This address is communicated using LLDP and the PE TLV (B.2.3).

Media access method specific procedures can require the transmission and reception of frames that use an individual MAC Address associated with the Bridge Port, but neither originate from nor are delivered to a MAC Service user. Where an individual MAC Address is associated with the provision of an instance of the MAC Service by the Port, that address can be used as the source and/or the destination address of such frames, unless the specification of the media access method specific procedures requires otherwise.

6.14.1 Unique identification of a Bridge Port Extender

A unique 48-bit Universally Administered MAC Address, termed the Bridge Port Extender Address, shall be assigned to each Bridge Port Extender. The Bridge Port Extender Address may be the individual MAC Address of the Upstream Port. This address is communicated using LLDP and the PE TLV (B.2.2).

6.14.2 Points of attachment and connectivity for Higher Layer Entities

The Higher Layer Entities in a Bridge Port Extender, such as the Control and Status Agent (6.1), are modeled as attaching directly to one or more individual LANs connected by the Bridge's Ports, in the same way that any distinct end station is attached to the network.

6.15 Bridge Port Extender Initialization

Bridge Port Extenders shall be initialized upon power-on and when specified by the Port Extender Control and Status Protocol (Clause 9). Initialization shall be accomplished by setting the Bridge Port Extender parameters to the values indicated in Table 6-1 and the parameters associated with each Bridge Port Extender Port to the values indicated in Table 6-2.

.

Table 6-1—Bridge Port Extender Initialization

Parameter	Initial value
Member set for each E-channel	Empty

Table 6-2—Bridge Port Extender Port Initialization

Parameter	Initial value
PCID	One
Transmission Selection Algorithm Table	Each entry set to strict priority (see Table 8-5 of IEEE Std 802.1Q)
Priority to traffic class mapping table	Recommended values in Table 8-4 of IEEE Std 802.1Q
Priority-based Flow Control	Disabled for all priorities
use_dei	Zero
useDefault	FALSE
untagged_vlan_list	Empty

6.16 Support of Congestion Points

A Bridge Port Extender can optionally support Congestion Points in support of congestion notification as specified by Clause 30 through Clause 33 of IEEE Std 802.1Q. The Congestion Point shall operate as specified in those clauses except that:

a) References to the EISS refer instead to the PEISS;

b) References to EM_UNITDATA refer instead to PEM_UNITDATA;

c) The GenerateCnmPdu() procedure specified in 32.9.4 of IEEE Std 802.1Q copies all unspecified PEM_UNITDATA parameters from the PEM_UNITDATA.request to the CNM PDU PEM_UNITDATA.indication; and

d) The GenerateCnmPdu() procedure specified in 32.9.4 of IEEE Std 802.1Q inserts a C-TAG in the mac_service_data_unit parameter of the PEM_UNITDATA.indication.

7. Tagged frame format

This clause specifies the format of the tags added to and removed from user data frames by the tag encoding and decoding functions that support the Port Extender Internal Sublayer Service (PEISS) (6.10).

7.1 Representation and encoding of tag fields

The representation and encoding of tag fields in this clause conform to that specified in 9.2 of IEEE Std 802.1Q.

7.2 Tag format

Each tag comprises the sequential information elements specified in 9.3 of IEEE Std 802.1Q.

7.3 Tag Protocol Identifier (TPID) formats

The TPID in this clause conforms to 9.4 of IEEE Std 802.1Q.

7.4 Tag Protocol Identification

An E-TAG is specified that is used at Cascade and Upstream Ports of a Bridge Port Extender. A distinct EtherType (Table 7-1) has been allocated for use in the TPID field of the E-TAG so that it can be distinguished from other tag headers and other protocols.

Table 7-1—E-TAG EtherType allocation

Tag Type	Name	Value
E-TAG	IEEE 802.1BR E-Tag Type	89-3F

7.5 E-TAG Control Information

The E-TAG TCI field (Figure 7-1) is six octets in length.

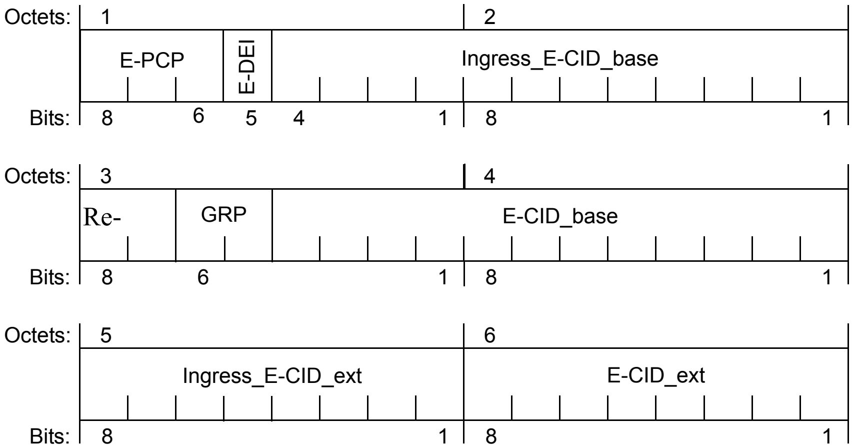


Figure 7-1—E-TAG TCI format

This tag encodes the priority, drop_eligible, ingress_echannel_identifier, and echannel_identifier parameters of the corresponding PEM_UNITDATA.request (6.10) as unsigned binary numbers in the E-TAG TCI fields as follows:

a) *Priority Code Point (E-PCP)*—This 3-bit field encodes the priority parameter of the service request primitive associated with this frame using the same encoding as specified for VLAN tags in IEEE Std 802.1Q.

b) *Drop Eligible Indicator (E-DEI)*—This 1-bit field encodes the drop_eligible parameter of the service request primitive associated with this frame.

c) *Ingress_E-CID_base*—This 12-bit field encodes part of the ingress_echannel_identifier parameter of the service request primitive associated with this frame (7.5.1).

d) *GRP*—This 2-bit field encodes part of the echannel_identifier parameter of the service request primitive associated with this frame (7.5.2).

e) *E-CID_base*—This 12-bit field encodes part of the echannel_identifier parameter of the service request primitive associated with this frame (7.5.2).

f) *Ingress_E-CID_ext*—This 8-bit field encodes part of the ingress_echannel_identifier parameter of the service request primitive associated with this frame (7.5.1).

g) *E-CID_ext*—This 8-bit field encodes part of the echannel_identifier parameter of the service request primitive associated with this frame (7.5.2).

h) *Reserved*—Set to zero on transmit, ignored on receive.

7.5.1 Encoding of the ingress_echannel_identifier

The ingress_echannel_identifier parameter is 22 bits in length. An aggregating Port Extender encodes the bits in the E-TAG as follows:

a) Bits 22-21 are always zero and are not encoded in the E-TAG;

b) Bits 20-13 are encoded in the Ingress_E-CID_ext field; and

c) Bits 12-1 are encoded in the Ingress_E-CID_base field.

A base Port Extender encodes the bits in the E-TAG as follows:

d) Bits 22-21 are always zero and are not encoded in the E-TAG;

e) Bits 20-13 are set to zero in response to a PEM_UNITDATA.indication and ignored in a PEM_UNITDATA.request;

f) The Ingress_E-CID_ext field is set to zero in response to a PEM_UNITDATA.request and ignored in a PEM_UNITDATA.indication; and

g) Bits 12-1 are encoded in the Ingress_E-CID_base field.

7.5.2 Encoding of the echannel_identifier

The echannel_identifier parameter is 22 bits in length. An aggregating Port Extender encodes the bits in the E-TAG as follows:

a) Bits 22-21 are encoded in the GRP field;

b) Bits 20-13 are encoded in the E-CID_ext field; and

c) Bits 12-1 are encoded in the E-CID_base field.

A base Port Extender encodes the bits in the E-TAG as follows:

d) Bits 22-21 are encoded in the GRP field;

e) Bits 20-13 are set to zero in response to a PEM_UNITDATA.indication and ignored in a PEM_UNITDATA.request;

f) The E-CID_ext field is set to zero in response to a PEM_UNITDATA.request and ignored in a PEM_UNITDATA.indication; and

g) Bits 12-1 are encoded in the E-CID_base field.

8. Support of Bridge Port Extension by C-VLAN components

Bridge Port Extension describes the cascaded attachment of one or more Bridge Port Extenders to a Controlling Bridge such that the collection of devices provides the functionality of a single Bridge (referred to as an *Extended Bridge*). Figure 8-1 illustrates the interconnection of a C-VLAN component and an Internal Bridge Port Extender forming a Controlling Bridge, and its interconnection with External Bridge Port Extenders forming an Extended Bridge.

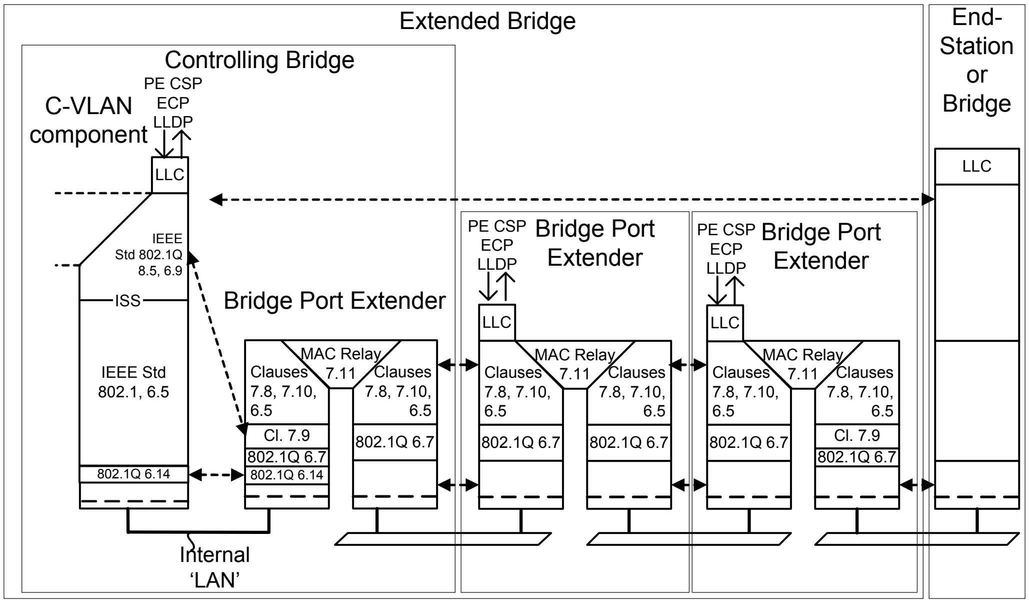


Figure 8-1—Internal organization of the MAC sublayer in an Extended Bridge

This clause describes and specifies the configuration of the following aspects of Bridge Port Extension:

a) Use of tags

b) Bridge Port Extension Port types

c) Controlling Bridge Cascade Ports

d) Bridge Port Extender Upstream Ports

e) Bridge Port Extender Extended Ports

f) Bridge Port Extender Cascade Ports

g) Traffic isolation

h) Remote replication

An Internal Bridge Port Extender refers to the Bridge Port Extender that is part of the Controlling Bridge. External Bridge Port Extender refers to Bridge Port Extenders external to the Controlling Bridge that make up the Extended Bridge.

This clause specifies the configuration actions the Controlling Bridge performs on the Bridge Port Extenders under its control. Configuration of Bridge Port Extenders that are part of the Controlling Bridge is done directly by the Controlling Bridge utilizing a Layer Management Interface. Configuration of Bridge Port Extenders external to the Controlling Bridge is done utilizing the Bridge Port Extender Control and Status Protocol (PE CSP) specified in Clause 9. In addition, this clause specifies actions that the Controlling Bridge takes upon detection of the attachment of a Bridge Port Extender. This detection shall be accomplished utilizing the Port Extension TLV (Clause D of IEEE Std 802.1Q) and LLDP. The destination address of all

LLDP PDUs carrying the Port Extension TLV shall be set to the Nearest non-TPMR Bridge group address. Finally, this clause specifies actions that the Controlling Bridge takes upon detection of or deletion of Extended and Cascade Ports on the attached Bridge Port Extenders.

8.1 Use of Tags

An E-TAG is used between Bridge Port Extenders and the Controlling Bridge. E-TAGs are used to uniquely identify the Extended Port as the frame travels between Bridge Port Extenders and the Controlling Bridge. On egress, the E-TAG may identify a group of Extended Ports to which the frame is destined. This identification is accomplished by an E-channel Identifier (E-CID) that is carried in the E-TAG. An E-CID value in the range 0x00 0001 to 0x0F FFFF identifies a point-to-point E-channel to either carry control and status PDUs between the Internal Bridge Port Extender and an External Bridge Port Extender (8.3), or to carry unicast frames between the Extended Ports of an External Bridge Port Extender and an Internal Bridge Port Extender (8.5). An E-TAG with an E-CID value in the range 0x10 0000 to 0x3F FFFE identifies a point-to-multipoint E-channel to carry multicast frames from the Extended Port of an Internal Bridge Port Extender to the Extended Ports of an External Bridge Port Extender(s) (8.10).

E-TAGs are also used by Bridge Port Extenders to identify the Extended Port from which the frame entered the Extended Bridge so that the frame on egress may be filtered from this Port.

E-TAGs are added and removed by the Extended Ports of Bridge Port Extenders. Therefore, they are present exclusively on the LANs between Bridge Port Extenders. They are not present on the LANs that connect the C-VLAN component to the Internal Bridge Port Extender within a Controlling Bridge.

8.2 Bridge Port Extension Port Types

Figure 6-1 illustrates the Ports utilized in an example Extended Bridge.

Bridge Port Extension defines a number of Port types, each providing the different capabilities needed to construct an Extended Bridge. Initially, the Controlling Bridge provides interfaces via its internal C-VLAN component. Upon detection of the connection of an External Bridge Port Extender, the Controlling Bridge instantiates an Internal Bridge Port Extender between the C-VLAN component Port and the Upstream Port of the External Bridge Port Extender. The instantiation of the Internal Bridge Port Extender provides the capability of receiving and transmitting E-tagged frames. Bridge Port Extenders provide three types of Ports, as follows:

a) Upstream Port: The Bridge Port Extender Upstream Port provides connectivity to the Controlling Bridge Cascade Port or to the Cascade Port of another Bridge Port Extender;

b) Cascade Port: The Cascade Port is used exclusively to provide connectivity to the Upstream Port of a cascaded Bridge Port Extender; and

c) Extended Port: Internal Extended Ports provide connectivity to the Ports of the C-VLAN component. External Extended Ports operate as Ports of the Extended Bridge. Each internal Extended Port is linked via an E-channel to an external Extended Port. Additional E-channels provide linkage between an internal Extended Port and multiple external Extended Ports in support of multicast frame delivery.

PE CSP PDUs flow between the C-VLAN component and External Bridge Port Extenders via the Cascade Port of the instantiated Internal Bridge Port Extender.

8.3 Internal Bridge Port Extender Cascade Ports

Figure 8-2 illustrates an example of the connection of an External Bridge Port Extender to a Controlling Bridge.

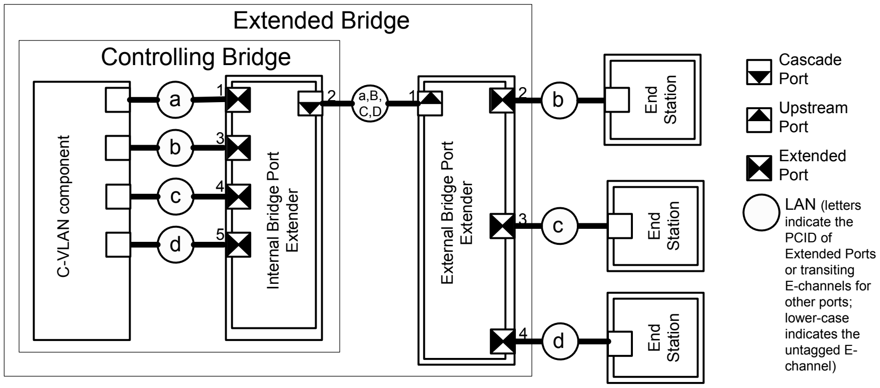

Extended Bridge
Controlling Bridge
C-VLAN component
a
b
c
d
1
3
4
5
Internal Bridge Port Extender
2
a,B,
C,D
1
External Bridge Port Extender
2
3
4
b
c
d
End Station
End Station
End Station
Cascade Port
Upstream Port
Extended Port
LAN (letters indicate the PCID of Extended Ports or transiting E-channels for other ports; lower-case indicates the untagged E-channel)

Figure 8-2—Extended Bridge Interconnection

For each directly attached External Bridge Port Extender, the Controlling Bridge shall:

a) Instantiate an Internal Bridge Port Extender between the Port of the C-VLAN component to which the External Bridge Port Extender was connected and the External Bridge Port Extender;

b) Allocate an E-CID as specified in 8.11 to identify the E-channel that is to carry frames through the instantiated Internal Bridge Port Extender to and from the External Bridge Port Extender; and

NOTE 1—The scope of the allocated E-CID is local to the Internal Bridge Port Extender and all External Bridge Port Extenders connected to it either directly or through a cascade, therefore, it is permissible to use the same E-CID for this purpose across multiple Internal Bridge Port Extenders.

NOTE 2—This E-channel carries frames to and from the External Bridge Port Extender itself for management purposes. Additional E-channels are allocated to carry frames through the External Bridge Port Extender to its Extended and Cascade Ports.

c) Maintain the internal Extended Port, the Cascade Port of the Internal Bridge Port Extender, and the Internal Bridge Port Extender parameters as specified in Table 8-1.

NOTE 3—As a result of this process, a separate Internal Bridge Port Extender is instantiated for each directly attached External Bridge Port Extender. Bridge Port Extenders connected to Cascade Ports of other External Bridge Port Extenders do not create additional Internal Bridge Port Extenders within the Controlling Bridge.

Table 8-1—Bridge Port Extender parameter settings

Object	Parameter	Set to	PE CSP Command
Internal Extended Port, external Extended Port, and Cascade Port			
	PCID	Allocated E-CID	Extended Port Create
	Transmission Selection Algorithm Table	Transmission Selection Algorithm Table of the C-VLAN component Port (8.6.8 of IEEE Std 802.1Q)	Port Parameters Set
	Priority to traffic class mapping table	Priority to traffic class mapping table of the C-VLAN component Port (8.6.6 of IEEEStd 802.1Q)	Port Parameters Set
	PFC (36.1.3.2 of IEEE Std 802.1Q)	PFC setting of the C-VLAN component Port (36.1.3.2 of IEEE Std 802.1Q)	Port Parameters Set
	Enhanced Transmission Selection Bandwidth Table	Set to match the Enhanced Transmission Selection Bandwidth Table of the C-VLAN component Port (37.2 of IEEE Std 802.1Q)	Port Parameters Set
	use_dei	The use_dei parameter of the C-VLAN component Port (6.9.3 of IEEE Std 802.1Q)	Port Parameters Set
	Priority Code Point Selection (6.9.3 of IEEE Std 802.1Q)	Priority Code Point Selection of the C-VLAN component Port (6.9.3 of IEEE Std 802.1Q)	Port Parameters Set
	Priority Code Point Decoding Table	Set to match the Priority Code Point Decoding Table in the C-VLAN component Port (6.9.3 of IEEE Std 802.1Q)	Port Parameters Set
	useDefault [a]	Set to TRUE if this is a Cascade Port and the extension bits of the E-CID are to be determined from the PCID	Port Parameters Set
Upstream Port			
	Transmission Selection Algorithm Table	Same as peer Cascade Port	Port Parameters Set
	Priority to traffic class mapping table	Same as peer Cascade Port	Port Parameters Set
	PFC (36.1.3.2 of IEEE Std 802.1Q)	Same as peer Cascade Port	Port Parameters Set
	Enhanced Transmission Selection Bandwidth Table	Same as peer Cascade Port	Port Parameters Set
Internal Extended Port			
	untagged_vlan_list	empty	NA
	MAC_Enabled (6.6 of IEEE Std 802.1Q)	Initialize to FALSE, then as specified by PE CSP	Status Parameter Set

Table 8-1—Bridge Port Extender parameter settings *(continued)*

Object	Parameter	Set to	PE CSP Command
External Extended Port			
	untagged_vlan_list	Include all the VLANs for which the C-VLAN component Port is a member of the untagged set (8.8.2 of IEEE Std 802.1Q).	Port Parameters Set
External Bridge Port Extender			
	Member set of the E-channels identified by the allocated E-CID	Include Extended Port, remove all other Ports.	Extended Port Create
	Bridge Port Extender transit delay	Maximum bridge transit delay (see 6.5 of IEEE Std 802.1Q)	Transit Delay Set
Internal Bridge Port Extender			
	Member set of the E-channels identified by the allocated E-CID	Include Extended Port and Cascade Port, remove all other Extended Ports.	NA
	Bridge Port Extender transit delay	Maximum bridge transit delay (6.5 of IEEE Std 802.1Q)	NA
Intervening Bridge Port Extenders (i.e., External Bridge Port Extenders in a cascade between the Controlling Bridge and other External Bridge Port Extenders.			
	Member set of the E-channels identified by the allocated E-CID	Include intervening Cascade Ports	E-channel Register

[a] This parameter applies only to Cascade Ports of aggregating Bridge Port Extenders.

8.4 Bridge Port Extender Upstream Ports

An External Bridge Port Extender provides exactly one Upstream Port. This Port attaches to the Cascade Port of another Bridge Port Extender (either internal or external). If attached to any other Port, the External Bridge Port Extender does not establish communication utilizing the PE CSP. In this case and due to the Bridge Port Extender initialization requirements, the Bridge Port Extender will not relay frames.

The Controlling Bridge shall maintain the parameters of the Upstream Port in accordance with Table 8-1.

NOTE 1—This requirement does not preclude the use of link aggregation in the Upstream Port and Cascade Port. If link aggregation is utilized, the Upstream Port and Cascade Port refer to the aggregated Ports.

NOTE 2—Since the Upstream Port is implicitly a member of all E-channel member sets, it requires no specific configuration by the Controlling Bridge. The E-channel previously allocated to the Extended Port to which a new Bridge Port Extender is attached, and therefore now becomes a Cascade Port, becomes the E-channel used to carry the Bridge PE CSP and LLDP PDUs between the Controlling Bridge and the Control and Status Agent in the Bridge Port Extender.

8.5 External Extended Ports

Figure 8-3 illustrates an example Extended Bridge consisting of a Controlling Bridge and four External Bridge Port Extenders along with the E-channel configuration.

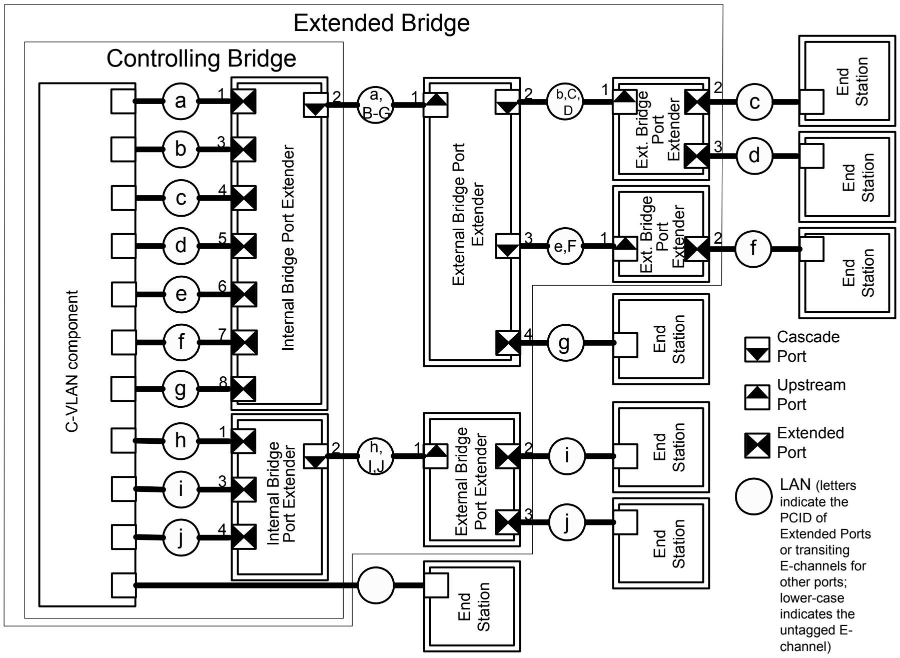

Extended Bridge
Controlling Bridge
C-VLAN component
a
b
c
d
e
f
g
h
i
j
1
3
4
5
6
7
8
Internal Bridge Port Extender
2
a,
B-G
1
External Bridge Port Extender
2
3
4
b,C,
D
e,F
g
1
Ext. Bridge Port Extender
2
3
c
d
1
Ext. Bridge Port Extender
2
f
End Station
End Station
End Station
End Station
1
3
4
Internal Bridge Port Extender
2
h,
I,J
1
External Bridge Port Extender
2
3
i
j
End Station
End Station
End Station
Cascade Port
Upstream Port
Extended Port
LAN (letters indicate the PCID of Extended Ports or transiting E-channels for other ports; lower-case indicates the untagged E-channel)

Figure 8-3—Cascaded Bridge Port Extenders

External Extended Ports are those Ports that are connected to stations or other bridges and establish connectivity to the Extended Bridge. These include all External Bridge Port Extender Ports except for the single Upstream Port (8.4) and Ports connected to other Bridge Port Extenders (i.e., Cascade Ports).

For each external Extended Port, the Controlling Bridge shall:

a) Instantiate a Port on the C-VLAN component;

b) Instantiate an Extended Port on the Internal Bridge Port Extender connected to the Port instantiated on the C-VLAN component utilizing an internal LAN (6.14);

c) Allocate an E-CID as specified in 8.11 that is unique within the scope of the Internal Bridge Port Extender to identify the E-channel between the instantiated Port on the Internal Bridge Port Extender and the external Extended Port; and

d) Maintain the internal Extended Port, the external Extended Port, the Internal Bridge Port Extender, and the External Bridge Port Extender parameters as specified in Table 8-1.

NOTE 1—This standard does not specify if or when the de-instantiation of these Ports, Bridge Port Extenders, and internal links occurs as a result of a link to a Bridge Port Extender becoming inactive.

NOTE 2—This standard does not specify the mechanism by which the initial port configuration settings such as flow control, transmission selection, etc., are determined. Some possibilities include standard bridge defaults, management configuration, through the execution of the Data Center Bridging eXchange protocol (Clause 38 of IEEE Std 802.1Q).

Frames transiting this E-channel contain an E-TAG that is inserted by the first Bridge Port Extender and removed by the last Bridge Port Extender along the channel.

8.6 External Bridge Port Extender Cascade Ports

When a Bridge Port Extender Upstream Port is connected to an Extended Port on another Bridge Port Extender, the Extended Port becomes, by definition, a Cascade Port. The Controlling Bridge shall set the useDefault parameter, if applicable, to TRUE if the extension bits of the E-CID are to be determined from the Cascade Port’s PCID. No other configuration changes are required to effect this transition. The Controlling Bridge shall perform ongoing configuration to maintain consistency between Cascade Ports and their corresponding C-VLAN Port as specified in Table 8-1.

8.7 Traffic isolation

Figure 8-4 illustrates the traffic isolation provided within an Extended Bridge.

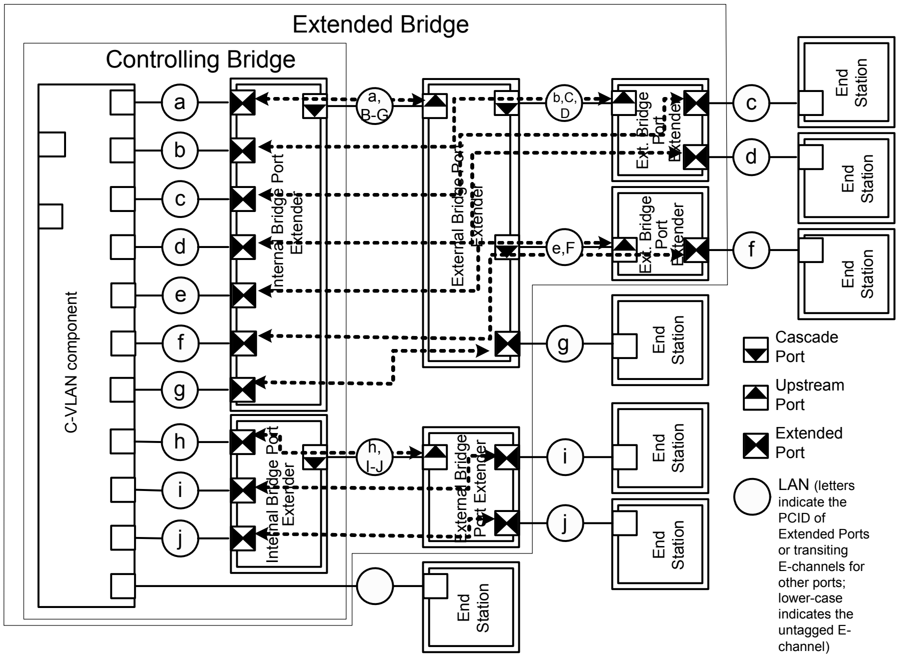

Extended Bridge
Controlling Bridge
C-VLAN component
a
b
c
d
e
f
g
h
i
j
Internal Bridge Port Extender
Internal Bridge Port Extender
a, B-G
External Bridge Port Extender
b,C, D
e,F
Ext. Bridge Port Extender
Ext. Bridge Port Extender
h, I-J
External Bridge Port Extender
End Station
Cascade Port
Upstream Port
Extended Port
LAN (letters indicate the PCID of Extended Ports or transiting E-channels for other ports; lower-case indicates the untagged E-channel)

Figure 8-4—Extended Bridge traffic isolation

Isolation of data frames belonging to different C-VLAN component Ports is achieved by creating a unique E-channel for each Port and:

a) Ensuring that each Extended Port is configured with a PCID that represents the E-CID of the E-channel associated with that Port;

b) On ingress, ensuring that all frames transferred through Cascade and Upstream Ports of the Extended Bridge carry E-TAGs with the E-CID set to the PCID of the Extended Bridge ingress Port; and

c) On egress, ensuring that all frames transferred through Cascade and Upstream Ports of the Extended bridge carry E-TAGs with the E-CID identifying the E-channel whose member set includes the Extended Bridge egress Port or the set of Extended Bridge egress Ports.

8.8 Support of Port Extension by a C-VLAN component MAC Relay

This subclause specifies the additional requirements related to the MAC Relay for frames that are not being forwarded using the remote replication capability. See 8.10.3 for the additional requirements related to the MAC Relay for frames that utilize the remote replication capability.

In support of Port Extension within the C-VLAN component of a Controlling Bridge, the connection_identifier in the EM_UNITDATA.indication primitive is used to carry two additional pieces of information:

a) ingress_echannel_identifier; and

b) echannel_identifier

These fields are present in all EM_UNITDATA.indications sent to the MAC Relay corresponding to frames that were received by a Bridge Port Extender.

8.9 Remote replication

Remote replication is a capability provided to the Controlling Bridge by Bridge Port Extenders within an Extended Bridge. Utilizing this capability, a Controlling Bridge directs the replication of frames within the Bridge Port Extenders to multiple Ports (e.g., frames addressed to group addresses or flooded frames).

This capability is provided using E-channels. An E-channel configured for remote replication forms a point-to-multipoint bi-directional channel between the internal Extended Port, through one or more Bridge Port Extenders, and a set of external Extended Ports. E-channels used by remote replication are identified by an E-CID with a value in the range of 0x10 0000 to 0x3F FFFE (values less than this are reserved for E-CIDs associated with point-to-point E-channels). The Upstream Port is a member of all E-channel member sets. The Extended and Cascade Ports within the E-channel member set comprise the Ports to which a frame carrying the corresponding E-CID is to be replicated. The E-TAG of a replicated frame is removed by each external Extended Port.

The set of C-VLAN component Ports used for remote replication that originate from a single Internal Bridge Port Extender is referred to as a *Replication Group.* The Controlling Bridge's assignment of E-CIDs for remote replication shall be unique within a Replication Group.

A C-VLAN component establishes E-channels through the attached Bridge Port Extenders for every combination of paths over which a frame may need to be replicated based on the current state of the filtering database.

NOTE—This does not imply that an E-channel must be created for every possible combination of Extended and Cascade Ports. An E-channel is only required for each combination of Extended Ports to which a frame may be forwarded based on the current state of the filtering database and other internal bridge operations beyond the scope of this standard that may impact the final Port set.

To utilize remote replication, the C-VLAN component maintains a remote replication registration table (8.10.1) in each Internal Bridge Port Extender. This table contains a list of port maps and corresponding E-CIDs. For each frame that is to be delivered using remote replication, a port map is provided in the connection_identifier parameter of the EM_UNITDATA.request and is passed through the EISS to the ISS M_UNITDATA.request. The frame is then transmitted on the corresponding Ports within the Replication Group across the internal LANs to the Internal Bridge Port Extender. The connection_identifier is also passed on this internal LAN (6.14). These frames are filtered from all but one of the Extended Ports on the receiving Internal Bridge Port Extender based on the configuration of the E-channel member set. The Internal Bridge Port Extender finds an entry in the remote replication registration table with a matching port map. The E-CID from this entry is used as the echannel_identifier parameter in the PEM_UNITDATA.request for the construction of the E-Tag. In addition, an ingress_echannel_identifier is provided by the E-Tag if necessary to filter the frame from one of the external Extended Ports (6.11.4). The External Bridge Port Extenders then replicate the frame to all Cascade and Extended Ports that are in the member set of the E-channel.

8.10 Support of Remote Replication by a Controlling Bridge

8.10.1 Remote Replication Registration Table

A remote replication registration table is contained within each Internal Bridge Port Extender. This table is configured by the C-VLAN component to which it is attached. It is used by the Bridge Port Extender to determine the E-CID to be used for remote replication.

Each entry in the Remote Replication Registration Table comprises:

a) The E-CID of the E-channel to which the filtering information applies; and
b) A Port Map, with a control element for each outbound Port in the Replication Group. This Port Map operates as the key to identify the Remote Replication Registration Entry. Each control element may be set to *filter* or *forward*.

The addition, modification, or removal of entries in the filtering database of the C-VLAN component of the Controlling Bridge can change the combination of Ports from which a frame is to be filtered within the Ports of a Replication Group.

For each combination that contains at least two Ports within the same Replication Group to which a frame is to be forwarded, the C-VLAN component shall maintain a Remote Replication Registration entry in the corresponding Internal Bridge Port Extender. For each combination that contains at least two ports, regardless of their membership in Replication Groups, the C-VLAN component may maintain a Remote Replication Registration entry in the corresponding Internal Bridge Port Extender(s).

To maintain a Remote Replication Registration entry, the C-VLAN component shall:

c) Allocate an E-CID for the entry as specified in 8.11 that is unique among all of the other E-CIDs in the table;
d) Set the Port Map control elements that correspond to the Ports from which the frame is to be filtered to *filter*; and
e) Set the remaining Port Map control elements to *forward*.

8.10.2 Support of Remote Replication by a C-VLAN component active topology enforcement

A C-VLAN component that supports Remote Replication shall perform active topology enforcement as specified in 8.6.1 of IEEE Std 802.1Q except that the reception Port shall be identified as a potential transmission port if all of the following are true:

a) The reception Port corresponds to an external Extended Port; and
b) The frame to be forwarded is utilizing the Remote Replication Service.

NOTE—The above identification as a potential transmission Port is in addition to the Ports so identified in 8.6.1 of IEEE Std 802.1Q.

8.10.3 Support of Remote Replication by a C-VLAN component MAC Relay

This subclause specifies the additional requirements related to the MAC Relay of a C-VLAN component for frames that are forwarded using the remote replication capability.

The C-VLAN component shall utilize remote replication if the frame is to be forwarded to two or more ports within a Replication Group. The C-VLAN component may utilize remote replication if the frame is to be forwarded to two or more ports regardless of their membership in a particular Replication Group.

In support of remote replication within the C-VLAN component of a Controlling Bridge, the connection_identifier in the PEM_UNITDATA.request primitive shall carry a port map and an ingress E-CID in addition to the information specified IEEE 802.1Q. This port map shall have one control entry for each external Extended Port reachable through the Internal Bridge Port Extender. The control element shall be set to *filter* if either of the following are true:

a) The port is not a potential transmission Port as specified in 8.6.1 of IEEE Std 802.1Q and is not the reception Port; or
b) The final filter and forwarding decision produced by querying the filtering database (8.8.9 of IEEE Std 802.1Q) indicates filter for the corresponding Port.

Otherwise, the control element shall be set to *forward*.

The ingress_echannel_identifier of the EM_UNITDATA.request shall be set to the echannel_identifier value of the EM_UNITDATA.indication if all of the following conditions are true:

c) The operReflectiveRelayControl parameter of the Port to which the frame is being forwarded is FALSE (8.6.1 of IEEE Std 802.1Q).
d) An echannel_identifier value is present in the connection_identifier of the EM_UNITDATA.indication; and
e) The C-VLAN component Port on which the frame is to be transmitted and the Port on which the frame was received are members of the same Replication Group.

Otherwise, the ingress_echannel_identifier value within the EM_UNITDATA.request is set to zero.

The C-VLAN component shall insert a C-TAG in all frames being forwarded utilizing the remote replication capability.

8.10.4 Bridge Port Extender Remote Replication Configuration

Within each Bridge Port Extender through which an E-channel allocated for remote replication passes, the Controlling Bridge shall:

a) Include the Bridge Port Extender Extended and Cascade Ports through which frames on that E-channel are to pass in the member set of the E-channel; and
b) Exclude all other Ports from the E-channel's member set.

This configuration shall be achieved utilizing the Bridge PE CSP (Clause 9).

8.11 Assignment of E-CIDs

The allocation of E-CIDs by a Controlling Bridge shall be consistent with the restrictions specified in this subclause.

E-CIDs are subdivided into three fields in the E-TAG (7.5.1, 7.5.2) as follows:

a) E-CID_base (12 bits)
b) E-CID_ext (8 bits)
c) GRP (2 bits)

The GRP bits indicate whether the E-CID identifies a point-to-point E-channel (GRP equal to zero) or a point-to-multipoint channel (GRP not equal to zero).

Base Bridge Port Extenders do not utilize the E-CID_ext bits. Therefore, in a cascade of all base Bridge Port Extenders, the E-CIDs assigned by the Controlling bridge have the E-CID_ext bits of E-CIDs set to zero.

Furthermore, a Bridge Port Extender implementation does not necessarily have the filtering database capacity to support the entire range of E-channels that can be represented by the E-CID. The filtering database capacity of a Bridge Port Extender is communicated to the Controlling Bridge in the CSP Open command (9.8.1). For an aggregating Bridge Port Extender, the Controlling Bridge can allocate any E-CID_base and E-CID_ext value to any E-channel so long as the E-channel capacity of the Bridge Port Extenders is not exceeded.

To enable a base Bridge Port Extender to implement a more simplified database model, the allocation of E-CIDs by the Controlling Bridge is more restrictive. In the case of a point-to-point E-channel, the E-CID_base bits must be in the range of 1 up to the number of point-to-point E-channels supported. For point-to-multipoint E-channels, the concatenation of the GRP bits with the E-CID_base bits must be in the range of 0x1000 up to 0x0fff plus the number of point-to-multipoint E-channels supported.

Aggregating Bridge Port Extenders and base Bridge Port Extenders can be cascaded together to provide a greater number of E-channels than that which can be provided by base Bridge Port Extenders alone. Aggregating Bridge Port Extenders provide an implied E-CID_ext field for a lower layer base Bridge Port Extender. This field is inserted into the E-TAG of every frame received by the Aggregating Bridge Port Extender from a base Bridge Port Extender based on the Port on which the frame was received. Thus, the implied E-CID_ext field shall be the same for every E-channel passing through a given Cascade Port at an aggregating/base Bridge Port Extender boundary, and shall be unique among other Cascade Ports at aggregating/base Bridge Port Extender boundaries.

8.12 Support of Congestion Notification

A Controlling Bridge can support congestion notification as specified by Clause 30 through Clause 33 of IEEE Std 802.1Q. Bridge Port Extenders optionally support the functionality of a congestion point (6.16). A Controlling Bridge can incorporate this support within the Extended Bridge through appropriate configuration of the Bridge Port Extender congestion points utilizing the PE CSP.

9. Port Extender Control and Status Protocol

The Port Extender Control and Status Protocol (PE CSP) provides the mechanism by which a Controlling Bridge configures the External Bridge Port Extenders under its control. It is also the mechanism by which the Controlling Bridge dynamically discovers the presence of external Extended Ports and obtains status information from the External Bridge Port Extenders. It is implemented as a simple command/response protocol. Information utilized within the protocol are packaged into Type, Length, Value (TLV) triples. A PE CSP Protocol Data Unit (PDU) consists of a Command TLV and zero or more additional TLVs, as specified by the protocol.

Figure 8-1 illustrates the PE CSP, Edge Control Protocol (ECP), and LLDP locations in the Extended Bridge architecture.

PE CSP executes as an upper layer protocol over the ECP (Clause 43 of IEEE Std 802.1Q). The PE CSP executes exclusively between a Controlling Bridge and Bridge Port Extenders that comprise an Extended Bridge. As each Bridge Port Extender is discovered, a separate E-channel is created between the Controlling Bridge and the Bridge Port Extender to carry frames between the Controlling Bridge and the Control and Status Agent within the Bridge Port Extender. The Control and Status Agent is the entity within a Bridge Port Extender responsible for executing the PE CSP. A separate instance of the ECP and PE CSP is executed over each of these E-channels.

The PE CSP creates PDUs that are passed to the ECP for transmission to the peer. Each PDU contains one or more TLVs specified in this clause. Likewise, ECP passes PE CSP PDUs to the PE CSP that were received from the peer. The ECP provides a basic acknowledgement and retransmit mechanism; therefore, PE CSP assumes that once a PDU is delivered to ECP, the PDU is reliably delivered to the peer PE CSP entity, if it still exists. PE CSP limits the number of outstanding commands to one and therefore the buffer space used to receive commands and responses is never exceeded.

NOTE—This implies that reserving one buffer to receive commands and an additional buffer to receive responses is all that is needed to prevent a buffer overflow between ECP and PE CSP.

The PE CSP PDU consists of one or more data units encoded in TLV triples. All PE CSP PDUs contain the Command TLV. Additional TLVs are included as required by each command.

9.1 Port Selection and Addressing

Within a Bridge Port Extender, PE CSP shall be executed only on the Upstream Port as selected by 6.13. Any PE CSP messages received on other Ports shall be ignored.

Either the Nearest non-TPMR Bridge group address (8.6.3 of IEEE Std 802.1Q) or individual MAC addresses shall be used to address the ECP frames that carry the PE CSP. The individual address is discovered utilizing Port Extension TLV within LLDP (D.2.1.5 of IEEE Std 802.1Q).

9.2 PE CSP State Machines

Four state machines define the transmission, reception, and processing of PE CSP PDUs, as follows:

a) PE CSP Receive PDU state machine (Figure 9-1) controls the reception of PE CSP PDUs;

b) PE CSP Transmit PDU state machine(Figure 9-2) controls the transmission of PE CSP PDUs;

c) PE CSP Local Request state machine (Figure 9-3) controls the transmission of PE CSP request PDUs and the associated response time-out processing;

d) PE CSP Remote Request state machine (Figure 9-4) controls the reception of remote PE CSP request PDUs and the transmission of the associated responses.

Each External Bridge Port Extender has one instance of these four state machines within the Port Extender Control and Status Agent. The C-VLAN component of a Controlling Bridge instantiates one instance of these four state machines for each Port that provides communication to a Port Extender Control and Status Agent.

Each state machine shall implement the functionality defined in their associated figure and attendant definitions in 9.2.1, 9.2.2, and 9.2.3. The notational conventions used in the state machines are as stated in Annex E of IEEE Std 802.1Q.

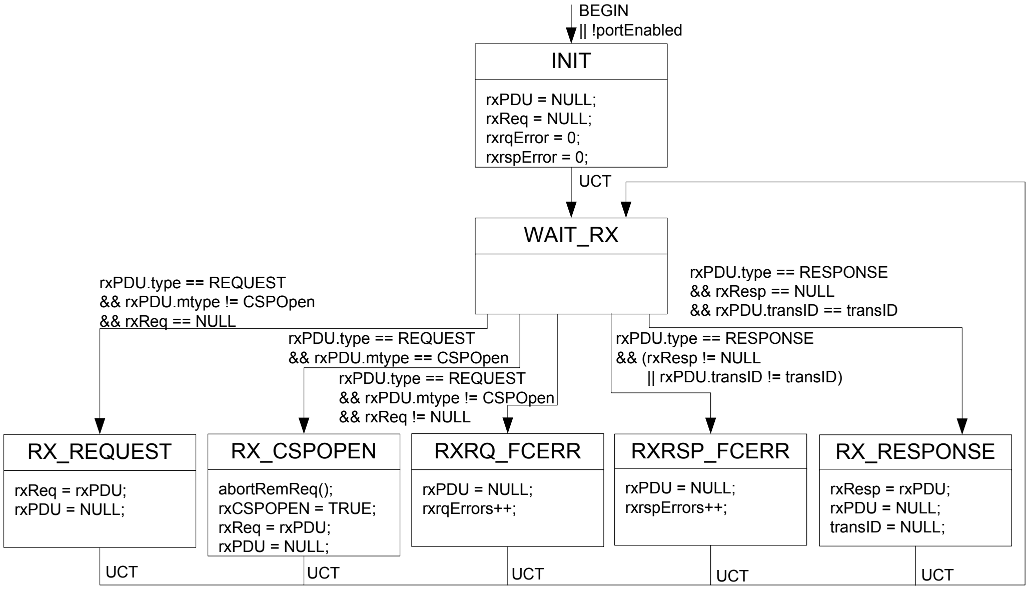


Figure 9-1—PE CSP Receive PDU state machine

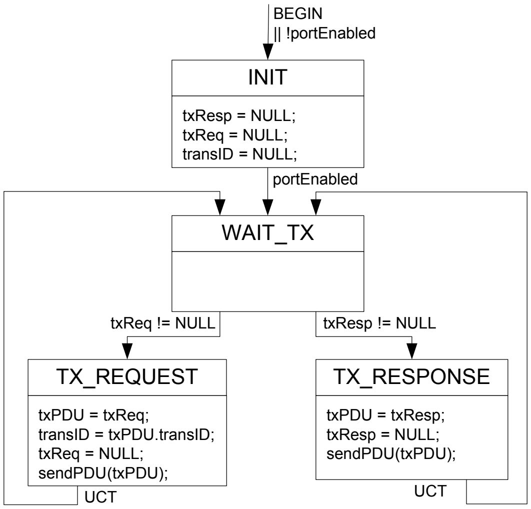


Figure 9-2—PE CSP Transmit PDU state machine

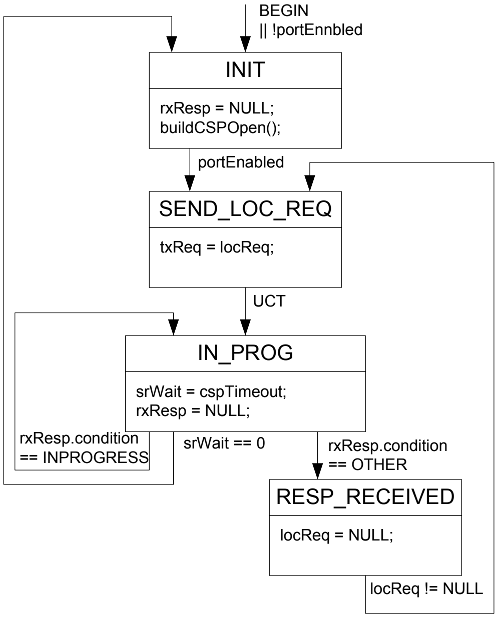


Figure 9-3—PE CSP Local Request state machine

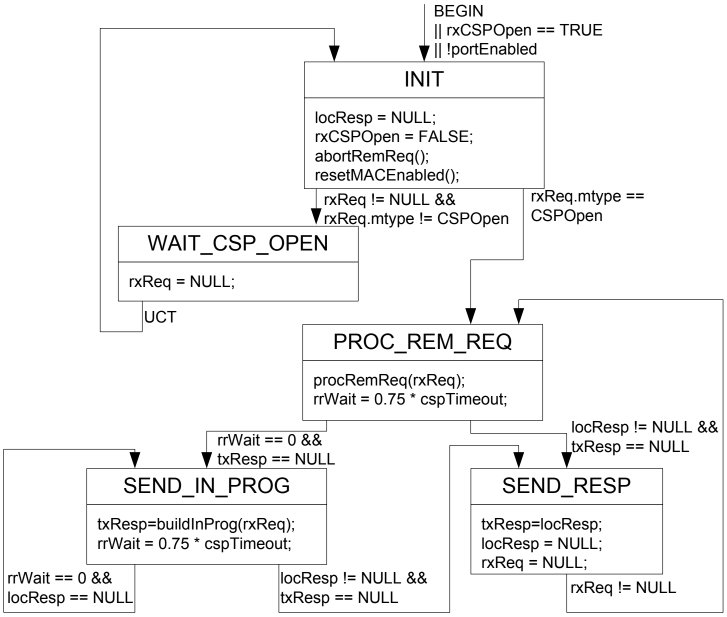


Figure 9-4—PE CSP Remote Request state machine

9.2.1 PE CSP state machine timers

A set of timers is used by the PE CSP state machines. These operate as countdown timers (i.e., they expire when their value reaches zero). These timers:

a) Have a resolution of one second;

b) Are loaded by an initial integer value;

c) Are decremented once per second until reaching zero;

d) Represent the remaining time in the period.

9.2.1.1 srWait

An instance of srWait exists for each instance of the PE CSP Local Request state machine. It is used to detect a time out waiting for a remote response following the transmission of a local request.

9.2.1.2 rrWait

An instance of rrWait exists for each instance of the PE CSP Remote Request state machine. It is used to determine when to send a local response with a completion code of In Progress (9.6.3.2).

9.2.2 PE CSP state machine procedures

9.2.2.1 abortRemReq()

The abortRemReq procedure aborts the processing of the request currently being processed by procRemReq() and sets the req parameter of procRemReq() to NULL. If procRemReq() is not currently processing a request, then this procedure has no effect.

9.2.2.2 buildCSPOpen()

The buildCSPOpen() procedure builds a CSP Open request in locReq.

9.2.2.3 buildInProg(req)

The buildInProg(req) procedure builds and returns an In Progress response for the request PDU passed to it in the req parameter.

9.2.2.4 procRemReq(req)

The procRemReq(req) procedure passes the remote request PDU from the state machine for processing. Once processing is complete, the locResp parameter is set with the response PDU to be sent to the remote device. This procedure is non-blocking, the state machine progresses based on the specified transitions while procRemReq() executes. Completion of execution is indicated by locResp being non-NULL.

9.2.2.5 resetMACEnabled()

The resetMACEnabled() procedure sets to MAC_Enabled parameter to FALSE for at least one second on all ports except the Upstream Port.

9.2.2.6 sendPDU(pdu)

The sendPDU() procedure causes the TLVs that make up the PDU in the pdu parameter to be transmitted.

9.2.3 PE CSP state machines variables and parameters

9.2.3.1 cspTimeout

The message_timeout value from Table 9-1.

9.2.3.2 locReq

A PDU containing a locally generated request. The value is set by the Controlling Bridge or the Port Extender Control and Status Agent outside the state machine. The state machine sets this value to NULL to indicate the PDU has been transmitted and a response received.

9.2.3.3 locResp

A PDU containing the locally generated response. See 9.2.2.3 and 9.2.2.4.

9.2.3.4 NULL

A value assigned to a variable to indicate that the variable does not contain a valid value.

9.2.3.5 portEnabled

This variable is externally controlled. Its value reflects the operational state of the MAC service supporting the port. Its value is TRUE if the MAC service supporting the Port is in an operable condition; otherwise, it is FALSE.

9.2.3.6 rxCSPOpen

A Boolean that is set TRUE in the PE CSP Receive PDU state machine. It causes the PC CSP Remote Request state machine to initialize. rxCSPOpen is set to FALSE as part of the initialization.

9.2.3.7 rxPDU

The last PE CSP PDU received.

9.2.3.8 rxPDU.transID

Contains the value of the Transaction ID field in the Command TLV (9.6.2) of the rxPDU.

9.2.3.9 rxPDU.type

Indicates the type of rxPDU. Valid values are REQUEST and RESPONSE corresponding to the D bit of the Command TLV within the PDU (9.6.4).

9.2.3.10 rxReq

The last request PE CSP PDU received.

9.2.3.11 rxReq.mtype

The message type contained in the last request PE CSP PDU received, as specified in Table 9-4.

9.2.3.12 rxResp

The last response PE CSP PDU received.

9.2.3.13 rxResp.condition

Indicates whether the Completion Code (9.6.3) is In Progress or some other value. Valid values are INPROGRESS and OTHER, respectively. This value is NULL when rxResp is NULL.

9.2.3.14 rxrqErrors

Count of the number of flow control errors detected on received requests.

9.2.3.15 rxrspErrors

Count of the number of flow control errors detected of received responses.

9.2.3.16 transID

Contains the Transaction ID from the Command TLV (9.6.2) from the last request transmitted. Set to NULL upon receipt of the response.

9.2.3.17 txPDU

The next PE CSP PDU to be transmitted.

9.2.3.18 txPDU.transID

Contains the value of the Transaction ID field of the Command TLV (9.6.2) in the txPDU.

9.2.3.19 txReq

The next local request PE CSP PDU to be transmitted.

9.2.3.20 txResp

The next local response PE CSP PDU to be transmitted.

9.2.4 External interaction of the PE State Machines

Within a External Bridge Port Extender, the Control and Status Agent executes the PE CSP. The Controlling Bridge executes one instance of PE CSP for each Bridge Port Extender under its control.

To send a PE CSP PDU, the PDU is placed in the state machine parameter locReq. Once sent, the state machine sets locReq to NULL indicating that it is ready to send another PDU.

A received PE CSP request PDU is processed by the ProcRemReq() procedure.

A received PE CSP response is placed in the rxResp parameter by the state machines. The PE CSP entity that handles the response must set rxResp to NULL to enable reception of the next response.

9.3 Protocol Errors

The PE CSP protocol utilizes the parameters as defined in Table 9-1.

Table 9-1—Port Extender Control and Status Protocol—Time out Values

Parameter	Value (s)
message_timeout	60

A PE CSP implementation waits a minimum message_timeout period without receiving a response to a request. If no response is received, a protocol error is detected.

A PE CSP implementation may send a response with a completion code of In Progress (Table 9-3) to a request that can potentially take a long time to service. Upon receiving such a response, the PE CSP peer waits again for a message_timeout period without receiving a response. Each time it receives a response with a completion code of In Progress, the peer must again wait for a message_timeout period to receive a response. If no response has been received during this period, a protocol error is detected.

If a Controlling Bridge detects a protocol error, recovery is attempted by restarting the PE CSP Local Request state machines and reestablishing communication by sending a CSP Open command. If communication is reestablished, this will result in initialization of the Bridge Port Extender. The Controlling Bridge proceeds as if a new Bridge Port Extender had been attached.

If a Bridge Port Extender detects a protocol error, recovery is attempted by restarting the PE CSP Local Request state machine and sending a CSP Open command

9.4 PE CSP PDUs

A PE CSP PDU is made up of a Command TLV and zero or more additional TLVs. The required additional TLVs are specified in Table 9-4. Any additional TLVs, including unknown TLVs, are ignored.

9.5 Basic TLV format

Figure 9-5 shows the basic TLV format.

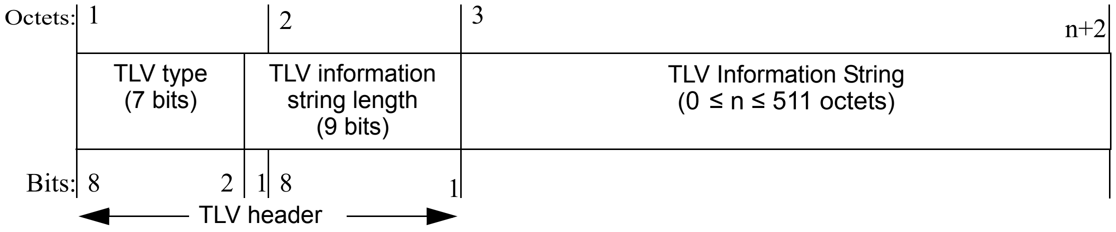

Octets: 1
2
3
n+2
TLV type (7 bits)
TLV information string length (9 bits)
TLV Information String (0 ≤ n ≤ 511 octets)
Bits: 8
2
1
8
1
TLV header

Figure 9-5—Basic TLV format

The TLV type field occupies the seven most significant bits of the first octet of the TLV format. The least significant bit in the first octet of the TLV format is the most significant bit of the TLV information string length field.

9.5.1 Use of reserved fields

Unless specified otherwise, all reserved fields in the PE CSP TLVs shall be set to zero and ignored on receive.

9.5.2 TLV Type

The TLV Type field shall be set to a valid value from Table 9-2.

Table 9-2—TLV type values

TLV type	TLV name	TLV reference
0	Reserved for future standardization	—
1	Command	9.6
2	Resource Limit Capability	9.9.1
3	Port Parameters	9.9.2
4	Port Array	9.9.3
5	VID Array	9.9.4
6	Port Status	9.9.5
7	Statistics	9.9.6
8	Object Name	9.9.7
9	Object Value	9.9.8
10	CN Parameters	9.9.9
11–126	Reserved for future standardization	—
127	Organizationally Specific TLVs	9.9.10

9.5.3 TLV information string length

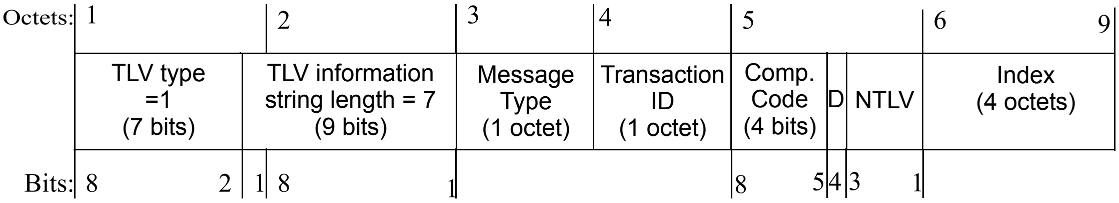
The TLV information string length field shall contain the length of the information string, in octets. If a TLV is received that is longer than expected, the excess content at the end of the TLV is ignored.

9.5.4 TLV information string

The TLV information string may be fixed or variable length and contains the information specified for each TLV.

9.6 Command TLV

The Command TLV shall be the first TLV in all PE CSP PDUs and shall be constructed and processed as specified in this subclause. Figure 9-6 illustrates the format of the Command TLV.



Octets:	1	2	3	4	5			6 9
	TLV type =1 (7 bits)	TLV information string length = 7 (9 bits)	Message Type (1 octet)	Transaction ID (1 octet)	Comp. Code (4 bits)	D	NTLV	Index (4 octets)
Bits:	8 2	1 8 1			8 5	4	3 1	

Figure 9-6—Command TLV

9.6.1 Message Type

The Message Type field indicates the type of message contained in this PE CSP PDU as indicated in Table 9-4. The Required TLVs corresponding to the Message Type listed in this table shall be provided in the PE CSP PDU. The use of each message type is described in 9.8.

9.6.2 Transaction ID

The Transaction ID field is used to ensure that responses to requests are properly matched as specified in the PE CSP State Machines (9.2). In a CSP Open request, Transaction ID shall be set to zero. Transaction ID shall be incremented by 1, and reset to zero upon reaching 256, for each successive request. The Transaction ID in a response shall be set to that of the corresponding request.

9.6.3 Completion Code

The Completion Code field is reserved in request messages. In response messages, it is set to one of the values in Table 9-3.

Table 9-3—Completion Codes

Completion Code	Condition
0	Success
1	In Progress
2	Failure—lack of resources
3	Failure—unknown message type
4	Other Failure
All others	Reserved for future standardization

9.6.3.1 Success

The Success completion code is returned to indicate the successful completion of a request. This completion code is also returned if no action was required to complete the request, e.g., deleting a nonexistent E-channel.

Table 9-4—Message Types

Message	Message Type	May be initiated by		Request TLVs	Response TLVs
		Controlling Bridge	Bridge Port Extender		
Reserved for future standardization	0	—	—	—	—
CSP Open	1	X	X	Resource Limit Capability	None
Extended Port Create	2	—	X	None	Port Parameters VID Array (optional)
Extended Port Delete	3	X	X	None	None
Port Parameters Set	4	X	—	Port Parameters (optional) VID Array (optional) (at least one TLV must be present)	None
Port Parameters Get	5	X	—	None	Port Parameters VID Array
Status Parameter Set	6		X	Port Status	None
E-channel Register	7	X	—	Port Array	None
E-channel Registration Get	8	X	—	None	Port Array
Statistics Get	9	X	—	None	Statistics
Transit Delay Set	10	X	—	None	None
Object Get	11	X	—	Object Name	Object Name Object Value
Object Set	12	X	—	Object Name Object Value	Object Name Object Value
CN Parameters Set	13	X	—	CN Parameters	None
CN Parameters Get	14	X	—	None	CN Parameters
Reserved for future standardization	13–254	—	—	—	—
Organizationally Specific	255	X	X	(See NOTE)	(See NOTE)

NOTE—At least one organizationally specific TLV is required to identify the organizationally specific command. This TLV shall be the first TLV following the Command TLV.

9.6.3.2 In Progress

The In Progress completion code is returned to indicate additional time is needed to process the request. See 9.3.

9.6.3.3 Failure—lack of resources

The Failure—lack of resources completion code is returned to indicate that a Command TLV was received that would have otherwise been successful except that the sufficient resources were not available to complete the command (e.g., exceeding the E-channel capacity).

9.6.3.4 Failure—unknown message type

The Failure—unknown message type completion code is returned to indicate that the Command TLV contained an unknown message type.

9.6.3.5 Other failure

The Other failure completion code is returned to indicate that the Command TLV was not processed for a reason other than lack of resources (e.g., malformed TLV) or unknown message type.

9.6.4 D

D: Set to 1 if this is a response message, 0 if this is a request message.

9.6.5 NTLV

This field contains the number of TLVs following the command TLV that form this PE CSP PDU.

9.6.6 Index

The Index field contains a command specific value. The value to be placed in the Index field is specified for each individual command in 9.8. If not specified, the value is reserved.

9.7 Flow Control

After the transmission of the first request PDU, a PE CSP entity does not transmit another request PDU until it has received the response from the previous request or a protocol error is detected.

9.8 Messages

The following subclauses describe each of the messages supported in the PE CSP.

9.8.1 CSP Open

The CSP Open message shall be sent by each Bridge Port Extender and Controlling Bridge to initialize PE CSP communication. The parameters in the associated TLVs are exchanged. The operational parameters are established based on the capabilities for each peer. In addition, a Bridge Port Extender shall initialize its parameters as specified in 6.15.

Upon completion of processing the request message, each peer shall send a CSP Open response message to the other peer.

Receipt of a CSP Open Message at any time other than the first message received indicates that the peer has reset. Therefore, to reestablish communication, a new CSP Open Message is sent.

This is the first message sent upon PE CSP initialization, and no other messages shall be sent until a successful response is received.

The Index field in the Command TLV of the PE CSP Open Message shall contain the value one, indicating the version of the PE CSP being executed. The value of this field shall be ignored in received CSP Open Messages.

NOTE—It is assumed that future versions of the protocol will remain backwards compatible. Therefore, it is not necessary for this, the first version of the protocol, to do anything other than set the value of this field. Future versions could need to check it to ensure that they emit PDUs that are compatible.

9.8.2 Extended Port Create

The Bridge Port Extender shall send an Extended Port Create request message to the Controlling Bridge to request the creation of a new E-channel for binding with an Extended Bridge Port with the Index field in the Command TLV set to a value that identifies the individual Port. The value zero is reserved to indicate the Upstream Port and is not used in the Extended Port Create command.

Upon receipt of the request, the Controlling Bridge shall send an Extended Port Create response message with the Index Field of the Command TLV set to the E-CID that identifies the newly created E-channel if the command is successful, otherwise the content of the Index field is reserved.

Upon receipt of the response message with a Completion Code (9.6.3) of Success, the Bridge Port Extender shall:

a) Enter the Extended Port in the member set of the E-channel identified by the E-CID;
b) Remove other Ports, if any, from the member set;
c) Set the PCID value of the Extended Port to the E-CID;
d) Set the MAC_Enabled parameter to TRUE; and
e) Configure the Extended Port parameters as specified in the Port Parameter and VID array TLVs.

NOTE—It is not an error for an Extended Port Create request to request the creation of an already existing Extended Port. If this occurs, the request is processed as specified above and a successful response is returned.

9.8.3 Extended Port Delete

The Extended Port Delete request shall be sent by the Bridge Port Extender or the Controlling Bridge to remove an Extended Port previously created via the Extended Port Create request from all E-channel member sets. The Index field of the Command TLV shall contain the E-CID identifying the E-channel associated with the Port to be deleted.

When a Bridge Port Extender receives the Extended Port Delete request, it shall:

a) Remove all Ports from the E-channel member set;
b) Remove the Extended Port associated with the E-channel from all other E-channel member sets;
c) Set MAC_Enabled in the PEISS to FALSE (6.10.2);
d) Upon completion of these operations, send the Extended Port Delete response message to the Controlling Bridge.

When a Controlling Bridge receives the Extended Port Delete request, it shall:

e) Remove the corresponding E-channel in any intervening Bridge Port Extenders using the E-channel Register message (9.8.7);
f) Upon completion of these operations, send the Extended Port Delete response message to the Bridge Port Extender.

When a Controlling Bridge receives an Extended Port Delete response, it shall:

g) Remove the corresponding E-channel in any intervening Bridge Port Extenders using the E-channel Register message (9.8.7);

When a Bridge Port Extender receives the Extended Port Delete response message, it shall:

h) Remove all Ports from the corresponding E-channel member set;
i) Remove the Extended Port associated with the E-channel from all member sets.

NOTE—It is not an error for an Extended Port Delete request to be issued for a nonexistent Extended Port. If this occurs, a successful response is returned.

9.8.4 Port Parameters Set

The Controlling Bridge shall send a Port Parameters Set message to a Bridge Port Extender to configure the parameters specified in the Port Parameters TLV and/or the VID Array TLV for an Extended Port or for the Upstream Port. The Index field in the Command TLV shall be set to the E-CID identifying the Extended or Cascade Port, or to zero to indicate the Upstream Port.

Upon completion, the Bridge Port Extender shall send a Port Parameters Set response message with Index field set to the E-channel identifying the Extended or Cascade Port, or to zero to indicate the Upstream Port.

9.8.5 Port Parameters Get

A Controlling Bridge shall send a Port Parameters Get request message to query the currently configured state for a Port on a Bridge Port Extender. Upon receiving this message, the Bridge Port Extender shall send the Port Parameters Get response message to the peer. The Index field in the Command TLV of both the request and the response shall be set to the E-CID identifying the Extended or Cascade Port, or to zero to indicate the Upstream Port. The Port Parameters and VID Array TLVs shall be populated with the parameters applicable to the Port if the E-CID is valid. The contents of the Port Parameters and VID Array TLVs is unspecified if the E-CID is invalid.

9.8.6 Status Parameter Set

A Bridge Port Extender shall send a Status Parameter Set request each time the value of MAC_Operational (6.6 of IEEE Std 802.1Q) changes on one of its Cascade or Extended Ports. The Index field in the Command TLV shall be set to the PCID of the Extended or Cascade Port.

The Controlling Bridge, upon reception of a Status Parameter Set, shall set the MAC_Enabled parameter (6.6 of IEEE Std 802.1Q) of the corresponding Extended Port within the Internal Bridge Port Extender to match the indication (TRUE or FALSE) received in the Port Status TLV.

NOTE—Setting the MAC_Enabled parameter on the Extended Port of the Internal Bridge Port Extender is reflected across the internal LAN to the MAC_Operational parameter of the Port in the C-VLAN component. This provides MAC_Operational propagation from the external Extended Port to the C-VLAN component.

Upon completion, the Controlling Bridge shall send a Status Parameter Set response message to the Bridge Port Extender with Index field set to the E-channel identifying the Extended or Cascade Port.

9.8.7 E-channel Register

The E-channel Register request message shall be sent by a Controlling Bridge to a Bridge Port Extender to configure a set of Ports within, or to remove them from, the member set of an E-channel.

The message shall be constructed as follows:

a) The Index field in the Command TLV is set to the E-CID identifying the E-channel;

b) The Port Array TLV is populated with a list of Port_Index elements, along with an indication of which sets the Port is to be added to or removed from.

Upon receipt of the message, the Bridge Port Extender shall:

c) Perform the specified action on all Ports in the Port Array TLV;

d) Upon completion, the Bridge Port Extender sends an E-channel Register response message.

If the E-CID is in the range of point-to-point E-CIDs (8.11), then only the first element of the Port Array TLV is processed and any other elements are ignored.

9.8.8 E-channel Registration Get

The E-channel Registration Get request message shall be sent by the Controlling Bridge to query E-channel member set population.

The Index field of the Command TLV for both the request and the response shall be set to the E-CID identifying the E-channel being queried.

Upon receipt of an E-channel Registration Get request, an E-channel Registration Get response shall be generated containing a Port Array TLV enumerating the Ports that are members of the member set of the E-channel.

9.8.9 Statistics Get

The Statistics Get request shall be sent by a Controlling Bridge to a Bridge Port Extender to retrieve the values of the statistics counters. The Index field of the Command TLV is reserved.

Upon receipt of a Statistics Get request message, the Bridge Port Extender shall send a Statistics Get response message with the Statistics TLV populated with the values from the Port’s statistics counters. The Index field of the Command TLV is reserved.

9.8.10 Transit Delay Set

The Transit Delay Set request shall be sent by the Controlling Bridge to set the Bridge Port Extender transit delay parameter (6.11.5), with the value, in seconds, included in the Index field of the Command TLV.

Upon receipt of a Transit Delay Set request, the Bridge Port Extender shall set the value of the Bridge Port Extender transit delay parameter to that in the Index field of the request. The Bridge Port Extender shall then send a Transit Delay Set response to the Controlling Bridge with the Index field of the Command TLV set to that of the request.

9.8.11 Object Get

A Bridge Port Extender may provide a variety of MAC layer interfaces and associated control facilities. The objects associated with the control and status of these interface are defined in IEEE Std 802.3.1. An Object Get request shall be sent by the Controlling Bridge to obtain the value of an object defined by IEEE Std 802.3.1 within a Bridge Port Extender. The Index field of the Command TLV is reserved.

Upon receipt of a Get Object command, the Bridge Port Extender shall respond with an Get Object response. The Index field of the Command TLV is reserved.

NOTE—The Controlling Bridge is responsible for the indexing of its Ports, including Extended Ports, for presentation to a management system. This indexing is independent of the indexing utilized by a particular Bridge Port Extender. The Controlling Bridge is responsible for the translation between the indexing schemes.

9.8.12 Object Set

A Bridge Port Extender may provide a variety of MAC layer interfaces and associated control facilities. The objects associated with the control and status of these interface are defined in IEEE Std 802.3.1. An Object Set request shall be sent by the Controlling Bridge to set the value of an object defined by IEEE Std 802.3.1 within a Bridge Port Extender. The Index field of the Command TLV is reserved.

Upon receipt of a Set Object command, the Bridge Port Extender shall respond with an Get Object response. The Index field of the Command TLV is reserved.

NOTE—The Controlling Bridge is responsible for the indexing of its Ports, including Extended Ports, for presentation to a management system. This indexing is independent of the indexing utilized by a particular Bridge Port Extender. The Controlling Bridge is responsible for the translation between the indexing schemes.

9.8.13 CN Parameters Set

This message shall be supported in a Controlling Bridge only if the implementation supports congestion notification in Bridge Port Extenders. This message shall be supported in a Bridge Port Extender only if the implementation support congestion notification (5.3).

The Controlling Bridge shall send a CN Parameters Set message to a Bridge Port Extender to configure the parameters specified in the CN Parameters TLV for a Port. The Index field in the Command TLV shall be set to the E-CID identifying the Extended or Cascade Port, or to zero to indicate the Upstream Port.

Upon reception, the Bridge Port Extender shall send a CN Parameters Set response message with Index field set to the E-channel identifying the Extended or Cascade Port, or to zero to indicate the Upstream Port.

9.8.14 CN Parameters Get

This message shall be supported only if the implementation supports congestion notification.

A Controlling Bridge shall send a CN Parameters Get request message to query the currently configured congestion notification state for a Port on a Bridge Port Extender.

Upon receiving this message, the Bridge Port Extender shall send the CN Parameters Get response message to the Controlling Bridge. The Index field in the Command TLV of both the request and the response shall be set to the E-CID identifying the Extended or Cascade Port, or to zero to indicate the Upstream Port. The CN Parameters TLVs shall be populated with the parameters applicable to the Port if the E-CID is valid. The contents of the CN Parameters TLV is unspecified if the E-CID is invalid.

9.8.15 Organizationally Specific

A Controlling Bridge or Bridge Port Extender may send an Organizationally Specific message. This message must contain at least one Organizationally Specific TLV. Received Organizationally Specific messages that are unknown shall be ignored.

9.9 Additional TLVs

This subclause describes the TLVs that are used in addition to the Command TLV to form complete messages as specified in Table 9-4. The TLVs shall be constructed as specified in the following subclauses.

9.9.1 Resource Limit Capability TLV

Figure 9-7 illustrates the format of the Resource Limit Capability TLV.

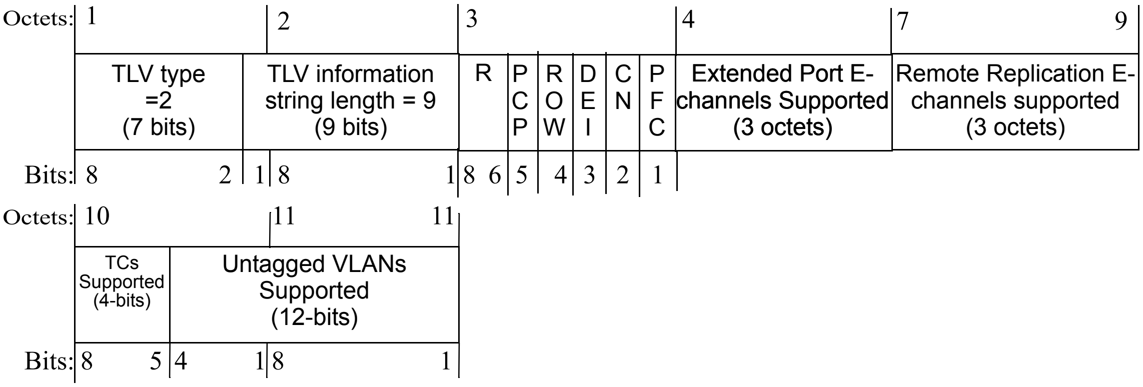

Octets: 1 | 2 | 3 | 4 | 7 9
TLV type =2 (7 bits)
TLV information string length = 9 (9 bits)
R
PCP
ROW
DEI
CN
PFC
Extended Port E-channels Supported (3 octets)
Remote Replication E-channels supported (3 octets)
Bits: 8 2 | 1 | 8 1 | 8 6 | 5 | 4 | 3 | 2 | 1
Octets: 10 | 11 11
TCs Supported (4-bits)
Untagged VLANs Supported (12-bits)
Bits: 8 5 | 4 1 | 8 1

Figure 9-7—Resource Limit Capability TLV

The fields have the following meanings:

R: Reserved.

PCP: Set to one if the implementation supports modification of the values in the Priority Code Point decoding table (6.9 of IEEE Std 802.1Q), zero otherwise.

ROW: Set to one if the implementation supports rows in the Priority Code Point Decoding table in addition to the 8P0D row (6.9 of IEEE Std 802.1Q), zero otherwise.

DEI: Set to one if the implementation supports the decoding of the Drop Eligible Indicator (6.9 of IEEE Std 802.1Q), zero otherwise.

CN: Set to one if the implementation supports congestion notification, zero otherwise.

PFC: Set to one if the implementation support Priority-based Flow Control, zero otherwise.

Extended Port E-channels Supported: When sourced by a Controlling Bridge, reserved. When sourced by a Bridge Port Extender, the number of point-to-point E-channels that can be allocated. These are E-CIDs that are assigned to E-channels associated with an Extended Port.

Remote Replication E-channels Supported: When source by a Controlling Bridge, reserved. When source by a Bridge Port Extender, set to the number of point-to-multipoint E-channels that can be allocated. These E-channels may have more than two Ports in their member set.

If Extended Port E-channels Supported is greater than 4095 or Remote Replication E-channels supported is greater than 12 288, then the Bridge Port Extender is an aggregating Bridge Port Extender. Otherwise, the Bridge Port Extender is a base Bridge Port Extender.

TCs Supported: The number of Traffic Classes supported, valid values are one through eight.

Untagged VLANs Supported: The number of untagged VLANs supported, valid values are one through 4094.

9.9.2 Port Parameters TLV

Figure 9-1 illustrates the format of the Port Parameters TLV.

This TLV provides parameters for use by the Bridge Port Extender Ports.

UD: The useDefault parameter (6.10.4). A value of 1 indicates TRUE and 0 indicates FALSE. This value applies only to Extended and Cascade Ports of aggregating Bridge Port Extenders. In other cases, it is set to zero and ignored on receive.

USE: The use_dei parameter (6.9).

PCS: Priority Code Point Selection (6.9.3 of IEEE Std 802.1Q) encoded as specified in Table 9-5.

PRG7 – PRG0: Contains the Priority Regeneration table for the Port (6.9.4 of IEEE Std 802.1Q). PRG7 contains the priority to which receive priority 7 maps continuing to PRG0 corresponding to receive priority 0.

PTC7 – PTC0: Contains the Priority to Traffic Class mapping for the Port (8.6.6 of IEEE Std 802.1Q). PTC7 contains the Traffic Class to which priority 7 maps continuing to PTC0 corresponding to priority 0.

PFC Enable: Contains one bit per priority (bit 8 corresponding to priority 7 through bit 1 corresponding to priority 0). A one indicates that Priority-based Flow Control (Clause 36 of IEEE Std 802.1Q) is enabled for the corresponding priority. A zero indicates that PFC is disabled for the corresponding priority.

TSA Table: Contains an eight entry table with one octet per entry. Each entry identifies a transmission selection algorithm for the corresponding traffic class. The code points for the Transmission Selection Algorithms are listed in Table 8-5 of IEEE Std 802.1Q. The first entry corresponds to traffic class 7 proceeding down to traffic class 0.

ETS Bandwidth Table: Contains an eight entry table with one octet per entry. Each entry contains a bandwidth allocated to the corresponding traffic class to be used by the Enhanced Transmission Selection (Clause 37 of IEEE Std 802.1Q) algorithm if enabled for the corresponding traffic class. Valid values for each entry are 0 through 100. The valid total of all values in the table is 100.

PD07 – PD00: The entries for the 8P0D row of the Priority Code Point decoding table (6.9.3 of IEEE Std 802.1Q) corresponding to priorities 7 through 0 respectively.

PD17 – PD10: The entries for the 7P1D row of the Priority Code Point decoding table (6.9.3 of IEEE Std 802.1Q) corresponding to priorities 7 through 0 respectively.

PD27 – PD20: The entries for the 6P2D row of the Priority Code Point decoding table (6.9.3 of IEEE Std 802.1Q) corresponding to priorities 7 through 0 respectively.

PD37 – PD30: The entries for the 5P3D row of the Priority Code Point decoding table (6.9.3 of IEEE Std 802.1Q) corresponding to priorities 7 through 0 respectively.

Reserved, R: Reserved

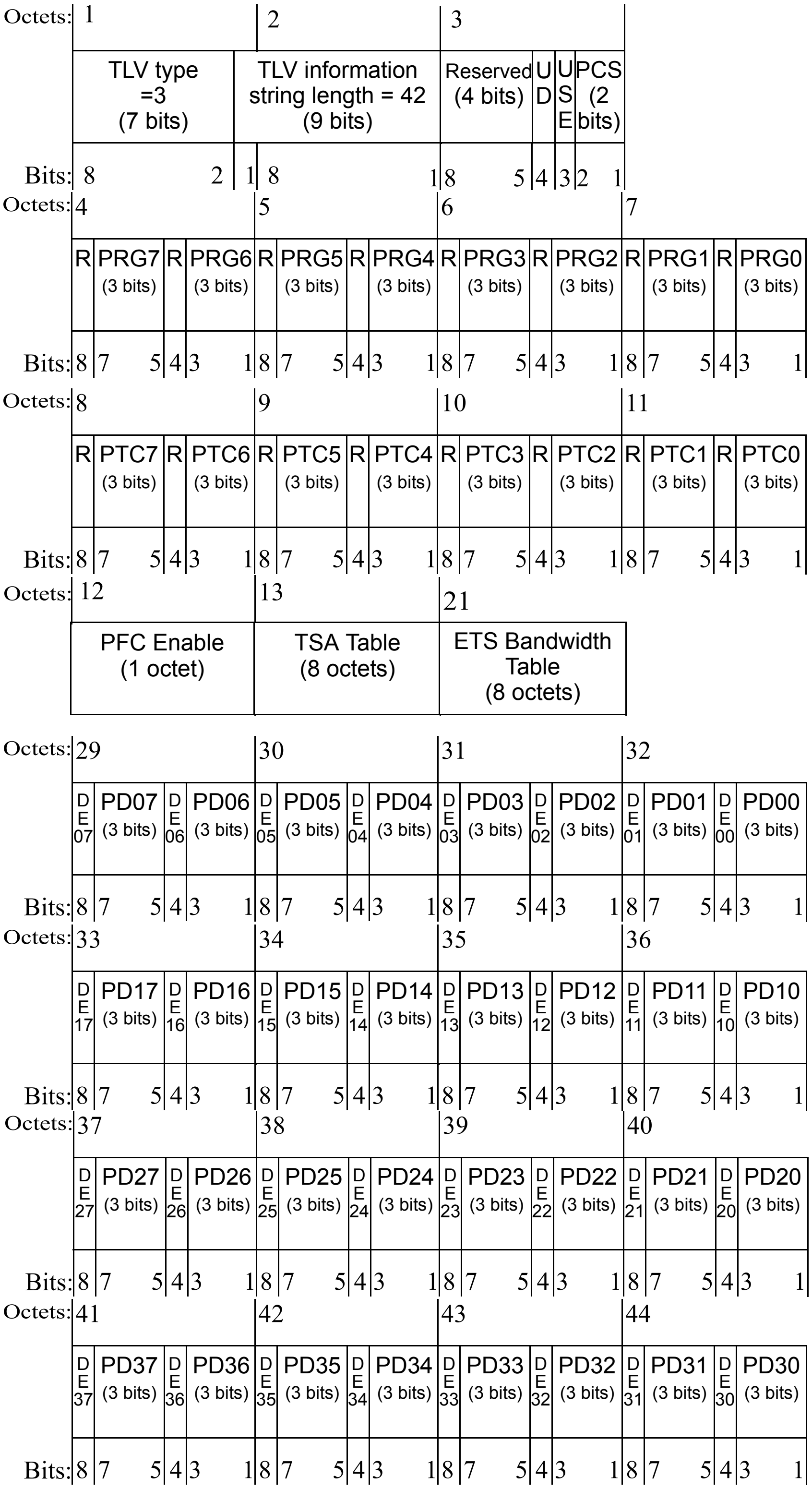

Octets: 1 2 3
TLV type =3 (7 bits)
TLV information string length = 42 (9 bits)
Reserved (4 bits)
UD
USE
PCS (2 bits)
Bits: 8 2 1 8 1 8 5 4 3 2 1
Octets: 4 5 6 7
R PRG7 (3 bits) R PRG6 (3 bits) R PRG5 (3 bits) R PRG4 (3 bits) R PRG3 (3 bits) R PRG2 (3 bits) R PRG1 (3 bits) R PRG0 (3 bits)
Bits: 8 7 5 4 3 1 8 7 5 4 3 1 8 7 5 4 3 1 8 7 5 4 3 1
Octets: 8 9 10 11
R PTC7 (3 bits) R PTC6 (3 bits) R PTC5 (3 bits) R PTC4 (3 bits) R PTC3 (3 bits) R PTC2 (3 bits) R PTC1 (3 bits) R PTC0 (3 bits)
Bits: 8 7 5 4 3 1 8 7 5 4 3 1 8 7 5 4 3 1 8 7 5 4 3 1
Octets: 12 13 21
PFC Enable (1 octet)
TSA Table (8 octets)
ETS Bandwidth Table (8 octets)
Octets: 29 30 31 32
DE07 PD07 (3 bits) DE06 PD06 (3 bits) DE05 PD05 (3 bits) DE04 PD04 (3 bits) DE03 PD03 (3 bits) DE02 PD02 (3 bits) DE01 PD01 (3 bits) DE00 PD00 (3 bits)
Bits: 8 7 5 4 3 1 8 7 5 4 3 1 8 7 5 4 3 1 8 7 5 4 3 1
Octets: 33 34 35 36
DE17 PD17 (3 bits) DE16 PD16 (3 bits) DE15 PD15 (3 bits) DE14 PD14 (3 bits) DE13 PD13 (3 bits) DE12 PD12 (3 bits) DE11 PD11 (3 bits) DE10 PD10 (3 bits)
Bits: 8 7 5 4 3 1 8 7 5 4 3 1 8 7 5 4 3 1 8 7 5 4 3 1
Octets: 37 38 39 40
DE27 PD27 (3 bits) DE26 PD26 (3 bits) DE25 PD25 (3 bits) DE24 PD24 (3 bits) DE23 PD23 (3 bits) DE22 PD22 (3 bits) DE21 PD21 (3 bits) DE20 PD20 (3 bits)
Bits: 8 7 5 4 3 1 8 7 5 4 3 1 8 7 5 4 3 1 8 7 5 4 3 1
Octets: 41 42 43 44
DE37 PD37 (3 bits) DE36 PD36 (3 bits) DE35 PD35 (3 bits) DE34 PD34 (3 bits) DE33 PD33 (3 bits) DE32 PD32 (3 bits) DE31 PD31 (3 bits) DE30 PD30 (3 bits)
Bits: 8 7 5 4 3 1 8 7 5 4 3 1 8 7 5 4 3 1 8 7 5 4 3 1

Figure 9-1—Port Parameters TLV

Table 9-5—Priority Code Point Selection Encoding

PCS value	Meaning
0	8P0D
1	7P1D
2	6P2D
3	5P3D

9.9.3 Port Array TLV

Figure 9-8 illustrates the format of the Port Array TLV.

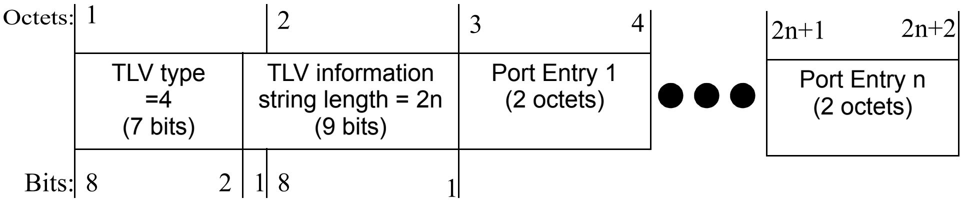


Figure 9-8—Port Array TLV

The Port Array TLV carries one or more Port Entries. Figure 9-9 illustrates the format of a Port Entry.

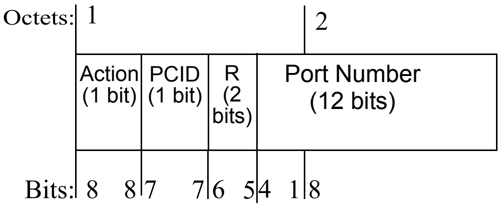


Figure 9-9—Port Entry

The fields of the Port Entry have the following meanings:

Action: The coding of this field is specified in Table 9-6.

PCID: If set to one, the PCID of the Port indicated by the Port Number shall be set to the E-channel specified in the Index field of the Command TLV. If set to zero, the PCID remains unchanged.

R: Reserved.

Port Number: Port to which this entry applies (this is the Port Number that is provided in the Extended Port Create request).

Table 9-6—Action Values

Action value	Action to be performed (request message):	Meaning (response message)
0	Add the Port indicated by the Port Number to the member set of the E-channel specified in the Index field of the Command TLV.	Set to zero, ignore on receive.
1	Delete the Port indicated by the Port Number from the member set of the E-channel specified in the Index field of the Command TLV.	

9.9.4 VID Array TLV

Figure 9-10 illustrates the format of the VID Array TLV.

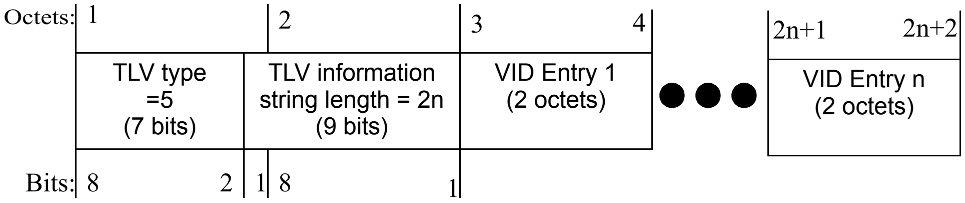


Figure 9-10—VID Array TLV

The VID Array TLV carries one or more VID Entries. Figure 9-11 illustrates the format of a VID Entry.

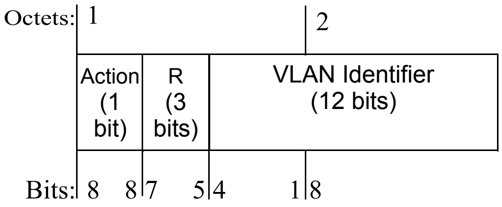


Figure 9-11—VID Entry

This TLV is used to set or retrieve the C-VLAN member sets and untagged sets used in the Tag Handler (6.9). The fields have the following meanings:

Action: The coding of this field is specified in Table 9-7.

R: Reserved.

VLAN Identifier: Identifier of the C-VLAN to which this entry applies.

Table 9-7—Action Values

Action Value	Action to be performed (request message):	Meaning (response message)
0	Add the Extended Port that is in the member set of the E-channel indicated in the Index field of the Command TLV to the untagged set of the C-VLAN indicated in this VID Entry (see 6.9).	The Extended Port that is in the member set of the E-channel indicated in the Index field of the Command TLV is in the untagged set of the C-VLAN indicated in this VID entry (6.9).
1	Delete the Extended Port that is in the member set of the E-channel indicated in the Index field of the Command TLV from the untagged set of the C-VLAN indicated in this VID Entry (see 6.9).	Reserved for future standardization.

9.9.5 Port Status TLV

Figure 9-12 illustrates the format of the Port Status TLV.

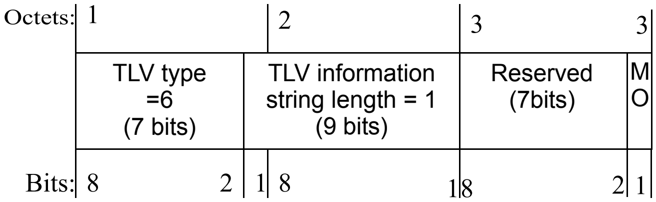


Figure 9-12—Port Status TLV

MO: Set to one if the value of MAC_Operational is TRUE and set to zero if the value of MAC_Operational is FLASE.

9.9.6 Statistics TLV

Figure 9-13 illustrates the format of the Statistics TLV.

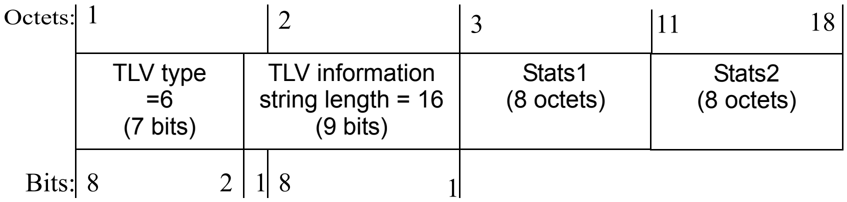


Figure 9-13—Statistics TLV

Table 9-8 specifies the content of the fields within the Statistics TLV.

Table 9-8—Statistics TLV contents

Field	Contents
Stats1	The value of rxrqErrors (9.2.3.14)
Stats2	The value of rxrspErrors (9.2.3.15)

9.9.7 Object Name TLV

Figure 9-14 illustrates the format of the Object Name TLV.

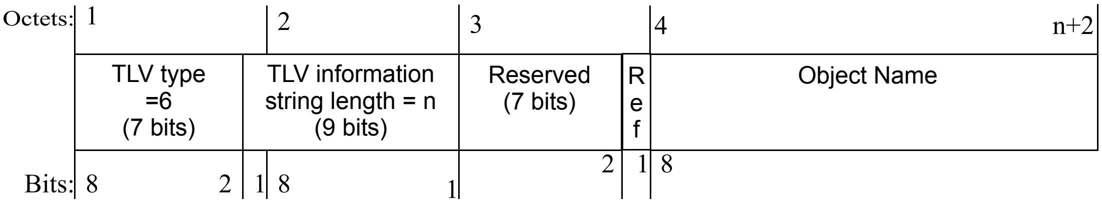


Figure 9-14—Object Name TLV

Ref: Specifies the object to which the Object Name refers in accordance with Table 9-9.

Table 9-9—Reference

Value	Meaning
0	The object referenced is that specified by the Object Name field.
1	The object referenced is that next in the lexical order following that specified in the Object Name field. Valid only in a Get Object Request message.

Object Name: Specifies the object name encoded using the Basic Encoding Rules specified in ASN.1.

9.9.8 Object Value TLV

Figure 9-15 illustrates the format of the Object Value TLV.

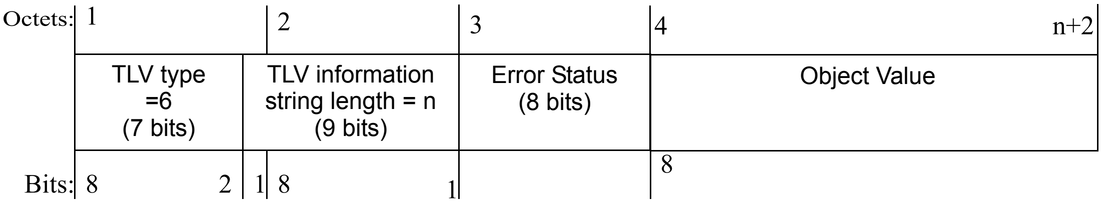


Figure 9-15—Object Value TLV

Error Status: If this TLV is part of a request message, Error Status is reserved. If this TLV is part of a Get Object Response Message, then Error Status is set one of the error codes as specified in Table 9-10. If this TLV is part of a Set Object Response Message, then Error Status is set to one of the error codes as specified in Table 9-11. The error code to be used is specified in 9.9.8.1.

Table 9-10—Get Objects Value Error Codes

Value	Enumeration
0	noError
1	noSuchObject
2	noSuchInstance or endOfMibView
3—255	Reserved for future standardization

Object Value: This value is encoded using the Basic Encoding Rules specified in ASN.1. It is set as follows:

a) In a Object Set request, set to the value that object referenced by Ref and Object Name in the Object Name TLV is to be set;

b) In an Object Set response, set to the current value of the object referenced by Ref and Object Name in the Object Name TLV. If the operation results in a noAccess or inconsistentName error, then Object Value is set to NULL;

c) In an Object Get response, set to the current value of the object referenced by Ref and Object Name in the Object Name TLV. If the operation results in a noSuchObject, noSuchInstance or endOfMibView error, then Object Value is set to NULL.

Table 9-11—Set Object Value Error Codes

Value	Enumeration
0	noError
1—5	Reserved for future standardization
6	noAccess
7	wrongType
8	wrongLength
9	wrongEncoding
10	wrongValue
11	noCreation
12	inconsistentValue
13	resourceUnavailable
14—16	Reserved for future standardization
17	notWritable
18	inconsistentName
19—255	Reserved for future standardization

9.9.8.1 Determining error values

The generation of error status specified here is consistent with IETF RFC 3416 [B4].

In the case of a Get Object response with Ref set to zero, the error status is set as follows:

a) If the Object Name exactly matches the name of a variable accessible by this request, then set to noError; otherwise,

b) If the Object Name does not have an OBJECT IDENTIFIER prefix which exactly matches the OBJECT IDENTIFIER prefix of any (potential) variable accessible by this request, then set to noSuchObject; otherwise,

c) Set to noSuchInstance.

In the case of a Get Object response with Ref set to one, the error status is set as follows:

d) If a variable is located that is in the lexicographically ordered list of the names of all variables that are accessible by this request and whose name is the first lexicographic successor of the Object Name, set to noError; otherwise,

e) Set to endOfMibView.

In the case of a Set Object response, the error status is set as follows (Object Name refers to the Object Name field of the Object Name TLV):

f) If the Object Name specifies an existing or nonexistent variable to which this request is/would be denied access because it is/would not be in the appropriate MIB view, then set to noAccess; otherwise,

g) If there are no variables that share the same OBJECT IDENTIFIER prefix as the Object Name, and that are able to be created or modified no matter what new value is specified, then set to notWritable; otherwise,

h) If the Object Value value field specifies, according to the ASN.1 language, a type that is inconsistent with that required for all variables that share the same OBJECT IDENTIFIER prefix as the variable binding's name, then set to wrongType; otherwise,

i) If the Object Value value field specifies, according to the ASN.1 language, a length that is inconsistent with that required for all variables that share the same OBJECT IDENTIFIER prefix as the variable binding's name, then set to wrongLength; otherwise,

j) If the Object Value value field contains an ASN.1 encoding that is inconsistent with that field's ASN.1 tag, then set to wrongEncoding; otherwise,

k) If the Object Value value field specifies a value that could under no circumstances be assigned to the variable, then set to wrongValue; otherwise,

l) If the Object Name specifies a variable that does not exist and could not ever be created (even though some variables sharing the same OBJECT IDENTIFIER prefix might under some circumstances be able to be created), then set to noCreation; otherwise,

m) If the Object Name specifies a variable that does not exist but can not be created under the present circumstances (even though it could be created under other circumstances), then set to inconsistentName; otherwise,

n) If the Object Name specifies a variable that exists but can not be modified no matter what new value is specified, then set to notWritable; otherwise,

o) If the Object Value value field specifies a value that could under other circumstances be held by the variable, but is presently inconsistent or otherwise unable to be assigned to the variable, then set to inconsistentValue; otherwise,

p) If the assignment of the value specified by the Object Value value field to the specified variable requires the allocation of a resource that is presently unavailable, then set to resourceUnavailable; otherwise,

q) Set to noError.

9.9.9 CN Parameters TLV

Figure 9-16 illustrates the format of the CN Parameters TLV.

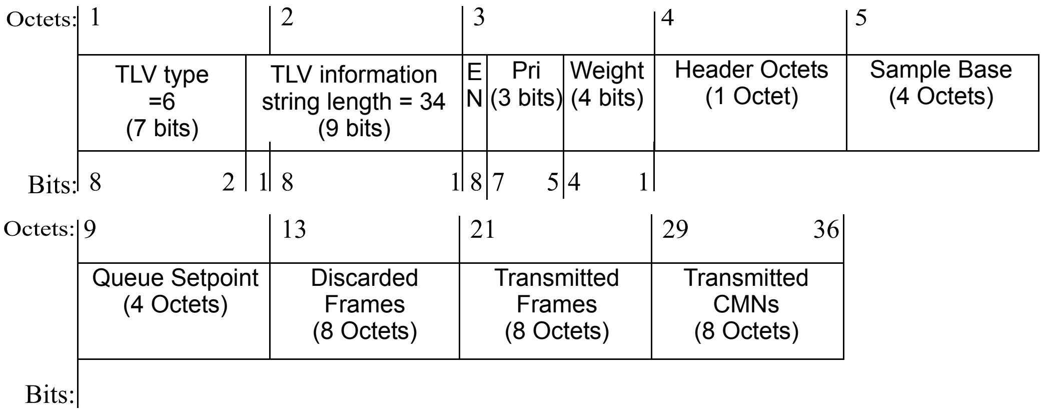

Octets: 1
2
3
4
5
TLV type =6 (7 bits)
TLV information string length = 34 (9 bits)
E N
Pri (3 bits)
Weight (4 bits)
Header Octets (1 Octet)
Sample Base (4 Octets)
Bits: 8 2 1 8 1 8 7 5 4 1
Octets: 9
13
21
29
36
Queue Setpoint (4 Octets)
Discarded Frames (8 Octets)
Transmitted Frames (8 Octets)
Transmitted CMNs (8 Octets)
Bits:

Figure 9-16—CN Parameters TLV

EN: Set to one if congestion notification is enabled on this Port, 0 otherwise.

The remaining fields correspond directly to Congestion Notification Parameters. Some of these fields are reserved when sourced from a Controlling Bridge. The corresponding parameter and whether the field is reserved is specified in Table 9-12.

Table 9-12—CN Parameter Fields

Field	CN Parameter	Reference in IEEE Std 802.1Q	Reserved when sourced by Controlling Bridge
Pri	cngCnmTransmitPriority	32.2.2	—
Weight (see NOTE)	cpW	32.8.6	—
Header Octets	cpMinHeaderOctets	32.8.15	—
Sample Base	cpSampleBase	32.8.11	—
Queue Setpoint	cpQSp	32.8.3	X
Discarded Frames	cpDiscardedFrames	32.8.12	X
Transmitted Frames	cpTransmittedFrames	32.8.13	X
Transmitted CMNs	cpTransmittedCmns	32.8.14	X
NOTE—The value of cpW is equal to two to the power of this field. This field is represented as a signed two's-compliment number. Thus, if this field is set to –1 (0xF), then cpW = 0.5.			

9.9.10 Organizationally Specific TLVs

Organizationally Specific TLVs provide a method by which other organizations, such as software and equipment vendors, may define TLVs that extend the capabilities of the PE CSP.

9.9.10.1 Basic Organizationally Specific TLV format

The basic format for Organizationally Specific TLVs is shown in Figure 9-17.

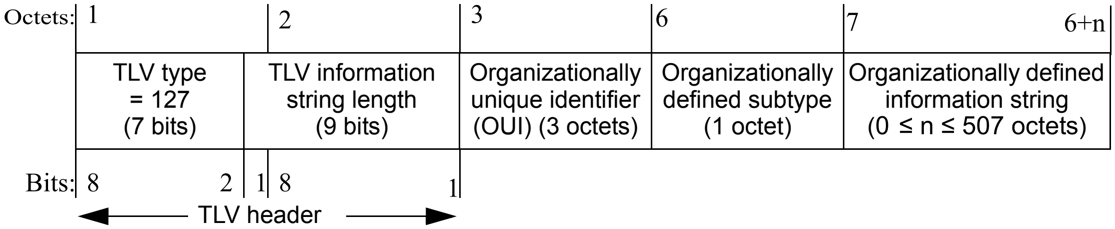

Octets: 1
2
3
6
7
6+n
TLV type = 127 (7 bits)
TLV information string length (9 bits)
Organizationally unique identifier (OUI) (3 octets)
Organizationally defined subtype (1 octet)
Organizationally defined information string (0 ≤ n ≤ 507 octets)
Bits: 8
2
1
8
1
TLV header

Figure 9-17—Basic format for Organizationally Specific TLVs

9.9.10.2 Organizationally unique identifier (OUI)

The Organizationally Unique Identifier is obtainable from IEEE.[9]

9.9.10.3 Organizationally unique subtype

The organizationally defined subtype field contains a unique subtype value assigned by the defining organization. The Subtype is required so that an additional OUI will not be required if more Organization-Specific TLVs are required by an owner of an OUI.

NOTE—Defining organizations are responsible for maintaining listings of organizationally defined subtypes in order to assure uniqueness.

9.9.10.4 Organizationally defined information string

The format of the organizationally defined information string is organizationally specific.

[9] Interested applicants should contact the IEEE Standards Department, Institute of Electrical and Electronics Engineers, 445 Hoes Lane, Piscataway, NJ 08854, USA, http://standards.ieee.org/develop/regauth/oui/index.html.

10. Bridge management

This clause defines the set of managed objects, and their functionality, that supplement those specified in Clause 12 of IEEE Std 802.1Q to allow administrative configuration of a Controlling Bridge and Extended Bridge. This includes the following:

a) The ability to monitor the functional elements of Bridge Port Extension; and

b) The ability to identify E-channels in use and through which Ports of the Controlling Bridge and Bridge Port Extenders they pass.

10.1 Data types

In addition to the data types defined in 12.3 of IEEE Std 802.1Q, the following data types are used:

a) E-channel Identifier (E-CID): an Unsigned value used to identify an E-channel. Valid values are in the range of 0x00 0001 through 0x3F FFFE.

b) Port Map: a set of control indicators, one of each Port of a Bridge or Bridge component, indicating that Port's inclusion within or exclusion from the specified set of Ports.

c) Time Stamp: time in hundredths of seconds that the management subsystem of the controlling bridge has been operational.

10.2 Bridge Port Extension Entries

The objects that comprise this managed resource are as follows:

a) Port Extension Port Table

b) Port Extension Remote Replication Table

10.2.1 Port Extension Port Table

There is one row of the Port Extension Port Table per Port of the C-VLAN component of a Controlling Bridge that connects to a Port on a Bridge Port Extender. Each table row contains the set of parameters detailed in Table 10-1.

The pepPortComponentID and pepPort parameters specify a Port in the Controlling Bridge. The pepPortType parameter specifies the type of Port on the Bridge Port Extender corresponding to the Port in the Controlling Bridge.

If the Bridge Port Extender Port is an Extended Port, then the pepUpstreamCSPAddress, pepRxrqErrorsBridge, pepRxrqErrorsPE, pepRxrspErrorsBridge, and pepRxrspErrorsPE objects will not exist. If the Bridge Port Extender Port is a Cascade Port, then the pepEcid object will not exist.

The pepPortNumber object refers to the port number on the Bridge Port Extender.

The pepRxrqErrorsBridge and pepRxrspErrorsBridge refer to the errors detected on the Controlling Bridge in the PE CSP Receive PDU state machine. The pepRxrqErrorsPE and pepRxrspErrorsPE refer to the errors detected in the PE CSP Receive PDU state machine on the Bridge Port Extender. These values are obtained by the Bridge using the PE CSP. If the PE CSP is unable to obtain these values, then the pepRxrqErrorsPE and pepRxrspErrorsPE objects do not exist.

Table 10-1—Port Extension Port Table row elements

Name	Data type	Operations supported[a]	Conformance[b]	References
pepPortComponentID	ComponentID		B	12.3 in IEEE Std 802.1Q
pepPort	Port Number		B	12.3 in IEEE Std 802.1Q
pepPortType	enum {pepCascade, pepUpstream, pepExetended}	R	B	8.2
pepUpstreamCSPAddress	MAC Address	R	B	9.1
pepEcid	E-channel identifier	R	B	6.1
pepPortNumber	Port Number	R	B	12.3 in IEEE Std 802.1Q
pepCounterDiscontinuityTime	TimeStamp	R	B	7.1.20 in IETF RFC 2578
pepRxrqErrorsBridge	Counter64	R	B	9.2.3.14
pepRxrspErrorsBridge	Counter64	R	B	9.2.3.15
pepRxrqErrorsPE	Counter64	R	B	9.2.3.14
pepRxspErrorsPE	Counter64	R	B	9.2.3.15
pepPCP	Boolean	R	B	9.9.1
pepROW	Boolean	R	B	9.9.1
pepDEI	Boolean	R	B	9.9.1
pepCN	Boolean	R	B	9.9.1
pepPFC	Boolean	R	B	9.9.1
pepExtPortEChannelsSupported	Unsigned [1..1 048 575]	R	B	9.9.1
pepRemoteRepEChannelsSupported	Unsigned [1..3 145 727]	R	B	9.9.1
pepTCsSupported	Unsigned [1..8]	R	B	9.9.1
pepUtVLANsSupported	Unsigned [1..4094]	R	B	9.9.1

[a] R = Read only access.
[b] B = Required for bridge or bridge component support of Bridge Port Extension.

The objects pepPCP, pepROW, pepDEI, pepCN, pepPFC, pepExtPortEChannelsSupported, pepRemoteRepEChannelsSupported, pepTCsSupported, and pepUtVLANsSupported appear only on C-VLAN component Ports associated with Cascade Ports. The values of these objects reflect the values discovered in the Resource Limit Capability TLV provided by the Bridge Port Extender (9.9.1).

10.2.2 Port Extension Remote Replication Table

There is one row of the Port Extension Remote Replication Table for each E-CID allocated by a Controlling Bridge for remote replication. The table row contains the set of parameters detailed in Table 10-2.

Table 10-2—Port Extension Remote Replication Table row elements

Name	Data type	Operations supported[a]	Conformance[b]	References
perrPortComponentID	ComponentID		B	12.3 in IEEE Std 802.1Q
perrE-CID	E-channel Identifier	R	B	8.10.1
perrPortMap	Port Map	R	B	8.10.1

[a] R = Read only access.

[b] B = Required for bridge or bridge component support of Bridge Port Extension.

The pepPortMap object contains a port map of all the C-VLAN component ports. The control entry for each port is set to forward if the corresponding port is attached to an internal Extended Port that corresponds to an external Extended Port that is part of the member set of the E-channel identified by the E-CID. The control element is set to filter otherwise.

10.2.3 Port Extension Upstream Port Enhanced Transmission Selection Table

There is one row of the of the Port Extension Upstream Port Enhanced Transmission Selection (ETS) Table for each traffic class (0–7) for each Cascade Port. These objects apply to the Upstream Port attached to the Cascade Port to enable asymmetric settings of Enhanced Transmission Selection.Each table row contains the set of parameters detailed in Table 10-3.

Table 10-3—Port Extension Upstream Port ETS Table row elements

Name	Data type	Operations supported[a]	Conformance[b]	References in IEEE Std 802.1Q
peetsPortComponentID	ComponentID		B	12.3
peetsPort	Port Number		B	12.3
peetsTrafficClass	Unsigned [0..7]		B	6.5.9
peetsTrafficSelectionAlgorthm	enum {peetsStrictPriority, peets-CreditBasedShaper, peets-EnhancedTransmission, peetsVendorSpecific}	RW	B	D.2.9.8
peetsETSBandwidth	Unsigned [0..100]	RW	B	D.2.9.7

[a] R = Read only access; RW = Read/Write access.

[b] B = Required for bridge or bridge component support of Bridge Port Extension.

11. Management Information Base (MIB)

This clause contains a complete SMIv2 Management Information Base (MIB) for the features of this standard. This MIB module supplements those specified in Clause 17 of IEEE Std 802.1Q, where a discussion of the Internet standard management framework may be found.

11.1 Structure of the IEEE8021-PE MIB

The IEEE8021-PE MIB module provides objects to determine the configuration a Controlling Bridge and Extended Bridge. Objects in this MIB module are arranged into subtrees. Each subtree is organized as a set of related objects. Table 11-1 indicates the structure of the IEEE8021-PE MIB module. The corresponding Clause 10 management reference is also included.

Table 11-1—PE MIB structure and object cross reference

MIB table	MIB object	References
ieee8021BridgePENotifications subtree		
ieee8021BridgePEObjects subtree		
ieee8021BridgePEPortTable		10.2.1
	ieee8021BridgePEPortComponentID[a]	—
	ieee8021BridgePEPort[a]	—
	ieee8021BridgePEPortType	—
	ieee8021BridgePEPortUpstreamCSPAddress	—
	ieee8021BridgePEPortEcid	—
	ieee8021BridgePEPortIndex	—
	ieee8021BridgePECounterDiscontinuityTime	—
	ieee8021BridgePEPortRxrqErrorsBridge	—
	ieee8021BridgePEPortRxrspErrorsBridge	—
	ieee8021BridgePEPortRxrqErrorsPE	—
	ieee8021BridgePEPortRxrspErrorsPE	—
	ieee8021BridgePEPCP	—
	ieee8021BridgePEROW	—
	ieee8021BridgePEDEI	—
	ieee8021BridgePECN	—

Table 11-1—PE MIB structure and object cross reference *(continued)*

MIB table	MIB object	References
	ieee8021BridgePEPFC	—
	ieee8021BridgePEExtPortEChannelsSupported	—
	ieee8021BridgePERemoteRepEChannelsSupported	—
	ieee8021BridgePETCsSupported	—
	ieee8021BridgePEUtVLANsSupported	—
ieee8021BridgePERemoteReplicationTable		10.2.2
	ieee8021BridgePEPortComponentID[a]	—
	ieee8021BridgePERREcid[a]	—
	ieee8021BridgePERRPortMap	—
ieee8021BridgePEETSTable		10.2.3
	ieee8021BridgePEPortComponentId[a]	—
	ieee8021BridgePEPort[a]	—
	ieee8021BridgePEETSTrafficClass[a]	—
	ieee8021BridgePEETSTrafficSelectionAlgorthm	—
	ieee8021BridgePEETSBandwidth	—
ieee8021BridgePEConformance subtree		—
ieee8021BridgePEGroups		—
	ieee8021BridgePEGroup	—
ieee8021BridgePECompliances		—
	ieee8021BridgePECompliance	—

[a] This object is an INDEX of the table in which it resides.

11.2 Relationship to other MIBs

The IEEE8021-PE MIB module provides objects that extend the core management functionality of a Bridge, as defined by the IEEE8021-BRIDGE MIB (IEEE Std 802.1™), in order to support the management functionality needed to support this standard. Support of the objects defined in the IEEE8021-PE MIB module also requires support of the IEEE8021-TC-MIB specified in IEEE Std 802.1.

11.3 Security considerations

All of the objects in this MIB module may be considered to be sensitive or vulnerable in some network environments.

Each of the readable objects are discussed as follows.

Many of the objects may be used to determine the topology and interconnection of an Extended Bridge. This knowledge could be used by an attacker to determine which attacks might be useful against a given device and how to target such an attack. These objects include the following:

a) ieee8021BridgePEPortComponentID
b) ieee8021BridgePEPort
c) ieee8021BridgePEPortType
d) ieee8021BridgePEPortUpstreamCSPAddress
e) ieee8021BridgePEPortEcid
f) ieee8021BridgePEPortIndex

Objects included in this MIB may be used to determine the routing of multicast frames. This knowledge could be used by an attacker to determine which attacks might be useful against a given device and how to target such an attack. These objects include the following:

g) ieee8021BridgePERREcid
h) ieee8021BridgePERRPortMap

Objects in this MIB provide counts of specific error conditions. This knowledge may be used by an attacker to access the success of an attack or its clandestinity. These objects include the following:

i) ieee8021BridgePECounterDiscontinuityTime
j) ieee8021BridgePEPortRxrqErrorsBridge
k) ieee8021BridgePEPortRxrspErrorsBridge
l) ieee8021BridgePEPortRxrqErrorsPE
m) ieee8021BridgePEPortRxrspErrorsPE

Objects in this MIB provide information on the capabilities of attached Bridge Port Extenders. This knowledge could be used by attacker to determine which attacks might be useful against a given device and how to target such an attack. These objects include the following:

n) ieee8021BridgePEPCP
o) ieee8021BridgePEROW
p) ieee8021BridgePEDEI
q) ieee8021BridgePECN
r) ieee8021BridgePEPFC
s) ieee8021BridgePEExtPortEChannelsSupported
t) ieee8021BridgePERemoteRepEChannelsSupported
u) ieee8021BridgePETCsSupported
v) ieee8021BridgePEUtVLANsSupported

The read-write objects are discussed as follows.

Objects in the MIB may be used to determine or configure the transmission selection algorithm and associated parameters of an Upstream Port. This could be used by an attacker to disrupt bandwidth allocations resulting in denial of service. These objects include the following:

w) ieee8021BridgePEETSTrafficSelectionAlgorthm

x) ieee8021BridgePEETSBandwidth

SNMP versions prior to SNMPv3 did not include adequate security. Even if the network itself is secure (for example by using IPSec), even then, there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in this MIB module.

Implementers should consider the security features as provided by the SNMPv3 framework (see RFC 3410, section 8 [B3]), including full support for the SNMPv3 cryptographic mechanisms (for authentication and privacy).

Further, implementers should not deploy SNMP versions prior to SNMPv3. Instead, implementers should deploy SNMPv3 to enable cryptographic security. It is then a customer/operator responsibility to ensure that the SNMP entity giving access to an instance of this MIB module is properly configured to give access to the objects only to those principals (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

11.4 Definition of the IEEE8021-PE MIB Module[10, 11]

```
IEEE8021-PE-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY,
    OBJECT-TYPE,
    Unsigned32,
    Counter64
        FROM SNMPv2-SMI
    MODULE-COMPLIANCE,
    OBJECT-GROUP
        FROM SNMPv2-CONF
    MacAddress,
    TEXTUAL-CONVENTION,
    TimeStamp,
    TruthValue
        FROM SNMPv2-TC
    ieee802dot1mibs,
    IEEE8021PbbComponentIdentifier,
    IEEE8021BridgePortNumber,
    IEEE8021BridgePortNumberOrZero
        FROM IEEE8021-TC-MIB
    PortList
        FROM Q-BRIDGE-MIB;

ieee8021BridgePEMib MODULE-IDENTITY
    LAST-UPDATED    "201201220000Z" -- January 22, 2012, 0000Z
    ORGANIZATION    "IEEE 802.1 Working Group"
    CONTACT-INFO
            "WG-URL: http:////www.ieee802.org/1/
            WG-EMail: stds-802-1-L@IEEE.ORG

             Contact: Tony Jeffree
              Postal: C/O IEEE 802.1 Working Group
```

[10] *Copyright release for MIBs:* Users of this standard may freely reproduce the MIB contained in this subclause so that it can be used for its intended purpose.

[11] An ASCII version of this MIB module can be obtained from the IEEE 802.1 Website at http://www.ieee802.org/1/pages/MIBS.html.

```
                      IEEE Standards Association
                      445 Hoes Lane
                      Piscataway
                      NJ 08854
                      USA
              E-mail: stds-802-1-L@IEEE.ORG"
    DESCRIPTION
        "The PE MIB module for managing devices that support
        Bridge Port Extension.

        Unless otherwise indicated, the references in this MIB
        module are to IEEE Std 802.1BR-2012.

        Copyright (C) IEEE.
        This version of this MIB module is part of
        IEEE 802.1BR-2012; see the specification itself
        for full legal notices."

    REVISION         "201201220000Z" -- January 22, 2012, 0000Z
    DESCRIPTION
        "Initial version published as part of IEEE Std. 802.1BR-2012"

    ::= { ieee802dot1mibs 25 }


-- =============================================================
-- subtrees in the PE MIB
-- =============================================================

ieee8021BridgePENotifications  OBJECT IDENTIFIER
    ::= { ieee8021BridgePEMib 1 }

ieee8021BridgePEObjects  OBJECT IDENTIFIER
    ::= { ieee8021BridgePEMib 2 }

ieee8021BridgePEConformance  OBJECT IDENTIFIER
    ::= { ieee8021BridgePEMib 3 }


-- Textual Conventions

IEEE802BridgePEEChannelIDTC ::= TEXTUAL-CONVENTION
    DISPLAY-HINT    "d"
    STATUS          current
    DESCRIPTION
        "Textual convention of an E-Channel Identifier."
    SYNTAX          Unsigned32 (1..4194302)

IEEE802BridgePETrafficClassValueTC ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "d"
    STATUS current
    DESCRIPTION
        "Indicates a traffic class. Values 0-7 correspond to
        traffic classes."
    SYNTAX Unsigned32 (0..7)

IEEE802BridgePETrafficSelectionAlgorithmTC ::= TEXTUAL-CONVENTION
    STATUS current
    DESCRIPTION
```

```
        "Indicates the Traffic Selection Algorithm
        0: Strict Priority
        1: Credit-based shaper
        2: Enhanced transmission selection
        3-254: Reserved for furture standardization
        255: Vendor specific"
    SYNTAX INTEGER {
        tsaStrictPriority(0),
        tsaCreditBasedShaper(1),
        tsaEnhancedTransmission(2),
        tsaVendorSpecific(255)
        }

IEEE802BridgePETrafficClassBandwidthValue ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "d"
    STATUS current
    DESCRIPTION
        "Indicates the bandwidth in percent assigned to a
        traffic class."
      SYNTAX Unsigned32 (0..100)

-- PE port table entry managed object

ieee8021BridgePEPortTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF Ieee8021BridgePEPortEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "A table that contains per port information
        related to Port Extension.  A row is created in this
        table for any port on a Controlling Bridge that is
        extended using Port Extension, including those ports
        that provide communication to the Port Extenders
        themselves."
    REFERENCE       "10.2.1"
    ::= { ieee8021BridgePEObjects 1 }

ieee8021BridgePEPortEntry OBJECT-TYPE
    SYNTAX          Ieee8021BridgePEPortEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "A list of per port Port Extension objects."
    INDEX           {
                        ieee8021BridgePEPortComponentId,
                        ieee8021BridgePEPort,
                        ieee8021BridgePEPortType
                    }
    ::= { ieee8021BridgePEPortTable 1 }

Ieee8021BridgePEPortEntry ::= SEQUENCE {
        ieee8021BridgePEPortComponentId
            IEEE8021PbbComponentIdentifier,
        ieee8021BridgePEPort
            IEEE8021BridgePortNumber,
        ieee8021BridgePEPortType
            INTEGER,
        ieee8021BridgePEPortUpstreamCSPAddress
            MacAddress,
        ieee8021BridgePEPortEcid
            IEEE802BridgePEEChannelIDTC,
        ieee8021BridgePEPortNumber
            IEEE8021BridgePortNumberOrZero,
        ieee8021BridgePECounterDiscontinuityTime
            TimeStamp,
        ieee8021BridgePEPortRxrqErrorsBridge
            Counter64,
        ieee8021BridgePEPortRxrspErrorsBridge
            Counter64,
        ieee8021BridgePEPortRxrqErrorsPE
            Counter64,
        ieee8021BridgePEPortRxrspErrorsPE
            Counter64,
        ieee8021BridgePEPCP
            TruthValue,
        ieee8021BridgePEROW
            TruthValue,
        ieee8021BridgePEDEI
            TruthValue,
        ieee8021BridgePECN
            TruthValue,
        ieee8021BridgePEPFC
            TruthValue,
        ieee8021BridgePEExtPortEChannelsSupported
            Unsigned32,
        ieee8021BridgePERemoteRepEChannelsSupported
            Unsigned32,
        ieee8021BridgePETCsSupported
            Unsigned32,
        ieee8021BridgePEUtVLANsSupported
            Unsigned32
}

ieee8021BridgePEPortComponentId OBJECT-TYPE
    SYNTAX          IEEE8021PbbComponentIdentifier
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "The component identifier is used to distinguish between the
        multiple virtual bridge instances within a PBB. In simple
        situations where there is only a single component the default
        value is 1."
    ::= { ieee8021BridgePEPortEntry 1 }

ieee8021BridgePEPort OBJECT-TYPE
    SYNTAX          IEEE8021BridgePortNumber
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "The port number of the port for which this entry
        contains bridge management information."
    ::= { ieee8021BridgePEPortEntry 2 }

ieee8021BridgePEPortType OBJECT-TYPE
    SYNTAX          INTEGER  {
                        pepCascade(1),
                        pepUpstream(2),
                        pepExtended(3)
```

```
                    }
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "The operational mode of a port participating in
        Port Exension.  The enumerated values are:
        pepCascade - the port is operating as a Cascade port
        pepUpstream - the port is operating as an Upstream port
        pepExtended - the port is operating as an Extended port"
    REFERENCE       "10.2.1"
    ::= { ieee8021BridgePEPortEntry 3 }

ieee8021BridgePEPortUpstreamCSPAddress OBJECT-TYPE
    SYNTAX          MacAddress
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "The MAC address used for communication of the PE CSP
        protocol of the device connected to the upstream port
        of the Port Extender (which may be the Controlling
        Bridge or an upstream Port Extender).  This provides
        the hierarchal relationship in a cascade of Port
        Extenders"
    REFERENCE       "10.2.1"
    ::= { ieee8021BridgePEPortEntry 4 }

ieee8021BridgePEPortEcid OBJECT-TYPE
    SYNTAX          IEEE802BridgePEEChannelIDTC
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "The default ECID assigend to this port and the port
        on the Port Extender to which this port corresponds."
    REFERENCE       "10.2.1"
    ::= { ieee8021BridgePEPortEntry 5 }

ieee8021BridgePEPortNumber OBJECT-TYPE
    SYNTAX          IEEE8021BridgePortNumberOrZero
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "The Port number on the of the Port on the Port Extender,
        or zero for the Upstream Port."
    REFERENCE       "10.2.1"
    ::= { ieee8021BridgePEPortEntry 6 }

ieee8021BridgePECounterDiscontinuityTime OBJECT-TYPE
    SYNTAX          TimeStamp
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "The value of sysUpTime on the most recent occasion at which
        any one or more of the counters in this conceptaul row
        suffered a discontinuity.  The relevant counters are the
        specific instances associated with this conceptual row of
        any Counter32 or Counter64 object. If no such discontinuities
        have occurred since the last re-initialization of the local
        management subsystem, then this object contains a zero value."
    ::= { ieee8021BridgePEPortEntry 7 }
```

```
ieee8021BridgePEPortRxrqErrorsBridge OBJECT-TYPE
    SYNTAX          Counter64
    UNITS           "frames"
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "The number of PE CSP flow control overflow errors
        that have occured for requests on the Bridge."
    REFERENCE       "10.2.1"
    ::= { ieee8021BridgePEPortEntry 8 }

ieee8021BridgePEPortRxrspErrorsBridge OBJECT-TYPE
    SYNTAX          Counter64
    UNITS           "octets"
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "The number of PE CSP flow control overflow errors
        that have occured for responses on the Bridge."
    REFERENCE       "10.2.1"
    ::= { ieee8021BridgePEPortEntry 9 }

ieee8021BridgePEPortRxrqErrorsPE OBJECT-TYPE
    SYNTAX          Counter64
    UNITS           "frames"
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "The number of PE CSP flow control overflow errors
        that have occured for requests on the Port Extender."
    REFERENCE       "10.2.1"
    ::= { ieee8021BridgePEPortEntry 10 }

ieee8021BridgePEPortRxrspErrorsPE OBJECT-TYPE
    SYNTAX          Counter64
    UNITS           "octets"
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "The number of PE CSP flow control overflow errors
        that have occured for responses on the Port Extender."
    REFERENCE       "10.2.1"
    ::= { ieee8021BridgePEPortEntry 11 }

ieee8021BridgePEPCP OBJECT-TYPE
    SYNTAX          TruthValue
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "Indicates whether the Port Exender supports
        modification of the priority code point
        table (true) or not (false)."
    REFERENCE       "10.2.1"
    ::= { ieee8021BridgePEPortEntry 12 }

ieee8021BridgePEROW OBJECT-TYPE
    SYNTAX          TruthValue
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "Indicates whether the Port Extender supports
        rows in the PCP table in addition to the 8P0D
        row (true)or not (false)."
    REFERENCE       "10.2.1"
    ::= { ieee8021BridgePEPortEntry 13 }

ieee8021BridgePEDEI OBJECT-TYPE
    SYNTAX          TruthValue
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "Indicates whether the Port Extender supports
        encoding of the Drop Eligible Indicatior (true)
        or not (false)."
    REFERENCE       "10.2.1"
    ::= { ieee8021BridgePEPortEntry 14 }

ieee8021BridgePECN  OBJECT-TYPE
    SYNTAX          TruthValue
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "Indicates whether the Port Extender supports
        Congestion Notification (true) or not (false)."
    REFERENCE       "10.2.1"
    ::= { ieee8021BridgePEPortEntry 15 }

ieee8021BridgePEPFC OBJECT-TYPE
    SYNTAX          TruthValue
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "Indicates whether the Port Extender supports
        Priority-based flow control(true) or
        not (false)."
    REFERENCE       "10.2.1"
    ::= { ieee8021BridgePEPortEntry 16 }

ieee8021BridgePEExtPortEChannelsSupported OBJECT-TYPE
    SYNTAX       Unsigned32 (1..1048575)
    UNITS          "E-channels"
    MAX-ACCESS     read-only
    STATUS         current
    DESCRIPTION
        "Indicates the number of Extended Port
        E-channels supported by the Port Extender."
    REFERENCE       "10.2.1"
    ::= { ieee8021BridgePEPortEntry 17 }

ieee8021BridgePERemoteRepEChannelsSupported  OBJECT-TYPE
    SYNTAX       Unsigned32 (1..3145727)
    UNITS          "E-channels"
    MAX-ACCESS     read-only
    STATUS         current
    DESCRIPTION
        "Indicates the number of Remote Replication
        E-channels supported by the Port Extender."
    REFERENCE       "10.2.1"
    ::= { ieee8021BridgePEPortEntry 18 }

ieee8021BridgePETCsSupported OBJECT-TYPE
    SYNTAX       Unsigned32 (1..8)
    UNITS          "traffic classes"
    MAX-ACCESS     read-only
    STATUS         current
    DESCRIPTION
        "Indicates the number of traffic clasees
        supported by the Port Extender."
    REFERENCE       "10.2.1"
    ::= { ieee8021BridgePEPortEntry 19 }

ieee8021BridgePEUtVLANsSupported OBJECT-TYPE
    SYNTAX       Unsigned32 (1..4094)
    UNITS          "VLANs"
    MAX-ACCESS     read-only
    STATUS         current
    DESCRIPTION
        "Indicates the number of untagged VLANs
        supported by the Port Extender."
    REFERENCE       "10.2.1"
    ::= { ieee8021BridgePEPortEntry 20 }

-- PE Remote Replication entry table managed object

ieee8021BridgePERemoteReplicationTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF Ieee8021BridgePERemoteReplicationEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "A table that contains one row for each Remote Replication
        entry in the filtering database."
    REFERENCE       "10.3.1"
    ::= { ieee8021BridgePEObjects 2 }

ieee8021BridgePERemoteReplicationEntry OBJECT-TYPE
    SYNTAX          Ieee8021BridgePERemoteReplicationEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "A list of Remote Replication objects."
    INDEX           {
                        ieee8021BridgePEPortComponentId,
                        ieee8021BridgePERREcid
                    }
    ::= { ieee8021BridgePERemoteReplicationTable 1 }

Ieee8021BridgePERemoteReplicationEntry ::= SEQUENCE {
        ieee8021BridgePERREcid    IEEE802BridgePEEChannelIDTC,
        ieee8021BridgePERRPortMap PortList
}

ieee8021BridgePERREcid OBJECT-TYPE
    SYNTAX          IEEE802BridgePEEChannelIDTC
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
```

```
        "The ECID assigend to this Remote Replication
        filtering entry."
    REFERENCE       "10.3.1"
    ::= { ieee8021BridgePERemoteReplicationEntry 1 }

ieee8021BridgePERRPortMap OBJECT-TYPE
    SYNTAX          PortList
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "The list of ports to which a frame is to be
        replicated."
    REFERENCE       "10.3.1"
    ::= { ieee8021BridgePERemoteReplicationEntry 2 }

--PE Enhanced Transmission Selection table managed object

ieee8021BridgePEETSTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF Ieee8021BridgePEETSEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "A table that contains per port information
        related to Enhanced Transmission Selection.
        A row is created in this table for any port on a
        Controlling Bridge that corresponds to a Cascade
        Port. These objects refer to the ETS configuration
        of the attached Upstream Port"
    REFERENCE       "10.2.2"
    ::= { ieee8021BridgePEObjects 3 }

ieee8021BridgePEETSEntry OBJECT-TYPE
    SYNTAX          Ieee8021BridgePEETSEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "A list of per Cascade Port ETS objects."
    INDEX           {
                        ieee8021BridgePEPortComponentId,
                        ieee8021BridgePEPort,
                        ieee8021BridgePEETSTrafficClass
                    }
    ::= { ieee8021BridgePEETSTable 1 }

Ieee8021BridgePEETSEntry ::= SEQUENCE {
        ieee8021BridgePEETSTrafficClass
            IEEE802BridgePETrafficClassValueTC,
        ieee8021BridgePEETSTrafficSelectionAlgorthm
            IEEE802BridgePETrafficSelectionAlgorithmTC,
        ieee8021BridgePEETSBandwidth
            IEEE802BridgePETrafficClassBandwidthValue
}

ieee8021BridgePEETSTrafficClass OBJECT-TYPE
    SYNTAX IEEE802BridgePETrafficClassValueTC
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Indicates the traffic class to
        which this bandwidth applies"
    REFERENCE
        "10.2.3"
    ::= { ieee8021BridgePEETSEntry 1 }

ieee8021BridgePEETSTrafficSelectionAlgorthm OBJECT-TYPE
    SYNTAX IEEE802BridgePETrafficSelectionAlgorithmTC
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "Inticates the Traffic Selection Algorthm
        assigned to this traffic class"
    REFERENCE
        "10.2.3"
::= { ieee8021BridgePEETSEntry 2 }

ieee8021BridgePEETSBandwidth OBJECT-TYPE
    SYNTAX IEEE802BridgePETrafficClassBandwidthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "Indicates the bandwidth assigned to this traffic class."
    REFERENCE
        "10.2.3"
::= { ieee8021BridgePEETSEntry 3 }

-- ===============================================================
-- Conformance Information
-- ===============================================================

ieee8021BridgePEGroups  OBJECT IDENTIFIER
    ::= { ieee8021BridgePEConformance 1 }

ieee8021BridgePECompliances  OBJECT IDENTIFIER
    ::= { ieee8021BridgePEConformance 2 }


-- ===============================================================
-- Units of conformance
-- ===============================================================

ieee8021BridgePEGroup OBJECT-GROUP
    OBJECTS         {
                        ieee8021BridgePEPortUpstreamCSPAddress,
                        ieee8021BridgePEPortEcid,
                        ieee8021BridgePEPortNumber,
                        ieee8021BridgePECounterDiscontinuityTime,
                        ieee8021BridgePEPortRxrqErrorsBridge,
                        ieee8021BridgePEPortRxrspErrorsBridge,
                        ieee8021BridgePEPortRxrqErrorsPE,
                        ieee8021BridgePEPortRxrspErrorsPE,
                        ieee8021BridgePEPCP,
                        ieee8021BridgePEROW,
                        ieee8021BridgePEDEI,
                        ieee8021BridgePECN,
                        ieee8021BridgePEPFC,
                        ieee8021BridgePEExtPortEChannelsSupported,
                        ieee8021BridgePERemoteRepEChannelsSupported,
                        ieee8021BridgePETCsSupported,
                        ieee8021BridgePEUtVLANsSupported,
                        ieee8021BridgePERRPortMap,
                        ieee8021BridgePEETSTrafficSelectionAlgorthm,
                        ieee8021BridgePEETSBandwidth
                    }
    STATUS          current
    DESCRIPTION
        "The collection of objects used to represent
        Port Extension management objects."
    ::= { ieee8021BridgePEGroups 1 }

-- =============================================================
-- compliance statements
-- =============================================================

ieee8021BridgePECompliance MODULE-COMPLIANCE
    STATUS          current
    DESCRIPTION
        "The compliance statement for devices supporting PE
        as defined in IEEE 802.1BR."
    MODULE          -- this module
    MANDATORY-GROUPS { ieee8021BridgePEGroup }
    ::= { ieee8021BridgePECompliances 1 }

END
```

Annex A

(normative)

PICS proforma[12]

A.1 Introduction

The supplier of a protocol implementation that is claimed to conform to this standard shall complete the following 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 the protocol have been implemented. The PICS can have a number of uses, including use by the following:

a) Protocol implementers, as a checklist to reduce the risk of failure to conform to the standard through oversight

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

c) User—or potential user—of the implementation, as a basis for initially checking the possibility of interworking with another implementation (note that although interworking can never be guaranteed, failure to interwork can often be predicted from incompatible PICs)

d) Protocol tester, as the basis for selecting appropriate tests against which to assess the claim for conformance of the implementation

A.2 Abbreviations and special symbols

A.2.1 Status symbols

M	mandatory
O	optional
O.n	optional, but support of at least one of the group of options labeled by the same numeral n is required
X	prohibited
pred:	conditional-item symbol, including predicate identification: see A.3.4
¬	logical negation, applied to a conditional item’s predicate

A.2.2 General abbreviations

N/A	not applicable
PICS	Protocol Implementation Conformance Statement

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

A.3 Instructions for completing the PICS proforma

A.3.1 General structure of 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 several subclauses, each containing a number of individual items. Answers to the questionnaire items are to be provided in the rightmost column, either by simply marking an answer to indicate a restricted choice (usually Yes or No) 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 records the status of the item—whether support is mandatory, optional, or conditional: see also A.3.4. The fourth column contains the reference or references to the material that specifies the item in the main body of this standard, and the fifth column provides the space for the answers.

A 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 Ai or Xi, 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 and presentation.

A completed PICS proforma, including any Additional Information and Exception Information, is the Protocol Implementation Conformation Statement for the implementation in question.

NOTE—Where an implementation is capable of being configured in more than one way, a 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, in case that makes for easier and clearer presentation of the information.

A.3.2 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 will be supplied, and a 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 information about aspects of the implementation that are outside the scope of this standard but that have a bearing on the answers to some items.

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.

A.3.3 Exception Information

It may occasionally happen that a supplier will wish to answer an item with mandatory status (after any conditions have been applied) in a way that conflicts with the indicated requirement. No pre-printed answer will be found in the Support column for this item. Instead, the supplier shall write the missing answer into the Support column, together with an Xi reference to an item of Exception Information, and shall 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—A possible reason for the situation described previously is that a defect in this standard has been reported, a correction for which is expected to change the requirement not met by the implementation.

A.3.4 Conditional status

A.3.4.1 Conditional items

The PICS proforma contains a number of conditional items. These are items for which both the applicability of the item itself, and its status if it does apply—mandatory or optional—are dependent on whether certain other items are supported.

Where a group of items is subject to the same condition for applicability, a separate preliminary question about the condition appears at the head of the group, with an instruction to skip to a later point in the questionnaire if the "Not Applicable" answer is selected. Otherwise, individual conditional items are indicated by a conditional symbol in the Status column.

A conditional symbol is of the form "pred: S" where pred is a predicate as described in A.3.4.2, and S is a status symbol, M, O, or X.

If the value of the predicate is true (see A.3.4.2), the conditional item is applicable, and its status is indicated by the status symbol following the predicate: The answer column is to be marked in the usual way. If the value of the predicate is false, the "Not Applicable" (N/A) answer is to be marked.

A.3.4.2 Predicates

A predicate is one of the following:

a) An item-reference for an item in the PICS proforma: The value of the predicate is true if the item is marked as supported and is false otherwise;

b) A predicate-name, for a predicate defined as a Boolean expression constructed by combining item references using the Boolean operator OR: The value of the predicate is true if one or more of the items is marked as supported;

c) The logical negation symbol "¬" prefixed to an item-reference or predicate-name: The value of the predicate is true if the value of the predicate formed by omitting the "¬" symbol is false, and vice versa.

Each item whose reference is used in a predicate or predicate definition, or in a preliminary question for grouped conditional items, is indicated by an asterisk in the Item column.

A.4 PICS proforma for IEEE Std 802.1BR—Bridge Port Extension

A.4.1 Implementation identification

Supplier	
Contact point for queries about the PICS	
Implementation Name(s) and Version(s)	
Other information necessary for full identification, e.g., name(s) and version(s) of machines and/or operating system names	

NOTE 1—Only the first three items are required for all implementations; other information may be completed as appropriate in meeting the requirement for full identification.

NOTE 2—The terms "Name" and "Version" should be interpreted appropriately to correspond with a supplier's terminology (e.g., Type, Series, Model).

A.4.2 Protocol summary, IEEE Std 802.1BR

Identification of protocol specification	IEEE Std 802.1BR-2012, IEEE Standard for Local and metropolitan area networks—Virtual Bridged Local Area Networks—Bridge Port Extension
Identification of amendments and corrigenda to the PICS proforma that have been completed as part of the PICS	Amd. : Corr. : Amd. : Corr. :
Have any Exception items been required? (See A.3.3: the answer "Yes" means that the implementation does not conform to IEEE Std 802.1BR.)	No [ ] Yes [ ]

Date of Statement	

A.5 Bridge Port Extender

Item	Feature	Status	References	Support
PEXT-1	If this implementation is not a Bridge Port Extender, mark N/A and ignore the rest of this table.			N/A [ ]
PEXT-2	Does the implementation meet the requirements of a conformant implementation listed in 5.3?	M	5.3	Yes [ ]
PEXT-3	If the implementation is an External Bridge Port Extender, does the implementation meet the requirements of a conformant implementation of such a Bridge Port Extender listed in 5.3?	M	5.3	Yes [ ] N/A [ ]
PEXT-4	If the implementation is an External Bridge Port Extender, does the implementation support Priority-based Flow Control (PFC)?	O	5.3	Yes [ ] No [ ] N/A [ ]
PEXT-4	Does the implementation support congestion notification?	O	5.3	Yes [ ] No [ ]
PEXT-5	Does the implementation meet the requirements for support of congestion notification?	PEXT-4:M	5.3	Yes [ ] N/A [ ]
PEXT-6	Does the implementation support at least one entry in the untagged set of the Tag Handler?	M	6.9	Yes [ ]
PEXT-7	Does the implementation process data indications as specified in 6.9.1?	M	6.9.1	Yes [ ]
PEXT-8	Does the implementation process data requests destined to external Extended Ports as specified in 6.9.2?	M	6.9.2	Yes [ ]
PEXT-9	Does the implementation process data requests destined to internal Extended Ports as specified in 6.9.2?	M	6.9.2	Yes [ ]
PEXT-10	Does the implementation support a PCID for each Port supported by the PEISS?	M	6.10.4	Yes [ ]
PEXT-11	Are E-TAGs formatted as specified?	M	6.10.6	Yes [ ]
PEXT-12	Does the implementation provide a remote replication registration table?	M	6.10.7	Yes [ ]
PEXT-13	Does each entry in the Remote Replication Registration Table comprise the required elements?	M	6.10.7	Yes [ ]
PEXT-14	Is each frame submitted to the MAC Relay Entity forwarded subject to the constituent functions of the Forwarding Process?	M	6.11	Yes [ ]

A.5 Bridge Port Extender *(continued)*

Item	Feature	Status	References	Support
PEXT-15	Does the Forwarding Process queue each received frame to each of the potential transmission Ports that is present in the member set?	M	6.11.3	Yes [ ]
PEXT-16	Is frame ordering preserved?	M	6.11.4	Yes [ ]
PEXT-17	Does the implementation provide a global Bridge Port Extender transit delay parameter with the value specified?	M	6.11.5	Yes [ ]
PEXT-18	Does the implementation remove frames as specified in 6.11.5?	M	6.11.5	Yes [ ]
PEXT-19	Does the implementation provide a queue management algorithm that attempts to improve the QoS provided by deterministically or probabilistically managing the queue depth based on the current and past queue depth?	O	6.11.5	Yes [ ] No [ ]
PEXT-20	Does the implementation ensure that the probability of removing a frame with drop_eligible set shall not be less than that of removing a frame with drop_eligible False, all other conditions being equal?	M	6.11.5	Yes [ ]
PEXT-21	If a queue management algorithm is implemented, does it preferentially discard frames with drop_eligible True?	O	6.11.5	Yes [ ] No [ ] N/A [ ]
PEXT-22	Is the strict priority transmission selection algorithm supported as the default algorithm for selecting frames for transmission?	M	6.11.6	Yes [ ]
PEXT-23	Is the Enhanced Transmission Selection algorithm supported?	O	6.11.6	Yes [ ] No [ ]
PEXT-24	If the Bridge Port Extender is an External Bridge Port Extender, does the implementation provide exactly one Upstream Port at any given time?	M	6.13	Yes [ ] N/A [ ]
PEXT-25	Does the implementation support more than one Port that is capable of being the Upstream Port?	O	6.13	Yes [ ] No [ ]
PEXT-26	Does the implementation support the required mechanism for selecting the Upstream Port?	PEXT-33:M	6.13	Yes [ ] N/A [ ]
PEXT-27	Does the implementation support additional mechanisms to select the Upstream Port	PEXT-33:O	6.13	Yes [ ] No [ ]

A.5 Bridge Port Extender *(continued)*

Item	Feature	Status	References	Support
PEXT-28	If the Bridge Port Extender is an Internal Bridge Port Extender, does the implementation provide no Upstream Ports?	M	6.13	Yes [ ] N/A [ ]
PEXT-29	Is a unique 48-bit Universally Administered MAC Address assigned to the implementation?	M	6.14.1	Yes [ ]
PEXT-30	Is the implementation initialized when specified?	M	6.15	Yes [ ]
PEXT-31	Is the implementation initialized as specified?	M	6.15	Yes [ ]
PEXT-32	Does the implementation support Congestion Points as specified?	PEXT-4:M	6.16	Yes [ ] N/A [ ]
PEXT-33	Does the implementation support Congestion Points as specified?	PEXT-4:M ¬PEXT-4:X	5.3	Yes [ ] No [ ]

A.6 Controlling Bridge

Item	Feature	Status	References	Support
	If this implementation does not support Controlling Bridge functionality, mark N/A and ignore the rest of this table.			N/A [ ]
PECB-1	Does the implementation comprise at least one C-VLAN component that supports the required functionality specified in 5.4?	M	5.4	Yes [ ]
PECB-2	Does the implementation support the instantiation of one or more Bridge Port Extenders connected as specified in Clause 8?	M	5.4	Yes [ ]
PECB-3	Does the implementation implement the PE CSP (Clause 9)?	M	5.4	Yes [ ]
PECB-4	Does the implementation implement LLDP (IEEE Std 802.1AB)?	M	5.4	Yes [ ]
PECB-5	Does the implementation implement the LLDP Port Extension TLV?	M	5.4	Yes [ ]
PECB-6	Does the implementation support the Bridge Port Extension requirements specified in Clause 8?	M	5.4	Yes [ ]
PECB-7	Does the implementation support the Bridge Port Extension management objects?	O	5.4	Yes [ ] No [ ]
PECB-8	Does the implementation support the IEEE8021-PE MIB module?	O	5.4	Yes [ ] No [ ]
PECB-9	Does the implementation support Congestion Points within Bridge Port Extenders?	O	5.4	Yes [ ] No [ ]

A.6 Controlling Bridge *(continued)*

Item	Feature	Status	References	Support
PECB-10	Does the implementation detect the attachment of Bridge Port Extenders using LLDP and the LLDP Port Extension TLV?	M	Clause 8	Yes [ ]
PECB-11	Is the destination address of all LLDP PDUs carrying the Port Extension TLV set to the Nearest non-TPMR Bridge group address?	M	Clause 8	Yes [ ]
PECB-12	Does the implementation perform the required actions for each directly attached Bridge Port Extender?	M	8.3	Yes [ ]
PECB-13	Does the implementation perform the required actions for Upstream Ports	M	8.4	Yes [ ]
PECB-14	Does the implementation perform the required actions for each Extended Port?	M	8.5	Yes [ ]
PECB-15	Does the implementation perform the required actions for each Bridge Port Extender Cascade Port?	M	8.6	Yes [ ]
PECB-16	Does the implementation exclude the connection_identifier parameter from EM_UNITDATA.requests?	M	8.8	Yes [ ]
PECB-17	Is the implementation's allocation of E-CIDs for remote replication unique within a Replication Group?	M	8.9	Yes [ ]
PECB-18	For each combination of Ports that contain at least two Extended Ports within the same Replication Group to which a frame is to be forwarded, does the implementation maintain a Remote Replication Registration entry in the corresponding Internal Bridge Port Extender?	M	8.10.1	Yes [ ]
PECB-19	For each combination of Ports that contain at least two Extended Ports to which a frame is to be forwarded, regardless of their membership in Replication Groups, does the implementation maintain a Remote Replication Registration entry in the corresponding Internal Bridge Port Extender(s)?	O	8.10.1	Yes [ ] No [ ]
PECB-20	Does the implementation maintain Remote Replication Entries as specified in 8.10.1?	M	8.10.1	Yes [ ]
PECB-21	Does the implementation identify potential transmission Ports as specified in 8.10.2?	M	8.10.2	Yes [ ]
PECB-22	Does the implementation utilize remote replication if the frame is to be forwarded to two or more ports within a Replication Group?	M	8.10.3	Yes [ ]
PECB-23	Does the implementation utilize remote replication if the frame is to be forwarded to two or more ports regardless of their membership in a particular Replication Group?	O	8.10.3	Yes [ ] No [ ]
PECB-24	Does the implementation include the port map in the connection_identifier parameter as specified in 8.10.3?	M	8.10.3	Yes [ ]

A.6 Controlling Bridge *(continued)*

Item	Feature	Status	References	Support
PECB-25	For each frame that is to be forwarded using remote replication, does the connection_identifier in the EM_UNITDATA.request primitive carry a port map?	M	8.10.3	Yes [ ]
PECB-26	Does the implementation set the control elements to filter as specified?	M	8.10.3	Yes [ ]
PECB-27	Does the implementation set the control elements to forward as specified?	M	8.10.3	Yes [ ]
PECB-28	Does the implementation set the ingress_echannel_identifier as specified?	M	8.10.3	Yes [ ]
PECB-29	Does the implementation insert a C-TAG is specified?	M	8.10.3	Yes [ ]
PECB-30	Does the implementation configure Bridge Port Extenders for remote replication as required?	M	8.10.4	Yes [ ]
PECB-31	Does the implementation utilize the PE CSP (Clause 9) to achieve the specified configurations?	M	8.10.4	Yes [ ]
PECB-32	Does the implementation assign E-CIDs in accordance with the specified restrictions?	M	8.11	Yes [ ]

A.7 PE CSP—Controlling Bridge

Item	Feature	Status	References	Support
	If this implementation is not a C-VLAN component that implements support for Bridge Port Extension, mark N/A and ignore the rest of this table.			N/A [ ]
CSPCB-1	Does PE CSP utilize the addresses specified?	M	9.1	Yes [ ]
CSPCB-2	Does the implementation support the state machines as specified in 9.2?	M	9.2	Yes [ ]
CSPCB-3	Does the implementation set all reserved fields to zero and ignore them on receive, unless otherwise specified?	M	9.5	Yes [ ]
CSPCB-4	Does each TLV contain a valid type value?	M	9.5.2	Yes [ ]
CSPCB-5	Does the length field of each TLV contain the length of the information string, in octets?	M	9.5.3	Yes [ ]
CSPCB-6	Is the Command TLV the first TLV in all PE CSP PDUs?	M	9.6	Yes [ ]
CSPCB-7	Is the Command TLV constructed and processed as specified in 9.6?	M	9.6	Yes [ ]

A.7 PE CSP—Controlling Bridge *(continued)*

Item	Feature	Status	References	Support
CSPCB-8	Does each PDU contain the required TLVs based on message type?	M	9.6.1	Yes [ ]
CSPCB-9	Does the implementation set the Transaction ID to zero on each CSP Open request?	M	9.6.2	Yes [ ]
CSPCB-10	Does the implementation increment the Transaction ID by 1, and reset to zero upon reaching 256, for each successive request?	M	9.6.2	Yes [ ]
CSPCB-11	Does the implementation set the Transaction ID in a response to that of the corresponding request?	M	9.6.2	Yes [ ]
CSPCB-12	Is a CSP Open Request Message sent to initialize CSP communication?	M	9.8.1	Yes [ ]
CSPCB-13	Is a CSP Open Response Message sent after completion of processing a received CSP Open Request Message?	M	9.8.1	Yes [ ]
CSPCB-14	Does the implementation refrain from sending CSP messages other than CSP Open Messages until a CSP Open Response is received?	M	9.8.1	Yes [ ]
CSPCB-15	Does the implementation place the value one in the Index field in CSP Open Messages?	M	9.8.1	Yes [ ]
CSPCB-16	Does the implementation ignore the value in the Index field of received CSP Open Messages	M	9.8.1	Yes [ ]
CSPCB-17	Does the implementation send an Extended Port Create Response message as specified in response to receiving an Extended Port Create Request message?	M	9.8.2	Yes [ ]
CSPCB-18	Does the implementation send the Extended Port Delete request message to remove an Extended Port from all E-channel member sets?	M	9.8.3	Yes [ ]
CSPCB-19	Does the implementation set the Index field of the Command TLV in Extended Port Delete request messages to the E-CID identifying the E-channel associated with the Port to be deleted?	M	9.8.3	Yes [ ]
CSPCB-20	Does the implementation perform the required processing upon receipt of an Extended Port Delete request?	M	9.8.3	Yes [ ]
CSPCB-21	Does the implementation perform the required processing upon receipt of an Extended Port Delete response?	M	9.8.3	Yes [ ]
CSPCB-22	Does the implementation send a Port Parameters Set request message to a Bridge Port Extender to configure the parameters specified in the Port Parameters TLV and/or the VID array TLV for an Extended Port or for the Upstream Port?	M	9.8.4	Yes [ ]

A.7 PE CSP—Controlling Bridge *(continued)*

Item	Feature	Status	References	Support
CSPCB-23	Does the implementation set the Index field in the Port Parameters Set request message to the E-channel identifying the Extended or Cascade Port, or to zero to indicate the Upstream Port?	M	9.8.4	Yes [ ]
CSPCB-24	Does the implementation send a Port Parameters Get request message to query the currently configured state for a Port on a Bridge Port Extender?	M	9.8.5	Yes [ ]
CSPCB-25	Upon receiving a Port Parameters Get request message, does the implementation send the Port Parameters Get response message to the peer?	M	9.8.5	Yes [ ]
CSPCB-26	Does the implementation set the Index field in the Command TLV of both the request and the response Port Parameters Get messages to the E-CID identifying the Extended or Cascade Port, or to zero to indicate the Upstream Port?	M	9.8.5	Yes [ ]
CSPCB-27	Does the implementation populate Port Parameters and VID Array TLVs with the parameters applicable to the Port when constructing an Port Parameters Get response message?	M	9.8.5	Yes [ ]
CSPCB-28	Does the implementation, upon reception of a Status Parameter Set, set the MAC_Enabled parameter of the corresponding Extended Port within the Internal Bridge Port Extender to match the indication (TRUE or FALSE) received in the Port Status PDU?	M	9.8.6	Yes [ ]
CSPCB-29	Does the implementation send a Status Parameter Set response message to the Bridge Port Extender with Index field set to the E-channel identifying the Extended or Cascade Port as specified?	M	9.8.6	Yes [ ]
CSPCB-30	Does the implementation send the E-channel Register request message to configure a set of Bridge Port Extender Ports within, or to remove them from, the member set of an E-channel?	M	9.8.7	Yes [ ]
CSPCB-31	Does the implementation construct the E-channel Register request message as required?	M	9.8.7	Yes [ ]
CSPCB-32	Does the implementation send the E-channel Registration Get request message to query E-channel member set population?	M	9.8.8	Yes [ ]
CSPCB-33	Does the implementation set the Index field of the Command TLV of the E-channel Registration get request to the E-CID identifying the E-channel being queried?	M	9.8.8	Yes [ ]
CSPCB-34	Does the implementation send the Statistics Get request to Bridge Port Extenders to retrieve the values of the statistics counters?	M	9.8.9	Yes [ ]

A.7 PE CSP—Controlling Bridge *(continued)*

Item	Feature	Status	References	Support
CSPCB-35	Does the implementation send the Transit Delay Set request to Controlling Bridges with the Index field of the Command TLV set to the Transit Delay value in order to set the Bridge Port Extender transit delay value?	M	9.8.10	Yes [ ]
CSPCB-36	Does the implementation send an Object Get request to obtain the value of an object defined by IEEE Std 802.3.1 within a Bridge Port Extender?	M	9.8.11	Yes [ ]
CSPCB-37	Does the implementation send an Object Set request to set the value of an object defined by IEEE Std 802.3.1 within a Bridge Port Extender?	M	9.8.12	Yes [ ]
CSPCB-38	Does the implementation support the CN Parameters Set message?	PECB-9:M ¬PECB-9:X	9.8.13	Yes [ ] No [ ]
CSPCB-39	Does the implementation send a CN Parameters Set message to a Bridge Port Extender to configure the parameters specified in the CN Parameters TLV for a Port?	CSPCB-37:M	9.8.13	Yes [ ] N/A [ ]
CSPCB-40	Does the implementation set the Index field in the Command TLV of a CN Parameters Set message to the E-CID identifying the Extended or Cascade Port, or to zero to indicate the Upstream Port?	CSPCB-37:M	9.8.13	Yes [ ] N/A [ ]
CSPCB-41	Does the implementation support the CN Parameters Get message?	PECB-9:M ¬PECB-9:X	9.8.14	Yes [ ] No [ ]
CSPCB-41	Does the implementation send a CN Parameter Get message to a Bridge Port Extender to query the currently configured congestion notification state for a Port on a Bridge Port Extender?	CSPCB-40:M	9.8.14	Yes [ ] N/A [ ]
CSPCB-42	Does the implementation set the Index field in the Command TLV of a CN Parameters Get message to the E-CID identifying the Extended or Cascade Port, or to zero to indicate the Upstream Port?	CSPCB-40:M	9.8.14	Yes [ ] N/A [ ]
CSPCB-43	Does the implementation construct the additional TLVs as specified?	M	9.9	Yes [ ]

A.8 PE CSP—Bridge Port Extender

Item	Feature	Status	References	Support
	If this implementation is not a Bridge Port Extender, mark N/A and ignore the rest of this table.			N/A [ ]
CSPPE-1	Is PE CSP executed only on the Upstream Port as selected by 6.13?	M	9.1	Yes [ ]
CSPPE-2	Does PE CSP utilize the addresses specified?	M	9.1	Yes [ ]
CSPPE-3	Does the implementation support the state machines as specified in 9.2?	M	9.2	Yes [ ]
CSPPE-4	Does the implementation set all reserved fields to zero and ignore them on receive, unless otherwise specified?	M	9.5	Yes [ ]
CSPPE-5	Does each TLV contain a valid type value?	M	9.5.2	Yes [ ]
CSPPE-6	Does the length field of each TLV contain the length of the information string, in octets?	M	9.5.3	Yes [ ]
CSPPE-7	Is the Command TLV the first TLV in all PE CSP PDUs?	M	9.6	Yes [ ]
CSPPE-8	Is the Command TLV constructed and processed as specified in 9.6?	M	9.6	Yes [ ]
CSPPE-9	Does each PDU contain the required TLVs based on message type?	M	9.6.1	Yes [ ]
CSPPE-10	Does the implementation set the Transaction ID to zero on each CSP Open request?	M	9.6.2	Yes [ ]
CSPPE-11	Does the implementation increment the Transaction ID by 1, and reset to zero upon reaching 256, for each successive request?	M	9.6.2	Yes [ ]
CSPPE-12	Does the implementation set the Transaction ID in a response to that of the corresponding request?	M	9.6.2	Yes [ ]
CSPPE-13	Is a CSP Open Request Message sent to initialize CSP communication?	M	9.8.1	Yes [ ]
CSPPE-14	Upon receipt of a CSP Open message, does the implementation initialize its parameters as specified in Table 44-7 of IEEE Std 802.1Q-2011?	M	9.8.1	Yes [ ]
CSPPE-15	Is a CSP Open Response Message sent after completion of processing a received CSP Open Request Message?	M	9.8.1	Yes [ ]
CSPPE-16	Does the implementation refrain from sending CSP messages other than CSP Open Messages until a CSP Open Response is received?	M	9.8.1	Yes [ ]

A.8 PE CSP—Bridge Port Extender *(continued)*

Item	Feature	Status	References	Support
CSPPE-17	Does the implementation place the value one in the Index field in CSP Open Messages?	M	9.8.1	Yes [ ]
CSPPE-18	Does the implementation ignore the value in the Index field of received CSP Open Messages	M	9.8.1	Yes [ ]
CSPPE-19	Does the implementation send an Extended Port Create message to request a new E-channel binding for an Extended Port with the Index field set to a value identifying the Extended Port?	M	9.8.2	Yes [ ]
CSPPE-20	Does the implementation perform the required processing upon receipt of an Extended Port Create Response message?	M	9.8.2	Yes [ ]
CSPPE-21	Does the implementation send the Extended Port Delete request message to remove an Extended Port from all E-channel member sets?	M	9.8.3	Yes [ ]
CSPPE-22	Does the implementation set the Index field of the Command TLV in Extended Port Delete request messages to the E-CID identifying the E-channel associated with the Port to be deleted?	M	9.8.3	Yes [ ]
CSPPE-23	Does the implementation perform the required processing upon receipt of an Extended Port Delete request?	M	9.8.3	Yes [ ]
CSPPE-24	Does the implementation perform the required processing upon receipt of an Extended Port Delete response?	M	9.8.3	Yes [ ]
CSPPE-25	Upon completion of processing a Port Parameters Set request message, does the implementation send a Port Parameters Set response message with Index field set to the E-channel identifying the Extended or Cascade Port, or to zero to indicate the Upstream Port?	M	9.8.4	Yes [ ]
CSPPE-26	Does the implementation send a Port Parameters Get request message to query the currently configured state for a Port on a Bridge Port Extender?	M	9.8.5	Yes [ ]
CSPPE-27	Upon receiving a Port Parameters Get request message, does the implementation send the Port Parameters Get response message to the peer?	M	9.8.5	Yes [ ]
CSPPE-28	Does the implementation set the Index field in the Command TLV of both the request and the response Port Parameters Get messages to the E-CID identifying the Extended or Cascade Port, or to zero to indicate the Upstream Port?	M	9.8.5	Yes [ ]

A.8 PE CSP—Bridge Port Extender *(continued)*

Item	Feature	Status	References	Support
CSPPE-29	Does the implementation populate Port Parameters and VID Array TLVs with the parameters applicable to the Port when constructing an Port Parameters Get response message?	M	9.8.5	Yes [ ]
CSPPE-30	Does the implementation send a Status Parameter Set request each time the value of MAC_Operational changes on one of its Cascade or Extended Ports?	M	9.8.6	Yes [ ]
CSPPE-31	Does the implementation set the Index field in the Command TLV of Status Parameter Set requests to the PCID of the Extended or Cascade Port?	M	9.8.6	Yes [ ]
CSPPE-32	Upon receipt of a E-channel Register request message, does the implementation perform the required processing?	M	9.8.7	Yes [ ]
CSPPE-33	Does the implementation set the Index field of the Command TLV for the E-channel Registration Get response to the E-CID identifying the E-channel being queried?	M	9.8.8	Yes [ ]
CSPPE-34	Upon receipt of an E-channel Registration Get request, does the implementation send an E-channel Registration Get response containing a Port Array TLV enumerating the Ports that are members of the member set of the E-channel?	M	9.8.8	Yes [ ]
CSPPE-35	Upon receipt of a Statistics Get request message, does the implementation send a Statistics Get response message with the Index field set to that received and the Statistics TLV populated with the values from the Port’s statistics counters?	M	9.8.9	Yes [ ]
CSPPE-36	Upon receipt of a Transit Delay Set request, does the implementation set the value of the Bridge Port Extender transit delay parameter to that in the Index field of the request?	M	9.8.10	Yes [ ]
CSPPE-37	Does the implementation send a Transit Delay Set response to the Controlling Bridge with the Index field of the Command TLV set to that of the request upon completion of setting the transit delay parameter in response to the reception of a Transit Delay Set request?	M	9.8.10	Yes [ ]
CSPPE-38	Upon receipt of a Get Object command, does the implementation respond with an Get Object response?	M	9.8.11	Yes [ ]
CSPPE-39	Upon receipt of a Set Object command, does the implementation respond with an Get Object response?	M	9.8.12	Yes [ ]

A.8 PE CSP—Bridge Port Extender *(continued)*

Item	Feature	Status	References	Support
CSPPE-40	Does the implementation support the CN Parameters Set message?	PEEXT-4:M ¬PEEXT-4:X	9.8.13	Yes [ ] No [ ]
CSPPE-41	Upon reception of a CN Parameter Set message, does the implementation send a CN Parameters Set response message with Index field set to the E-channel identifying the Extended or Cascade Port, or to zero to indicate the Upstream Port?	CSPPE-38:M	9.8.13	Yes [ ] N/A [ ]
CSPPE-42	Does the implementation support the CN Parameters Get message?	PEEXT-4:M ¬PEEXT-4:X	9.8.14	Yes [ ] No [ ]
CSPPE-43	Upon receiving a CN Parameters Get message, does the implementation send the CN Parameters Set response message to the Controlling Bridge?	CSPPE-40:M	9.8.14	Yes [ ] N/A [ ]
CSPPE-44	Does the implementation set the Index field in the Command TLV CN Parameters Get response to the E-CID identifying the Extended or Cascade Port, or to zero to indicate the Upstream Port?	CSPPE-40:M	9.8.14	Yes [ ] N/A [ ]
CSPPE-45	Does the implementation populate the CN Parameters TLV with the parameters applicable to the Port?	CSPPE-40:M	9.8.14	Yes [ ] N/A [ ]
CSPPE-46	Does the implementation construct the additional TLVs as specified?	M	9.9	Yes [ ]

Annex B

(normative)

IEEE 802.1 Organizationally Specific TLVs

B.1 Requirements of IEEE 802.1 Organizationally Specific TLV sets

See D.1 of IEEE Std 802.1Q.

Table B.1 identifies the IEEE 802.1 Organizationally Specific TLV set related to Port Extension:

Table B.1—IEEE 802.1 Organizationally Specific TLVs

IEEE 802.1 subtype	TLV name	TLV set name	TLV reference	Feature clause reference
0F	Port Extension	peSet	B.2	B.2 in IEEE Std 802.1BR

B.2 Port Extension TLV

The Port Extension TLV is a TLV that allows a Bridge or Bridge Port Extender to advertise support for Bridge Port Extension on a given Port. Transmission by a Controlling Bridge indicates that the Port is, or is capable of, operating as a Cascade Port. Transmission by a Extended Bridge through an Extended Port indicates that the Extended Port is, or is capable of, operating as a Cascade Port. Transmission by a Bridge Port Extender indicates that the Port is, or is capable of, operating as an Upstream Port. The value of Cascade Port Priority differentiates between Ports that operate as an Upstream Port versus those that operate as a Cascade Port.

Figure B.1 shows the Port Extension TLV format.

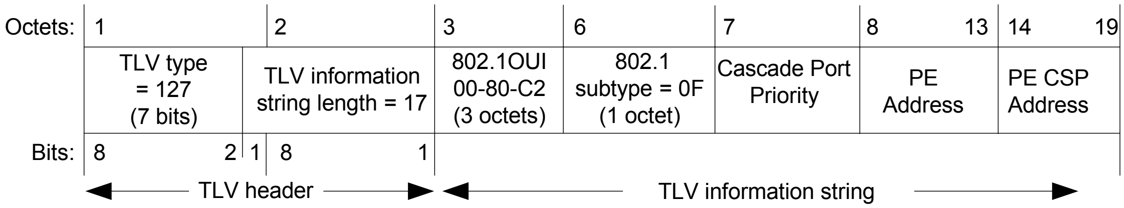


Figure B.1—Port Extension TLV format

B.2.1 Cascade Port Priority

When transmitted from a Port capable of operating as a cascade Port (e.g., Ports of a Controlling Bridge or Extended Ports of an Extended Bridge), this parameter indicates the cascade_port_priority used in determining which Port is to be used by a Bridge Port Extender as its Upstream Port. Valid values are the range from 0 to 254.

When transmitted from a Bridge Port Extender on an Upstream Port or a Port capable of becoming an Upstream Port, this parameter shall be set to 255.

B.2.2 PE Address

When emitted from a Bridge Port Extender, the PE Address contains an unique MAC address that identifies the Bridge Port Extender. This may be the same as the PE CSP address.

When emitted from a Controlling Bridge, the PE Address contains an unique MAC address that identifies the Internal Bridge Port Extender.

B.2.3 PE CSP Address

Contains the MAC address that is to be used for transmission of the Port Extension Control and Status Protocol to the device emitting this TLV. An unique address is emitted from each Port.

B.3 Structure of IEEE 802.1/LLDP Port Extension MIB module

Table B.2—IEEE 802.1/LLDP extension MIB object cross reference

MIB table	MIB object	LLDP reference
Configuration group		
lldpXdot1PeCofigPortExtensionTable		Augments lldpV2Xdot1LocPortExtensionEntry
	lldpXdot1PeConfigPortExtensionTxEnable	B.2
Local system information		
lldpXdot1PeLocPortExtensionTable		
	lldpV2LocPortIfIndex	(Table index)
	lldpXdot1LocPeCascadePortPriority	B.2.1
	lldpXdot1LocPeAddress	B.2.2
	lldpXdot1LocPeCSPAddress	B.2.3
Remote system information		
lldpXdot1PeRemPortExtensionTable		
	lldpV2RemTimeMark	(Table index)
	lldpV2RemLocalIfIndex	(Table index)
	lldpV2RemLocalDestMACAddress	(Table index)
	lldpV2RemIndex	(Table index)
	lldpXdot1PeCascadePortPriority	B.2.1
	lldpXdot1PeAddress	B.2.2
	lldpXdot1PeCSPAddress	B.2.3

B.4 Security considerations for IEEE 802.1 LLDP Port Extension MIB module

There are management objects defined in this MIB module with a MAX-ACCESS clause of read-write.[13] Such objects may be considered sensitive or vulnerable in some network environments. The support for SET operations in a non-secure environment without proper protection can have a negative effect on network operations.

Setting the following objects to incorrect values can result in improper operation of LLDP when in the transmit mode:

a) lldpXdot1PeConfigPortExtensionTxEnable

b) lldpXdot1PeLocPECascadePortPriority

The following readable objects in this MIB module may be considered to be sensitive or vulnerable in some network environments:

c) MIB objects that are related to transmit mode:

   1) lldpV2Xdot1LocPECascadePortPriority
   2) lldpV2Xdot1LocPEAddress
   3) lldpV2Xdot1LocPECSPAddress

d) MIB objects that are related to the receive mode:

   1) lldpV2Xdot1RemPECascadePortPriority
   2) lldpV2Xdot1RemPEAddress
   3) lldpV2Xdot1RemPECSPAddress

This concern applies both to objects that describe the configuration of the local host, as well as for objects that describe information from the remote hosts, acquired via LLDP and displayed by the objects in this MIB module. It is thus important to control even GET and/or NOTIFY access to these objects and possibly to even encrypt the values of these objects when sending them over the network via SNMP.

SNMP versions prior to SNMPv3 did not include adequate security. Even if the network itself is secure (for example by using IPSec), even then, there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in this MIB module.

Implementers should consider the security features as provided by the SNMPv3 framework (see RFC 3410, section 8 [B3]), including full support for the SNMPv3 cryptographic mechanisms (for authentication and privacy).

Further, implementers should not deploy SNMP versions prior to SNMPv3. Instead, implementers should deploy SNMPv3 to enable cryptographic security. It is then a customer/operator responsibility to ensure that the SNMP entity giving access to an instance of this MIB module is properly configured to give access to the objects only to those principals (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

[13] In IETF MIB definitions, the MAX-ACCESS clause defines the type of access that is allowed for particular data elements in the MIB. An explanation of the MAX-ACCESS mappings is given in section 7.3 of IETF RFC 2578.

B.5 IEEE 802.1 LLDP Port Extension MIB module[14,15]

```
LLDP-EXT-DOT1-PE-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY,
    OBJECT-TYPE,
    Unsigned32
        FROM SNMPv2-SMI
    TruthValue,
    MacAddress
        FROM SNMPv2-TC
    MODULE-COMPLIANCE,
    OBJECT-GROUP
        FROM SNMPv2-CONF
    ifGeneralInformationGroup
        FROM IF-MIB
    lldpXdot1StandAloneExtensions
        FROM LLDP-EXT-DOT1-EVB-EXTENSIONS-MIB
    lldpV2Extensions,
    lldpV2LocPortIfIndex,
    lldpV2RemTimeMark,
    lldpV2RemLocalIfIndex,
    lldpV2RemLocalDestMACAddress,
    lldpV2RemIndex,
    lldpV2PortConfigEntry
        FROM LLDP-V2-MIB;

lldpXDot1PEExtensions MODULE-IDENTITY
    LAST-UPDATED “201201230000Z” -- January 23, 2012
    ORGANIZATION “IEEE 802.1 Working Group”
    CONTACT-INFO
           “WG-URL: http://www.ieee802.org/1/
            WG-EMail: stds-802-1-L@IEEE.ORG

          Contact: Tony Jeffree
           Postal: C/O IEEE 802.1 Working Group
                   IEEE Standards Association
                   445 Hoes Lane
                   Piscataway
                   NJ 08854
                   USA
           E-mail: stds-802-1-L@IEEE.ORG”
    DESCRIPTION
            “The LLDP Management Information Base extension module for
            IEEE 802.1 organizationally defined discovery information
            to support Port Extension.

            This MIB module is rooted under the
            lldpXdot1StandAloneExtensions OID arc, in order to allow
            it to be defined independently of other 802.1 LLDP
            extension MIBs.

            Unless otherwise indicated, the references in this
```

[14] *Copyright release for MIBs:* Users of this standard may freely reproduce the MIB contained in this subclause so that it can be used for its intended purpose.

[15] An ASCII version of this MIB module can be obtained from the IEEE 802.1 Website at http://www.ieee802.org/1/pages/MIBS.html.

```
            MIB module are to IEEE Std 802.1BR-2012.

            Copyright (C) IEEE.  This version of this MIB module
            is published as Annex B.5 of IEEE Std 802.1BR-2012;
            see the standard itself for full legal notices.”

    REVISION “201201230000Z” -- January 23, 2012
    DESCRIPTION
            “Initial version published as part of IEEE Std. 802.1BR-2012”

    ::= { lldpXdot1StandAloneExtensions 2 }
-----------------------------------------------------------------------
-----------------------------------------------------------------------
--
-- Organizationally Defined Information Extension - IEEE 802.1
-- Definitions to support Port Extension
-- peSet TLV set (IEEE Std 802.1Q Table D-1)
--
-----------------------------------------------------------------------
-----------------------------------------------------------------------

lldpXdot1PeMIB OBJECT IDENTIFIER
    ::= { lldpXDot1PEExtensions 1 }
lldpXdot1PeObjects OBJECT IDENTIFIER ::= { lldpXdot1PeMIB 1 }

-- Port Extension 802.1 MIB Extension groups

lldpXdot1PeConfig     OBJECT IDENTIFIER ::= { lldpXdot1PeObjects 1 }
lldpXdot1PeLocalData  OBJECT IDENTIFIER ::= { lldpXdot1PeObjects 2 }
lldpXdot1PeRemoteData OBJECT IDENTIFIER ::= { lldpXdot1PeObjects 3 }

-----------------------------------------------------------------------
-- IEEE 802.1 - Configuration for the peSet TLV set
-----------------------------------------------------------------------

--
-- lldpV2Xdot1PeConfigPortExtensionTable : configure the transmission
-- of the Port Extension TLVs on a set of ports.
--
lldpXdot1PeConfigPortExtensionTable OBJECT-TYPE
     SYNTAX SEQUENCE OF LldpXdot1PeConfigPortExtensionEntry
     MAX-ACCESS not-accessible
     STATUS current
     DESCRIPTION
          “A table that controls selection of LLDP Port Extension
          TLVs to be transmitted on individual ports.”
::= { lldpXdot1PeConfig 1 }

lldpXdot1PeConfigPortExtensionEntry OBJECT-TYPE
     SYNTAX LldpXdot1PeConfigPortExtensionEntry
     MAX-ACCESS not-accessible
     STATUS current
     DESCRIPTION
          “LLDP configuration information that specifies Port
          Exension configuration.
          This configuration object augments the
          lldpV2Xdot1LocPortExtensionEntry, therefore it is
          only present along with the associated
          lldpV2Xdot1LocPortExtensionEntry entry.
          Each active lldpV2Xdot1ConfigPortExensionEntry must be
          restored from non-volatile storage (along with the
          corresponding lldpV2Xdot1LocPortExtensionEntry) after a
          re-initialization of the management system.”
     AUGMENTS      { lldpV2PortConfigEntry }
::= { lldpXdot1PeConfigPortExtensionTable 1 }

LldpXdot1PeConfigPortExtensionEntry ::= SEQUENCE {
     lldpXdot1PeConfigPortExtensionTxEnable TruthValue
     }

lldpXdot1PeConfigPortExtensionTxEnable OBJECT-TYPE
     SYNTAX TruthValue
     MAX-ACCESS read-write
     STATUS current
     DESCRIPTION
          “The lldpXdot1PeConfigPortExtensionTxEnable, which is
          defined as a truth value and configured by the network
          management, determines whether the IEEE 802.1
          organizationally defined Port Extension TLV transmission
          is allowed on a given LLDP transmission capable port.
          The value of this object must be restored from
          non-volatile storage after a re-initialization of the
          management system.”
     REFERENCE
          “D.8 of 802.1Q”
     DEFVAL { true }
::= { lldpXdot1PeConfigPortExtensionEntry 1 }

------------------------------------------------------------------------
-- IEEE 802.1 - Port Extension Local System Information
------------------------------------------------------------------------
---
---
--- lldpXdot1PeLocPortExtensionTable: Port Extension Information Table
---
---
lldpXdot1PeLocPortExtensionTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF LldpXdot1PeLocPortExtensionEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            “This table contains one row per port of Port Extension
            information (as a part of the LLDP 802.1 organizational
            extension) on the local system known to this agent.”
    ::= { lldpXdot1PeLocalData 1 }

lldpXdot1PeLocPortExtensionEntry OBJECT-TYPE
    SYNTAX      LldpXdot1PeLocPortExtensionEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            “Port Extension information about a particular
            Port Extender Port.”
    INDEX   { lldpV2LocPortIfIndex }
    ::= { lldpXdot1PeLocPortExtensionTable 1 }

LldpXdot1PeLocPortExtensionEntry ::= SEQUENCE {
      lldpXdot1PeLocPECascadePortPriority  Unsigned32,
      lldpXdot1PeLocPEAddress              MacAddress,
      lldpXdot1PeLocPECSPAddress           MacAddress
}

lldpXdot1PeLocPECascadePortPriority OBJECT-TYPE
    SYNTAX      Unsigned32 (0..255)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
            "Contains the cascade port priority."
    REFERENCE
            "D.8"
    ::= { lldpXdot1PeLocPortExtensionEntry 1 }

lldpXdot1PeLocPEAddress OBJECT-TYPE
    SYNTAX      MacAddress
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "This object contains the MAC address that
            uniquely identifies the Port Extender."
    REFERENCE
            "D.8"
    ::= { lldpXdot1PeLocPortExtensionEntry 2 }

lldpXdot1PeLocPECSPAddress OBJECT-TYPE
    SYNTAX      MacAddress
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "This object contains the MAC address to be used
            for the Port Extension control and status protocol."
    REFERENCE
            "D.8"
    ::= { lldpXdot1PeLocPortExtensionEntry 3 }

-------------------------------------------------------------------------
-- IEEE 802.1 - Port Extension Remote System Information
-------------------------------------------------------------------------
---
---
--- lldpXdot1PeRemPortExtensionTable: Port Extension Information Table
---
---
lldpXdot1PeRemPortExtensionTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF LldpXdot1PeRemPortExtensionEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "This table contains Port Extension information
            (as a part of the LLDP IEEE 802.1 organizational extension)
            of the remote system."
    ::= { lldpXdot1PeRemoteData 1 }

lldpXdot1PeRemPortExtensionEntry OBJECT-TYPE
    SYNTAX      LldpXdot1PeRemPortExtensionEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
```

```
            “Port Extension information about remote systems port
            component.”
    INDEX   { lldpV2RemTimeMark,
              lldpV2RemLocalIfIndex,
              lldpV2RemLocalDestMACAddress,
              lldpV2RemIndex }
    ::= { lldpXdot1PeRemPortExtensionTable 1 }

LldpXdot1PeRemPortExtensionEntry ::= SEQUENCE {
            lldpXdot1PeRemPECascadePortPriority Unsigned32,
            lldpXdot1PeRemPEAddress             MacAddress,
            lldpXdot1PeRemPECSPAddress          MacAddress
}

lldpXdot1PeRemPECascadePortPriority OBJECT-TYPE
    SYNTAX      Unsigned32 (0..255)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            “The cascade port priority.”
    REFERENCE
            “D.8”
    ::= { lldpXdot1PeRemPortExtensionEntry 1 }

lldpXdot1PeRemPEAddress OBJECT-TYPE
    SYNTAX      MacAddress
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            “This object contains the MAC address that
            uniquely identifies the Port Extender.”
    REFERENCE
            “D.8”
    ::= { lldpXdot1PeRemPortExtensionEntry 2 }

lldpXdot1PeRemPECSPAddress OBJECT-TYPE
    SYNTAX      MacAddress
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            “This object contains the MAC address to be used
            for the Port Extension Control and Status Protocol.”
    REFERENCE
            “D.8”
    ::= { lldpXdot1PeRemPortExtensionEntry 3 }

------------------------------------------------------------------------
-- IEEE 802.1 - Port Extension Conformance Information
------------------------------------------------------------------------

lldpXdot1PeConformance OBJECT IDENTIFIER ::= { lldpXDot1PEExtensions 2 }

lldpXdot1PeCompliances
    OBJECT IDENTIFIER ::= { lldpXdot1PeConformance 1 }
lldpXdot1PeGroups OBJECT IDENTIFIER ::= { lldpXdot1PeConformance 2 }

--
-- Port Extension - Compliance Statements
--
```

```
lldpXdot1PeCompliance MODULE-COMPLIANCE
    STATUS        current
    DESCRIPTION
         “A compliance statement for entities that implement
         the IEEE 802.1 organizationally defined Port Extension
         LLDP extension MIB.

         This group is mandatory for agents that implement the
         Port Extension peSet TLV set.”
    MODULE        -- this module
        MANDATORY-GROUPS  { lldpXdot1PeGroup,
                            ifGeneralInformationGroup }
    ::= { lldpXdot1PeCompliances 1 }

--
-- Port Extension - MIB groupings
--

lldpXdot1PeGroup  OBJECT-GROUP
    OBJECTS {
        lldpXdot1PeConfigPortExtensionTxEnable,
        lldpXdot1PeLocPECascadePortPriority,
        lldpXdot1PeLocPEAddress,
        lldpXdot1PeLocPECSPAddress,
        lldpXdot1PeRemPECascadePortPriority,
        lldpXdot1PeRemPEAddress,
        lldpXdot1PeRemPECSPAddress
    }
    STATUS  current
    DESCRIPTION
        “The collection of objects that support the
        Port Extension peSet TLV set.”
    ::= { lldpXdot1PeGroups 1 }

END
```

Annex C

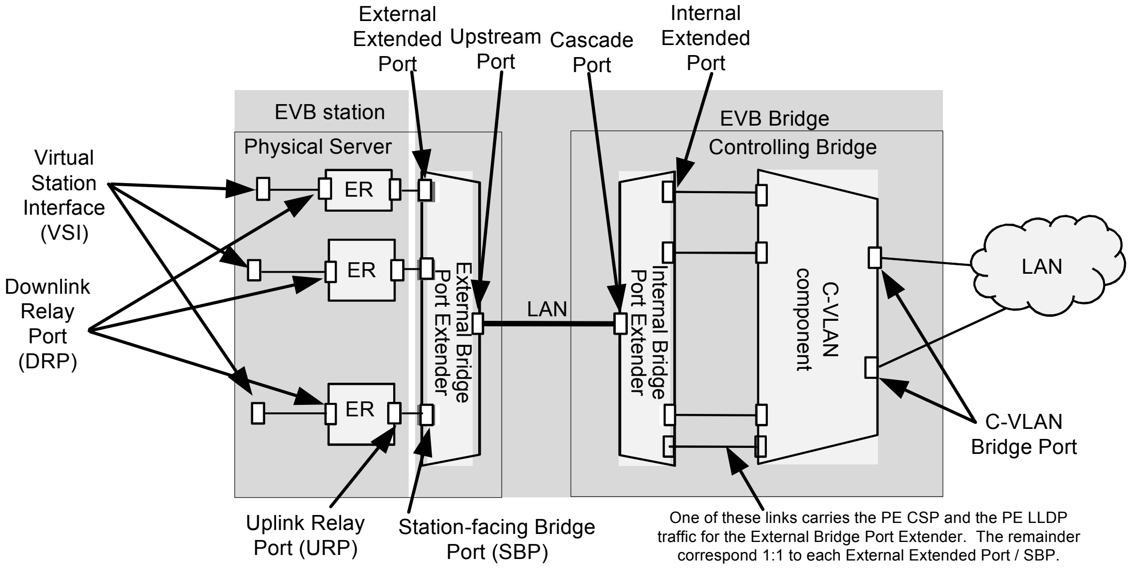
(informative)

Utilizing VDP with Port Extension

C.1 VDP and Port Extension

Clause 41 of IEEE Std 802.1Q defines the Virtual Station Interface (VSI) discovery and configuration protocol (VDP). VDP associates (registers) a VSI instance with an Station-facing Bridge Port (SBP) of an Edge Virtual Bridging (EVB) Bridge (see Clause 40 of IEEE Std 802.1Q). VDP simplifies and automates virtual station configuration and enables the movement of a VSI instance (and its related VSI Type information) from one virtual station to another or from one EVB Bridge to another. VDP supports VSI discovery and configuration across a channel interconnecting an EVB station and an EVB Bridge. VDP TLVs are exchanged between the station and the Bridge in support of this protocol.

An Extended Bridge can operate as an EVB Bridge. In this case, the external Extended Ports become SBPs of the EVB Bridge. On the server side, a two-Port edge relay may be inserted to provide support for VDP. An example of the relationship of the Port Extension and EVB components is illustrated in Figure C.1.



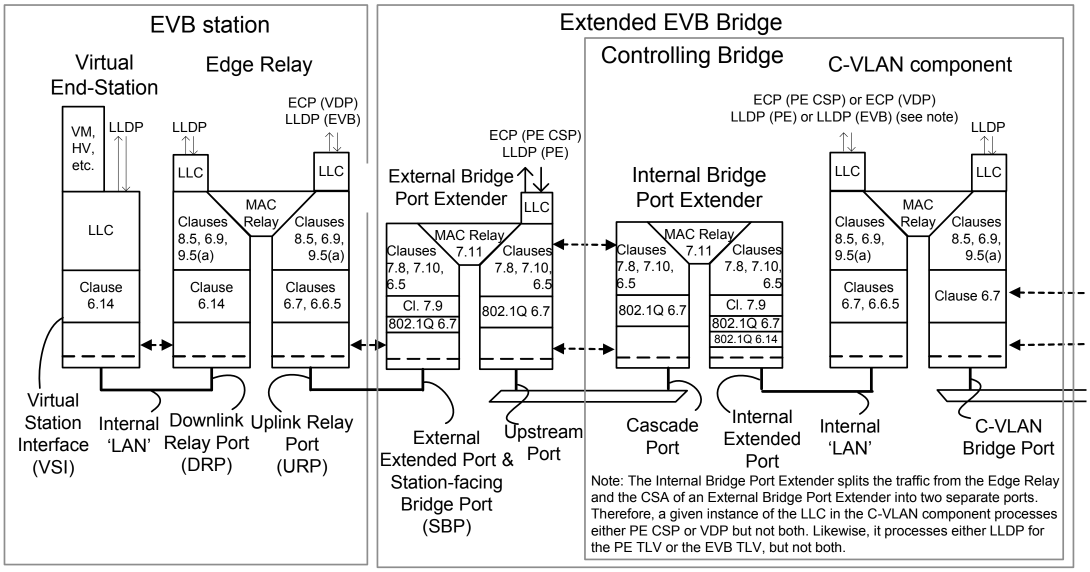
One of these links carries the PE CSP and the PE LLDP traffic for the External Bridge Port Extender. The remainder correspond 1:1 to each External Extended Port / SBP.

Figure C.1—Relationship of Port Extension and EVB

In this example, the Edge Relay components are present to inject the VDP protocol and the EVB TLV. In addition, these components terminate LLDP (and filter the reserved group addresses specified in Table 8-1 of IEEE Std 802.1Q). The Edge Relay components can be configured to be otherwise transparent. Also note that in this example, the External Bridge Port Extender is integrated within the physical server. As a result, the Extended Bridge, and therefore the EVB Bridge, extends across the physical device boundaries into the physical server. The remainder of the physical server provides the EVB station functionality.

Figure C.2 illustrates the relationship between the Port Extension and EVB entities to the Bridge architecture. As illustrated, the ECP carrying VDP and the LLDP carrying the EVB TLV pass transparently through the Bridge Port Extenders to one port on the C-VLAN component for each virtual end-station. An additional port on the C-VLAN component participates in ECP carrying PE CSP and LLDP carrying the PE TLV for each Bridge Port Extender.

.



Note: The Internal Bridge Port Extender splits the traffic from the Edge Relay and the CSA of an External Bridge Port Extender into two separate ports. Therefore, a given instance of the LLC in the C-VLAN component processes either PE CSP or VDP but not both. Likewise, it processes either LLDP for the PE TLV or the EVB TLV, but not both.

References within the Virtual End Station, Edge Relay, and C-VLAN component refer to IEEE Std 802.1Q.

Figure C.2—Port Extension and EVB combined architecture

This example describes one deployment that is expected to be common; however, it does not imply that this is the only valid deployment of these technologies.

Annex D

(informative)

Extended Bridge Initialization

D.1 Introduction

This specification provides the protocol necessary to form an Extended Bridge as Bridge Port Extenders are attached to a Controlling Bridge. However, the precise sequence of messages to achieve this is not specified to facilitate flexibility in implementation. This annex provides an example of how an Extended Bridge might be initialized for a specific topology and sequence of attachment.

D.2 Attachment of a physical Bridge Port Extender

Figure D.1 provides an example of the attachment of an External Bridge Port Extender.

Figure D.1(a) illustrates a simple two-port Bridge that is capable of acting as a Controlling Bridge.

Figure D.1(b) illustrates the attachment of a physical Bridge Port Extender to the top port of the two-port Bridge. At this point, the Bridge and the Bridge Port Extender execute LLDP. The Bridge learns that a Bridge Port Extender is directly attached when it receives the Port Extension TLV from the Bridge Port Extender.

Upon detection of the directly attached Bridge Port Extender, the Controlling Bridge instantiates an Internal Bridge Port Extender between the C-VLAN component and the External Bridge Port Extender. An E-channel is established for communication between the Bridge Port Extender and the C-VLAN component. This is illustrated in Figure D.1(c). The E-channel used for communication between the C-VLAN component and the Bridge Port Extender is identified as E-channel “a” in this example.

Next both the C-VLAN component and the Bridge Port Extender initiate communication with each other using the Bridge Port Extender Control and Status Protocol (PE CSP). This is accomplished using the CSP Open message. Since the E-channel is not tagged, the communication is established without the Bridge Port Extender needing knowledge of the E-CID of the E-channel. After completion of the CSP Open, the Controlling Bridge informs the Bridge Port Extender of the proper E-CID, which is “a” in this example, using the E-channel Register message.

Following the establishment of communication, the Controlling Bridge may issue a Port Parameters Set message to change the configuration of the Bridge Port Extender’s Upstream Port from the default.

Note that the Bridge Port Extender in Figure D.1(c) has two Extended Ports. The Bridge Port Extender issues an Extended Port Create message for each of these Ports. The Controlling Bridge establishes a Port on the C-VLAN component and an E-channel for each of these Extended Ports and informs the Bridge Port Extender of the E-CIDs for these E-channels in the response to the Extended Port Create message. The Bridge Port Extender configures the member set of these E-channels to include the corresponding Extended Port. This is illustrated in Figure D.1(d). In this example, E-CIDs “b” and “c” are allocated for the Extended Ports. Also note that these E-channels are not tagged on the Extended Ports.

.

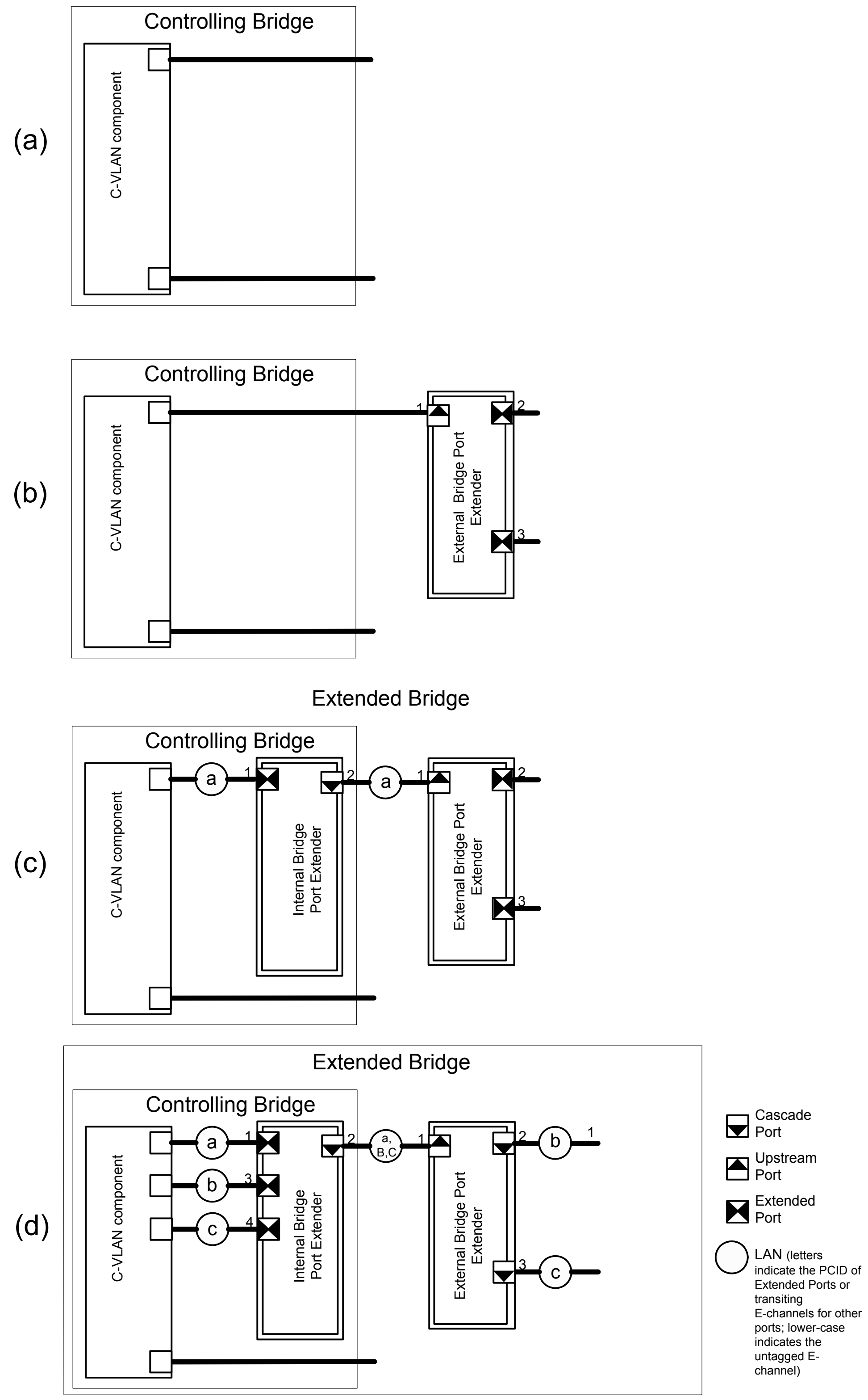

(a)
Controlling Bridge
C-VLAN component
(b)
Controlling Bridge
C-VLAN component
External Bridge Port Extender
1
2
3
Extended Bridge
(c)
Controlling Bridge
C-VLAN component
a
1
Internal Bridge Port Extender
2
a
1
External Bridge Port Extender
2
3
Extended Bridge
(d)
Controlling Bridge
C-VLAN component
a
b
c
1
3
4
Internal Bridge Port Extender
2
a, B,C
1
External Bridge Port Extender
2
b
1
3
c
Cascade Port
Upstream Port
Extended Port
LAN (letters indicate the PCID of Extended Ports or transiting E-channels for other ports; lower-case indicates the untagged E-channel)

Figure D.1—Attachment of a physical Bridge Port Extender

D.3 Attachment of a downstream Bridge Port Extender

Figure D.2 illustrates the attachment of a downstream Bridge Port Extender to the Extended Bridge illustrated in Figure D.1(d). In this example, the downstream Bridge Port Extender happens to be part of a virtualized server in which two virtual machines are to be instantiated.

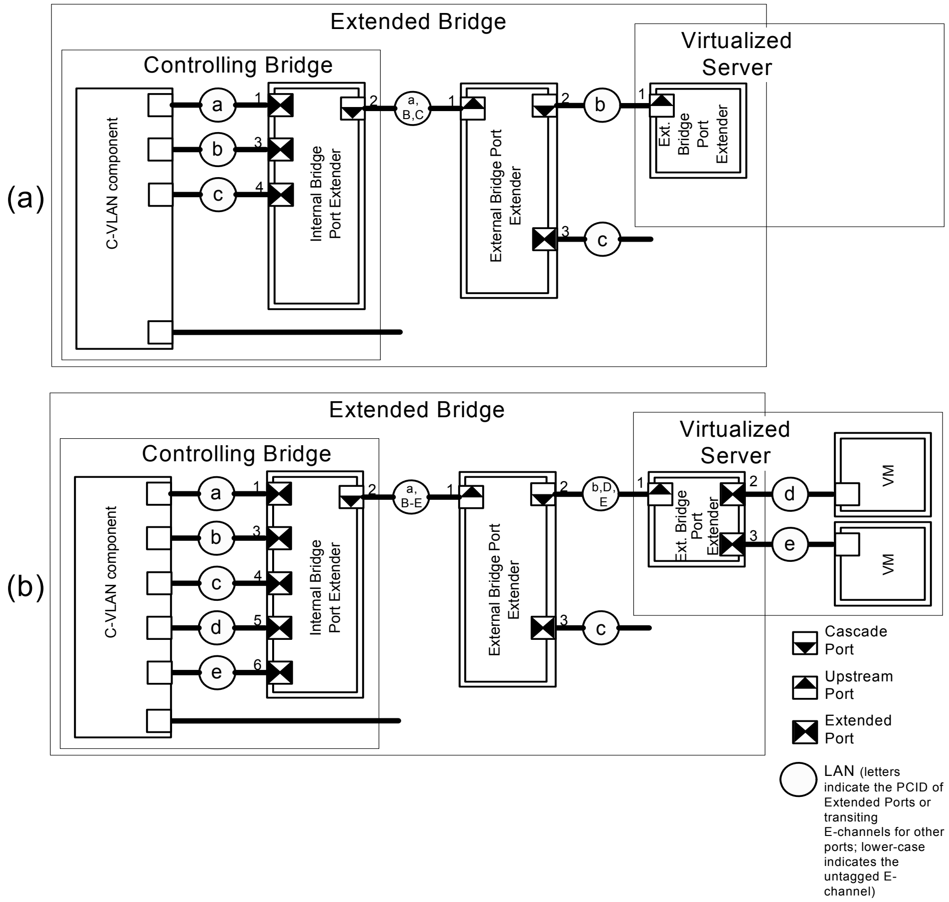

(a)
Extended Bridge
Controlling Bridge
Virtualized Server
C-VLAN component
Internal Bridge Port Extender
External Bridge Port Extender
Ext. Bridge Port Extender
a
b
c
a, B,C
1
2
3
4
(b)
VM
d
e
b,D, E
a, B-E
5
6
Cascade Port
Upstream Port
Extended Port
LAN (letters indicate the PCID of Extended Ports or transiting E-channels for other ports; lower-case indicates the untagged E-channel)

Figure D.2—Attachment of a downstream Bridge Port Extender

The Controlling Bridge detects the attachment of the new Bridge Port Extender by receiving the Bridge Port Extender LLDP TLV on E-channel “b.” At this point, the extended Port on the first Bridge Port Extender, by definition, becomes a Cascade Port. E-channel “b” is used for communication between the C-VLAN component and the new Bridge Port Extender. The new Bridge Port Extender and the C-VLAN component establish communication by issuing a CSP Open message. After completion of the CSP Open, the Controlling Bridge informs the new Bridge Port Extender of the proper E-CID, which is “b” in this example, using the E-channel Register message. Following the establishment of communication, the Controlling Bridge may issue a Port Parameters Set message to change the configuration of the Bridge Port Extender’s Upstream Port form the default.

Note that in Figure D.2(a), the Extended Ports have not been instantiated. Extended Ports are not necessarily instantiated at the same time the Bridge Port Extender itself is instantiated. For example, the Extended Ports may be instantiated coincident with the instantiation of virtual machines.

Figure D.2(b) illustrates the instantiation of the virtual machines and the corresponding Extended Ports. When the Extended Ports are instantiated, the new Bridge Port Extender informs the controlling bridge by issuing an Extended Port create message for each extended Port. The Controlling Bridge allocates a Port on the C-VLAN component and an E-channel for each new Extended Port, and informs the new Bridge Port Extender of the E-CID for these E-channels. E-CIDs “d” and “e” are established in this example. In addition, the Controlling Bridge issues E-channel Register messages to the first Bridge Port Extender to establish the new E-channels through the first Bridge Port Extender. At this point, the virtual machines have connectivity to the network.

D.3.1 Message Flow

Figure D.3 provides an example of a possible message flow that might take place to initialize the Extended Bridge illustrated in Figure D.2(b).

.

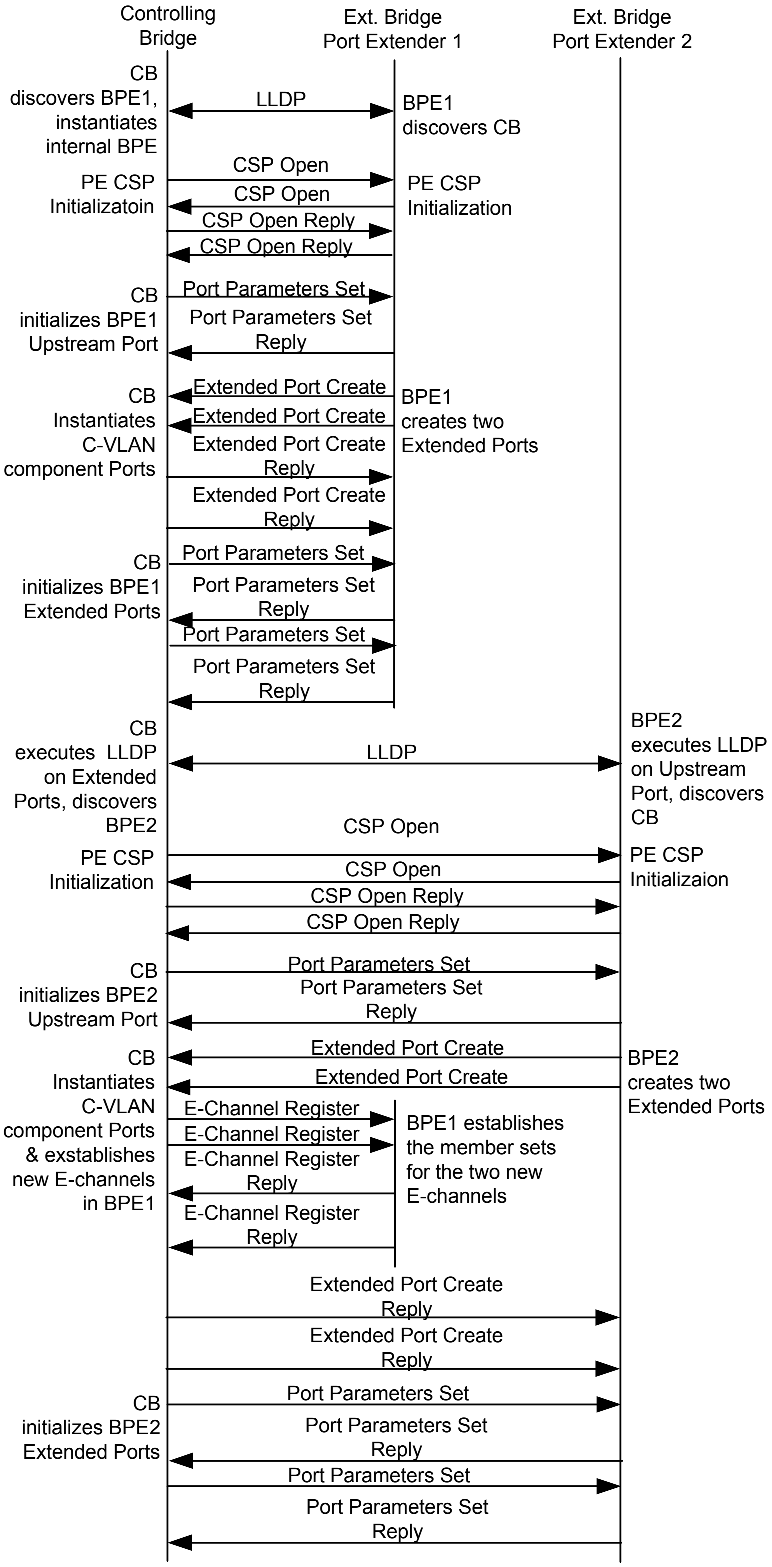

Controlling Bridge
Ext. Bridge Port Extender 1
Ext. Bridge Port Extender 2
CB discovers BPE1, instantiates internal BPE
LLDP
BPE1 discovers CB
PE CSP Initializatoin
CSP Open
CSP Open
CSP Open Reply
CSP Open Reply
PE CSP Initialization
CB initializes BPE1 Upstream Port
Port Parameters Set
Port Parameters Set Reply
CB Instantiates C-VLAN component Ports
Extended Port Create
Extended Port Create
Extended Port Create Reply
Extended Port Create Reply
BPE1 creates two Extended Ports
CB initializes BPE1 Extended Ports
Port Parameters Set
Port Parameters Set Reply
Port Parameters Set
Port Parameters Set Reply
CB executes LLDP on Extended Ports, discovers BPE2
LLDP
BPE2 executes LLDP on Upstream Port, discovers CB
PE CSP Initialization
CSP Open
CSP Open
CSP Open Reply
CSP Open Reply
PE CSP Initializaion
CB initializes BPE2 Upstream Port
Port Parameters Set
Port Parameters Set Reply
CB Instantiates C-VLAN component Ports & exstablishes new E-channels in BPE1
Extended Port Create
Extended Port Create
BPE2 creates two Extended Ports
E-Channel Register
E-Channel Register
E-Channel Register Reply
E-Channel Register Reply
BPE1 establishes the member sets for the two new E-channels
Extended Port Create Reply
Extended Port Create Reply
CB initializes BPE2 Extended Ports
Port Parameters Set
Port Parameters Set Reply
Port Parameters Set
Port Parameters Set Reply

Figure D.3—Example Initialization Message Flow

Annex E

(informative)

Bibliography

Bibliographical references are resources that provide additional or helpful material but do not need to be understood or used to implement this standard. Reference to these resources is made for informational use only.

[B1] IEEE Std 802.1D™, IEEE Standard for Local and Metropolitan Area Networks—Media Access Control (MAC) Bridges.[16, 17]

[B2] IEEE Std 802.3™, IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications.

[B3] IETF RFC 3410, Introduction and Applicability Statements for Internet-Standard Management Framework, Dec. 2002.[18]

[B4] IETF RFC 3416, STD 62, Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP), Dec. 2002.

[B5] ISO/IEC 15802-1, Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Common specifications—Part 1: Medium Access Control (MAC) service definition.[19]

[16] IEEE publications are available from The Institute of Electrical and Electronics Engineers (http://standards.ieee.org/).

[17] The IEEE standards or products referred to in Annex E are trademarks of The Institute of Electrical and Electronics Engineers, Inc.

[18] IETF documents (i.e., RFCs) are available for download at http://www.rfc-archive.org/.

[19] ISO/IEC publications are available from the ISO Central Secretariat (http://www.iso.org/). ISO publications are also available in the United States from the American National Standards Institute (http://www.ansi.org/).