IEEE STANDARDS ASSOCIATION

IEEE Standard for
Local and metropolitan area networks—

Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks—

Amendment 21: Edge Virtual Bridging

IEEE Computer Society

Sponsored by the
LAN/MAN Standards Committee

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

5 July 2012

IEEE Std 802.1Qbg™-2012
(Amendment to IEEE Std 802.1Q™-2011
as amended by IEEE Std 802.1Qbe™-2011,
IEEE Std 802.1Qbc™-2011, IEEE Std 802.1Qbb™-2011,
IEEE Std 802.1Qaz™-2011, IEEE Std 802.1Qbf™-2011,
and IEEE Std 802.aq™-2012)

IEEE Std 802.1Qbg™-2012
(Amendment to
IEEE Std 802.1Q™-2011
as amended by IEEE Std 802.1Qbe™-2011,
IEEE Std 802.1Qbc™-2011, IEEE Std 802.1Qbb™-2011,
IEEE Std 802.1Qaz™-2011, IEEE Std 802.1Qbf™-2011,
and IEEE Std 802.1aq™-2012)

IEEE Standard for
Local and metropolitan area networks—

Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks—

Amendment 21: Edge Virtual Bridging

Sponsor

LAN/MAN Standards Committee
of the
IEEE Computer Society

Approved 14 May 2012

IEEE-SA Standards Board

Abstract: This amendment to IEEE Std 802.1Q-2011 specifies protocols, procedures, and managed objects that

— Provide for the discovery, configuration, and control of a pair of direct-attached limited-function Service VLAN (S-VLAN) components to extend the services of a customer bridge to remote ports and enable coexistence of multiple services onstation-resident ports (e.g., embedded bridging).

— Provide for discovery, configuration, and control of a Reflective Relay Service for a bridge port.

— Provide for discovery of, and coordinated configuration of, edge relays (ERs) and other devices that utilize the reflective relay service.

— Provide for dynamic profile-driven port configuration.

Keywords: Bridged Local Area Networks, edge virtual bridging, IEEE 802.1Qbg, LANs, local area networks, MAC Bridges, metropolitan area networks, Reflective Relay Service, SPB, Virtual Bridged Local Area Networks, virtual LANs

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

PDF: **ISBN 978-0-7381-7273-6** **STD97248**
Print: **ISBN 978-0-7381-7379-5** **STDPD97248**

Introduction

This introduction is not part of IEEE Std 802.1Qbg-2012, IEEE Standard for Local and metropolitan area networks—Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks—Amendment 21: Edge Virtual Bridging.

This amendment to IEEE Std 802.1Q-2011 specifies protocols, procedures, and managed objects that

— Provide for the discovery, configuration, and control of a pair of direct-attached limited-function Service VLAN (S-VLAN) components to extend the services of a customer bridge to remote ports and enable coexistence of multiple services onstation-resident ports (e.g., embedded bridging).

— Provide for discovery, configuration, and control of a Reflective Relay Service for a bridge port.

— Provide for discovery of, and coordinated configuration of, edge relays (ERs) and other devices that utilize the reflective relay service.

— Provide for dynamic profile-driven port configuration.

The purpose of this standard is to allow multiple virtual stations to share a common bridge port to obtain the services of bridge relay. The standard enables coordinated configuration and management of bridge services for virtual stations.

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

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

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 or contact the IEEE at the address listed previously. For more information about the IEEE Standards Association or the IEEE standards development process, visit the IEEE-SA website.

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 http://standards.ieee.org/about/sasb/patcom/patents.html. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

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

Participants

At the time this amendment was submitted to the IEEE-SA Standards Board for approval, the IEEE 802.1 Working Group had the following membership:

Tony Jeffree, *Chair*
Paul Congdon, *Vice Chair*
Patricia Thaler, *Chair, Data Center Bridging Task Group*
Paul Bottorff, *Editor, Clauses 12 and 17*

Zehavit Alon
Yafan An
Ting Ao
Peter Ashwood-Smith
Christian Boiger
Rudolf Brandner
Craig Carlson
Xin Chang
Weiying Cheng
Rodney Cummings
Claudio DeSanti
Zhemin Ding
Donald Eastlake
Janos Farkas
Donald Fedyk
Norman Finn
Geoffrey Garner
Anoop Ghanwani
Franz Goetz
Mark Gravel
Eric Gray
Yingjie Gu
Craig Gunther
Stephen Haddock
Hitoshi Hayakawa
Daya Kamath
Hal Keen
Girault Jones
Srikanth Keesara
Yongbum Kim
Philippe Klein
Oliver Kleineberg
Michael Krause
Lin Li
Jeff Lynch
Thomas Mack-Crane
David Martin
John Messenger
John Morris
Eric Multanen
Yukihiro Nakagawa
David Olsen
Donald Pannell
Glenn Parsons
Mark Pearson
Joseph Pelissier
Rene Raeber
Karen T. 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
Chait Tumuluri
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
Richard Alfvin
Mark Anderson
Hugh Barrass
Paul Bottorff
Nancy Bravin
William Byrd
Craig Carlson
Keith Chow
Charles Cook
Rodney Cummings
Claudio DeSanti
Patrick Diamond
Thomas Dineen
Sourav Dutta
Richard Edgar
John Egan
Yukihiro Fujimoto
Ilango Ganga
Anoop Ghanwani
Randall Groves
Stephen Haddock
David Hunter
Noriyuki Ikeuchi
Atsushi Ito
Raj Jain
Tony Jeffree
Peter Jones
Shinkyo Kaku
Piotr Karocki
Stuart Kerry
Yukihiro Kikuchi
Yongbum Kim
Paul Lambert
Brian L'Ecuyer
Greg Luri
Michael Lynch
Thomas Mack-Crane
Elvis Maculuba
David Martin
John Messenger
Gary Michel
Jose Morales
Shimon Muller
Yukihiro Nakagawa
Robert Needy
Michael S. Newman
Nick S.A. Nikjoo
Satoshi Obara
Glenn Parsons
Maximilian Riegel
Jessy Rouyer
Herbert Ruck
Panagiotis Saltsidis
Bartien Sayogo
Gil Shultz
Kapil Sood
Matthew Squire
Manikantan Srinivasan
Thomas Starai
Walter Struppler
Joseph Tardo
William Taylor
Michael Johas Teener
Patricia Thaler
James Tomcik
Dmitri Varsanofiev
Prabodh Varshney
John Vergis
Karl Weber
Yuehua Wei
Ludwig Winkel
Oren Yuen

When the IEEE-SA Standards Board approved this amendment 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 Kuyn 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*

Catherine Berger
IEEE Standards Program Manager, Document Development

Kathryn Bennett
IEEE Standards Program Manager, Technical Program Development

Contents

Figures

Tables

IEEE Standard for
Local and metropolitan area networks—

Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks—

Amendment 21: Edge Virtual Bridging

[This amendment is based on IEEE Std 802.1Q™-2011, as modified by those amendments that had been approved, but not incorporated into the base text of the standard, at the time that this amendment was approved, namely (in chronological order) IEEE Std 802.1Qbe, IEEE Std 802.1Qbc, IEEE Std 802.1Qbb, IEEE Std 802.1Qaz, IEEE Std 802.1Qbf, and IEEE Std 802.1aq.]

NOTE—The editing instructions contained in this amendment define how to merge the material contained here into the base document and its other amendments to form the new comprehensive standard.

Editing instructions are shown in ***bold italic***. Four editing instructions are used: change, delete, insert, and replace. ***Change*** is used to make corrections in existing text or tables. The editing instruction specifies the location of the change and describes what is being changed either by using ~~strikethrough~~ (to remove old material) and underscore (to add new material). ***Delete*** removes existing material. ***Insert*** adds new material without disturbing the existing material. Insertions may require renumbering. If so, renumbering instructions are given in the editing instruction. ***Replace*** is used to make changes in figures or equations by removing the existing figure or equation and replacing it with a new one. Editing instructions, change markings, and this NOTE will not be carried over into future editions because the changes will be incorporated into the base standard.[1]

[1]Notes in text, tables, and figures are given for information only, and do not contain requirements needed to implement the standard.

1. Overview

1.3 Introduction

Insert the following text at the end of 1.3, renumbering the list items as necessary:

This standard specifies protocols, procedures, and managed objects that

a) Provide for the discovery, configuration, and control of a pair of direct-attached Port-mapping Service VLAN (S-VLAN) components to extend the operation of a customer bridge to remote ports and enable coexistence of multiple services on station-resident ports (e.g., embedded bridging).
b) Provide for discovery, configuration, and operation of reflective relay (8.6.1) for a bridge port.
c) Provide for discovery of, and coordinated configuration of, edge relays (ERs) and other devices that utilize the reflective relay service.
d) Provide for dynamic profile-driven port configuration.

2. Normative references

Insert the following references into Clause 2, in appropriate collating sequence:

IEEE Std 802.1BR™, IEEE Standard for Local and Metropolitan Area Networks—Virtual Bridged Local Area Networks—Bridge Port Extension.[2]

IETF RFC 4122, A Universally Unique IDentifier (UUID) URN Namespace.

IETF RFC 4291, IP Version 6 Addressing Architecture.

[2]IEEE publications are available from the Institute of Electrical and Electronics Engineers, Inc., 445 Hoes Lane, Piscataway, NJ 08854, USA (http://standards.ieee.org/).

3. Definitions

Insert the following definitions into Clause 3, in appropriate collating sequence, renumbering existing definitions as needed:

3.1 Downlink relay port (DRP): A port of an edge relay that is capable of supporting at least one VSI.

3.2 Edge Control Protocol (ECP): A protocol that provides reliable delivery of control SDUs.

3.3 edge relay (ER): A bridge supporting the transfer of frames between one or more downlink relay ports (DRPs) and one uplink relay port (URP).

3.4 Edge Virtual Bridging (EVB): The set of functions supporting VSIs in Bridges and attached end stations.

3.5 Edge Virtual Bridging Bridge (EVB Bridge): A C-VLAN Bridge that supports the Virtual Station Interface (VSI) discovery and configuration protocol (VDP).

3.6 Edge Virtual Bridging station (EVB station): An end station containing one or more edge relays.

3.7 GroupID: A service instance identifier used in VDP.

3.8 Reflective relay: A mode of operation of the active topology enforcement function in which a received frame on a port that supports reflective operation can be forwarded on the same port on which it was received.

3.9 S-channel: A point-to-point S-VLAN established between a Port-mapping S-VLAN component in an EVB Bridge and a Port-mapping S-VLAN component in an EVB station.

3.10 S-channel Access Port (CAP): The Port that terminates an S-channel.

3.11 S-channel Discovery and Configuration Protocol (CDCP): A protocol that is used to configure S-VLAN components to create S-channels.

3.12 Station-facing Bridge Port (SBP): A Port of a Bridge that supports the EVB status parameters (6.6.6) with an EVBMode parameter value of “EVB Bridge”.

3.13 Uplink Access Port (UAP): A Port on a Port-mapping S-VLAN component that connects an EVB Bridge with an EVB station.

3.14 Uplink relay port (URP): A port of an edge relay that supports the EVB status parameters (6.6.6) with an EVBMode parameter value of “EVB station”.

3.15 Virtual edge bridge (VEB): An edge relay that requires reflective relay service to be disabled on the station-facing Bridge Port (SBP) of the attached Bridge.

3.16 Virtual edge port aggregator (VEPA): An edge relay that always forwards frames through its uplink relay port (URP) and that can make use of reflective relay service provided by the station-facing Bridge Port (SBP) of the attached Bridge.

3.17 Virtual station: An end station instantiated within an EVB station.

3.18 Virtual Station Interface (VSI): An interface to a to a virtual station that is attached to a DRP of an edge relay.

3.19 Virtual Station Interface (VSI) Discovery and Configuration Protocol (VDP): A protocol that supports the association of a VSI with a bridge port.

4. Abbreviations

Insert the following abbreviations into Clause 4, in appropriate collating sequence:

ACK	acknowledgement
CAP	S-channel Access Port
CDCP	S-channel discovery and configuration protocol
DCN	data center network
DRP	downlink relay port
ECP	edge control protocol
ECPDU	edge control protocol data unit
ER	edge relay
EVB	Edge Virtual Bridging
IP	Internet protocol
IPv6	Internet protocol version 6
LLDP	link layer discovery protocol
OUI	organizationally unique identifier
SBP	Station-facing Bridge Port
SCID	S-channel identifier
SDU	service data unit
TLV	type, length, value
UAP	Uplink Access Port
ULP	upper layer protocol
ULPDU	upper layer PDU
URP	uplink relay port
UUID	Universally Unique Identifier
VDP	VSI discovery and configuration protocol
VEB	virtual edge bridge
VEPA	virtual edge port aggregator
VSIID	VSI Instance Identifier
VSI	Virtual Station Interface
VTID	VSI Type identifier

5. Conformance

Change subclause 5.2 as follows:

5.2 Conformant components and equipment

This subclause specifies requirements and options for the following core components:

a) VLAN-aware Bridge component (5.4);
b) VLAN-unaware Bridge component (5.12);

for the following components that use that core functionality:

c) C-VLAN component (5.5);
d) S-VLAN component (5.6);
e) I-component (5.7);
f) B-component (5.8);
g) TPMR component (5.13);
h) T-component (5.15);
i) Edge relay (5.23.1);

and for the following systems that include instances of the above components:

j) VLAN Bridge (5.9);
k) S-VLAN Bridge (5.10.1);
l) Provider Edge Bridge (5.10.2);
m) Backbone Edge Bridge (5.11);
n) TPMR (5.14);
o) Edge Virtual Bridging Bridge (5.22);
p) Edge Virtual Bridging station (5.23).

NOTE—A VLAN Bridge can also be referred to as a Customer Bridge or a C-VLAN Bridge. Both S-VLAN Bridges and Provider Edge Bridges are examples of Provider Bridges.

Insert new subclauses 5.22 and 5.23, renumbering existing subclauses as necessary, as follows:

5.22 Edge Virtual Bridging (EVB) Bridge requirements

An EVB Bridge shall comprise a single conformant C-VLAN component (5.5) and zero or one Port-mapping S-VLAN component (5.6) per externally accessible port.

Each externally accessible port shall be capable of being configured as one of, and may be capable of being configured as any of the following:

a) A C-VLAN Bridge Port
b) A Station-facing Bridge Port (SBP)
c) An Uplink Access Port (UAP)

as specified in Clause 40.

A conformant EVB Bridge implementation shall

d) Support the functionality of a C-VLAN component (5.5).

e) Support at least one SBP on the C-VLAN component (Clause 40).
f) Support the EVB status parameters for EVBMode = EVB Bridge (6.6.6).
g) Support an LLDP nearest Customer Bridge database (Clause 40).
h) Support the EVB TLV on each SBP (D.2.13).
i) Support ECP on each SBP (Clause 43).
j) Support the Bridge role of VDP on each SBP (Clause 41).

A conformant EVB Bridge may support S-channels. A conformant EVB Bridge with S-channel support shall

k) Support at least one Port-mapping S-VLAN component (22.6.4) and associated UAP, configured as specified in 40.2 a)–d).
l) Support CDCP, as specified in Clause 42, operating in Bridge mode.
m) Support the enhanced filtering utility criteria (8.7.2) and not support the default filtering utility criteria (8.7.1).

A conformant EVB Bridge implementation may

n) Support configuration of reflective relay on each SBP of the C-VLAN component (8.6.1).
o) Support management for the EVB components (12.4–12.12, 12.26).
p) Support the MIB module defined in 17.7.20.
q) Support assignment of VIDs to GroupIDs (41.2.9).
r) Support the use of the M and S bits in VDP (41.2.3).

5.23 Edge Virtual Bridging (EVB) station requirements

An EVB station shall comprise one or more conformant ERs (5.23.1) and zero or one Port-mapping S-VLAN component (5.6) per externally accessible port.

Each externally accessible port shall be capable of being configured as one of, and may be capable of being configured as any of the following:

a) An Uplink Access Port (UAP)
b) An Uplink Relay Port (URP)

as specified in Clause 40.

Each DRP shall be capable of attaching to one or more VSIs.

Each URP shall be capable of attaching its ER to the LAN connecting to an EVB Bridge, or in the case where a Port-mapping S-VLAN component is present, to an internal LAN (6.14) connecting the URP to a CAP.

A conformant EVB station implementation shall

c) Support at least one ER (5.23.1, Clause 40).
d) Support the EVB status parameters for EVBMode = EVB station on each URP (6.6.6).
e) Support an LLDP nearest Customer Bridge database (Clause 40).
f) Support the EVB TLV on each URP of each ER (D.2.13).
g) Support ECP on each URP of each ER (Clause 43).
h) Support the station role of VDP for each URP of each ER (Clause 41).

In addition, a conformant EVB station implementation that supports a Port-mapping S-VLAN components shall

i) Support a Port-mapping S-VLAN component (22.6.4) on each port configured as a UAP (42.1.2) configured as specified in 40.2 (a)–(d).
j) Support CDCP, as specified in Clause 42, operating in Station mode.
k) Support the enhanced filtering utility criteria (8.7.2) and not support the default filtering utility criteria (8.7.1).

A conformant EVB station implementation may

l) Support multiple ERs (Clause 40).
m) Support management for the EVB components (12.4–12.12, 12.26).
n) Support the MIB module defined in 17.7.20.
o) Support assignment of VIDs to GroupIDs (41.2.9).
p) Support the use of the M and S bits in VDP (41.2.3).

5.23.1 Edge relay requirements

A conformant implementation of an ER shall

a) Conform to the relevant standard for the Media Access Control technology implemented at each Port in support of the MAC ISS, as specified in 6.6, 6.7, and 6.14.
b) Support the MAC Enhanced Internal Sublayer Service at each Port, as specified in 6.8 and 6.9.
c) Recognize and use C-TAGs (6.9).
d) Relay and filter frames as described in 8.1 and specified in 8.5, 8.6, and 8.8.
e) Support the following on each DRP Port that supports untagged and priority-tagged frames:
    1) A Port VLAN Identifier (PVID) value (6.9)
    2) Configuration of at least one VID whose untagged set includes that Port (8.8.2)
f) Support setting the Acceptable Frame Types parameter (6.9) to *Admit Only VLAN Tagged Frames* on the URP.
g) Allow tag headers to be inserted, modified, and removed from relayed frames, as specified in 8.1 and Clause 9, as required by the value(s) of the Acceptable Frame Types parameter supported on each Port, and by the ability of each Port to transmit VLAN-tagged and/or untagged frames.
h) Support at least one Filtering Identifier (FID) (6.6, 8.8.3, 8.8.8, and 8.8.9).
i) Allow allocation of at least one VID to each FID that is supported (6.6, 8.8.3, 8.8.8, and 8.8.9).
j) Support exactly one URP (Clause 40) supporting the parameters of 6.6.6 for EVBMode = EVB station.
k) Support one or more DRPs each supporting access to VSIs (Clause 40).
l) Filter the Reserved MAC Addresses specified in Table 8-1.
m) If more than one DRP is supported, support setting the Enable Ingress Filtering parameter (8.6.2) on each DRP and the URP.
n) Support the requirements of either a VEB ER (5.23.1.1), or a VEPA ER (5.23.1.2).

A conformant implementation of an ER may

o) Support the following if the URP supports untagged and priority-tagged frames:
    1) A Port VLAN Identifier (PVID) value (6.9);
    2) Configuration of at least one VID whose untagged set includes that Port (8.8.2).
p) Comprise a single conformant C-VLAN component (5.4).
q) Support disabling of learning on each DRP (8.6.1).
r) Support the ability to discard frames received at each DRP if there is no entry in the filtering database that specifies forwarding on that Port for the frame's source MAC address and VLAN.
s) Support the operation of the learning process as described in 8.7.

5.23.1.1 VEB ER requirements

In addition to the requirements stated in 5.23.1, a conformant VEB ER implementation shall

a) Request that reflective relay service not be provided by setting adminReflectiveRelayRequest to FALSE (6.6.6).

5.23.1.2 VEPA ER requirements

In addition to the requirements stated in 5.23.1, a conformant VEPA ER implementation shall

a) Disable learning on the URP (8.6.1).
b) Filter frames as specified in 8.6.3.1.

A conformant VEPA ER implementation may

c) Filter frames received at each DRP that are destined for the URP until reflective relay is enabled (6.6.6).

A conformant VEPA ER implementation should

d) Request the provision of reflective relay service by setting adminReflectiveRelayRequest to TRUE (6.6.6).

NOTE—This item is optional because there can be cases where an EVB station is configured to prohibit VSIs from communicating with each other in VEPA mode.

6. Support of the MAC Service

6.6 Internal Sublayer Service

Insert the following new subclause 6.6.6, renumbering existing subclauses as necessary:

6.6.6 EVB status parameters

The Internal Sublayer Service optionally makes available parameters that control and represent the EVB status of each instance of the service provided.

The **EVBMode** parameter determines whether EVB functionality is supported, and in what mode. The parameter can take one of the following three values:

a) **EVB Bridge.** The service supports the functionality of an EVB Bridge.
b) **EVB station.** The service supports the functionality of an EVB station.
c) **Not Supported.** The service does not support EVB functionality. This value is assumed if the EVB status parameters are not implemented.

6.6.6.1 EVBMode = Not supported

If the value of the **EVBMode** parameter is **Not Supported**, then no further status parameters are available, EVB functionality is not supported, and the operation of the service follows the normal forwarding rules for a Bridge.

6.6.6.2 EVBMode = EVB Bridge

If the value of the **EVBMode** parameter is **EVB Bridge**, then further parameters are available, as follows:

a) **reflectiveRelayCapable.** If this parameter is TRUE, then the active topology enforcement function is capable of performing reflective relay, as specified in 8.6.1; if FALSE, the active topology enforcement function is not capable of performing reflective relay.

NOTE 1—The value of the reflectiveRelayCapable parameter is an inherent property of the implementation and is not subject to administrative control.

b) **operReflectiveRelayControl**. If this parameter is TRUE, then reflective relay is enabled; if FALSE, reflective relay is disabled.

NOTE 2—Reflective relay is enabled if a remote EVB station has requested that it be provided (as determined by protocol exchanges between the EVB station and EVB Bridge) and the EVB Bridge is capable of providing it, or disabled if the EVB station has not requested that it be provided or the EVB Bridge is not capable of providing it.

6.6.6.3 EVBMode = EVB station

If the value of the **EVBMode** parameter is **EVB station**, then further parameters are available, as follows:

a) **adminReflectiveRelayRequest**. This parameter can take one of two values:
   1) **TRUE**. The attached EVB Bridge is requested to enable Reflective relay.
   2) **FALSE**. The attached EVB Bridge is requested to disable Reflective relay.

NOTE 1—The value of adminReflectiveRelayRequest is used in the EVB TLV exchanges described in D.2.13 to indicate to an attached EVB Bridge that the EVB station needs reflective relay to be provided. A given EVB station is not required to support both possible values of adminReflectiveRelayRequest.

b) **operReflectiveRelayStatus**. This parameter can take one of three values:
   1) **TRUE.** The EVB Bridge has enabled reflective relay.
   2) **FALSE.** The EVB Bridge has disabled reflective relay.
   3) **Unknown.** It is not known whether the EVB Bridge has enabled reflective relay or not.

NOTE 2—The value of operReflectiveRelayStatus indicates whether or not the EVB Bridge has enabled reflective relay, or whether the EVB Bridge status is not currently known, as determined by protocol exchanges between the EVB station and EVB Bridge. The EVB Bridge status can be unknown during initialization or until the protocol exchanges have completed.

6.11.4 Regenerating priority

Change the NOTE as follows:

NOTE—IEEE 802 LAN technologies signal a maximum of eight priority values. ~~Annex G~~Annex I further explains the use of priority values and how they map to traffic classes.

8. Principles of bridge operation

8.6 The Forwarding Process

8.6.1 Active topology enforcement

Change the initial paragraph as follows:

To prevent data loops and unwanted learning of source MAC addresses, the Forwarding Process determines the values (TRUE, or FALSE) of the learning and forwarding controls (8.4) appropriate to each received frame and Bridge Port, If learning is ~~true~~ TRUE for the receiving Port and ingress filtering (8.6.2) would not cause the received frame to be discarded, the source address and VID are submitted to the Learning Process. If forwarding is ~~true~~ TRUE for the receiving Port, and either the EVBMode parameter value (6.6.6) for the Port is not "EVB Bridge" or the value of the operReflectiveRelayControl parameter for the Port is FALSE, each Bridge Port, other than the reception Port, with forwarding ~~true~~ TRUE is identified as a potential transmission Port. If forwarding is TRUE for the receiving Port and the EVBMode parameter value (6.6.6) for the Port is "EVB Bridge" and the operReflectiveRelayControl parameter value for the Port is TRUE, each Bridge Port, including the reception Port, with forwarding TRUE is identified as a potential transmission Port.

Insert a new paragraph between the first and second paragraphs, as follows:

In an edge relay (ER), the forwarding process may set learning FALSE for all frames.

Insert new subclause 8.6.1.1 at the end of 8.6.1, as follows:

8.6.1.1 Requirements for the use of reflective relay

VEPA ERs (8.6.3.1) used in Edge Virtual Bridging (Clause 40) require reflective relay (6.6.6, 8.6.1) to be enabled on an attached EVB Bridge SBP in order to ensure that all VSIs connected to the VEPA ER are able to receive frames transmitted on one of the other VSIs. The following requirements ensure that a device operates correctly when using reflective relay:

a) The operation of the device shall be such that it prevents the establishment of loops, i.e., the device shall not both request and provide reflective relay on the same port.
b) The device shall ensure that any frame that it transmits on a given port, and that is reflected back to the device through the attached Bridge Port, is filtered by the device in order to prevent it being delivered to the originating port.
c) Any device requesting reflective relay is responsible for performing frame replication as necessary for delivery to multiple ports.

NOTE—Information that can be used to assist in meeting these requirements is the source MAC address and the FID derived from the VLAN ID.

8.6.3 Frame filtering

Change the first sentence of the second paragraph of 8.6.3 as follows:

Each of the Reserved MAC Addresses specified in Table 8-1 shall be permanently configured in the Filtering Database in C-VLAN components and Edge Relays.

Insert new subclause 8.6.3.1 at the end of 8.6.3, as follows:

8.6.3.1 Virtual edge port aggregator (VEPA) filtering

A virtual edge port aggregator (VEPA) ER filters frames as follows.

If the receiving port is a DRP, then the URP shall be selected as the only transmission Port.

If the receiving port is a URP, then in addition to the filtering specified in 8.6.3, if there is a filtering entry that specifies forwarding for the source MAC address and VLAN of the frame for any DRP, then that DRP shall be removed from the list of potential transmission Ports.

8.7 The Learning Process

Change the last paragraph, and insert a new paragraph after it, as follows:

The purpose of filtering utility criteria is to reduce the capacity requirements of the Filtering Database and to reduce the time for which service can be denied (6.5.1) by retaining filtering information learned prior to a change in the physical topology of the network. Filtering utility criteria shall be applied to the learning and retention of information for each Filtering Identifier (FID) (8.8.8). In Bridges other than EVB Bridges (5.22), ~~E~~enhanced filtering utility criteria may be implemented for any Bridge Port as specified below (8.7.2); if implemented, both the default (8.7.1) and the enhanced criteria shall be selectable by management. In EVB Bridges, the enhanced filtering utility criteria shall be implemented for all Bridge Ports, and the default filtering utility criteria shall not be implemented.

Figure 8-4 illustrates the operation of the Learning Process in the inclusion of station location information carried by a single frame, received on one of the Ports of a Bridge, in the Filtering Database.

8.7.2 Enhanced filtering utility criteria

Change 8.7.2 as follows:

The enhanced criteria are satisfied if at least one VID that uses the FID includes the reception Port and at least one other Port with a Port State of Learning or Forwarding in its member set, and:

a) The operPointToPointMAC parameter is false for the reception Port; or
b) Ingress for the VID is permitted through a third Port~~.~~; or
c) The reception Port has reflective relay enabled (6.6.6.2).

NOTE—The third port can, but is not required to, be in the member set.

~~Figure 8-4 illustrates the operation of the Learning Process in the inclusion of station location information carried by a single frame, received on one of the Ports of a Bridge, in the Filtering Database.~~

10. Multiple Registration Protocol (MRP) and Multiple MAC Registration Protocol (MMRP)

10.6 Protocol operation

Change the second paragraph after NOTE 4 as follows:

When two MRP Participants are connected by a point-to-point medium or service instance delaying MRPDU transmission provides no benefit. In bridged networks it is desirable to transmit without delay, minimizing the denial of service that might occur while registration changes propagate after reconfiguration, and maximizing the benefit from using protocols such as RSTP and MSTP. When operPointToPointMAC (~~6.4.3~~6.6.3) is TRUE, transmit opportunities are scheduled immediately on request, subject to rate limiting (10.7.4).

12. Bridge management

12.1 Management functions

12.1.1 Configuration management

Insert the following list item, re-lettering if necessary, to follow the existing list:

l) The ability to configure the functional elements of Edge Virtual Bridging and to control their operation.

12.2 VLAN-aware bridge objects

Insert the following list item, re-lettering if necessary, to follow the existing list:

r) The Edge Virtual Bridging entities (12.26).

12.3 Data types

Insert the following list items, re-lettering if necessary, and NOTE, to follow the existing list:

s) Timer exp, an unsigned value from 0–31 representing a positive integer for the exponent of 2, which forms the multiplier of 10 μs, used for EVB protocol timeout parameters.

NOTE—For example, a value of 4 represents $2^4 \times 10$ μs, or 160 μs.

t) Boolean array, an array of Boolean values.

12.4 Bridge Management Entity

Change subclause 12.4.1 as follows:

12.4.1 Bridge Configuration

The Bridge Configuration object models the operations that modify, or inquire about, the configuration of the Bridge's resources. There is a single Bridge Configuration object per Bridge.

The management operations that can be performed on the Bridge Configuration are

a) Discover Bridge (12.4.1.1);
b) Read Bridge (12.4.1.2);
c) Set Bridge Name (12.4.1.3);
d) Reset Bridge (12.4.1.4)~~.~~;
e) Read component table entry (12.4.1.5);
f) Update component table entry (12.4.1.5).

Insert the following new subclauses 12.4.1.5, 12.4.1.5.1, and 12.4.1.5.2, and Table 12-1 after subclause 12.4.1.4, renumbering subsequent tables as necessary.

12.4.1.5 Bridge component configuration

There is a single Bridge component table per system. Each entry in the component table represents a component of the system (Table 12-1). The entries hold the parameters for each component including the component type and capabilities.

The operations that can be implemented on component table entries are as follows:

a) Read component table entry;
b) Update component table entry.

Table 12-1—Component table entry managed object

Name	Data type	Operations supported*	References
compComponentId	ComponentID	R	12.3 l)
compMACAddress	MAC Address	R	8.13.8, 13.24
compNumberPorts	unsigned (1..4095)	R	12.4.1.1.3 c)
compComponentType	ComponentType	R	12.3 m)
compDeviceCapabilities	Boolean array (0..7)	R	12.10.1.1.3 b)
compTrafficClassesEnabled	Boolean	RW	—
compMmrpEnabledStatus	Boolean	RW	—

*R= Read-only access; RW = Read/Write access

NOTE—The Bridge component table is implemented in Clause 17 as the BridgeBaseTable (see Table 17-3).

12.4.1.5.1 Component type enumeration

The compComponentType parameter can be assigned the following values:

a) iComponent—An I-component (5.7);
b) bComponent—A B-component (5.8);
c) cVlanComponent—A C-VLAN component (5.5);
d) sVlanComponent—An S-VLAN component (5.6);
e) dBridgeComponent—A VLAN unaware component (5.12);
f) edgeRelay—An EVB station edge relay (5.23.1).

12.4.1.5.2 Component device capabilities

The compDeviceCapabilities parameter contains an array of Boolean values for the following:

g) ExtendedFilteringServices;
h) TrafficClasses;

i) StaticEntryIndividualPort;
j) IVLCapable;
k) SVLCapable;
l) HybridCapable;
m) ConfigurablePvidTagging;
n) LocalVlanCapable.

Change subclause 12.4.2 and insert new Table 12-2, as follows:

12.4.2 Port configuration

The Port Configuration object models the operations that modify, or inquire about, the configuration of the Ports of a Bridge. Unless the system explicitly supports the ability to dynamically create and/or delete ports, there ~~There~~ are a fixed set of Bridge Ports per Bridge (one for each MAC interface), and each is identified by a permanently allocated Port Number.

The allocated Port Numbers are not required to be consecutive. Also, some Port Numbers can be dummy entries, with no actual LAN Port (for example, to allow for expansion of the Bridge by addition of further MAC interfaces in the future). Such dummy Ports can support the Port Configuration management operations and other Port-related management operations in a manner consistent with the Port being permanently disabled.

The information provided by the Port Configuration consists of summary data indicating its name and type. Specific counter information pertaining to the number of packets forwarded, filtered, and in error is maintained by the Forwarding Process resource. The management operations supported by the Bridge Protocol Entity allow for controlling the states of each Port.

A port table entry can be implemented by a Bridge for each Port of each component (Table 12-2). It comprises the parameters for each Port including the port type, capabilities, and statistics.

The management operations that can be implemented on the port ~~Configuration are~~ table are

a) ~~Read Port~~ Read port table entry (12.4.2.1);
b) ~~Set Port Name (12.4.2.2)~~Update port table entry (12.4.2.1).

Delete subclause 12.4.2.1 through 12.4.2.2.3 and insert the following new subclause 12.4.2.1.

12.4.2.1 Port type capabilities and enumeration

The portTypeCapabilities array has a bit for each port type the Port can take, while the portType is an enumeration. The portTypeCapabilities can take any combination of port types while the portType can take exactly one of the following values:

a) C-VLAN Bridge Port;
b) Provider Network Port (PNP);
c) Customer Network Port (CNP);
d) Customer Edge Port (CEP);
e) Customer Backbone Port (CBP);
f) Virtual Instance Port (VIP);
g) D-Bridge Port;
h) Remote Customer Access Port (RCAP)—(12.13.4);
i) Station-facing Bridge Port (SBP)—(6.6.5, 12.26.2);
j) Uplink Access Port (UAP)—(12.26.4);
k) Uplink Relay Port (URP)—(6.6.6, 12.26.5).

Table 12-2—Port table entry

Name	Data type	Operations supported*	References
portComponentId	ComponentID	R	12.4.1.5
portPortNumber	Port Number	R	13.25
portMACAddress	MAC Address	R	12.4.1.1.3 a)
portDelayExceededDiscards	counter	R	—
portMtuExceededDiscards	counter	R	—
portCapabilities	unsigned	R	—
portTypeCapabilities	unsigned	R	—
portType	enumerated	R	12.4.2.1
portExternal	Boolean	R	—
portAdminPointToPoint	unsigned	RW	6.6.3
portOperPointToPoint	Boolean	R	6.6.3
portName	Latin1 String (SIZE(0..32))	RW	—

*R= Read-only access; RW = Read/Write access

NOTE—A portType is not required for a Downlink Relay Port (DRP) or an S-channel Access Port (CAP) as no special EVB objects are necessary. A DRP is type C-VLAN Bridge Port while a CAP is type Customer Network Port.

12.5 MAC entities

Insert the following new subclause 12.5.1 and Table 12-3 after 12.5:

12.5.1 ISS Port Number table managed object (optional)

An instance of the ISS Port Number table can be implemented by a Bridge system to identify the ISS interfaces that can be assigned to Bridge Ports. The ISS table is required when the Bridge Port assigned to an ISS and the ISS itself are referenced using different Port Numbers. The ISS table is keyed on the ISS Port Number. Each ISS table entry identifies a mapping from the ISS Port Number to a Bridge Port's ComponentID and Port Number. An issToComponentID value of 0 indicates the ISS is not bound to a Bridge Port. The issToComponentID and issToPortNumber parameters are updated indirectly as a result of creating or updating other system specific Port objects.

The operation that can be implemented on an ISS Port Number table is

a) Read ISS Port Number table entry.

Table 12-3—ISS Port Number table entry

Name	Data type	Operations supported*	References
issPortNumber	Port Number	R	12.3 i)
issMACAddress	MAC Address	R	8.13.2
issToComponentID	ComponentID, 0	R	12.4.2
issToPortNumber	Port Number, 0	R	12.4.2

*R= Read-only access; RW = Read/Write access

Insert the following new subclauses, tables, and figures, following all existing subclauses, tables, and figures, renumbering as necessary:

12.26 Edge Virtual Bridging management

The conformance requirements for EVB Bridges and EVB stations are defined in 5.22 and 5.23, respectively. Each C-VLAN component, edge relay (5.23.1), and Port-mapping S-VLAN component can be managed using the managed objects of 12.4 through 12.12 along with the EVB managed objects specified in this subclause.

An EVB Bridge system (Figure 12-4) supports the EVB managed objects defined in 12.26.1 through 12.26.3 and 12.27. Optionally, an EVB Bridge supports the managed objects for S-channels defined in 12.26.4.

An EVB station system (Figure 12-5) supports the EVB managed objects defined in 12.26.1, 12.26.3, 12.26.5, and 12.27. Optionally, an EVB station supports the managed objects for S-channels defined in 12.26.4.

The EVB specific managed objects defined here

a) Provide managed objects for identifying and configuring an EVB system and its system-wide default parameters for LLDP, Virtual Station Interface Discovery and Configuration Protocol (VDP), and S-channel (12.26.1);
b) Provide managed objects for configuring the Station-facing Bridge Ports of EVB Bridges (12.26.2);
c) Provide a Virtual Station Interface (VSI) table that contains the current VSI and VDP state for each VSI that is active in the EVB system (12.26.3);
d) Provide managed objects for configuring the Port-mapping S-VLAN components and S-channels (12.26.4).
e) Provide managed objects for configuring the Uplink Relay Ports of the edge relays (12.26.5).

For each EVB system, an EVB system configuration managed object exists containing the system-wide defaults used to initialize the other EVB objects.

Every Port of the EVB system (40.1, 40.2, 40.3) is uniquely identified by a ComponentID (12.3) and Port Number (12.3) [which together identify a Port Index (12.3)]. In EVB systems where the ISS and bound Bridge Port have different Port Numbers, an ISS table (12.5.1) allows determining the associated Bridge Port's ComponentID and Port Number from the scalar ISS Port Number. An example where the ISS table

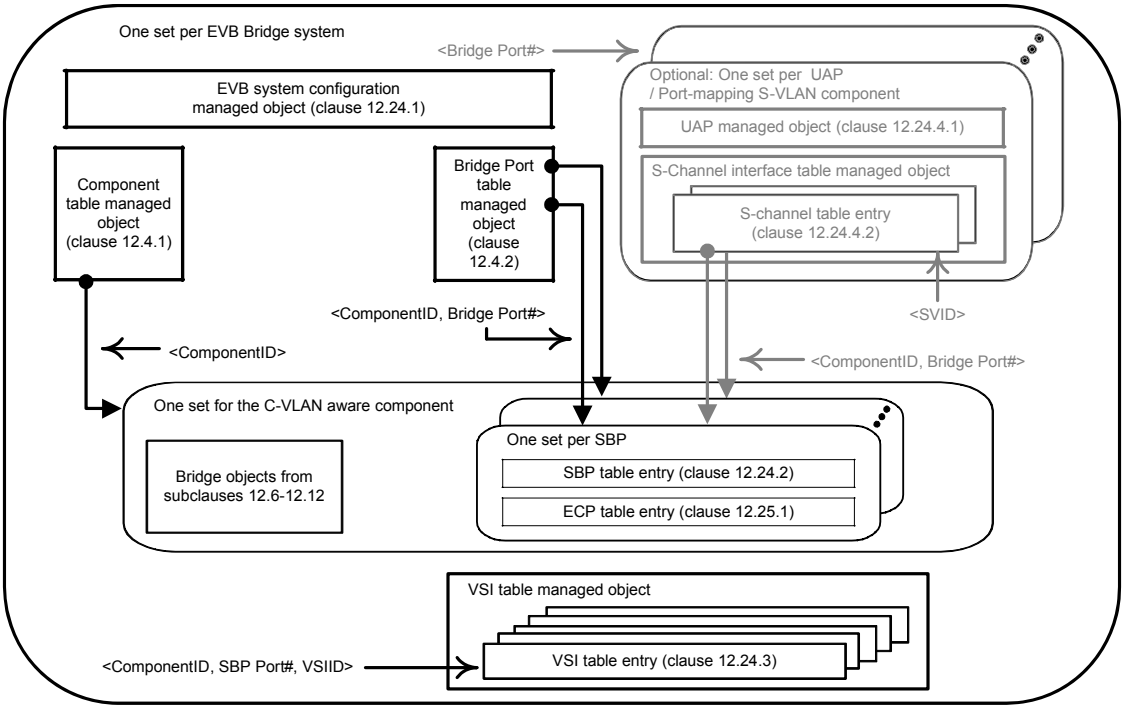
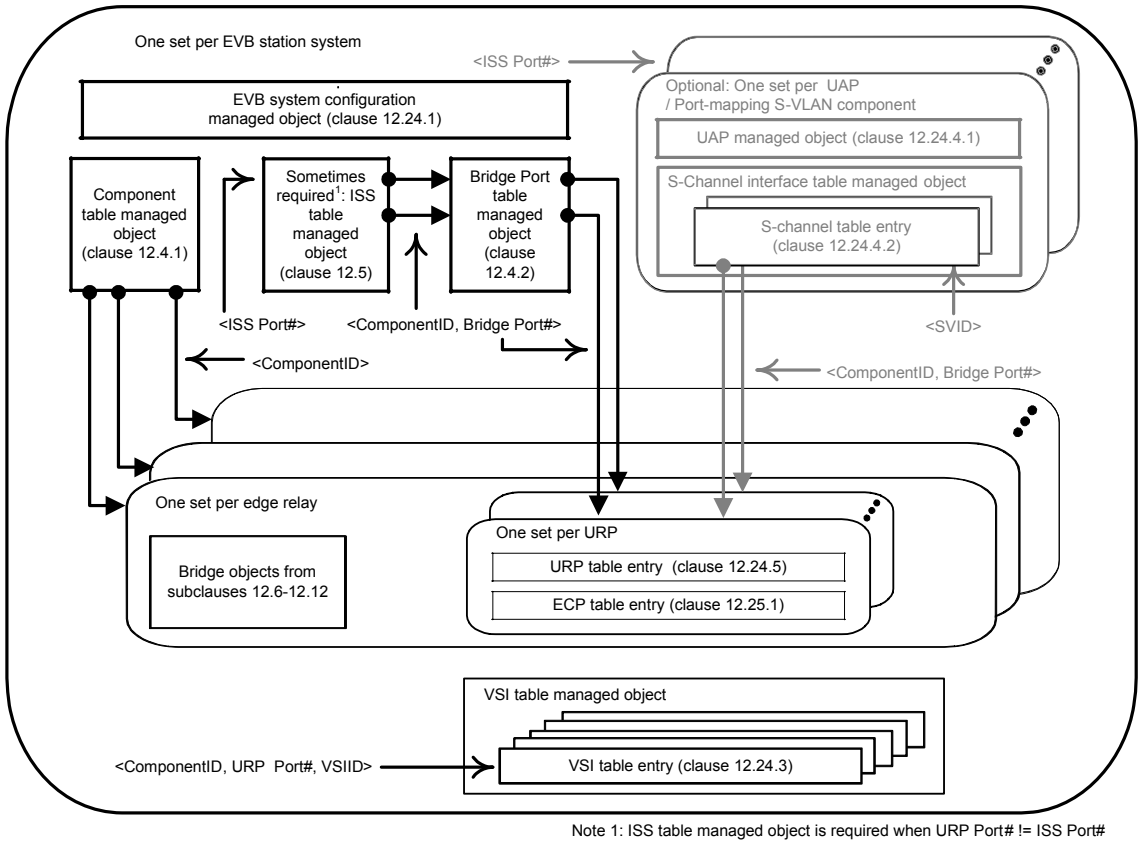


Figure 12-4—Relationships among EVB Bridge managed objects



Note 1: ISS table managed object is required when URP Port# != ISS Port#

Figure 12-5—Relationship among EVB station managed objects

would be needed is an EVB station where the station has multiple LANs each attaching to different edge relays and where the edge relays each use the same Port Number (however different ComponentIDs) to identify their Uplink Relay Ports (Figure 40-3). The ISS table is normally not necessary in an EVB Bridge since the single C-VLAN aware component provides a Bridge Port Number that is unique within the EVB Bridge for every ISS, allowing the ISS and the Bridge Port to use the same Port Number independent from the ComponentID.

The edge relays of an EVB station have two types of ports. Each edge relay DRP provides attachment to one or more Virtual Station Interfaces. No special EVB objects are used to manage the DRPs. URPs can attach to a LAN that can be externally accessible or attach through an internal LAN to a CAP of an S-channel. URPs are managed using the URP table. Each URP table entry identifies a URP within the system and allows configuration and monitoring of LLDP and VDP for the URP (12.26.5). For each URP there is also a ECP table entry that allows configuration and monitoring of ECP (12.27).

NOTE 1—The most common case is that a single VSI is associated with each DRP.

The C-VLAN component of an EVB Bridge has two types of Ports. Each C-VLAN Bridge Port is externally accessible and does not provide any EVB capabilities. No special EVB objects are used to manage the C-VLAN Bridge Ports. Each SBP attaches to a LAN that can be externally accessible or attach though an internal LAN to a CAP of an S-channel. SBPs are managed using the SBP table. Each SBP table entry identifies a single SBP within the system and allows configuration and monitoring of LLDP and VDP for the SBP (12.26.2). For each SBP there is also a ECP table entry that allows configuration and monitoring of ECP (12.27).

Within each EVB station and EVB Bridge, a table holding the VSIs allows monitoring of the active VSIs (12.26.3). VSI table entries are keyed on the VSI Instance Identifier (VSIID) and the URP's or SBP's ComponentID and Port Number. During the movement of VSIs within the DCN it is possible to have a transient condition where the same VSIID exists at two locations. When these two instances are at the same URP or SBP, they are indistinguishable and therefore the table holds only the state for the most recent VSI command; however, when the movement is between different URPs or SBPs, the table will hold two copies of the same VSIID differentiated by the ComponentID and Port Number of the URPs or SBPs.

NOTE 2—In the case where a VSI is migrated between DRPs of the same ER, there is no reason to de-associate, as the table does not maintain any state that identifies the DRP of the ER to which the VSI is attached.

An association is identified by its URP and VSIID. If an association is established for a VSI, and a new association is established on the same URP for the same VSIID, then the new association replaces the old one. A subsequent de-association for that VSIID will remove the current association, regardless of the parameters.

In EVB systems supporting S-channels, the optional UAP table and S-channel interface table are used to configure and control the Port-mapping S-VLAN components and CDCP protocol (12.26.4).

Each ISS that is bound to an Uplink Access Port (UAP) automatically has a Port-mapping S-VLAN component with a default S-channel (using S-VID 1). Each S-channel is identified by the ISS Port Number of the UAP and the S-VID, for SBPs, or SCID, for URPs, identifying the S-channel. The default S-channel carries all the un-S-tagged traffic that traverses the Port-mapping S-VLAN component.

In an EVB Bridge system, S-channels are automatically connected to SBPs when the S-Channel is created. If the S-channel is the default S-channel, it is connected to the SBP with the same Port Number as the ISS bound to the UAP. If the S-channel is not the default S-channel, a new SBP is allocated to the S-channel (which has a Port Number different from any ISS Port Number where a UAP or C-VLAN Bridge Port could be bound) and connected to the CAP through an internal LAN (6.14). With CDCP enabled and operating in

the ‘B’ role, the S-channel interface table entries are automatically created (deleted) by CDCP to fill requests from the peer CDCP operating the ‘S’ role.

In an EVB station, S-channels are connected to URPs by the operating system. The URP Port Numbers are not reserved, so the UAP’s ISS Port Number may not be the same as the URP Port Number. The operating system is not required to automatically attach each S-channel to a URP; however, it can if it chooses. The URP Port Numbers have no restrictions beyond the normal requirement that every relay port be uniquely identified in the EVB system by a ComponentID and Port Number pair. With CDCP enabled and operating in the ‘S’ role, CDCP uses the S-channel interface table to build requests for creating (deleting) S-channels from the peer CDCP operating in the ‘B’ role and enables these S-channels after the peer grants an S-VID for use by the S-channel.

Management of the UAP, Port-mapping S-VLAN component and CDCP is achieved through the UAP table (12.26.4.1). The management of CAPs and their LAN attachments is accomplished by the S-channel interface table entries (12.26.4.2). A UAP table entry along with the Port-mapping S-VLAN creation rules specified in 12.26.4 are sufficient to manage the Port-mapping S-VLAN component. The Bridge managed objects in 12.4 through 12.12 can optionally be used, in addition to the objects of 12.26.4, to manage the Port-mapping S-VLAN components.

The following managed objects, illustrated in Figure 12-4 and Figure 12-5, define the semantics of the management operations specific to EVB Bridges and stations:

a) The EVB system base managed object (12.26.1);
b) The SBP table entry managed object (12.26.2);
c) The VSI table entry managed object (12.26.3);
d) The MAC/VID pair table entry managed object (12.26.3);
e) The UAP table entry managed object (12.26.4);
f) The S-channels interface table entry managed object (12.26.4);
g) The URP table entry managed object (12.26.5).

12.26.1 EVB system base table

An instance of the EVB system base table (Table 12-17) can be implemented by an EVB Bridge or EVB station. It comprises the identifiers for an EVB system (40) and system-wide default parameters used to support EVB services (D.2.13, 6.6.5), VSI discovery (42.2), and ECP (43.3).

Table 12-17—EVB system base table

Name	Data type	Operations supported*	Conformance†	References
evbSysType	enumerated {sysB, sysS}	R	BE	5.22, 5.23
evbSysNumExternalPorts	unsigned [1...4095]	R	BE	12.4.2, 12.5.1
evbSysEvbLldpTxEnable	Boolean	RW	BE	D.2.13
evbSysEvbLldpManual	Boolean	RW	BE	D.2.13
evbSysEvbLldpGidCapable	Boolean	RW	BE	D.2.13

Table 12-17—EVB system base table *(continued)*

Name	Data type	Operations supported*	Conformance†	References
evbSysEcpDfltAckTimerInit	timer exp	RW	BE	D.2, 43.3.6.1
evbSysEcpDfltMaxTries	unsigned [0...7]	RW	BE	D.2, 43.3.7.4
evbSysVdpDfltRsrcWaitDelay	timer exp	RW	BE	D.2.13, 41.5.5.7
evbSysVdpDfltReinitKeepAlive	timer exp	RW	BE	D.2.13, 41.5.5.5

*R= Read-only access; RW = Read/Write access
†B = Required for an EVB Bridge system; E = Required for an EVB station system

The management operations that can be performed on an EVB system table entry are as follows:

a) Read EVB system table entry;
b) Update EVB system table entry.

12.26.1.1 System identifiers

The evbSysType identifies the system type. The enumerated types for evbSysType are

a) sysB—EVB Bridge;
b) sysS—EVB station.

12.26.1.2 System defaults for EVB

The parameters evbSysEvbLldpTxEnable and evbSysEvbLldpManual are used to initialize the LLDP EVB objects for new SBPs and URPs. When evbSysLldpTxEnable is TRUE, a new SBP or URP will place the local EVB objects in the LLDP nearest Customer database; when FALSE, a new SBP or URP will not place the local EVB objects in the LLDP database. When evbSysLldpManual is FALSE, the operating configuration will be determined by the comparison between the local and remote LLDP EVB objects (automatic), regardless of the setting of evbSysLldpTxEnable. When evbSysLldpManual is TRUE, the configuration will be determined by the setting of the local EVB objects only (manual).

The evbSysLldpGidCapable parameter indicates if the port is capable of processing GroupIDs. GroupIDs can be used if both the EVB Bridge and EVB station indicate they are capable by setting the EVB LLDP object for GroupID capable to TRUE. On an EVB Station this means the station can provide GroupIDs rather than VIDs in VDP requests and accept VIDs in response from the EVB Bridge. On an EVB Bridge this means the Bridge can provide a VID corresponding to the GroupID provided by the EVB station.

The default value for evbSysEcpDfltAckTimerInit is 14, which provides a time of 164 msec. Systems are not required to implement smaller times and can reject requests to set the timers to small times, however are required to implement 14 and above to allow fall back to the longest time proposed by the EVB Bridge or EVB station.

The default value for evbSysVdpDfltRsrcWaitDelay and evbSysVdpDfltReinitKeepAlive is system dependant. All systems support the values of 20 and above, which provide times of 10.5 seconds and greater. Systems are not required to implement smaller times and can reject requests to set the timers to small times; however, systems are required to implement 20 and above to allow fall back to the longest time proposed by the EVB Bridge or EVB station.

Table 12-18 shows how the system defaults are used to initialize the parameters in the SBP, URP, and ECP table entries.

Table 12-18—EVB system parameter defaults

System parameter	Default value	LLDP, SBP, URP, ECP entry parameter
evbSysEvbLldpTxEnable	TRUE	LLDP Transmit Enable
evbSysEvbLldpManual	FALSE	sbpLldpManual, urpLldpManual
evbSysEvbLldpGidCapable	system dependant	LLDP GID Capable
evbSysEcpDfltAckTimerInit	14, for 164 milliseconds	ecpAdminAckTimerInit
evbSysEcpDfltMaxTries	3	ecpAdminMaxTries
evbSysVdpDfltRsrcWaitDelay	system dependant	sbpVdpAdminRsrcWaitDelay, urpVdpAdminRsrcWaitDelay
evbSysVdpDfltReinitKeepAlive	system dependant	sbpVdpAdminReinitKeepAlive, urpVdpAdminReinitKeepAlive

12.26.2 SBP table entry

SBP table entries may be created explicitly or implicitly as a result of creating an entry in the S-channel interface table. When an SBP table entry is created, the port type in the Port table entry (Table 12-2) changes to type ‘Station-facing Bridge Port’. When an SBP table entry is deleted, the Port table entry returns to type C-VLAN Bridge Port.

Whenever a new SBP table entry is created, a new entry is also created in the ECP table (12.27) keyed under the ComponentID and Port Number of the SBP. Whenever an SBP table entry is deleted, the corresponding entry in the ECP table is deleted.

The management operations that can be performed on an SBP table entry are as follows:

a) Read SBP table entry;
b) Update SBP table entry;
c) Create SBP table entry;
d) Delete SBP table entry.

Table 12-19—SBP table entry

Name	Data type	Operations supported*	Conformance†	References
sbpComponentID	ComponentID	R	B	12.4.1.5
sbpPortNumber	Port Number	R	B	12.4.2
sbpLldpManual	Boolean	RW	B	—
sbpVdpOperRsrcWaitDelay	timer exp	R	B	D.2.13, 41.5.5.7
sbpVdpOperReinitKeepAlive	timer exp	R	B	D.2.13, 41.5.5.5
sbpVdpOperToutKeepAlive	unsigned	R	B	D.2.13, 41.5.5.13

*R = Read-only access; RW = Read/Write access
†B = Required for an EVB Bridge system

12.26.3 VSI table entry

Each EVB system maintains a table of the active Virtual Station Interfaces. The structure of a VSI table entry is shown in Table 12-20. This read-only table provides the current operation parameters of each VSI along with the VDP state associated with the VSI. The table is keyed on the SBP's or URP's ComponentID and Port Number and on the VSIID. The operations that can be performed on the VSI table are

a) Read entry for a ComponentID, Port Number and VSIID.

Each EVB Bridge or EVB station maintains a table of the VID/MACs on each Virtual Station Interface. This read-only table provides the current GroupID/VID/MAC assignments for each VSI. The operations that can be performed on the VSI table are

b) Read entries for a ComponentID, Port Number and VSIID
c) Read entries for a ComponentID and Port Number

Table 12-20—VSI table entry

Name	Data type	Operations supported*	Conformance†	References
evbVsiComponentID	ComponentID	R	BE	12.4.1.5
evbVsiPortNumber	Port Number	R	BE	12.4.2
evbVsiIDType	enumerated	R	BE	41.2.6, Table 41-5
evbVsiID	Latin1 String (SIZE(16))	R	BE	41.2.7
evbVsiTimeSinceCreate	time interval	R	BE	41
evbVsiVdpOperCmd	enumerated	R	BE	41.2.1, Table 41-1
evbVsiOperRevert	Boolean	R	BE	41.2.3
evbVsiOperHard	Boolean	R	BE	41.2.3
evbVsiOperReason	unsigned (0..15)	R	BE	41.2.3
evbVsiMgrID	Latin1 String (SIZE(1))	R	BE	41.1.3
evbVsiType	Latin1 String (SIZE(3))	R	BE	41.2.4
evbVsiTypeVersion	Latin1 String (SIZE(1))	R	BE	41.2.5
evbVsiMvFormat	Latin1 String (SIZE(1))	R	BE	41.2.8
evbVsiNumMACs	unsigned	R	BE	41.2.9
evbVdpMachineState	enumerated	R	BE	41.5.5.14
evbVdpCmdsSucceeded	counter	R	BE	41.5
evbVdpCmdsFailed	counter	R	BE	41.5
evbVdpCmdsReverts	counter	R	BE	41.5

*R = Read-only access; RW = Read/Write access
†B = Required for an EVB Bridge system; E = Required for an EVB station system

Table 12-21—VSI MAC/VLAN table entry

Name	Data type	Operations supported*	Conformance†	References
evbMvComponentID	ComponentID	R	BE	12.4.1.5
evbMvPortNumber	Port Number	R	BE	12.4.2
evbMvVsiIDType	enumerated	R	BE	41.2.6, Table 41-5
evbMvVsiID	Latin1 String (SIZE(16))	R	BE	41.2.7
evbMvVsiGroupID	unsigned	R	BE	41.2.9
evbMvVsiVID	unsigned (1..4094)	R	BE	41.2.9
evbMvVsiMAC	MAC Address	R	BE	41.2.9

*R= Read-only access; RW = Read/Write access
†B = Required for an EVB Bridge system; E = Required for an EVB station system

12.26.4 S-channel configuration and management

The S-channel managed objects are not required unless the system implements S-channels.

Creating an UAP table entry (40.2) causes a Port-mapping S-VLAN component (15.6) to be instantiated and sets the portType parameter of the Port table entry (12.4.2) to type ‘Uplink Access Port’. The Port-mapping S-VLAN component automatically includes a default S-channel with one CAP that can attach to an SBP or a URP through an internal LAN (6.14). Each UAP is identified by the ISS Port Number where the UAP is attached. The default S-channel and CAP shall be identified by the S-VID 1 and SCID 1.

12.26.4.1 UAP table entry

The management operations that can be performed on the UAP table entry (Table 12-22) managed object are as follows:

a) Read UAP table entry;
b) Update UAP table entry;
c) Create UAP table entry;
d) Delete UAP table entry.

Table 12-22—UAP table entry

Name	Data type	Operations supported*	Conformance†	References
uapISSPortNumber	Port Number	R	BE	12.4.2, 12.5.1
uapComponentID	ComponentID	R	be	12.4.1.5
uapPortNumber	Port Number	R	be	12.4.2
uapSchCdcpAdminEnable	Boolean	RW	BE	42.4.2
uapSchCdcpAdminRole	enumerated	RW	BE	42.4.2
uapSchCdcpAdminChnCap	unsigned [1...167]	RW	BE	42.4.1
uapSchCdcpOperChnCap	unsigned [1...167]	R	BE	42.4.8
uapSchAdminCdcpSvidPoolLow	unsigned [0,2...4094]	RW	BE	42.4.7
uapSchAdminCdcpSvidPoolHigh	unsigned [0,2...4094]	RW	BE	42.4.7
uapSchOperState	enumerated	R	BE	42.4.15
uapSchCdcpRemoteEnabled	Boolean	R	BE	42.4.14
uapSchCdcpRemoteRole	enumerated	R	BE	42.4.12

*R = Read-only access; RW = Read/Write access
†B = Required for an EVB Bridge system; E = Required for an EVB station system; b = Optional for an EVB Bridge system; e = Optional for an EVB station system

Table 12-23—UAP table entry parameters

UAP table name	Default values
uapSchCdcpAdminEnable	TRUE
uapSchCdcpAdminRole	schS if EVB station and schB if EVB Bridge
uapSchCdcpAdminChnCap	1
uapSchAdminCdcpSvidPoolLow	0
uapSchAdminCdcpSvidPoolHigh	0

The available SVIDs determined by the range of uapSchAdminCdcpSvidPoolLow and uapSchAdminCdcpSvidPoolHigh limit the S-channel capacity indicated by uapSchCdcpAdminChnCap to the available SVIDs plus one for the default S-channel. If the capacity is greater than the VID range plus one, then the VID range overrides the capacity (i.e., the actual capacity is never bigger than the available VID range plus one).

12.26.4.2 S-channel interface table entry

The S-channel interface table entry applies to each internal S-channel configured on an EVB Bridge or EVB station, as shown in Table 12-24. The management operations that can be performed on an S-channel interface table entry are as follows:

a) Read S-channel interface table entry;
b) Update S-channel interface table entry;
c) Create S-channel interface table entry;
d) Delete S-channel interface table entry.

If the S-channel interface table is being used with a UAP operating in the cdcpS role, then the table is keyed on schUapISSPortNumber and schScid. If the S-channel interface table is being used with a UAP operating in the cdcpB role, then the table is keyed on the schUapISSPortNumber and schSvid.

The schComponentID and schCapPortNumber refer to the Port-mapping S-VLAN component and the Port which may also be identified by the schISSPortNumber and S-VID (or SCID).

The schSbpOrUrpComponentID and schSbpOrUrpPortNumber are the ComponentID and Port Number of the attached SBP or URP.

Table 12-24—S-channel interface table entry

Name	Data type	Operations supported*	Conformance†	References
schUapISSPortNumber	Port Number	R	BE	12.4.2,12.5.1
schScid	unsigned [1...4094]	R	bE	42.4.3
schSvid	unsigned [0...4094]	R	BE	42.4.3
schComponentID	ComponentID	R	be	42.1, 12.4.1.5
schCapPortNumber	Port Number	R	be	42.1, 12.4.2
schSbpOrUrpComponentID	ComponentID	RW	BE	12.4.1.5
schSbpOrUrpPortNumber	Port Number	RW	BE	12.4.2

*R= Read-only access; RW = Read/Write access
†B = Required for an EVB Bridge system; E = Required for an EVB station system;
b = Optional for an EVB Bridge system; e = Optional for an EVB station system

12.26.5 Edge relay management

Edge relays can be built dynamically or statically within an EVB station. Each edge relay is assigned a ComponentID that is unique for the EVB station system. An edge relay always has a single Uplink Relay Port (URP) that exists as long as the edge relay exists. The Downlink Relay Ports (DRPs) of an edge relay may be built either along with the edge relay or on demand.

12.26.5.1 URP table entry

When a URP table entry is created, a corresponding Component table entry is created for the edge relay along with a Port table entry for the URP. When a URP table entry is deleted, the corresponding Component table entry is deleted along with the corresponding Port table entry.

When each URP table entry is created, a corresponding entry is created in the ECP table (12.27) and keyed under the ComponentID and Port Number of the URP. Whenever a URP table entry is deleted, the corresponding entry in the ECP table is deleted.

The management operations that can be performed on the URP table entry (Table 12-25) managed object are as follows:

a) Read URP table entry;
b) Update URP table entry;
c) Create URP table entry;
d) Delete URP table entry.

Table 12-25—URP table entry

Name	Data type	Operations supported*	Conformance†	References
urpComponentID	ComponentID	R	E	12.4.1.5
urpPortNumber	Port Number	R	E	12.4.2
urpBindToISSPortNumber	unsigned [0...4095]	RW	e	12.5.1
urpLldpManual	Boolean	RW	E	
urpVdpOperRsrcWaitDelay	timer exp	R	E	D.2.13, 41.5.5.7
urpVdpOperRespWaitDelay	unsigned	R	E	D.2.13, 41.5.5.9
urpVdpOperReinitKeepAlive	timer exp	R	E	D.2.13, 41.5.5.5

*R = Read-only access; RW = Read/Write access
†E = Required for an EVB station system; e = Optional for an EVB station system

12.27 Edge Control Protocol management

12.27.1 ECP table entry

The management operations that can be performed on the ECP table entry managed object are as follows:

a) Read ECP table entry.

ECP table entries are created or deleted implicitly as a result of the creation or deletion of other port objects.

Table 12-26—ECP table entry

Name	Data type	Operations supported*	Conformance†	References
ecpComponentID	ComponentID	R	BE	12.4.1.5
ecpPortNumber	Port Number	R	BE	12.4.2
ecpOperAckTimerInit	timer exp	R	BE	D.2.13, 43.3.7.1
ecpOperMaxTries	unsigned [0...7]	R	BE	D.2.13, 43.3.7.4
ecpTxFrameCount	counter	R	BE	43
ecpTxRetryCount	counter	R	BE	43
ecpTxFailures	counter	R	BE	43
ecpRxFrameCount	counter	R	BE	43

*R = Read-only access; RW = Read/Write access
†B = Required for an EVB Bridge system; E = Required for an EVB station system

17. Management protocol

17.2 Structure of the MIB

Insert the following new row at the end of Table 17-1:

Table 17-1—Structure of the MIB modules

IEEE8021-EVB-MIB	17.7.20	802.1Qbg	5.22, 5.23	Initial version in IEEE Std 802.1Qbg

17.2.2 Structure of the IEEE8021-Bridge MIB

Insert the following new rows at the end of Table 17-3:

Table 17-3—IEEE8021-BRIDGE MIB structure and relationship to IETF RFC 4188 and this standard

ieee8021BridgePortTable		12.5.1
	ieee8021BridgePhyPort	
	ieee8021BridgePhyIfIndex	
	8021BridgePhyMacAddress	
	ieee8021BridgePhyPortToComponentId	
	ieee8021BridgePhyPortToInternalPort	
ieee8021BridgeBaseIfToPortTable		17.3.2.2
	ieee8021BridgeBaseIfToPortComponentID	
	ieee8021BridgeBaseIfIndexPort	

Insert new subclause 17.2.20 as shown, following all existing subclauses of 17.2, renumbering as necessary:

17.2.20 Structure of the IEEE8021-EVB MIB

The IEEE8021-EVB MIB provides objects to configure and manage an EVB station system or EVB Bridge system. Objects in this MIB module are arranged into subtrees. Each subtree is organized as a set of related objects. Where appropriate, the corresponding Clause 12 management reference is also included. Table 17-26 indicates the structure of the IEEE8021-EVB MIB module.

Table 17-26—EVB MIB structure and object cross reference

MIB table	MIB object	References
ieee8021BridgeEvbNotifications subtree		
ieee8021BridgeEvbObjects subtree		
ieee8021BridgeEvbSys		12.26.1
	ieee8021BridgeEvbSysType	
	ieee8021BridgeEvbSysNumExternalPorts	
	ieee8021BridgeEvbSysEvbLldpTxEnable	
	ieee8021BridgeEvbSysEvbLldpManual	
	ieee8021BridgeEvbSysEvbLldpGidCapable	
	ieee8021BridgeEvbSysEcpAckTimer	
	ieee8021BridgeEvbSysEcpMaxRetries	
	ieee8021BridgeEvbSysVdpDfltRsrcWaitDelay	
	ieee8021BridgeEvbSysVdpDfltReinitKeepAlive	
ieee8021BridgeEvbSbpTable		12.26.2
	ieee8021BridgeEvbSbpComponentID	
	ieee8021BridgeEvbSbpPortNumber	
	ieee8021BridgeEvbSbpLldpManual	
	ieee8021BridgeEvbSbpVdpOperRsrcWaitDelay	
	ieee8021BridgeEvbSbpVdpOperReinitKeepAlive	
	ieee8021BridgeEvbSbpVdpOperToutKeepAlive	
	ieee8021BridgeEvbSbpRowStatus	
ieee8021BridgeEvbVsiDbTable		12.26.3
	ieee8021BridgeEvbVsiComponentID	
	ieee8021BridgeEvbVsiPortNumber	
	ieee8021BridgeEvbVsiID	
	ieee8021BridgeEvbVsiTimeSinceCreate	
	ieee8021BridgeEvbVsiVdpOperCmd	
	ieee8021BridgeEvbVsiOperRevert	
	ieee8021BridgeEvbVsiOperHard	
	ieee8021BridgeEvbVsiOperReason	
	ieee8021BridgeEvbVsiMgrID	
	ieee8021BridgeEvbVsiType	
	ieee8021BridgeEvbVsiTypeVersion	

Table 17-26—EVB MIB structure and object cross reference *(continued)*

MIB table	MIB object	References
	ieee8021BridgeEvbVsiMvFormat	
	ieee8021BridgeEvbVsiNumMACs	
	ieee8021BridgeEvbVDPMachineState	
	ieee8021BridgeEvbVDPCommandsSucceeded	
	ieee8021BridgeEvbVDPCommandsFailed	
	ieee8021BridgeEvbVDPCommandReverts	
ieee8021BridgeEvbVsiDbMacTable		12.26.3
	ieee8021BridgeEvbVsiComponentID	
	ieee8021BridgeEvbVsiPortNumber	
	ieee8021BridgeEvbVsiID	
	ieee8021BridgeEvbGroupID	
	ieee8021BridgeEvbVsiMac	
	ieee8021BridgeEvbVsiVlanId	
ieee8021BridgeEvbUapConfigTable		12.26.4.1
	ieee8021BridgePort	
	ieee8021BridgeEvbUapComponentId	
	ieee8021BridgeEvbUapPort	
	ieee8021BridgeEvbUapConfigIfIndex	
	ieee8021BridgeEvbUapCdcpAdminEnable	
	ieee8021BridgeEvbUapAdminCdcpRole	
	ieee8021BridgeEvbUapAdminCdcpChanCap	
	ieee8021BridgeEvbUapOperCdcpChanCap	
	ieee8021BridgeEvbUapAdminCdcpSVIDPoolLow	
	ieee8021BridgeEvbUapAdminCdcpSVIDPoolHigh	
	ieee8021BridgeEvbUapOperState	
	ieee8021BridgeEvbUapCdcpRemoteEnabled	
	ieee8021BridgeEvbUapCdcpRemoteRole	
	ieee8021BridgeEvbUapConfigRowStatus	
ieee8021BridgeEvbCapConfigTable		12.26.4.2
	ieee8021BridgeBridgePort	
	ieee8021BridgeEvbCapSchID	
	ieee8021BridgeEvbCapComponentId	
	ieee8021BridgeEvbCapIfIndex	

Table 17-26—EVB MIB structure and object cross reference *(continued)*

MIB table	MIB object	References
	ieee8021BridgeEvbCapPortNumber	
	ieee8021BridgeEvbCapSChannelID	
	ieee8021BridgeEvbCapAssociateSbpOrUrpCompID	
	ieee8021BridgeEvbCapAssociateSbpOrUrpPort	
	ieee8021BridgeEvbCapRowStatus	
ieee8021BridgeEvbUrpTable		12.26.5
	ieee8021BridgeEvbUrpComponentID	
	ieee8021BridgeEvbUrpPort	
	ieee8021BridgeEvbUrpIfIndex	
	ieee8021BridgeEvbUrpBindToIssPort	
	ieee8021BridgeEvbUrpLldpManual	
	ieee8021BridgeEvbUrpVdpOperRsrcWaitDelay	
	ieee8021BridgeEvbUrpVdpOperRespWaitDelay	
	ieee8021BridgeEvbUrpVdpOperReinitKeepAlive	
	ieee8021BridgeEvbUrpRowStatus	
ieee8021BridgeEvbEcpTable		12.27.1
	ieee8021BridgeEvbEcpComponentID	
	ieee8021BridgeEvbEcpPort	
	ieee8021BridgeEvbEcpOperAckTimerInit	
	ieee8021BridgeEvbEcpOperMaxTries	
	ieee8021BridgeEvbEcpTxFrameCount	
	ieee8021BridgeEvbEcpTxRetryCount	
	ieee8021BridgeEvbEcpFailures	
	ieee8021BridgeEvbEcpRxFrameCount	
ieee8021BridgeEvbConformance subtree		
ieee8021BridgeEvbGroups		
	ieee8021BridgeEvbSysGroup	
	ieee8021BridgeEvbSbpConfigGroup	
	ieee8021BridgeEvbVsiDbGroup	
	ieee8021BridgeEvbUapConfigGroup	
	ieee8021BridgeEvbCapConfigGroup	

Table 17-26—EVB MIB structure and object cross reference *(continued)*

MIB table	MIB object	References
	ieee8021BridgeEvbUrpConfigGroup	
	ieee8021BridgeEvbsEcpConfigGroup	
ieee8021BridgeEvbCompliances		
	ieee8021BridgeEvbbCompliance	
	ieee8021BridgeEvbsCompliance	

17.3 Relationship to other MIBs

Insert new subclause 17.3.20 as shown, following all existing subclauses of 17.3, renumbering as necessary:

17.3.20 Relationship of the IEEE8021-EVB MIB to other MIB modules

The IEEE8021-EVB MIB provides objects that extend the core management functionality of a Bridge, as defined by the IEEE8021-BRIDGE MIB (17.7.2), in order to support the management functionality needed for Edge Virtual Bridging (5.22, 5.23), as defined in Clause 40, Clause 41, Clause 42, and Clause 43. As support of the objects defined in the IEEE8021-EVB MIB also requires support of the IEEE8021-TC-MIB and IEEE8021-BRIDGE-MIB, the provisions of 17.3.2 apply to implementations claiming support of the IEEE8021-EVB MIB.

17.4 Security considerations

Insert new subclause 17.4.20 as shown, following all existing subclauses of 17.4, renumbering as necessary:

17.4.20 Security considerations of the IEEE8021-EVB MIB

The purpose of EVB is to coordinate Virtual Station Interfaces within an EVB station with a DCN. In this environment the EVB station and the EVB Bridge may be under different management authorities. Access to the objects within the IEEE8021-EVB MIB module of the EVB Bridge by the EVB station and access to objects within the IEEE8021-EVB MIB module of the EVB station by the EVB Bridge may therefore need to be restricted.

Access to the objects within the IEEE8021-EVB MIB module, whether they have MAX-ACCESS of read-write, read-create, or read-only, can reveal sensitive information in some network environments. Very serious health and safety situations could arise if EVB systems were involved in configuring network resources for an emergency public safety announcement and the EVB Bridge system behavior of the bridged network was allowed to be modified unexpectedly.

With these considerations in mind it is thus important to control all types of access (including GET and/or NOTIFY) to these objects. 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.

It is recommended that implementers consider the security features as provided by the SNMPv3 framework (see [RFC3410], section 8), including full support for the SNMPv3 cryptographic mechanisms (for authentication and privacy).

Further, deployment of SNMP versions prior to SNMPv3 is not recommended. Instead, it is recommended to deploy SNMPv3 and 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.

There are a number of management objects defined in IEEE8021-EVB MIB module with a MAX-ACCESS clause of read-write and/or read-create. 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. These are the tables and objects and their sensitivity/vulnerability:

Table 17-31—Sensitive managed objects (of EVB): tables and notifications

Table or object	Reason for sensitivity to security considerations
evbSysEvbLldpTxEnable evbSysEvbLldpManual evbSysEvbLldpGidCapable	The EVB TLV exchange controls how the VDP, ECP and reflective relay parameters are set. These parameters allow manual configurations of the EVB parameters, which can be used to disable the operation of these protocols.
ieee8021BridgeEvbSysEcpAckTimer ieee8021BridgeEvbSysEcpMaxRetires	The ECP timer and re-try count are set to provide reliable delivery of control frames. Incorrect settings can cause failures of the control protocols.
ieee8021BridgeEvbSysVdpDfltRsrcWaitDelay	The VDP resource timer determines the time required by the system to locate a profile. Setting this time too short may make VDP requests always fail.
ieee8021BridgeEvbSbpRowStatus ieee8021BridgeEvbUapRowStatus ieee8021BridgeEvbCapRowStatus ieee8021BridgeEvbUrpRowStatus	These variables allow creations of new ports on an EVB Bridge or EVB station. Inappropriate use of these may allow un-authorized interception of data.

Some of the readable objects in this MIB module (i.e., objects with a MAX-ACCESS other than not-accessible) may be considered sensitive or vulnerable in some network environments. 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. The following tables provide the identity of VSIs and the addresses used to access them that could be used to identify the user application and what addresses can be used to interfere with or intercept their traffic.

Table 17-32—Sensitive managed objects (of EVB) for read

ieee8021BridgeEvbVsiDbTable	The VSI database provides a list of all operating VSIs along with their primary system keys. This information may be used to spy on the operating applications and the activity associated with each application.
ieee8021BridgeEvbVsiDbMacTable	The VSI MAC table provides the network addresses associated with operating applications. This information may be used in system attacks.

17.5 Dynamic component and Port creation

17.5.1 Overview of the dynamically created Bridge entities

Change subclauses 17.5.1.1, 17.5.1.2, and 17.5.1.3 as follows:

17.5.1.1 Components

A component contains a relay function whose purpose is to move frames between interfaces to the relay called a Bridge Port~~s~~ or an Edge Relay Port. Different types of components are provided in IEEE Std 802.1Q that are used to construct different types of Bridges.

17.5.1.2 Bridge Ports

A Bridge Port or Edge Relay Port is a frame source or sink directly attached to the relay function of a Bridge component.

17.5.1.3 Internal LAN connections

A Backbone Edge Bridge (BEB) is composed of zero or one B-component and zero or more I-components. When the BEB has both a B-component and some I-components, the PIPs and CBPs of these I-components are connected by internal LAN connections. There needs to be a way to specify the interconnections between the PIPs on the I-component and the Customer Backbone Port on the B-component. This is done via the ieee8021BridgeILanIfTable defined in the IEEE8021-BRIDGE MIB.

Essentially, this table allows for the creation of a new “interface” to represent the connection and then the ifStackTable is used to specify the interconnection.

These interconnections are used in multi-component Bridges, such as:

Provider Edge Bridges ~~and~~,
Backbone Edge Bridges,
EVB stations,
EVB Bridges,

to specify the relationship between two interfaces in the following manner.

Two Port interfaces are interconnected if the invocation of a request operation at one of the interfaces causes an indication operation with the same parameters to happen at the other interface.

Change subclause 17.5.2 as follows:

17.5.2 Component creation

A component is created by making an entry in the ieee8021BridgeBaseTable with the ieee8021BridgeBaseComponentType set to the proper value. Components can also be created indirectly by making entries in other system specific tables which then automatically create entries in the ieee8021BridgeBaseTable.

A Bridge component consists of a relay function and related Bridge Ports or Edge Relay Ports. The component type determines if the relay operates on untagged, C-tagged, or S-tagged frames. It also determines which specific type(s) of Bridge Ports or Edge Relay Ports may be created on the component.

Change subclause 17.5.2.2 and 17.5.2.3 as follows:

17.5.2.2 C-VLAN component creation

C-VLAN components are used in ~~two~~the following different types of Bridges:

a) ~~The first is the C-VLAN component of a~~ customer VLAN Bridges,
b) ~~The second is as the C-VLAN component of a~~ Provider Edge Bridges,
c) EVB Bridges.

Provider Edge Bridge C-VLAN components are created implicitly by the creation of a Customer Edge Port on the S-VLAN component of the Provider Edge Bridge. C-VLAN components that belong to customer Bridges or to EVB Bridges are created by a management station performing a row-create on the component table or by implicit action such as the insertion of blades into a system.

17.5.2.3 S-component creation

S-VLAN components are used in ~~two~~three different ways in Bridges. The first is as the S-VLAN component of an S-VLAN Bridge or Provider Edge Bridge or the foundation for an I-component or B-component. The second is as a Port-mapping S-VLAN component in a Provider Edge Bridge. The third is as a Port-mapping S-VLAN component in an EVB station or an EVB Bridge.

Provider Edge Bridge Port-mapping S-VLAN components are created implicitly by the creation of a Remote Customer Access Port on the primary S-VLAN component of the Provider Edge Bridge.

EVB Bridge and EVB station Port-mapping S-VLAN components are created implicitly by the creation of an Uplink Access Port on an ISS of an EVB Bridge or an EVB station. For an EVB Bridge the ISS is allocated to a Bridge Port of the primary C-VLAN component. On an EVB station the ISS has no permanent Bridge Port or Edge Relay Port assignment.

Insert new subclause 17.5.2.6 as shown, following all existing subclauses of 17.5.2, renumbering as necessary:

17.5.2.6 Edge relay creation

Edge relays of an EVB station are created implicitly by the creation of an Uplink Relay Port, or actions such as the insertion of blades into a system or the installation of a software driver on the EVB station.

Change subclause 17.5.3 as follows:

17.5.3 Port creation

This subclause of the document discusses how Ports of each relevant Port type are created on each relevant component type.

The general procedure is for the network administrator to perform an SNMP row-create operation on a table specific to the type of Port being created. If the operation succeeds, an entry will be implicitly created in the ieee8021BridgeBasePortTable by the agent.

The specific details are outlined in the 17.5.3.1 through 17.5.3.~~5~~6.

Change subclause 17.5.3.2 as follows:

17.5.3.2 Port creation on C-VLAN components

C-VLAN components support four different types of Bridge Ports. These are Customer VLAN Ports, Customer Edge Ports, ~~and~~ Provider Edge Ports, and Station-facing Bridge Ports.

A C-VLAN component that is not part of a Provider Edge Bridge may have Customer VLAN Ports. A C-VLAN component that is part of a Provider Edge Bridge has exactly 1 Customer Edge Port and any number of Provider Edge Ports.

The only type of Ports that may be created by operating on the C-VLAN component that is not part of an EVB Bridge are the Customer VLAN Ports. On an EVB Bridge it is possible to create both C-VLAN Bridge Ports and Station-facing Bridge Ports.

Customer Edge Ports are created by a management action on the S-VLAN component of a Provider Edge Bridge. From a management perspective, these entities are managed through the S-VLAN component or via management operations specific to the Provider Edge Bridge.

Provider Edge Ports are created as a side effect of adding a CEP to the member set of an S-VID in a Provider Edge Bridge.

17.5.3.2.1 Creating Customer VLAN Ports

Insert new subclause 17.5.3.2.2 after the existing 17.5.3.2.1 as follows:

17.5.3.2.2 Creating Station-facing Bridge Ports

SBPs are created by performing a row-create operation on the ieee8021BridgeEvbSbpConfigTable for a C-VLAN component that is configured to act as an EVB Bridge. The required columns are the component ID and the Port Number to use for the newly created Port.

The type of the component referred to by the component ID parameter is a cVlanComponent configured for Q-Bridge operation.

The implicitly constructed ieee8021BridgeBasePortTable entry will have the following fields filled in:

ieee8021BridgeBasePortComponentId	- As per ieee8021BridgeEvbSbpConfigTable
ieee8021BridgeBasePort	- As per ieee8021BridgeEvbSbpConfigTable
ieee8021BridgeBasePortIfIndex	- Implementation Specific Action
ieee8021BridgeBasePortDelayExceededDiscards	- Statistic, reset at creation
ieee8021BridgeBasePortMtuExceededDiscards	- Statistic, reset at creation
ieee8021BridgeBasePortCapabilties	- Implementation Specific

ieee8021BridgeBasePortTypeCapabilities	- Implementation Specific: bit *Station-facing Bridge Port* (8) is set
ieee8021BridgeBasePortType	- Station-facing Bridge Port (8)
ieee8021BridgeBasePortExternal	- Implementation Specific

17.5.3.3 Port creation on S-components

Insert new subclause 17.5.3.3.5 after existing 17.5.3.3.4 as follows:

17.5.3.3.5 Creating an Uplink Access Port (UAP)

UAPs are created by doing a row-create operation on the EVB Bridge's or EVB station's ieee8021BridgeEvbUapConfigTable. The required column is the ISS Port Number to use for the newly created Port. The ComponentID and PortNumber are specified if the system chooses to build an implicit ieee8021BridgeBasePortTable entry. The ieee8021BridgeEvbUapConfigTable contains the following columns:

ieee8021BridgeEvbUapIssPortNumber
ieee8021BridgeEvbUapComponentId
ieee8021BridgeEvbUapPortNumber
ieee8021BridgeEvbUapCdcpAdminEnable
ieee8021BridgeEvbUapAdminCDCPRole
ieee8021BridgeEvbUapAdminCDCPChanCap
ieee8021BridgeEvbUapOperCDCPChanCap
ieee8021BridgeEvbUapAdminCDCPSVIDPoolLow
ieee8021BridgeEvbUapAdminCDCPSVIDPoolHigh
ieee8021BridgeEvbUapOperState
ieee8021BridgeEvbUapCdcpRemoteEnabled
ieee8021BridgeEvbUapCdcpRemoteRole
ieee8021BridgeEvbUapConfigRowStatus

The optional implicitly constructed ieee8021BridgeBasePortTable entry will have the following fields filled in:

ieee8021BridgeBasePortComponentId	- As per ieee8021BridgeEvbUapConfigTable
ieee8021BridgeBasePort	- As per ieee8021BridgeEvbUapConfigTable UapPortNumber
ieee8021BridgeBasePortIfIndex	- Implementation Specific Action
ieee8021BridgeBasePortDelayExceededDiscards	- Statistic, reset to 0 by creation
ieee8021BridgeBasePortMtuExceededDiscards	- Statistic, reset to 0 by creation
ieee8021BridgeBasePortCapabilties	- Implementation Specific
ieee8021BridgeBasePortTypeCapabilities	- Implementation Specific: bit *Uplink Access Port* (9) is set
ieee8021BridgeBasePortType	- Uplink Access Port
ieee8021BridgeBasePortExternal	- Implementation Specific

Insert new subclauses 17.5.3.7, 17.5.3.7.1, and 17.5.3.7.2 as shown, following all existing subclauses of 17.5.3, renumbering as necessary:

17.5.3.7 Port creation on edge relays

Creating an Uplink Relay Port implicitly creates the edge relay itself. When an Uplink Relay Port is created it is not necessarily bound to an ISS. The operating environment of the EVB system may choose to bind the URP to either an S-channel or to an ISS when it chooses. The edge relay may be created with a Downlink Relay Port or it may dynamically create them on demand from the operating environment.

Creating a Downlink Relay Port on an edge relay works the same way as creating a Port on a C-VLAN aware component (17.5.3.1) with the exception that the rules for determining the legality of the Port type are different.

17.5.3.7.1 Creating DRPs

Downlink Relay Ports are created by performing a row-create operation on the ieee8021EvbPortTable for an edge relay. The required columns are the component ID and the Port Number to use for the newly created DRP.

17.5.3.7.2 Creating URPs

URPs are created by performing a row-create operation on the ieee8021BridgeEvbUrpConfigTable of an EVB station. The required columns are the component ID and the Port Number to use for the newly created Port.

The implicitly constructed ieee8021BridgeBaseTable entry will have the following fields filled in:

ieee8021BridgeBaseComponentId	- As per ieee8021BridgeEvbUrpConfigTabl
ieee8021BridgeBaseBridgeAddress	- As per ieee8021BridgeEvbUrpConfigTabl
ieee8021BridgeBaseNumPorts	- Implementation Specific Action
ieee8021BridgeBaseComponentType	- Type edge relay
ieee8021BridgeBaseDeviceCapabilities	- Implementation specific
ieee8021BridgeBaseTrafficClassesEnabled	- Implementation Specific
ieee8021BridgeBaseMmrpEnabledStatus	- Implementation Specific
ieee8021BridgeBaseRowStatus	

The implicitly constructed ieee8021BridgeBasePortTable entry will have the following fields filled in:

ieee8021BridgeBasePortComponentId	- As per ieee8021BridgeEvbUrpConfigTabl
ieee8021BridgeBasePort	- As per ieee8021BridgeEvbUrpConfigTabl
ieee8021BridgeBasePortIfIndex	- Implementation Specific Action
ieee8021BridgeBasePortDelayExceededDiscards	- Statistic, reset to 0 by creation
ieee8021BridgeBasePortMtuExceededDiscards	- Statistic, reset to 0 by creation
ieee8021BridgeBasePortCapabilties	- Implementation Specific
ieee8021BridgeBasePortTypeCapabilities	- Implementation Specific: bit Uplink Relay Port(10) is set
ieee8021BridgeBasePortType	- Uplink Relay Port (10)
ieee8021BridgeBasePortExternal	- Implementation Specific

17.7 MIB modules

17.7.1 Definitions for the IEEE8021-TC MIB Module

Delete the entire text of 17.7.1, and insert the following text:

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

-- ================================================================
-- TEXTUAL-CONVENTIONs MIB for IEEE 802.1
-- ================================================================

IMPORTS
    MODULE-IDENTITY, Unsigned32, org
        FROM SNMPv2-SMI -- RFC 2578
    TEXTUAL-CONVENTION
        FROM SNMPv2-TC; -- RFC 2579

ieee8021TcMib MODULE-IDENTITY
    LAST-UPDATED "201202150000Z" -- February 15, 2012
    ORGANIZATION "IEEE 802.1 Working Group"
    CONTACT-INFO
        "  WG-URL: http://grouper.ieee.org/groups/802/1/index.html
         WG-EMail: stds-802-1@ieee.org

          Contact: David Levi
           Postal: C/O IEEE 802.1 Working Group
                   IEEE Standards Association
                   445 Hoes Lane
                   P.O. Box 1331
                   Piscataway
                   NJ 08855-1331
                   USA
           E-mail: STDS-802-1-L@LISTSERV.IEEE.ORG

          Contact: Kevin Nolish
           Postal: C/O IEEE 802.1 Working Group
                   IEEE Standards Association
                   445 Hoes Lane
                   P.O. Box 1331
                   Piscataway
                   NJ 08855-1331
                   USA
           E-mail: STDS-802-1-L@LISTSERV.IEEE.ORG"
    DESCRIPTION
        "Textual conventions used throughout the various IEEE 802.1 MIB
         modules.

         Unless otherwise indicated, the references in this MIB
         module are to IEEE 802.1Q-2011.

         Copyright (C) IEEE.
         This version of this MIB module is part of IEEE802.1Q;
         see the draft itself for full legal notices."

    REVISION     "201202150000Z" -- February 15, 2012
    DESCRIPTION
         "Modified IEEE8021BridgePortType textual convention to
```

```
          include stationFacingBridgePort,
          uplinkAccessPort, and uplinkRelayPort types."
    REVISION     "201108230000Z" -- August 23, 2011
    DESCRIPTION
         "Modified textual conventions  to support the IEEE 802.1
          MIBs for PBB-TE Infrastructure Protection Switching."
    REVISION     "201104060000Z" -- April 6, 2011
    DESCRIPTION
         "Modified textual conventions  to support Remote Customer
          Service Interfaces."
    REVISION     "201102270000Z" -- February 27, 2011
    DESCRIPTION
         "Minor edits to contact information etc. as part of
          2011 revision of IEEE Std 802.1Q."

    REVISION     "200811180000Z" -- November 18, 2008
    DESCRIPTION
         "Added textual conventions needed to support the IEEE 802.1
          MIBs for PBB-TE.  Additionally, some textual conventions were
          modified for the same reason."

    REVISION     "200810150000Z" -- October 15, 2008
    DESCRIPTION
         "Initial version."
    ::= { org ieee(111) standards-association-numbers-series-standards(2)
          lan-man-stds(802) ieee802dot1(1) 1 1 }

ieee802dot1mibs OBJECT IDENTIFIER
    ::= { org ieee(111) standards-association-numbers-series-standards(2)
          lan-man-stds(802) ieee802dot1(1) 1 }

-- ================================================================
-- Textual Conventions
-- ================================================================

IEEE8021PbbComponentIdentifier ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "d"
    STATUS       current
    DESCRIPTION
        "The component identifier is used to distinguish between the
        multiple virtual bridge instances within a PB or PBB.  Each
        virtual bridge instance is called a component.  In simple
        situations where there is only a single component the default
        value is 1.  The component is identified by a component
        identifier unique within the BEB and by a MAC address unique
        within the PBBN.  Each component is associated with a Backbone
        Edge Bridge (BEB) Configuration managed object."
    REFERENCE "12.3 l)"
    SYNTAX       Unsigned32 (1..4294967295)

IEEE8021PbbComponentIdentifierOrZero ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "d"
    STATUS       current
    DESCRIPTION
        "The component identifier is used to distinguish between the
        multiple virtual bridge instances within a PB or PBB.  In simple
        situations where there is only a single component the default
        value is 1.  The component is identified by a component
        identifier unique within the BEB and by a MAC address unique
```

```
        within the PBBN.  Each component is associated with a Backbone
        Edge Bridge (BEB) Configuration managed object.

        The special value '0' means 'no component identifier'.  When
        this TC is used as the SYNTAX of an object, that object must
        specify the exact meaning for this value."
    REFERENCE "12.3 l)"
    SYNTAX       Unsigned32 (0 | 1..4294967295)

IEEE8021PbbServiceIdentifier ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "d"
    STATUS       current
    DESCRIPTION
        "The service instance identifier is used at the Customer
        Backbone Port of a PBB to distinguish a service instance
        (Local-SID). If the Local-SID field is supported, it is
        used to perform a bidirectional 1:1 mapping between the
        Backbone I-SID and the Local-SID. If the Local-SID field
        is not supported, the Local-SID value is the same as the
        Backbone I-SID value."
    REFERENCE "12.16.3, 12.16.5"
    SYNTAX       Unsigned32 (256..16777214)

IEEE8021PbbServiceIdentifierOrUnassigned ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "d"
    STATUS       current
    DESCRIPTION
        "The service instance identifier is used at the Customer
        Backbone Port of a PBB to distinguish a service instance
        (Local-SID). If the Local-SID field is supported, it is
        used to perform a bidirectional 1:1 mapping between the
        Backbone I-SID and the Local-SID. If the Local-SID field
        is not supported, the Local-SID value is the same as the
        Backbone I-SID value.

        The special value of 1 indicates an unassigned I-SID."
    REFERENCE "12.16.3, 12.16.5"
    SYNTAX       Unsigned32 (1|256..16777214)

IEEE8021PbbIngressEgress ::= TEXTUAL-CONVENTION
    STATUS       current
    DESCRIPTION
        "A 2 bit selector which determines if frames on this VIP may
        ingress to the PBBN but not egress the PBBN, egress to the
        PBBN but not ingress the PBBN, or both ingress and egress
        the PBBN."
    REFERENCE "12.16.3, 12.16.5, 12.16.6"
    SYNTAX       BITS {
                     ingress(0),
                     egress(1)
                 }

IEEE8021PriorityCodePoint ::= TEXTUAL-CONVENTION
    STATUS       current
    DESCRIPTION
        "Bridge ports may encode or decode the PCP value of the
        frames that traverse the port. This textual convention
        names the possible encoding and decoding schemes that
        the port may use.  The priority and drop_eligible
```

```
        parameters are encoded in the Priority Code Point (PCP)
        field of the VLAN tag using the Priority Code Point
        Encoding Table for the Port, and they are decoded from
        the PCP using the Priority Code Point Decoding Table."
    REFERENCE "12.6.2.6"
    SYNTAX       INTEGER {
                     codePoint8p0d(1),
                     codePoint7p1d(2),
                     codePoint6p2d(3),
                     codePoint5p3d(4)
                 }

IEEE8021BridgePortNumber ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "d"
    STATUS       current
    DESCRIPTION
        "An integer that uniquely identifies a bridge port, as
        specified in 17.3.2.2 of IEEE 802.1ap.
        This value is used within the spanning tree
        protocol to identify this port to neighbor bridges."
    REFERENCE "17.3.2.2"
    SYNTAX       Unsigned32 (1..65535)

IEEE8021BridgePortNumberOrZero ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "d"
    STATUS       current
    DESCRIPTION
        "An integer that uniquely identifies a bridge port, as
        specified in 17.3.2.2 of IEEE 802.1ap.  The value 0
        means no port number, and this must be clarified in the
        DESCRIPTION clause of any object defined using this
        TEXTUAL-CONVENTION."
    REFERENCE "17.3.2.2"
    SYNTAX       Unsigned32 (0..65535)

IEEE8021BridgePortType ::= TEXTUAL-CONVENTION
    STATUS       current
    DESCRIPTION
        "A port type.  The possible port types are:

             customerVlanPort(2) - Indicates a port is a C-tag
                 aware port of an enterprise VLAN aware bridge.

             providerNetworkPort(3) - Indicates a port is an S-tag
                 aware port of a Provider Bridge or Backbone Edge
                 Bridge used for connections within a PBN or PBBN.

             customerNetworkPort(4) - Indicates a port is an S-tag
                 aware port of a Provider Bridge or Backbone Edge
                 Bridge used for connections to the exterior of a
                 PBN or PBBN.

             customerEdgePort(5) - Indicates a port is a C-tag
                 aware port of a Provider Bridge used for connections
                 to the exterior of a PBN or PBBN.

             customerBackbonePort(6) - Indicates a port is a I-tag
                 aware port of a Backbone Edge Bridge's B-component.
```

```
            virtualInstancePort(7) - Indicates a port is a virtual
                S-tag aware port within a Backbone Edge Bridge's
                I-component which is responsible for handling
                S-tagged traffic for a specific backbone service
                instance.

            dBridgePort(8) - Indicates a port is a VLAN-unaware
                member of an 802.1D bridge.

            remoteCustomerAccessPort (9) - Indicates a port is an
                S-tag aware port of a Provider Bridge used for
                connections to remote customer interface LANs
                through another PBN.

            stationFacingBridgePort (10) - Indicates a port of a
                Bridge that supports the EVB status parameters
                (6.6.5) with an EVBMode parameter value of
                EVB Bridge.

            uplinkAccessPort (11) - Indicates a port on a
                Port-mapping S-VLAN component that connects an EVB
                Bridge with an EVB station.

            uplinkRelayPort (12) - Indicates a port of an edge relay
                that supports the EVB status parameters (6.6.5)
                with an EVBMode parameter value of EVB station."
    REFERENCE   "12.16.1.1.3 h4), 12.16.2.1/2,
                12.13.1.1, 12.13.1.2, 12.15.2.1, 12.15.2.2"
    SYNTAX      INTEGER {
                    none(1),
                    customerVlanPort(2),
                    providerNetworkPort(3),
                    customerNetworkPort(4),
                    customerEdgePort(5),
                    customerBackbonePort(6),
                    virtualInstancePort(7),
                    dBridgePort(8),
                    remoteCustomerAccessPort(9),
                    stationFacingBridgePort(10),
                    uplinkAccessPort(11),
                    uplinkRelayPort(12)
                    }

IEEE8021VlanIndex ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "d"
    STATUS      current
    DESCRIPTION
        "A value used to index per-VLAN tables: values of 0 and
        4095 are not permitted.  If the value is between 1 and
        4094 inclusive, it represents an IEEE 802.1Q VLAN-ID with
        global scope within a given bridged domain (see VlanId
        textual convention).  If the value is greater than 4095,
        then it represents a VLAN with scope local to the
        particular agent, i.e., one without a global VLAN-ID
        assigned to it.  Such VLANs are outside the scope of
        IEEE 802.1Q, but it is convenient to be able to manage them
        in the same way using this MIB."
    REFERENCE   "9.6"
    SYNTAX      Unsigned32 (1..4094|4096..4294967295)
```

```
IEEE8021VlanIndexOrWildcard ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "d"
    STATUS      current
    DESCRIPTION
        "A value used to index per-VLAN tables.  The value 0 is not
        permitted, while the value 4095 represents a 'wildcard'
        value.  An object whose SYNTAX is IEEE8021VlanIndexOrWildcard
        must specify in its DESCRIPTION the specific meaning of the
        wildcard value.  If the value is between 1 and
        4094 inclusive, it represents an IEEE 802.1Q VLAN-ID with
        global scope within a given bridged domain (see VlanId
        textual convention).  If the value is greater than 4095,
        then it represents a VLAN with scope local to the
        particular agent, i.e., one without a global VLAN-ID
        assigned to it.  Such VLANs are outside the scope of
        IEEE 802.1Q, but it is convenient to be able to manage them
        in the same way using this MIB."
    REFERENCE   "9.6"
    SYNTAX      Unsigned32 (1..4294967295)

IEEE8021MstIdentifier ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "d"
    STATUS      current
    DESCRIPTION
        "In an MSTP Bridge, an MSTID, i.e., a value used to identify
        a spanning tree (or MST) instance.  In the PBB-TE environment
        the value 4094 is used to identify VIDs managed by the PBB-TE
        procedures."
    SYNTAX      Unsigned32 (1..4094)

IEEE8021ServiceSelectorType ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
       "A value that represents a type (and thereby the format)
        of a IEEE8021ServiceSelectorValue.  The value can be one of
        the following:

        ieeeReserved(0)   Reserved for definition by IEEE 802.1
                          recommend to not use zero unless
                          absolutely needed.
        vlanId(1)         12-Bit identifier as described in IEEE802.1Q.
        isid(2)           24-Bit identifier as described in IEEE802.1ah.
        tesid(3)          32 Bit identifier as described below.
        segid(4)          32 Bit identifier as described below.
        ieeeReserved(xx)  Reserved for definition by IEEE 802.1
                          xx values can be [5..7].

        To support future extensions, the IEEE8021ServiceSelectorType
        textual convention SHOULD NOT be sub-typed in object type
        definitions.  It MAY be sub-typed in compliance statements in
        order to require only a subset of these address types for a
        compliant implementation.

        The tesid is used as a service selector for MAs that are present
        in bridges that implement PBB-TE functionality.  A selector of
        this type is interpreted as a 32 bit unsigned value of type
        IEEE8021PbbTeTSidId.  This type is used to index the
        Ieee8021PbbTeTeSidTable to find the ESPs which comprise the TE
```

```
        Service Instance named by this TE-SID value.

        The segid is used as a service selector for MAs that are present
        in bridges that implement IPS functionality.  A selector of
        this type is interpreted as a 32 bit unsigned value of type
        IEEE8021TeipsSegid.  This type is used to index the
        Ieee8021TeipsSegTable to find the SMPs which comprise the
        Infrastructure Segment named by this segid value.

        Implementations MUST ensure that IEEE8021ServiceSelectorType
        objects and any dependent objects (e.g.,
        IEEE8021ServiceSelectorValue objects) are consistent.  An
        inconsistentValue error MUST be generated if an attempt to
        change an IEEE8021ServiceSelectorType object would, for
        example, lead to an undefined IEEE8021ServiceSelectorValue value."
    SYNTAX      INTEGER {
                    vlanId(1),
                    isid(2),
                    tesid(3),
                    segid(4)
                }

IEEE8021ServiceSelectorValueOrNone ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "d"
    STATUS       current
    DESCRIPTION
        "An integer that uniquely identifies a generic MAC service,
         or none.  Examples of service selectors are a VLAN-ID
         (IEEE 802.1Q) and an I-SID (IEEE 802.1ah).

         An IEEE8021ServiceSelectorValueOrNone value is always
         interpreted within the context of an
         IEEE8021ServiceSelectorType value.  Every usage of the
         IEEE8021ServiceSelectorValueOrNone textual convention is
         required to specify the IEEE8021ServiceSelectorType object
         that provides the context.  It is suggested that the
         IEEE8021ServiceSelectorType object be logically registered
         before the object(s) that use the
         IEEE8021ServiceSelectorValueOrNone textual convention, if
         they appear in the same logical row.

         The value of an IEEE8021ServiceSelectorValueOrNone object
         must always be consistent with the value of the associated
         IEEE8021ServiceSelectorType object.  Attempts to set an
         IEEE8021ServiceSelectorValueOrNone object to a value
         inconsistent with the associated
         IEEE8021ServiceSelectorType must fail with an
         inconsistentValue error.

         The special value of zero is used to indicate that no
         service selector is present or used.  This can be used in
         any situation where an object or a table entry MUST either
         refer to a specific service, or not make a selection.

         Note that a MIB object that is defined using this
         TEXTUAL-CONVENTION SHOULD clarify the meaning of
         'no service' (i.e., the special value 0), as well as the
         maximum value (i.e., 4094, for a VLAN ID)."
    SYNTAX       Unsigned32 (0 | 1..4294967295)
```

```
IEEE8021ServiceSelectorValue ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "d"
    STATUS       current
    DESCRIPTION
        "An integer that uniquely identifies a generic MAC service.
         Examples of service selectors are a VLAN-ID (IEEE 802.1Q)
         and an I-SID (IEEE 802.1ah).

         An IEEE8021ServiceSelectorValue value is always interpreted
         within the context of an IEEE8021ServiceSelectorType value.
         Every usage of the IEEE8021ServiceSelectorValue textual
         convention is required to specify the
         IEEE8021ServiceSelectorType object that provides the context.
         It is suggested that the IEEE8021ServiceSelectorType object
         be logically registered before the object(s) that use the
         IEEE8021ServiceSelectorValue textual convention, if they
         appear in the same logical row.

         The value of an IEEE8021ServiceSelectorValue object must
         always be consistent with the value of the associated
         IEEE8021ServiceSelectorType object.  Attempts to set an
         IEEE8021ServiceSelectorValue object to a value inconsistent
         with the associated IEEE8021ServiceSelectorType must fail
         with an inconsistentValue error.

         Note that a MIB object that is defined using this
         TEXTUAL-CONVENTION SHOULD clarify the
         maximum value (i.e., 4094, for a VLAN ID)."
    SYNTAX       Unsigned32 (1..4294967295)

IEEE8021PortAcceptableFrameTypes ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
        "Acceptable frame types on a port."
    REFERENCE   "12.10.1.3, 12.13.3.3, 12.13.3.4"
    SYNTAX      INTEGER {
                    admitAll(1),
                    admitUntaggedAndPriority(2),
                    admitTagged(3)
                }

IEEE8021PriorityValue ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "d"
    STATUS      current
    DESCRIPTION
        "An 802.1Q user priority value."
   REFERENCE   "12.13.3.3"
    SYNTAX      Unsigned32 (0..7)

IEEE8021PbbTeProtectionGroupId ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "d"
    STATUS       current
    DESCRIPTION
        "The PbbTeProtectionGroupId identifier is used to distinguish
         protection group instances present in the B Component of
         an IB-BEB."
    REFERENCE "12.19.2"
    SYNTAX       Unsigned32 (1..429467295)
```

```
IEEE8021PbbTeEsp ::= TEXTUAL-CONVENTION
    STATUS current
    DESCRIPTION
       "This textual convention is used to represent the logical
        components that comprise the 3-tuple that identifies an
        Ethernet Switched Path.  The 3-tuple consists of a
        destination MAC address, a source MAC address and a VID.
        Bytes (1..6) of this textual convention contain the
        ESP-MAC-DA, bytes (7..12) contain the ESP-MAC-SA, and bytes
        (13..14) contain the ESP-VID."
    REFERENCE "802.1Qay 3.2"
    SYNTAX OCTET STRING ( SIZE(14))

IEEE8021PbbTeTSidId ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "d"
    STATUS current
    DESCRIPTION
       "This textual convention is used to represent an identifier
        that refers to a TE Service Instance.  Note that, internally
        a TE-SID is implementation dependent.  This textual convention
        defines the external representation of TE-SID values."
    REFERENCE
        "802.1Qay 3.11"
    SYNTAX  Unsigned32 (1..42947295)

IEEE8021PbbTeProtectionGroupConfigAdmin ::= TEXTUAL-CONVENTION
    STATUS current
    DESCRIPTION
        "This textual convention is used to represent administrative
        commands that can be issued to a protection group.  The value
        noAdmin(1) is used to indicate that no administrative action
        is to be performed."
    REFERENCE "26.10.3.3.5
               26.10.3.3.6
               26.10.3.3.7
               12.19.2.3.2"
    SYNTAX    INTEGER {
                 clear(1),
                 lockOutProtection(2),
                 forceSwitch(3),
                 manualSwitchToProtection(4),
                 manualSwitchToWorking(5)
              }

IEEE8021PbbTeProtectionGroupActiveRequests ::= TEXTUAL-CONVENTION
   STATUS current
   DESCRIPTION
       "This textual convention is used to represent the status of
        active requests within a protection group."
    REFERENCE
       "12.19.2.1.3 d)"
   SYNTAX   INTEGER {
               noRequest(1),
               loP(2),
               fs(3),
               pSFH(4),
               wSFH(5),
               manualSwitchToProtection(6),
```

```
                manualSwitchToWorking(7)
            }

IEEE8021TeipsIpgid ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "d"
    STATUS       current
    DESCRIPTION
        "The TEIPS IPG identifier is used to distinguish
         IPG instances present in a PBB."
    REFERENCE "12.24.1.1.3 a)"
    SYNTAX       Unsigned32 (1..429467295)

IEEE8021TeipsSegid ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "d"
    STATUS current
    DESCRIPTION
       "This textual convention is used to represent an
        identifier that refers to an Infrastructure Segment.
        Note that, internally a SEG-ID implementation
        dependent.  This textual convention defines the
        external representation of SEG-ID values."
    REFERENCE
        "26.11.1"
    SYNTAX  Unsigned32 (1..42947295)

IEEE8021TeipsSmpid ::= TEXTUAL-CONVENTION
    STATUS current
    DESCRIPTION
       "This textual convention is used to represent the logical
        components that comprise the 3-tuple that identifies a
        Segment Monitoring Path (SMP).  The 3-tuple consists of a
        destination MAC address, a source MAC address and a VID.
        Bytes (1..6) of this textual convention contain the
        SMP-MAC-DA, bytes (7..12) contain the SMP-MAC-SA, and bytes
        (13..14) contain the SMP-VID."
    REFERENCE "26.11.1"
    SYNTAX OCTET STRING ( SIZE(14))

IEEE8021TeipsIpgConfigAdmin ::= TEXTUAL-CONVENTION
    STATUS current
    DESCRIPTION
        "This textual convention is used to represent administrative
        commands that can be issued to an IPG.  The value
        clear(1) is used to indicate that no administrative action
        is to be performed."
    REFERENCE "12.24.2.1.3 h)"
    SYNTAX    INTEGER {
                clear(1),
                lockOutProtection(2),
                forceSwitch(3),
                manualSwitchToProtection(4),
                manualSwitchToWorking(5)
              }

IEEE8021TeipsIpgConfigActiveRequests ::= TEXTUAL-CONVENTION
   STATUS current
   DESCRIPTION
        "This textual convention is used to represent the status of
        active requests within an IPG."
```

```
    REFERENCE
       "12.24.2.1.3 d)"
   SYNTAX   INTEGER {
                noRequest(1),
                loP(2),
                fs(3),
                pSFH(4),
                wSFH(5),
                manualSwitchToProtection(6),
                manualSwitchToWorking(7)
            }

END
```

17.7.2 Definitions for the IEEE8021-BRIDGE MIB module

Delete the entire text of 17.7.2, and insert the following text:

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

-- ================================================================
-- MIB for IEEE 802.1D devices
-- ================================================================
IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE,
    Integer32, Counter64
        FROM SNMPv2-SMI
    RowStatus, MacAddress, TruthValue, TimeInterval
        FROM SNMPv2-TC
    MODULE-COMPLIANCE, OBJECT-GROUP
        FROM SNMPv2-CONF
    ifIndex, InterfaceIndexOrZero, ifGeneralInformationGroup
        FROM IF-MIB
    ieee802dot1mibs, IEEE8021PbbComponentIdentifier,
    IEEE8021BridgePortNumber, IEEE8021PriorityCodePoint,
    IEEE8021BridgePortType, IEEE8021PriorityValue,
    IEEE8021PbbComponentIdentifierOrZero,
    IEEE8021BridgePortNumberOrZero
        FROM IEEE8021-TC-MIB
    SnmpAdminString
        FROM SNMP-FRAMEWORK-MIB
    systemGroup
        FROM SNMPv2-MIB
    ;

ieee8021BridgeMib MODULE-IDENTITY
    LAST-UPDATED "201202150000Z" -- February 15, 2012
    ORGANIZATION "IEEE 802.1 Working Group"
    CONTACT-INFO
        "  WG-URL: http://grouper.ieee.org/groups/802/1/index.html
         WG-EMail: stds-802-1@ieee.org

          Contact: David Levi
           Postal: C/O IEEE 802.1 Working Group
                   IEEE Standards Association
                   445 Hoes Lane
                   P.O. Box 1331
                   Piscataway
```

```
                    NJ 08855-1331
                    USA
            E-mail: STDS-802-1-L@LISTSERV.IEEE.ORG"
    DESCRIPTION
        "The Bridge MIB module for managing devices that support
        IEEE 802.1D.  This MIB module is derived from the IETF
        BRIDGE-MIB, RFC 4188.

        Unless otherwise indicated, the references in this MIB
        module are to IEEE Std 802.1Q-2011.

        Copyright (C) IEEE.
        This version of this MIB module is part of IEEE802.1Q;
        see the draft itself for full legal notices."

    REVISION     "201202150000Z" -- February 15, 2012
    DESCRIPTION  "Extended ieee8021BridgeBaseComponentType to
                  include erComponent and
                  ieee8021BridgeBasePortTypeCapabilities to include
                  stationFacingBridgePort, uplinkAccessPort and
                  uplinkRelayPort.
                  Added tables ieee8021BridgeBaseIfToPortTable and
                  ieee8021BridgePortTable
                  as part of IEEE Std 802.1Qbg."

    REVISION     "201104060000Z" -- April 6, 2011
    DESCRIPTION
         "Modifications to support Remote Customer Service
          Interfaces."
    REVISION     "201102270000Z" -- February 27, 2011
    DESCRIPTION
         "Minor edits to contact information etc. as part of
          2011 revision of IEEE Std 802.1Q."

    REVISION     "200810150000Z" -- October 15, 2008
    DESCRIPTION
         "Initial revision, derived from RFC 4188."
    ::= { ieee802dot1mibs 2 }

-- ================================================================
-- subtrees in the Bridge MIB
-- ================================================================

ieee8021BridgeNotifications
    OBJECT IDENTIFIER ::= { ieee8021BridgeMib 0 }

ieee8021BridgeObjects
    OBJECT IDENTIFIER ::= { ieee8021BridgeMib 1 }

ieee8021BridgeConformance
    OBJECT IDENTIFIER ::= { ieee8021BridgeMib 2 }

ieee8021BridgeBase
    OBJECT IDENTIFIER ::= { ieee8021BridgeObjects 1 }
ieee8021BridgeTp
    OBJECT IDENTIFIER ::= { ieee8021BridgeObjects 2 }
ieee8021BridgePriority
    OBJECT IDENTIFIER ::= { ieee8021BridgeObjects 3 }
ieee8021BridgeMrp
```

```
    OBJECT IDENTIFIER ::= { ieee8021BridgeObjects 4 }
ieee8021BridgeMmrp
    OBJECT IDENTIFIER ::= { ieee8021BridgeObjects 5 }
ieee8021BridgeInternalLan
    OBJECT IDENTIFIER ::= { ieee8021BridgeObjects 6 }
ieee8021BridgeDot1d
    OBJECT IDENTIFIER ::= { ieee8021BridgeObjects 7 }

-- ================================================================
-- the ieee8021BridgeBase subtree
-- ================================================================
-- Implementation of the ieee8021BridgeBase subtree is mandatory
-- for all bridges.
-- ================================================================

-- ================================================================
-- the ieee8021BridgeBaseTable
-- ================================================================
ieee8021BridgeBaseTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021BridgeBaseEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table that contains generic information about every
        bridge component.  All writable objects in this table
        MUST be persistent over power up restart/reboot."
    REFERENCE   "12.4.1"
    ::= { ieee8021BridgeBase 1 }

ieee8021BridgeBaseEntry OBJECT-TYPE
    SYNTAX      Ieee8021BridgeBaseEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A list of objects containing information for each bridge
        component."
    INDEX  { ieee8021BridgeBaseComponentId }
    ::= { ieee8021BridgeBaseTable 1 }

Ieee8021BridgeBaseEntry ::=
    SEQUENCE {
        ieee8021BridgeBaseComponentId
            IEEE8021PbbComponentIdentifier,
        ieee8021BridgeBaseBridgeAddress
            MacAddress,
        ieee8021BridgeBaseNumPorts
            Integer32,
        ieee8021BridgeBaseComponentType
            INTEGER,
        ieee8021BridgeBaseDeviceCapabilities
            BITS,
        ieee8021BridgeBaseTrafficClassesEnabled
            TruthValue,
        ieee8021BridgeBaseMmrpEnabledStatus
            TruthValue,
        ieee8021BridgeBaseRowStatus
            RowStatus
    }
```

```
ieee8021BridgeBaseComponentId 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."
    ::= { ieee8021BridgeBaseEntry 1 }

ieee8021BridgeBaseBridgeAddress OBJECT-TYPE
    SYNTAX      MacAddress
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The MAC address used by this bridge when it is
        referred to in a unique fashion.  It is recommended
        that this be the numerically smallest MAC address of
        all ports that belong to this bridge.  However, it is
        only required to be unique.  When concatenated with
        ieee8021SpanningTreePriority, a unique BridgeIdentifier
        is formed, which is used in the Spanning Tree Protocol.

        This object may not be modified while the corresponding
        instance of ieee8021BridgeBaseRowStatus is active(1).

        The value of this object MUST be retained across
        reinitializations of the management system."
    REFERENCE   "12.4.1.1.3 a)"
    ::= { ieee8021BridgeBaseEntry 2 }

ieee8021BridgeBaseNumPorts OBJECT-TYPE
    SYNTAX      Integer32
    UNITS       "ports"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of ports controlled by this bridging
        entity."
    REFERENCE   "12.4.1.1.3 c)"
    ::= { ieee8021BridgeBaseEntry 3 }

ieee8021BridgeBaseComponentType OBJECT-TYPE
    SYNTAX      INTEGER {
                    iComponent(1),
                    bComponent(2),
                    cVlanComponent(3),
                    sVlanComponent(4),
                    dBridgeComponent(5),
                    erComponent (6)
                }
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Indicates the component type(s) of this bridge.  The
        following component types are possible:

            iComponent(1) - An S-VLAN component of a Backbone
```

```
                Edge Bridge which performs encapsulation of customer
                frames.

            bComponent(2) - An S-VLAN component of a Backbone
                Edge Bridge which bundles backbone service instances
                into B-VLANs.

            cVlanComponent(3) - A C-VLAN component of an
                enterprise VLAN bridge or of a Provider Bridge used
                to process C-tagged frames.

            sVlanComponent(4) - An S-VLAN component of a
                Provider Bridge.

            dBridgeComponent(5) - A VLAN unaware component of an
                802.1D bridge.

            erComponent (6) - An Edge Relay component of an EVB Station.

        This object may not be modified while the corresponding
        instance of ieee8021BridgeBaseRowStatus is active(1).

        The value of this object MUST be retained across
        reinitializations of the management system."
    REFERENCE   "12.3 m)"
    ::= { ieee8021BridgeBaseEntry 4 }

ieee8021BridgeBaseDeviceCapabilities OBJECT-TYPE
    SYNTAX      BITS {
        dot1dExtendedFilteringServices(0),
        dot1dTrafficClasses(1),
        dot1qStaticEntryIndividualPort(2),
        dot1qIVLCapable(3),
        dot1qSVLCapable(4),
        dot1qHybridCapable(5),
        dot1qConfigurablePvidTagging(6),
        dot1dLocalVlanCapable(7)
    }
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Indicates the optional parts of IEEE 802.1D and 802.1Q
        that are implemented by this device and are manageable
        through this MIB.  Capabilities that are allowed on a
        per-port basis are indicated in
        ieee8021BridgeBasePortCapabilities.

        dot1dExtendedFilteringServices(0),
                            -- can perform filtering of
                            -- individual multicast addresses
                            -- controlled by MMRP.
        dot1dTrafficClasses(1),
                            -- can map user priority to
                            -- multiple traffic classes.
        dot1qStaticEntryIndividualPort(2),
                            -- dot1qStaticUnicastReceivePort &
                            -- dot1qStaticMulticastReceivePort
                            -- can represent non-zero entries.
        dot1qIVLCapable(3),   -- Independent VLAN Learning (IVL).
```

```
        dot1qSVLCapable(4),   -- Shared VLAN Learning (SVL).
        dot1qHybridCapable(5),
                              -- both IVL & SVL simultaneously.
        dot1qConfigurablePvidTagging(6),
                              -- whether the implementation
                              -- supports the ability to
                              -- override the default PVID
                              -- setting and its egress status
                              -- (VLAN-Tagged or Untagged) on
                              -- each port.
        dot1dLocalVlanCapable(7)
                              -- can support multiple local
                              -- bridges, outside of the scope
                              -- of 802.1Q defined VLANs.

        This object may not be modified while the corresponding
        instance of ieee8021BridgeBaseRowStatus is active(1).

        The value of this object MUST be retained across
        reinitializations of the management system."
    REFERENCE   "12.10.1.1.3 b)"
    ::= { ieee8021BridgeBaseEntry 5 }

ieee8021BridgeBaseTrafficClassesEnabled OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The value true(1) indicates that Traffic Classes are
        enabled on this bridge.  When false(2), the bridge
        operates with a single priority level for all traffic.

        This object may be modified while the corresponding
        instance of ieee8021BridgeBaseRowStatus is active(1).

        The value of this object MUST be retained across
        reinitializations of the management system."
    DEFVAL      { true }
    ::= { ieee8021BridgeBaseEntry 6 }

ieee8021BridgeBaseMmrpEnabledStatus OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The administrative status requested by management for
        MMRP.  The value true(1) indicates that MMRP should
        be enabled on this device, in all VLANs, on all ports
        for which it has not been specifically disabled.  When
        false(2), MMRP is disabled, in all VLANs and on all
        ports, and all MMRP packets will be forwarded
        transparently.  This object affects both Applicant and
        Registrar state machines.  A transition from false(2)
        to true(1) will cause a reset of all MMRP state
        machines on all ports.

        This object may be modified while the corresponding
        instance of ieee8021BridgeBaseRowStatus is active(1).
```

```
        The value of this object MUST be retained across
        reinitializations of the management system."
    DEFVAL      { true }
    ::= { ieee8021BridgeBaseEntry 7 }

ieee8021BridgeBaseRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The object indicates the status of an entry, and is used
        to create/delete entries.

        The following objects MUST be set prior to making a new
        entry active:
            ieee8021BridgeBaseBridgeAddress
            ieee8021BridgeBaseComponentType
            ieee8021BridgeBaseDeviceCapabilities
        It is recommended that these three objects not be allowed
        to be modified while the corresponding instance of
        ieee8021BridgeBaseRowStatus object is active(1).

        The following objects are not required to be set before
        making a new entry active (they will take their defaults),
        and they also may be modified while the corresponding
        instance of this object is active(1):
            ieee8021BridgeBaseTrafficClassesEnabled
            ieee8021BridgeBaseMmrpEnabledStatus

        The value of this object and all corresponding instances
        of other objects in this table MUST be retained across
        reinitializations of the management system."
    ::= { ieee8021BridgeBaseEntry 8 }

-- ============================================================
-- The Generic Bridge Port Table
-- ============================================================
ieee8021BridgePortTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021BridgePortEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table that contains generic information about every
        port that is associated with this bridge.  Transparent,
        and source-route ports are included."
    REFERENCE   "12.4.2"
    ::= { ieee8021BridgeBase 4 }

ieee8021BridgePortEntry OBJECT-TYPE
    SYNTAX      Ieee8021BridgePortEntry
    MAX-ACCESS  not-accessible
    STATUS      current

    DESCRIPTION
        "A list of objects containing information for each port
         of the bridge."
    INDEX  { ieee8021BridgePortComponentId,
             ieee8021BridgePort }
    ::= { ieee8021BridgePortTable 1 }
```

```
Ieee8021BridgeBasePortEntry ::=
    SEQUENCE {
        ieee8021BridgeBasePortComponentId
            IEEE8021PbbComponentIdentifier,
        ieee8021BridgeBasePort
            IEEE8021BridgePortNumber,
        ieee8021BridgeBasePortIfIndex
            InterfaceIndexOrZero,
        ieee8021BridgeBasePortDelayExceededDiscards
            Counter64,
        ieee8021BridgeBasePortMtuExceededDiscards
            Counter64,
        ieee8021BridgeBasePortCapabilities
            BITS,
        ieee8021BridgeBasePortTypeCapabilities
            BITS,
        ieee8021BridgeBasePortType
            IEEE8021BridgePortType,
        ieee8021BridgeBasePortExternal
            TruthValue,
        ieee8021BridgeBasePortAdminPointToPoint
            INTEGER,
        ieee8021BridgeBasePortOperPointToPoint
            TruthValue,
        ieee8021BridgeBasePortName
            SnmpAdminString
    }

ieee8021BridgeBasePortComponentId 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."
    ::= { ieee8021BridgeBasePortEntry 1 }

ieee8021BridgeBasePort 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."
    REFERENCE   "12.4.2.1.2 a)"
    ::= { ieee8021BridgeBasePortEntry 2 }

ieee8021BridgeBasePortIfIndex OBJECT-TYPE
    SYNTAX      InterfaceIndexOrZero
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
       "The value of the instance of the IfIndex object,
         defined in the IF-MIB, for the interface corresponding
         to this port, or the value 0 if the port has not been
         bound to an underlying frame source and sink.
```

```
          It is an implementation specific decision as to whether this object
          may be modified if it has been created or if 0 is a legal value.

          The underlying IfEntry indexed by this column MUST be persistent
          across reinitializations of the management system."
    ::= { ieee8021BridgeBasePortEntry 3 }

ieee8021BridgeBasePortDelayExceededDiscards OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "frames"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of frames discarded by this port due
        to excessive transit delay through the bridge.  It
        is incremented by both transparent and source
        route bridges.

        Discontinuities in the value of the counter can occur
        at re-initialization of the management system, and at
        other times as indicated by the value of
        ifCounterDiscontinuityTime object of the associated
        interface (if any)."
    REFERENCE   "12.6.1.1.3 f)"
    ::= { ieee8021BridgeBasePortEntry 4 }

ieee8021BridgeBasePortMtuExceededDiscards OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "frames"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of frames discarded by this port due
        to an excessive size.  It is incremented by both
        transparent and source route bridges.

        Discontinuities in the value of the counter can occur
        at re-initialization of the management system, and at
        other times as indicated by the value of
        ifCounterDiscontinuityTime object of the associated
        interface (if any)."
    REFERENCE   "12.6.1.1.3 g)"
    ::= { ieee8021BridgeBasePortEntry 5 }

ieee8021BridgeBasePortCapabilities OBJECT-TYPE
    SYNTAX      BITS {
        dot1qDot1qTagging(0),
        dot1qConfigurableAcceptableFrameTypes(1),
        dot1qIngressFiltering(2)
    }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Indicates the parts of IEEE 802.1D and 802.1Q that are
        optional on a per-port basis, that are implemented by
        this device, and that are manageable through this MIB.

        dot1qDot1qTagging(0), -- supports 802.1Q VLAN tagging of
```

```
                            -- frames and MVRP.
        dot1qConfigurableAcceptableFrameTypes(1),
                            -- allows modified values of
                            -- dot1qPortAcceptableFrameTypes.
        dot1qIngressFiltering(2)
                            -- supports the discarding of any
                            -- frame received on a Port whose
                            -- VLAN classification does not
                            -- include that Port in its Member
                            -- set."
    REFERENCE   "12.10.1.1.3 c)"
    ::= { ieee8021BridgeBasePortEntry 6 }

ieee8021BridgeBasePortTypeCapabilities OBJECT-TYPE
    SYNTAX      BITS {
                    customerVlanPort(0),
                    providerNetworkPort(1),
                    customerNetworkPort(2),
                    customerEdgePort(3),
                    customerBackbonePort(4),
                    virtualInstancePort(5),
                    dBridgePort(6),
                    remoteCustomerAccessPort(7),
                    stationFacingBridgePort (8),
                    uplinkAccessPort (9),
                    uplinkRelayPort(10)
                }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Indicates the capabilities of this port.  The corresponding
         instance of ieee8021BridgeBasePortType can potentially take
         any of the values for which the corresponding bit in this
         object is 1.  The possible port types are as follows:

             customerVlanPort(0) - Indicates the port can be a C-tag
                 aware port of an enterprise VLAN aware bridge.

             providerNetworkPort(1) - Indicates the port can be an
                 S-tag aware port of a Provider Bridge or Backbone
                 Edge Bridge used for connections within a PBN or
                 PBBN.

             customerNetworkPort(2) - Indicates the port can be an
                 S-tag aware port of a Provider Bridge or Backbone
                 Edge Bridge used for connections to the exterior of
                 a PBN or PBBN.

             customerEdgePort(3) - Indicates the port can be a C-tag
                 aware port of a Provider Bridge used for connections
                 to the exterior of a PBN or PBBN.

             customerBackbonePort(4) - Indicates the port can be a
                 I-tag aware port of a Backbone Edge Bridge's
                 B-component.

             virtualInstancePort(5) - Indicates the port can be a
                 virtual S-tag aware port within a Backbone Edge
                 Bridge's I-component which is responsible for
```

```
                handling S-tagged traffic for a specific backbone
                service instance.

            dBridgePort(6) - Indicates the port can be a VLAN-unaware
                member of an 802.1D bridge.

            remoteCustomerAccessPort(7) - Indicates the port can be an
                S-tag aware port of a Provider Bridge capable of providing
                Remote Customer Service Interfaces.

            stationFacingBridgePort(8) - Indicates the station-facing
                Bridge Port in a EVB Bridge.

            uplinkAccessPort(9)  -  Indicates the uplink access port
                in an EVB Bridge or EVB station.

            uplinkRelayPort (10)  - Indicates the uplink relay port
                in an EVB station."

    REFERENCE   "12.16.1.1.3 h4), 12.16.2.1/2,
                 12.13.1.1, 12.13.1.2, 12.15.2.1, 12.15.2.2,
                 12.26.2, 12.26.4.1, 12.26.5.1"
    ::= { ieee8021BridgeBasePortEntry 7 }

ieee8021BridgeBasePortType OBJECT-TYPE
    SYNTAX      IEEE8021BridgePortType
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The port type.  This value MUST be persistent over power up
         restart/reboot."
    REFERENCE   "12.16.1.1.3 h4), 12.16.2.1/2,
                 12.13.1.1, 12.13.1.2, 12.15.2.1, 12.15.2.2,
                 12.26.2, 12.26.4.1, 12.26.5.1"
    ::= { ieee8021BridgeBasePortEntry 8 }

ieee8021BridgeBasePortExternal OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "A boolean indicating whether the port is external.  A value of
        true(1) means the port is external.  A value of false(2) means
        the port is internal."
    REFERENCE   "12.16.1.1.3 h4)"
    ::= { ieee8021BridgeBasePortEntry 9 }

ieee8021BridgeBasePortAdminPointToPoint OBJECT-TYPE
    SYNTAX      INTEGER {
                    forceTrue(1),
                    forceFalse(2),
                    auto(3)
                }
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
         "For a port running spanning tree, this object represents the
         administrative point-to-point status of the LAN segment
         attached to this port, using the enumeration values of
```

```
          6.4.3.  A value of forceTrue(1) indicates
          that this port should always be treated as if it is
          connected to a point-to-point link.  A value of
          forceFalse(2) indicates that this port should be treated as
          having a shared media connection.  A value of auto(3)
          indicates that this port is considered to have a
          point-to-point link if it is an Aggregator and all of its
          members are aggregatable, or if the MAC entity
          is configured for full duplex operation, either through
          auto-negotiation or by management means.  Manipulating this
          object changes the underlying adminPointToPointMAC.

          For a VIP, the adminPointToPointMAC parameter controls
          the mechanism by which the Default Backbone Destination
          parameter for the VIP is determined.  For a backbone
          service instance that includes only 2 VIPs, the value
          may be set to forceTrue(1) which permits dynamic learning
          of the Default Backbone Destination parameter.  For a
          backbone service instance that includes more than 2 VIPs,
          the value MUST be set to ForceFalse(2) or auto(3).

          When this object is set to forceTrue(1) for a VIP, the
          Default Backbone Destination parameter is modified by
          the subsequent M_UNITDATA.indications as specified in
          6.10.1 (and described in 26.4.1).  Whenever the parameter
          is set to ForceFalse(2) or auto(3), the value for the
          Default Backbone Destination parameter is set to the
          Backbone Service Instance Group Address for the VIP-ISID.

          The value of this object MUST be retained across
          reinitializations of the management system."
      REFERENCE   "6.6.3, 6.10, 12.8.2.1.3 o), 12.8.2.3.2 f), 26.4.1"
      DEFVAL { forceFalse }
      ::= { ieee8021BridgeBasePortEntry 10 }

ieee8021BridgeBasePortOperPointToPoint OBJECT-TYPE
      SYNTAX      TruthValue
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
          "For a port running spanning tree, this object represents
          the operational point-to-point status of the LAN segment
          attached to this port.  It indicates whether a port is
          considered to have a point-to-point connection.
          If adminPointToPointMAC is set to auto(2), then the value
          of operPointToPointMAC is determined in accordance with the
          specific procedures defined for the MAC entity concerned,
          as defined in 6.5 of IEEE 802.1w.  The value is
          determined dynamically; that is, it is re-evaluated whenever
          the value of adminPointToPointMAC changes, and whenever
          the specific procedures defined for the MAC entity evaluate
          a change in its point-to-point status.

          For a VIP, this object simply reflects the value of the
          corresponding instance of ieee8021BridgeBasePortAdminPointToPoint.
          The value will be true(1) if that object is forceTrue(1), and
          the value will be false(2) if the value of that object is either
          forceFalse(2) or auto(3)."
      REFERENCE   "6.6.3, 6.10, 12.8.2.1.3 p), 12.8.2.3.2 f), 26.4.1"
```

```
    ::= { ieee8021BridgeBasePortEntry 11 }

ieee8021BridgeBasePortName OBJECT-TYPE
    SYNTAX      SnmpAdminString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
             "A  text  string  of  up  to  32  characters,  of  locally  determined
significance."
    REFERENCE   "12.4.2.1.3 a)"
    ::= { ieee8021BridgeBasePortEntry 12 }

-- ============================================================
-- The Generic Bridge ifIndex  to Port Table
-- ============================================================
ieee8021BridgeBaseIfToPortTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021BridgeBaseIfToPortEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table that contains generic information about every
        ifIndex that is associated with this bridge."
    REFERENCE   "17.2.2"
    ::= { ieee8021BridgeBase 5 }

ieee8021BridgeBaseIfToPortEntry OBJECT-TYPE
    SYNTAX      Ieee8021BridgeBaseIfToPortEntry
    MAX-ACCESS  not-accessible
    STATUS      current

    DESCRIPTION
        "A list of objects containing information for each ifIndex
         of the bridge."
    INDEX  {   ifIndex  }
    ::= { ieee8021BridgeBaseIfToPortTable 1 }

Ieee8021BridgeBaseIfToPortEntry ::=
    SEQUENCE {
        ieee8021BridgeBaseIfIndexComponentId
            IEEE8021PbbComponentIdentifier,
        ieee8021BridgeBaseIfIndexPort
            IEEE8021BridgePortNumber
     }

 ieee8021BridgeBaseIfIndexComponentId     OBJECT-TYPE
     SYNTAX      IEEE8021PbbComponentIdentifier
     MAX-ACCESS  read-only
     STATUS      current
     DESCRIPTION
         "The component ID for this ifIndex."
     ::= { ieee8021BridgeBaseIfToPortEntry 1 }

 ieee8021BridgeBaseIfIndexPort            OBJECT-TYPE
     SYNTAX      IEEE8021BridgePortNumber
     MAX-ACCESS  read-only
     STATUS      current
     DESCRIPTION
         "The port for this ifIndex."
     ::= { ieee8021BridgeBaseIfToPortEntry 2 }
```

```
-- ================================================================
-- port number table section 12.5.1
-- ================================================================

ieee8021BridgePhyPortTable OBJECT-TYPE
   SYNTAX      SEQUENCE OF Ieee8021BridgePhyPortEntry
   MAX-ACCESS  not-accessible
   STATUS      current
   DESCRIPTION
   "A table that contains ISS port number to bridge
    componentID and port number mapping."
   REFERENCE   "12.5.1"
   ::= { ieee8021BridgeBase 6}

ieee8021BridgePhyPortEntry OBJECT-TYPE
   SYNTAX      Ieee8021BridgePhyPortEntry
   MAX-ACCESS  not-accessible
   STATUS      current
   DESCRIPTION
   "A list of objects containing mapping for ISS port
    numbers to bridge componentID and port numbers "
   INDEX  { ieee8021BridgePhyPort }
   ::= { ieee8021BridgePhyPortTable 1 }

   Ieee8021BridgePhyPortEntry ::=
   SEQUENCE {
         ieee8021BridgePhyPort
                  IEEE8021BridgePortNumber,
         ieee8021BridgePhyPortIfIndex
                  InterfaceIndexOrZero,
         ieee8021BridgePhyMacAddress
                  MacAddress,
         ieee8021BridgePhyPortToComponentId
                  IEEE8021PbbComponentIdentifierOrZero,
         ieee8021BridgePhyPortToInternalPort
                  IEEE8021BridgePortNumberOrZero
   }


ieee8021BridgePhyPort  OBJECT-TYPE
   SYNTAX       IEEE8021BridgePortNumber
   MAX-ACCESS   not-accessible
   STATUS       current
   DESCRIPTION
   "The ISS port."
   REFERENCE   "12.26"
   ::= { ieee8021BridgePhyPortEntry 1 }


ieee8021BridgePhyPortIfIndex  OBJECT-TYPE
    SYNTAX      InterfaceIndexOrZero
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
       "The value of the instance of the IfIndex object,
        defined in the IF-MIB, for the interface corresponding
        to this port, or the value 0 if the port has not been
```

```
        bound to an underlying frame source and sink.

        The underlying IfEntry indexed by this column MUST
        be persistent across reinitializations of the
        management system."
    ::= { ieee8021BridgePhyPortEntry 2 }

ieee8021BridgePhyMacAddress OBJECT-TYPE
   SYNTAX      MacAddress
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION  "The mac address"
   ::= { ieee8021BridgePhyPortEntry 3 }

ieee8021BridgePhyPortToComponentId OBJECT-TYPE
   SYNTAX      IEEE8021PbbComponentIdentifierOrZero
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
   "The component ID that this ISS port belongs to."

   ::= { ieee8021BridgePhyPortEntry 4 }


ieee8021BridgePhyPortToInternalPort   OBJECT-TYPE
   SYNTAX      IEEE8021BridgePortNumberOrZero
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
   "The port number to which this ISS port maps to."

   ::= { ieee8021BridgePhyPortEntry 5 }

-- ================================================================
-- the ieee8021BridgeTp subtree
-- ================================================================
-- This is implemented by those bridges that support the
-- transparent bridging mode.  A transparent bridge will
-- implement this subtree.
-- ================================================================

-- ================================================================
--  Port Table for Transparent Bridges
-- ================================================================

ieee8021BridgeTpPortTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021BridgeTpPortEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table that contains information about every port that
        is associated with this transparent bridge."
    REFERENCE   "12.4.2, C.4"
    ::= { ieee8021BridgeTp 1 }

ieee8021BridgeTpPortEntry OBJECT-TYPE
    SYNTAX      Ieee8021BridgeTpPortEntry
    MAX-ACCESS  not-accessible
    STATUS      current
```

```
    DESCRIPTION
        "A list of objects containing information for each port of
         a transparent bridge."
    INDEX   { ieee8021BridgeTpPortComponentId,
              ieee8021BridgeTpPort }
    ::= { ieee8021BridgeTpPortTable 1 }

Ieee8021BridgeTpPortEntry ::=
    SEQUENCE {
        ieee8021BridgeTpPortComponentId
            IEEE8021PbbComponentIdentifier,
        ieee8021BridgeTpPort
            IEEE8021BridgePortNumber,
        ieee8021BridgeTpPortMaxInfo
            Integer32,
        ieee8021BridgeTpPortInFrames
            Counter64,
        ieee8021BridgeTpPortOutFrames
            Counter64,
        ieee8021BridgeTpPortInDiscards
            Counter64
    }

ieee8021BridgeTpPortComponentId 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."
    ::= { ieee8021BridgeTpPortEntry 1 }

ieee8021BridgeTpPort OBJECT-TYPE
    SYNTAX      IEEE8021BridgePortNumber
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The port number of the port for which this entry
        contains Transparent bridging management information."
    ::= { ieee8021BridgeTpPortEntry 2 }

ieee8021BridgeTpPortMaxInfo OBJECT-TYPE
    SYNTAX      Integer32
    UNITS       "bytes"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The maximum size of the INFO (non-MAC) field that
        this port will receive or transmit."
    ::= { ieee8021BridgeTpPortEntry 3 }

ieee8021BridgeTpPortInFrames OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "frames"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
```

```
        "The number of frames that have been received by this
        port from its segment.  Note that a frame received on the
        interface corresponding to this port is only counted by
        this object if and only if it is for a protocol being
        processed by the local bridging function, including
        bridge management frames.

        Discontinuities in the value of the counter can occur
        at re-initialization of the management system, and at
        other times as indicated by the value of
        ifCounterDiscontinuityTime object of the associated
        interface (if any)."
    REFERENCE   "12.6.1.1.3 a)"
    ::= { ieee8021BridgeTpPortEntry 4 }

ieee8021BridgeTpPortOutFrames OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "frames"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of frames that have been transmitted by this
        port to its segment.  Note that a frame transmitted on
        the interface corresponding to this port is only counted
        by this object if and only if it is for a protocol being
        processed by the local bridging function, including
        bridge management frames.

        Discontinuities in the value of the counter can occur
        at re-initialization of the management system, and at
        other times as indicated by the value of
        ifCounterDiscontinuityTime object of the associated
        interface (if any)."
    REFERENCE   "12.6.1.1.3 d)"
    ::= { ieee8021BridgeTpPortEntry 5 }

ieee8021BridgeTpPortInDiscards OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "frames"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Count of received valid frames that were discarded
        (i.e., filtered) by the Forwarding Process.

        Discontinuities in the value of the counter can occur
        at re-initialization of the management system, and at
        other times as indicated by the value of
        ifCounterDiscontinuityTime object of the associated
        interface (if any)."
    REFERENCE   "12.6.1.1.3 c)"
    ::= { ieee8021BridgeTpPortEntry 6 }

-- ================================================================
-- the ieee8021BridgePriority subtree
-- ================================================================

-- ================================================================
-- Port Priority Table
```

```
-- ================================================================

ieee8021BridgePortPriorityTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021BridgePortPriorityEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table that contains information about every port that
        is associated with this transparent bridge."
    ::= { ieee8021BridgePriority 1 }

ieee8021BridgePortPriorityEntry OBJECT-TYPE
    SYNTAX      Ieee8021BridgePortPriorityEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A list of Default User Priorities for each port of a
        transparent bridge.  This is indexed by
        ieee8021BridgeBasePortComponentId and
        ieee8021BridgeBasePort."
    AUGMENTS { ieee8021BridgeBasePortEntry }
    ::= { ieee8021BridgePortPriorityTable 1 }

Ieee8021BridgePortPriorityEntry ::=
    SEQUENCE {
        ieee8021BridgePortDefaultUserPriority
            IEEE8021PriorityValue,
        ieee8021BridgePortNumTrafficClasses
            Integer32,
        ieee8021BridgePortPriorityCodePointSelection
            IEEE8021PriorityCodePoint,
        ieee8021BridgePortUseDEI
            TruthValue,
        ieee8021BridgePortRequireDropEncoding
            TruthValue,
        ieee8021BridgePortServiceAccessPrioritySelection
            TruthValue
    }

ieee8021BridgePortDefaultUserPriority OBJECT-TYPE
    SYNTAX      IEEE8021PriorityValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The default ingress User Priority for this port.  This
        only has effect on media, such as Ethernet, that do not
        support native User Priority.

        The value of this object MUST be retained across
        reinitializations of the management system."
    ::= { ieee8021BridgePortPriorityEntry 1 }

ieee8021BridgePortNumTrafficClasses OBJECT-TYPE
    SYNTAX      Integer32 (1..8)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The number of egress traffic classes supported on this
        port.  This object may optionally be read-only.
```

```
        The value of this object MUST be retained across
        reinitializations of the management system."
    ::= { ieee8021BridgePortPriorityEntry 2 }

ieee8021BridgePortPriorityCodePointSelection OBJECT-TYPE
    SYNTAX      IEEE8021PriorityCodePoint
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        " This object identifies the rows in the PCP encoding and
          decoding tables that are used to remark frames on this
          port if this remarking is enabled."
    REFERENCE   "12.6.2.6, 12.6.2.7"
    ::= { ieee8021BridgePortPriorityEntry 3 }

ieee8021BridgePortUseDEI OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "If the Use_DEI is set to true(1) for the Port then the
        drop_eligible parameter is encoded in the DEI of transmitted
        frames, and the drop_eligible parameter shall be true(1) for a
        received frame if the DEI is set in the VLAN tag or the Priority
        Code Point Decoding Table indicates drop_eligible True for
        the received PCP value. If the Use_DEI parameter is false(2),
        the DEI shall be transmitted as zero and ignored on receipt.
        The default value of the Use_DEI parameter is false(2)."
    REFERENCE   "12.6.2.12, 12.6.2.13"
    ::= { ieee8021BridgePortPriorityEntry 4 }

ieee8021BridgePortRequireDropEncoding OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "If a Bridge supports encoding or decoding of drop_eligible
        from the PCP field of a VLAN tag (6.7.3) on any of its Ports,
        then it shall implement a Boolean parameter Require Drop
        Encoding on each of its Ports with default value false(2). If
        Require Drop Encoding is True and the Bridge Port cannot
        encode particular priorities with drop_eligible, then frames
        queued with those priorities and drop_eligible true(1) shall
        be discarded and not transmitted."
    REFERENCE   "12.6.2.14, 12.6.2.15"
    DEFVAL { false }
    ::= { ieee8021BridgePortPriorityEntry 5 }

ieee8021BridgePortServiceAccessPrioritySelection OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "Indication of if the Service Access Priority Selection
        function is supported on the Customer Bridge Port to request
        priority handling of the frame from a Port-based service
        interface."
    REFERENCE   "12.6.2.16, 12.6.2.17"
```

```
    ::= { ieee8021BridgePortPriorityEntry 6 }

-- ============================================================
-- User Priority Regeneration Table
-- ============================================================

ieee8021BridgeUserPriorityRegenTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021BridgeUserPriorityRegenEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A list of Regenerated User Priorities for each received
        User Priority on each port of a bridge.  The Regenerated
        User Priority value may be used to index the Traffic
        Class Table for each input port.  This only has effect
        on media that support native User Priority.  The default
        values for Regenerated User Priorities are the same as
        the User Priorities."
    REFERENCE   "6.5"
    ::= { ieee8021BridgePriority 2 }

ieee8021BridgeUserPriorityRegenEntry OBJECT-TYPE
    SYNTAX      Ieee8021BridgeUserPriorityRegenEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A mapping of incoming User Priority to a Regenerated
        User Priority."
    INDEX   { ieee8021BridgeBasePortComponentId,
              ieee8021BridgeBasePort,
              ieee8021BridgeUserPriority }
    ::= { ieee8021BridgeUserPriorityRegenTable 1 }

Ieee8021BridgeUserPriorityRegenEntry ::=
    SEQUENCE {
        ieee8021BridgeUserPriority
            IEEE8021PriorityValue,
        ieee8021BridgeRegenUserPriority
            IEEE8021PriorityValue
    }

ieee8021BridgeUserPriority OBJECT-TYPE
    SYNTAX      IEEE8021PriorityValue
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The User Priority for a frame received on this port."
    ::= { ieee8021BridgeUserPriorityRegenEntry 1 }

ieee8021BridgeRegenUserPriority OBJECT-TYPE
    SYNTAX      IEEE8021PriorityValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The Regenerated User Priority that the incoming User
        Priority is mapped to for this port.

        The value of this object MUST be retained across
        reinitializations of the management system."
```

```
    ::= { ieee8021BridgeUserPriorityRegenEntry 2 }

-- ================================================================
-- Traffic Class Table
-- ================================================================

ieee8021BridgeTrafficClassTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021BridgeTrafficClassEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table mapping evaluated User Priority to Traffic
        Class, for forwarding by the bridge.  Traffic class is a
        number in the range (0..(ieee8021BridgePortNumTrafficClasses-1))."
    REFERENCE   "Table 8-3"
    ::= { ieee8021BridgePriority 3 }

ieee8021BridgeTrafficClassEntry OBJECT-TYPE
    SYNTAX      Ieee8021BridgeTrafficClassEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "User Priority to Traffic Class mapping."
    INDEX   { ieee8021BridgeBasePortComponentId,
              ieee8021BridgeBasePort,
              ieee8021BridgeTrafficClassPriority }
    ::= { ieee8021BridgeTrafficClassTable 1 }

Ieee8021BridgeTrafficClassEntry ::=
    SEQUENCE {
        ieee8021BridgeTrafficClassPriority
            IEEE8021PriorityValue,
        ieee8021BridgeTrafficClass
            Integer32
    }

ieee8021BridgeTrafficClassPriority OBJECT-TYPE
    SYNTAX      IEEE8021PriorityValue
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The Priority value determined for the received frame.
        This value is equivalent to the priority indicated in
        the tagged frame received, or one of the evaluated
        priorities, determined according to the media-type.
        For untagged frames received from Ethernet media, this
        value is equal to the ieee8021BridgePortDefaultUserPriority value
        for the ingress port.

        For untagged frames received from non-Ethernet media,
        this value is equal to the ieee8021BridgeRegenUserPriority value
        for the ingress port and media-specific user priority."
    ::= { ieee8021BridgeTrafficClassEntry 1 }

ieee8021BridgeTrafficClass OBJECT-TYPE
    SYNTAX      Integer32 (0..7)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
```

```
        "The Traffic Class the received frame is mapped to.

        The value of this object MUST be retained across
        reinitializations of the management system."
    ::= { ieee8021BridgeTrafficClassEntry 2 }

-- ============================================================
-- Outbound Access Priority Table
-- ============================================================

ieee8021BridgePortOutboundAccessPriorityTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021BridgePortOutboundAccessPriorityEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table mapping Regenerated User Priority to Outbound
        Access Priority.  This is a fixed mapping for all port
        types, with two options for 802.5 Token Ring, and three
        options for 802.17 RPR."
    REFERENCE   "Table 8-3"
    ::= { ieee8021BridgePriority 4 }

ieee8021BridgePortOutboundAccessPriorityEntry OBJECT-TYPE
    SYNTAX      Ieee8021BridgePortOutboundAccessPriorityEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Regenerated User Priority to Outbound Access Priority
        mapping."
    INDEX   { ieee8021BridgeBasePortComponentId,
              ieee8021BridgeBasePort,
              ieee8021BridgeRegenUserPriority }
    ::= { ieee8021BridgePortOutboundAccessPriorityTable 1 }

Ieee8021BridgePortOutboundAccessPriorityEntry ::=
    SEQUENCE {
        ieee8021BridgePortOutboundAccessPriority
            IEEE8021PriorityValue
    }

ieee8021BridgePortOutboundAccessPriority OBJECT-TYPE
    SYNTAX      IEEE8021PriorityValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The Outbound Access Priority the received frame is
        mapped to."
    ::= { ieee8021BridgePortOutboundAccessPriorityEntry 1 }

-- ============================================================
-- ieee8021BridgePortDecodingTable:
-- ============================================================

ieee8021BridgePortDecodingTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021BridgePortDecodingEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table that contains information about Priority Code
```

```
        Point Decoding Table for a Port of a provider bridge.
        Alternative values for each table are specified as rows
        in Table 6-4 (6.7.3), with each alternative labeled by
        the number of distinct priorities that can be communicated,
        and the number of these for which drop precedence can
        be communicated.  All writable objects in this table MUST
        be persistent over power up restart/reboot."
    ::= { ieee8021BridgePriority 5 }

ieee8021BridgePortDecodingEntry OBJECT-TYPE
    SYNTAX      Ieee8021BridgePortDecodingEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A list of objects containing Priority Code Point Decoding
        information for a port of a provider bridge."
    INDEX { ieee8021BridgePortDecodingComponentId,
            ieee8021BridgePortDecodingPortNum,
            ieee8021BridgePortDecodingPriorityCodePointRow,
            ieee8021BridgePortDecodingPriorityCodePoint }
    ::= { ieee8021BridgePortDecodingTable 1 }

Ieee8021BridgePortDecodingEntry ::= SEQUENCE {
      ieee8021BridgePortDecodingComponentId
          IEEE8021PbbComponentIdentifier,
      ieee8021BridgePortDecodingPortNum
          IEEE8021BridgePortNumber,
      ieee8021BridgePortDecodingPriorityCodePointRow
          IEEE8021PriorityCodePoint,
      ieee8021BridgePortDecodingPriorityCodePoint
          Integer32,
      ieee8021BridgePortDecodingPriority
          IEEE8021PriorityValue,
      ieee8021BridgePortDecodingDropEligible
          TruthValue
}

ieee8021BridgePortDecodingComponentId 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."
    ::= { ieee8021BridgePortDecodingEntry 1 }

ieee8021BridgePortDecodingPortNum OBJECT-TYPE
    SYNTAX      IEEE8021BridgePortNumber
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A unique identifier of a port controlled by this VLAN
        bridging entity."
    ::= { ieee8021BridgePortDecodingEntry 2 }

ieee8021BridgePortDecodingPriorityCodePointRow OBJECT-TYPE
    SYNTAX      IEEE8021PriorityCodePoint
```

```
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The specific row in Table 6-3 (6.7.3) indicating the PCP."
    ::= { ieee8021BridgePortDecodingEntry 3 }

ieee8021BridgePortDecodingPriorityCodePoint OBJECT-TYPE
    SYNTAX      Integer32 (0..7)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The specific PCP value in Table 6-3 (6.7.3)."
    ::= { ieee8021BridgePortDecodingEntry 4 }

ieee8021BridgePortDecodingPriority OBJECT-TYPE
    SYNTAX      IEEE8021PriorityValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The specific priority value in Table 6-3 (6.7.3)."
    REFERENCE   "12.6.2.8, 12.6.2.9"
    ::= { ieee8021BridgePortDecodingEntry 5 }

ieee8021BridgePortDecodingDropEligible OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The drop eligibility value in Table 6-3 (6.7.3)."
    REFERENCE   "12.6.2.8, 12.6.2.9"
    ::= { ieee8021BridgePortDecodingEntry 6 }

-- ============================================================
-- ieee8021BridgePortEncodingTable:
-- ============================================================

ieee8021BridgePortEncodingTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021BridgePortEncodingEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table that contains information about Priority Code
        Point Decoding Table for a Port of a provider bridge.
        Alternative values for each table are specified as rows
        in Table 6-3 (6.7.3), with each alternative labeled by
        the number of distinct priorities that can be communicated,
        and the number of these for which drop precedence can be
        communicated.  All writable objects in this table MUST be
        persistent over power up restart/reboot."
    ::= { ieee8021BridgePriority 6 }

ieee8021BridgePortEncodingEntry OBJECT-TYPE
    SYNTAX      Ieee8021BridgePortEncodingEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A list of objects containing Priority Code Point Encoding
        information for a port of a provider bridge."
    INDEX { ieee8021BridgePortEncodingComponentId,
```

```
            ieee8021BridgePortEncodingPortNum,
            ieee8021BridgePortEncodingPriorityCodePointRow,
            ieee8021BridgePortEncodingPriorityCodePoint,
            ieee8021BridgePortEncodingDropEligible }
    ::= { ieee8021BridgePortEncodingTable 1 }

Ieee8021BridgePortEncodingEntry ::= SEQUENCE {
      ieee8021BridgePortEncodingComponentId
          IEEE8021PbbComponentIdentifier,
      ieee8021BridgePortEncodingPortNum
          IEEE8021BridgePortNumber,
      ieee8021BridgePortEncodingPriorityCodePointRow
          IEEE8021PriorityCodePoint,
      ieee8021BridgePortEncodingPriorityCodePoint
          Integer32,
      ieee8021BridgePortEncodingDropEligible
          TruthValue,
      ieee8021BridgePortEncodingPriority
          IEEE8021PriorityValue
}

ieee8021BridgePortEncodingComponentId 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."
    ::= { ieee8021BridgePortEncodingEntry 1 }

ieee8021BridgePortEncodingPortNum OBJECT-TYPE
    SYNTAX      IEEE8021BridgePortNumber
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A unique identifier of a port controlled by this VLAN bridging
        entity."
    ::= { ieee8021BridgePortEncodingEntry 2 }

ieee8021BridgePortEncodingPriorityCodePointRow OBJECT-TYPE
    SYNTAX      IEEE8021PriorityCodePoint
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The specific row in Table 6-3 (6.7.3)indicating the PCP row.
         (i.e. 8P0D, 7P1D, 6P2D, 5P3D)"
    ::= { ieee8021BridgePortEncodingEntry 3 }

ieee8021BridgePortEncodingPriorityCodePoint OBJECT-TYPE
    SYNTAX      Integer32 (0..7)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The specific row in Table 6-3 (6.7.3) indicating the PCP.
         (i.e., 0,1,2,3,4,5,6,7)."
    ::= { ieee8021BridgePortEncodingEntry 4 }
```

```
ieee8021BridgePortEncodingDropEligible OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The specific row in Table 6-3 (6.7.3) indicating the drop
         eligibility.  A value of true(1) means eligible for drop."
    ::= { ieee8021BridgePortEncodingEntry 5 }

ieee8021BridgePortEncodingPriority OBJECT-TYPE
    SYNTAX      IEEE8021PriorityValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The encoding priority in Table 6-3 (6.7.3)."
    REFERENCE   "12.6.2.10, 12.6.2.11"
    ::= { ieee8021BridgePortEncodingEntry 6 }

-- ================================================================
-- ieee8021BridgeServiceAccessPriorityTable:
-- ================================================================

ieee8021BridgeServiceAccessPriorityTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021BridgeServiceAccessPriorityEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table that contains information about the  Service Access
        Priority Selection function for a provider bridge.  The use
        of this table enables a mechanism for a Customer Bridge
        attached to a Provider Bridged Network to request priority
        handling of frames.  All writable objects in this table MUST
        be persistent over power up restart/reboot."
    ::= { ieee8021BridgePriority 7 }

ieee8021BridgeServiceAccessPriorityEntry OBJECT-TYPE
    SYNTAX      Ieee8021BridgeServiceAccessPriorityEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A list of objects containing information about the Service
         Access Priority Selection function for a provider bridge."
    INDEX { ieee8021BridgeServiceAccessPriorityComponentId,
            ieee8021BridgeServiceAccessPriorityPortNum,
            ieee8021BridgeServiceAccessPriorityReceived }
    ::= { ieee8021BridgeServiceAccessPriorityTable 1 }

Ieee8021BridgeServiceAccessPriorityEntry ::= SEQUENCE {
      ieee8021BridgeServiceAccessPriorityComponentId
          IEEE8021PbbComponentIdentifier,
      ieee8021BridgeServiceAccessPriorityPortNum
          IEEE8021BridgePortNumber,
      ieee8021BridgeServiceAccessPriorityReceived
          IEEE8021PriorityValue,
      ieee8021BridgeServiceAccessPriorityValue
          IEEE8021PriorityValue
}

ieee8021BridgeServiceAccessPriorityComponentId 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."
    ::= { ieee8021BridgeServiceAccessPriorityEntry 1 }

ieee8021BridgeServiceAccessPriorityPortNum OBJECT-TYPE
    SYNTAX      IEEE8021BridgePortNumber
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A unique identifier of a port controlled by this VLAN bridging
        entity."
    ::= { ieee8021BridgeServiceAccessPriorityEntry 2 }

ieee8021BridgeServiceAccessPriorityReceived OBJECT-TYPE
    SYNTAX      IEEE8021PriorityValue
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The default received priority value in Table 6-3 (6.7.3).
         (i.e., 0,1,2,3,4,5,6,7)"
    ::= { ieee8021BridgeServiceAccessPriorityEntry 3 }

ieee8021BridgeServiceAccessPriorityValue OBJECT-TYPE
    SYNTAX      IEEE8021PriorityValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The regenerated priority value in Table 6-3 (6.7.3).
         (i.e., 0,1,2,3,4,5,6,7)"
    REFERENCE   "12.6.2.18, 12.6.2.19"
    ::= { ieee8021BridgeServiceAccessPriorityEntry 4 }

-- ================================================================
-- the ieee8021BridgeMrp subtree
-- ================================================================

-- ================================================================
-- The MRP Port Table
-- ================================================================

ieee8021BridgePortMrpTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021BridgePortMrpEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table of MRP control information about every bridge
        port.  This is indexed by ieee8021BridgeBasePortComponentId
        and ieee8021BridgeBasePort."
    ::= { ieee8021BridgeMrp 1 }

ieee8021BridgePortMrpEntry OBJECT-TYPE
    SYNTAX      Ieee8021BridgePortMrpEntry
    MAX-ACCESS  not-accessible
```

```
    STATUS      current
    DESCRIPTION
        "MRP control information for a bridge port."
    AUGMENTS { ieee8021BridgeBasePortEntry }
    ::= { ieee8021BridgePortMrpTable 1 }

Ieee8021BridgePortMrpEntry ::=
    SEQUENCE {
        ieee8021BridgePortMrpJoinTime
            TimeInterval,
        ieee8021BridgePortMrpLeaveTime
            TimeInterval,
        ieee8021BridgePortMrpLeaveAllTime
            TimeInterval
    }

ieee8021BridgePortMrpJoinTime OBJECT-TYPE
    SYNTAX      TimeInterval
    UNITS       "centi-seconds"
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The MRP Join time, in centiseconds.

        The value of this object MUST be retained across
        reinitializations of the management system."
    DEFVAL      { 20 }
    ::= { ieee8021BridgePortMrpEntry 1 }

ieee8021BridgePortMrpLeaveTime OBJECT-TYPE
    SYNTAX      TimeInterval
    UNITS       "centi-seconds"
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The MRP Leave time, in centiseconds.

        The value of this object MUST be retained across
        reinitializations of the management system."
    DEFVAL      { 60 }
    ::= { ieee8021BridgePortMrpEntry 2 }

ieee8021BridgePortMrpLeaveAllTime OBJECT-TYPE
    SYNTAX      TimeInterval
    UNITS       "centi-seconds"
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The MRP LeaveAll time, in centiseconds.

        The value of this object MUST be retained across
        reinitializations of the management system."
    DEFVAL      { 1000 }
    ::= { ieee8021BridgePortMrpEntry 3 }

-- ============================================================
-- The MMRP Port Configuration and Status Table
-- ============================================================
```

```
ieee8021BridgePortMmrpTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021BridgePortMmrpEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table of MMRP control and status information about
        every bridge port.  Augments the ieee8021BridgeBasePortTable."
    ::= { ieee8021BridgeMmrp 1 }

ieee8021BridgePortMmrpEntry OBJECT-TYPE
    SYNTAX      Ieee8021BridgePortMmrpEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "MMRP control and status information for a bridge port."
    AUGMENTS { ieee8021BridgeBasePortEntry }
    ::= { ieee8021BridgePortMmrpTable 1 }

Ieee8021BridgePortMmrpEntry ::=
    SEQUENCE {
        ieee8021BridgePortMmrpEnabledStatus
            TruthValue,
        ieee8021BridgePortMmrpFailedRegistrations
            Counter64,
        ieee8021BridgePortMmrpLastPduOrigin
            MacAddress,
        ieee8021BridgePortRestrictedGroupRegistration
            TruthValue
    }

ieee8021BridgePortMmrpEnabledStatus OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The administrative state of MMRP operation on this port.  The
        value true(1) indicates that MMRP is enabled on this port
        in all VLANs as long as ieee8021BridgeMmrpEnabledStatus is
        also true(1).  A value of false(2) indicates that MMRP is
        disabled on this port in all VLANs: any MMRP packets received
        will be silently discarded, and no MMRP registrations will be
        propagated from other ports.  Setting this to a value of
        true(1) will be stored by the agent but will only take
        effect on the MMRP protocol operation if
        ieee8021BridgeMmrpEnabledStatus
        also indicates the value true(1).  This object affects
        all MMRP Applicant and Registrar state machines on this
        port.  A transition from false(2) to true(1) will
        cause a reset of all MMRP state machines on this port.

        The value of this object MUST be retained across
        reinitializations of the management system."
    DEFVAL      { true }
    ::= { ieee8021BridgePortMmrpEntry 1 }

ieee8021BridgePortMmrpFailedRegistrations OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "failed MMRP registrations"
    MAX-ACCESS  read-only
```

```
    STATUS      current
    DESCRIPTION
        "The total number of failed MMRP registrations, for any
        reason, in all VLANs, on this port."
    ::= { ieee8021BridgePortMmrpEntry 2 }

ieee8021BridgePortMmrpLastPduOrigin OBJECT-TYPE
    SYNTAX      MacAddress
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The Source MAC Address of the last MMRP message
        received on this port."
    ::= { ieee8021BridgePortMmrpEntry 3 }

ieee8021BridgePortRestrictedGroupRegistration OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The state of Restricted Group Registration on this port.
         If the value of this control is true(1), then creation
         of a new dynamic entry is permitted only if there is a
         Static Filtering Entry for the VLAN concerned, in which
         the Registrar Administrative Control value is Normal
         Registration.

        The value of this object MUST be retained across
        reinitializations of the management system."
    REFERENCE   "11.2.3.2.3, 12.11.1.3"
    DEFVAL      { false }
    ::= { ieee8021BridgePortMmrpEntry 4 }

-- ============================================================
-- I-LAN Interface configuration table
-- ============================================================

ieee8021BridgeILanIfTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021BridgeILanIfEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
         "This table is a sparse augmentation of ifTable and controls
          the creation of the I-LAN Interface.  An I-LAN Interface is
          used to create internal connections between bridge ports in a
          802.1 device.  An I-LAN Interfaces can be directly associated
          with a set of bridge ports.  An I-LAN Interfaces can also be
          used as a stacking interface to relate other interfaces before
          association to bridge ports.

          For example, an I-LAN interface can be created to link traffic
          between a PIP and a CBP.  In this case a CBP is created on the
          B-Component and the CBP's related IfEntry is stacked upon the
          IfEntry of the I-LAN.  The PIP is stacked upon the I-LAN using
          the IfStackTable.  Finally, a VIP is created on the I-Component
          and is associated with the PIP, thus completing the path from
          the I-Component's MAC relay to the CBP on the B-Component.

          Entries in this table MUST be persistent over power up
```

```
          restart/reboot."
    REFERENCE   "17.3.2.2"
    ::= { ieee8021BridgeInternalLan 1 }

ieee8021BridgeILanIfEntry OBJECT-TYPE
    SYNTAX      Ieee8021BridgeILanIfEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Each entry consists of a Row Status to control creation."
    INDEX       { ifIndex }
    ::= { ieee8021BridgeILanIfTable 1 }

Ieee8021BridgeILanIfEntry ::=
    SEQUENCE {
        ieee8021BridgeILanIfRowStatus
            RowStatus
    }

ieee8021BridgeILanIfRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This object is used to create and delete entries in this
         table and the Interface table."
    ::= { ieee8021BridgeILanIfEntry 1 }

-- ============================================================
-- 802.1D Dynamic Port Creation table
-- ============================================================

ieee8021BridgeDot1dPortTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021BridgeDot1dPortEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
         "This table provides the capability to dynamically create and
          delete 802.1D bridge ports.  Each entry in this table MUST
          have a corresponding entry in the ieee8021BridgeBasePortTable.

          Entries in this table MUST be persistent over power up
          restart/reboot."
    REFERENCE   "17.5.3"
    ::= { ieee8021BridgeDot1d 1 }

ieee8021BridgeDot1dPortEntry OBJECT-TYPE
    SYNTAX      Ieee8021BridgeDot1dPortEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Each entry consists of a Row Status to control creation."
    INDEX { ieee8021BridgeBasePortComponentId,
            ieee8021BridgeBasePort }
    ::= { ieee8021BridgeDot1dPortTable 1 }

Ieee8021BridgeDot1dPortEntry ::=
    SEQUENCE {
        ieee8021BridgeDot1dPortRowStatus
```

```
            RowStatus
    }

ieee8021BridgeDot1dPortRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This object is used to create and delete entries in this
         table and the ieee8021BridgeBasePortTable."
    ::= { ieee8021BridgeDot1dPortEntry 1 }


-- ================================================================
-- IEEE 802.1D MIB - Conformance Information
-- ================================================================

ieee8021BridgeCompliances
    OBJECT IDENTIFIER ::= { ieee8021BridgeConformance 1 }
ieee8021BridgeGroups
    OBJECT IDENTIFIER ::= { ieee8021BridgeConformance 2 }

-- ================================================================
-- units of conformance
-- ================================================================

-- ================================================================
-- the ieee8021BridgeBase group
-- ================================================================

ieee8021BridgeBaseBridgeGroup OBJECT-GROUP
    OBJECTS {
        ieee8021BridgeBaseBridgeAddress,
        ieee8021BridgeBaseNumPorts,
        ieee8021BridgeBaseComponentType
    }
    STATUS      current
    DESCRIPTION
        "Bridge level information for this device."
    ::= { ieee8021BridgeGroups 1 }

ieee8021BridgeBasePortGroup OBJECT-GROUP
    OBJECTS {
        ieee8021BridgeBasePortIfIndex,
        ieee8021BridgeBasePortDelayExceededDiscards,
        ieee8021BridgeBasePortMtuExceededDiscards,
        ieee8021BridgeBasePortType,
        ieee8021BridgeBasePortExternal,
        ieee8021BridgeBasePortAdminPointToPoint,
        ieee8021BridgeBasePortOperPointToPoint,
        ieee8021BridgeBasePortName
    }
    STATUS      current
    DESCRIPTION
        "Information for each port on this device."
    ::= { ieee8021BridgeGroups 2 }

ieee8021BridgeCapGroup OBJECT-GROUP
    OBJECTS {
```

```
        ieee8021BridgeBaseDeviceCapabilities,
        ieee8021BridgeBasePortCapabilities,
        ieee8021BridgeBasePortTypeCapabilities
    }
    STATUS      current
    DESCRIPTION
        "A collection of objects indicating the optional
        capabilities of the device."
    ::= { ieee8021BridgeGroups 3 }

ieee8021BridgeDeviceMmrpGroup OBJECT-GROUP
    OBJECTS {
        ieee8021BridgeBaseMmrpEnabledStatus
    }
    STATUS      current
    DESCRIPTION
        "A collection of objects providing device-level control
        for the Multicast Filtering extended bridge services."
    ::= { ieee8021BridgeGroups 4 }

-- ================================================================
-- the ieee8021BridgeTp group
-- ================================================================

ieee8021BridgeTpPortGroup OBJECT-GROUP
    OBJECTS {
        ieee8021BridgeTpPortMaxInfo,
        ieee8021BridgeTpPortInFrames,
        ieee8021BridgeTpPortOutFrames,
        ieee8021BridgeTpPortInDiscards
    }
    STATUS      current
    DESCRIPTION
        "Dynamic Filtering Database information for each port of
        the Bridge."
    ::= { ieee8021BridgeGroups 6 }

-- ================================================================
-- Bridge Priority groups
-- ================================================================

ieee8021BridgeDevicePriorityGroup OBJECT-GROUP
    OBJECTS {
        ieee8021BridgeBaseTrafficClassesEnabled
    }
    STATUS      current
    DESCRIPTION
        "A collection of objects providing device-level control
        for the Priority services."
    ::= { ieee8021BridgeGroups 7 }

ieee8021BridgeDefaultPriorityGroup OBJECT-GROUP
    OBJECTS {
        ieee8021BridgePortDefaultUserPriority,
        ieee8021BridgePortPriorityCodePointSelection,
        ieee8021BridgePortUseDEI,
        ieee8021BridgePortRequireDropEncoding,
        ieee8021BridgePortServiceAccessPrioritySelection
    }
```

```
    STATUS      current
    DESCRIPTION
        "A collection of objects defining the User Priority
        applicable to each port for media that do not support
        native User Priority."
    ::= { ieee8021BridgeGroups 8 }

ieee8021BridgeRegenPriorityGroup OBJECT-GROUP
    OBJECTS {
        ieee8021BridgeRegenUserPriority
    }
    STATUS      current
    DESCRIPTION
        "A collection of objects defining the User Priorities
        applicable to each port for media that support native
        User Priority."
    ::= { ieee8021BridgeGroups 9 }

ieee8021BridgePriorityGroup OBJECT-GROUP
    OBJECTS {
        ieee8021BridgePortNumTrafficClasses,
        ieee8021BridgeTrafficClass
    }
    STATUS      current
    DESCRIPTION
        "A collection of objects defining the traffic classes
        within a bridge for each evaluated User Priority."
    ::= { ieee8021BridgeGroups 10 }

ieee8021BridgeAccessPriorityGroup OBJECT-GROUP
    OBJECTS {
        ieee8021BridgePortOutboundAccessPriority
    }
    STATUS      current
    DESCRIPTION
        "A collection of objects defining the media-dependent
        outbound access level for each priority."
    ::= { ieee8021BridgeGroups 11 }

ieee8021BridgePortMrpGroup OBJECT-GROUP
    OBJECTS {
        ieee8021BridgePortMrpJoinTime,
        ieee8021BridgePortMrpLeaveTime,
        ieee8021BridgePortMrpLeaveAllTime
    }
    STATUS      current
    DESCRIPTION
        "A collection of objects providing port level control
        and status information for MRP operation."
    ::= { ieee8021BridgeGroups 12 }

ieee8021BridgePortMmrpGroup OBJECT-GROUP
    OBJECTS {
        ieee8021BridgePortMmrpEnabledStatus,
        ieee8021BridgePortMmrpFailedRegistrations,
        ieee8021BridgePortMmrpLastPduOrigin,
        ieee8021BridgePortRestrictedGroupRegistration
    }
    STATUS      deprecated
```

```
    DESCRIPTION
        "A collection of objects providing port level control
        and status information for MMRP operation."
    ::= { ieee8021BridgeGroups 13 }

ieee8021BridgePortDecodingGroup OBJECT-GROUP
    OBJECTS {
        ieee8021BridgePortDecodingPriority,
        ieee8021BridgePortDecodingDropEligible
    }
    STATUS      current
    DESCRIPTION
        "A collection of objects providing statistics counters for
        decoding priority and drop eligibility for bridge ports."
    ::= { ieee8021BridgeGroups 14 }

ieee8021BridgePortEncodingGroup OBJECT-GROUP
    OBJECTS {
        ieee8021BridgePortEncodingPriority
    }
    STATUS      current
    DESCRIPTION
        "A collection of objects providing statistics counters for
        encoding priority and drop eligibility for bridge ports."
    ::= { ieee8021BridgeGroups 15 }

ieee8021BridgeServiceAccessPriorityGroup OBJECT-GROUP
    OBJECTS {
        ieee8021BridgeServiceAccessPriorityValue
    }
    STATUS      current
    DESCRIPTION
        "A collection of objects providing statistics
        counters for service access priority."
    ::= { ieee8021BridgeGroups 16 }

-- ================================================================
-- Internal LAN group
-- ================================================================

ieee8021BridgeInternalLANGroup OBJECT-GROUP
    OBJECTS {
        ieee8021BridgeILanIfRowStatus
    }
    STATUS      current
    DESCRIPTION
        "A collection of objects providing control of internal
        LAN configuration."
    ::= { ieee8021BridgeGroups 17 }

-- ================================================================
-- Bridge Creation Group
-- ================================================================

ieee8021BridgeCreatableBaseBridgeGroup OBJECT-GROUP
    OBJECTS {
        ieee8021BridgeBaseRowStatus
    }
    STATUS      current
```

```
    DESCRIPTION
        "Controls the managment system directed creation of
         Bridge Components."
    ::= { ieee8021BridgeGroups 18 }

-- ================================================================
-- Dot1d Dynamic Port Creation group
-- ================================================================

ieee8021BridgeDot1dDynamicPortCreationGroup OBJECT-GROUP
    OBJECTS {
        ieee8021BridgeDot1dPortRowStatus
    }
    STATUS      current
    DESCRIPTION
        "A collection of objects providing dynamic creation and
        deletion of 802.1D bridge ports."
    ::= { ieee8021BridgeGroups 19 }


-- ================================================================
-- Bridge interface index to port table group
-- ================================================================

ieee8021BridgeBaseIfToPortGroup OBJECT-GROUP
    OBJECTS {
    ieee8021BridgeBaseIfIndexComponentId,
    ieee8021BridgeBaseIfIndexPort

    }
    STATUS      current
    DESCRIPTION
        "A collection of objects providing a map between interface
        index and component ID and bridge ports."
    ::= { ieee8021BridgeGroups 20 }



-- ================================================================
-- Bridge interface index to component group
-- ================================================================
ieee8021BridgePhyPortGroup   OBJECT-GROUP
   OBJECTS {
   ieee8021BridgePhyPortIfIndex,
   ieee8021BridgePhyMacAddress,
   ieee8021BridgePhyPortToComponentId,
   ieee8021BridgePhyPortToInternalPort
   }
   STATUS      current
   DESCRIPTION
   "The collection of objects used to represent a ISS port management objects."
   ::= { ieee8021BridgeGroups 21 }


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

ieee8021BridgeCompliance1 MODULE-COMPLIANCE
```

```
    STATUS      current
    DESCRIPTION
        "The compliance statement for devices supporting bridging
        services as defined in 802.1D-2004.  Such devices support
        path cost values of 32-bits, and bridge and port priority
        values are more restricted than in 802.1D-1995.

        Full support for the 802.1D management objects requires
        implementation of the objects listed in the systemGroup
        from the SNMPv2-MIB [RFC3418], as well as the objects
        listed in the ifGeneralInformationGroup from the
        IF-MIB [RFC2863]."

    MODULE SNMPv2-MIB -- The SNMPv2-MIB, RFC 3418
        MANDATORY-GROUPS {
            systemGroup
        }

    MODULE IF-MIB -- The interfaces MIB, RFC 2863
        MANDATORY-GROUPS {
            ifGeneralInformationGroup
        }

    MODULE
        MANDATORY-GROUPS {
            ieee8021BridgeBaseBridgeGroup,
            ieee8021BridgeBasePortGroup
        }

    GROUP ieee8021BridgeCreatableBaseBridgeGroup
    DESCRIPTION
        "Implementation of this group is mandatory for
         bridges that allow management systems to add and delete
         bridge components.  Provider Backbone Edge Bridges would
         typically fall in this category."

    GROUP   ieee8021BridgeTpPortGroup
    DESCRIPTION
        "Implementation of this group is mandatory for
        bridges that support the transparent bridging
        mode.  A transparent bridge will implement
        this group."

    GROUP   ieee8021BridgeInternalLANGroup
    DESCRIPTION
        "Implementation of this group is optional.  It can be supported
        to provide control over the relationship between interfaces and
        bridge ports where such relationships are more complex than a
        simple 1-to-1 mapping."

    GROUP   ieee8021BridgeDot1dDynamicPortCreationGroup
    DESCRIPTION
        "Implementation of this group is optional.  It can be supported
        to provide the ability to dynamically create and deleted 802.1D
        bridge ports."

    GROUP  ieee8021BridgeBaseIfToPortGroup
    DESCRIPTION
         "A collection of objects providing a map between interface
```

```
        index and component ID and bridge ports."
    GROUP   ieee8021BridgePhyPortGroup
    DESCRIPTION
         "A colelction of objects providing a map between port numbers
         to the component id, interface index."

    ::= { ieee8021BridgeCompliances 3 }


ieee8021BridgeCompliance MODULE-COMPLIANCE
    STATUS      current
    DESCRIPTION
        "The compliance statement for devices supporting bridging
        services as defined in 802.1D-2004.  Such devices support
        path cost values of 32-bits, and bridge and port priority
        values are more restricted than in 802.1D-1995.

        Full support for the 802.1D management objects requires
        implementation of the objects listed in the systemGroup
        from the SNMPv2-MIB [RFC3418], as well as the objects
        listed in the ifGeneralInformationGroup from the
        IF-MIB [RFC2863]."

    MODULE SNMPv2-MIB -- The SNMPv2-MIB, RFC 3418
        MANDATORY-GROUPS {
            systemGroup
        }

    MODULE IF-MIB -- The interfaces MIB, RFC 2863
        MANDATORY-GROUPS {
            ifGeneralInformationGroup
        }

    MODULE
        MANDATORY-GROUPS {
            ieee8021BridgeBaseBridgeGroup,
            ieee8021BridgeBasePortGroup
        }

    GROUP ieee8021BridgeCreatableBaseBridgeGroup
    DESCRIPTION
        "Implementation of this group is mandatory for
         bridges that allow management systems to add and delete
         bridge components.  Provider Backbone Edge Bridges would
         typically fall in this category."

    GROUP   ieee8021BridgeTpPortGroup
    DESCRIPTION
        "Implementation of this group is mandatory for
        bridges that support the transparent bridging
        mode.  A transparent bridge will implement
        this group."

    GROUP   ieee8021BridgeInternalLANGroup
    DESCRIPTION
        "Implementation of this group is optional.  It can be supported
        to provide control over the relationship between interfaces and
        bridge ports where such relationships are more complex than a
        simple 1-to-1 mapping."
```

```
    GROUP   ieee8021BridgeDot1dDynamicPortCreationGroup
    DESCRIPTION
        "Implementation of this group is optional.  It can be supported
        to provide the ability to dynamically create and deleted 802.1D
        bridge ports."

    ::= { ieee8021BridgeCompliances 1 }

ieee8021BridgePriorityAndMulticastFilteringCompliance MODULE-COMPLIANCE
    STATUS  deprecated
    DESCRIPTION
        "The compliance statement for device support of Priority
        and Multicast Filtering extended bridging services."

    MODULE
        MANDATORY-GROUPS { ieee8021BridgeCapGroup }

        GROUP       ieee8021BridgeDeviceMmrpGroup
        DESCRIPTION
            "This group is mandatory for devices supporting the MMRP
            application, defined by IEEE 802.1D Extended Filtering
            Services."

        GROUP       ieee8021BridgeDevicePriorityGroup
        DESCRIPTION
            "This group is mandatory only for devices supporting
            the priority forwarding operations defined by IEEE
            802.1D."

        GROUP       ieee8021BridgeDefaultPriorityGroup
        DESCRIPTION
            "This group is mandatory only for devices supporting
            the priority forwarding operations defined by the
            extended bridge services with media types, such as
            Ethernet, that do not support native User Priority."

        GROUP       ieee8021BridgeRegenPriorityGroup
        DESCRIPTION
            "This group is mandatory only for devices supporting
            the priority forwarding operations defined by IEEE 802.1D
            and that have interface media types that support
            native User Priority, e.g., IEEE 802.5."

        GROUP       ieee8021BridgePriorityGroup
        DESCRIPTION
            "This group is mandatory only for devices supporting
            the priority forwarding operations defined by IEEE 802.1D."

        GROUP       ieee8021BridgeAccessPriorityGroup
        DESCRIPTION
            "This group is optional and is relevant only for devices
            supporting the priority forwarding operations defined by
            IEEE 802.1D and that have interface media types that
            support native Access Priority, e.g., IEEE 802.5."

        GROUP       ieee8021BridgePortMrpGroup
        DESCRIPTION
            "This group is mandatory for devices supporting any
```

```
            of the MRP applications: e.g., MMRP, defined by the
            extended filtering services of 802.1D; or MVRP,
            defined by 802.1Q (refer to the Q-BRIDGE-MIB for
            conformance statements for MVRP).”

        GROUP       ieee8021BridgePortMmrpGroup
        DESCRIPTION
            “This group is mandatory for devices supporting the
            MMRP application, as defined by IEEE 802.1D Extended
            Filtering Services.”

        GROUP       ieee8021BridgePortDecodingGroup
        DESCRIPTION
            “This group is optional and supports Priority Code Point
            Decoding Table for a Port of a provider bridge.”

        GROUP       ieee8021BridgePortEncodingGroup
        DESCRIPTION
            “This group is optional and supports Priority Code Point
            Encoding Table for a Port of a provider bridge.”

        GROUP       ieee8021BridgeServiceAccessPriorityGroup
        DESCRIPTION
            “This group is optional and supports Priority Code Point
            Encoding Table for a Port of a provider bridge.”

        OBJECT      ieee8021BridgePortNumTrafficClasses
        MIN-ACCESS  read-only
        DESCRIPTION
            “Write access is not required.”

        OBJECT      ieee8021BridgeTrafficClass
        MIN-ACCESS  read-only
        DESCRIPTION
            “Write access is not required.”

        OBJECT      ieee8021BridgeRegenUserPriority
        MIN-ACCESS  read-only
        DESCRIPTION
            “Write access is not required.”

       ::= { ieee8021BridgeCompliances 2 }

END
```

Insert new subclause 17.7.20 as shown, following all existing subclauses of 17.7, renumbering as necessary:

17.7.20 Definitions of the IEEE8021-EVB MIB module

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

-- ================================================================
-- MIB for EVB Bridges and EVB Stations
-- ================================================================
IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE,
    Integer32,Counter32,Unsigned32, TimeTicks
        FROM SNMPv2-SMI
    MacAddress, TruthValue, RowStatus, StorageType
        FROM SNMPv2-TC

    ieee802dot1mibs, IEEE8021PbbComponentIdentifier,
    IEEE8021BridgePortNumber
        FROM IEEE8021-TC-MIB
    VlanIndex
        FROM Q-BRIDGE-MIB
    InterfaceIndexOrZero
        FROM IF-MIB
    ieee8021BridgePhyPort
        FROM IEEE8021-BRIDGE-MIB
    MODULE-COMPLIANCE, OBJECT-GROUP
        FROM SNMPv2-CONF;

ieee8021BridgeEvbMib  MODULE-IDENTITY
    LAST-UPDATED "201202150000Z" -- February 15, 2012
    ORGANIZATION "IEEE 802.1 Working Group"
    CONTACT-INFO
         "WG-URL: http://www.ieee802.org/1
          WG-EMail: stds-802-1@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@LISTSERV.IEEE.ORG"
    DESCRIPTION
        "The EVB MIB module for managing devices that support
        Ethernet Virtual Bridging.

        Unless otherwise indicated, the references in this MIB
        module are to IEEE Std 802.1Q-2011.

        Copyright (C) IEEE.
        This version of this MIB module is part of IEEE802.1Q;
        see the draft itself for full legal notices."
    REVISION     "201202150000Z" -- February 15, 2012
    DESCRIPTION
         "Initial version published in IEEE Std 802.1Qbg."
    ::= { ieee802dot1mibs 24 }
```

```
-- ================================================================
-- subtrees in the EVB MIB
-- ================================================================

ieee8021BridgeEvbNotifications
    OBJECT IDENTIFIER ::= { ieee8021BridgeEvbMib 0 }

ieee8021BridgeEvbObjects
    OBJECT IDENTIFIER ::= { ieee8021BridgeEvbMib 1 }

ieee8021BridgeEvbConformance
    OBJECT IDENTIFIER ::= { ieee8021BridgeEvbMib 2 }

--  ==========================
--  EVB Bridge managed object
--  ==========================

ieee8021BridgeEvbSys  OBJECT IDENTIFIER ::=   { ieee8021BridgeEvbObjects 1  }

ieee8021BridgeEvbSysType    OBJECT-TYPE
   SYNTAX        INTEGER {
                  evbBridge (1),
                  evbStation (2)
                 }
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION   "The evbSysType determines if this is an EVB Bridge
                  or EVB station."
   REFERENCE     "5.22,5.23"

   ::= { ieee8021BridgeEvbSys 1}

ieee8021BridgeEvbSysNumExternalPorts  OBJECT-TYPE
   SYNTAX        Unsigned32 (1..4095)
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION   "The evbSysNumExternalPorts parameter indicates how
                  many externally accessible port are available."
   REFERENCE     "12.4.2, 12.5.1"
   ::= {ieee8021BridgeEvbSys 2}

ieee8021BridgeEvbSysEvbLldpTxEnable  OBJECT-TYPE
   SYNTAX        TruthValue
   MAX-ACCESS    read-write
   STATUS        current
   DESCRIPTION   "This object is used to initialize the LLDP EVB
                  objects for new SBPs and URPS.
                  When set to 'true' a new SBP or URP will place the local
                  EVB objects in the LLDP nearest Customer database;
                  when set to 'false' a new SBP or URP will not place
                  the local EVB objects in the LLDP database."
   REFERENCE     "D.2.13"
   DEFVAL        { true }
   ::= {ieee8021BridgeEvbSys 3}


ieee8021BridgeEvbSysEvbLldpManual  OBJECT-TYPE
   SYNTAX        TruthValue
```

```
   MAX-ACCESS    read-write
   STATUS        current
   DESCRIPTION   "This object is used to initialize the LLDP EVB
                  objects for new SBPs and URPS.
                  When set to 'false' the operating configuration
                  will be determined by the comparison between
                  the local and remote LLDP EVB objects
                  (automatic), regardless of the setting of
                  ieee8021BridgeEvbSysLldpTxEnable.
                  When ieee8021BridgeEvbSysLldpManual is 'true' the
                  configuration will be determined by the setting
                  of the local EVB objects only (manual)."
   REFERENCE     "D.2.13"
   DEFVAL        { false }
   ::= {ieee8021BridgeEvbSys 4}

ieee8021BridgeEvbSysEvbLldpGidCapable  OBJECT-TYPE
   SYNTAX        TruthValue
   MAX-ACCESS    read-write
   STATUS        current
   DESCRIPTION   "The value of this object is used as the default
                  value of the BGID or SGID bit of the EVB LLDP TLV string."
   REFERENCE     "D.2.13"
   ::= {ieee8021BridgeEvbSys 5}

ieee8021BridgeEvbSysEcpAckTimer  OBJECT-TYPE
   SYNTAX        Integer32
   MAX-ACCESS    read-write
   STATUS        current
   DESCRIPTION
        "A value indicating the  Bridge Proposed ECP ackTimer."
   REFERENCE
       "D.2.13.6, 43.3.6.1"
   ::= { ieee8021BridgeEvbSys 6 }

ieee8021BridgeEvbSysEcpMaxRetries  OBJECT-TYPE
    SYNTAX       Integer32 (0..7)
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
    "A value indicating the Bridge ECP maxRetries."
    REFERENCE
    "D.2.13.5, 43.3.7.4"
    DEFVAL       { 3 }
    ::= { ieee8021BridgeEvbSys 7 }


ieee8021BridgeEvbSysVdpDfltRsrcWaitDelay OBJECT-TYPE
    SYNTAX       Integer32
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
    "A value indicating the  Bridge Resource VDP Timeout."
    REFERENCE   "D.2.13, 41.5.5.7"
    ::= { ieee8021BridgeEvbSys 8 }


ieee8021BridgeEvbSysVdpDfltReinitKeepAlive OBJECT-TYPE
    SYNTAX       Integer32
```

```
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
    "A value indicating the Bridge Proposed VDP Keep Alive Timeout."
    REFERENCE "D.2.13, 41.4.5.5"
    ::= { ieee8021BridgeEvbSys 9 }

-- ======================================
-- Station facing bridge port table
-- ======================================

ieee8021BridgeEvbSbpTable OBJECT-TYPE
     SYNTAX      SEQUENCE OF Ieee8021BridgeEvbSbpEntry
     MAX-ACCESS  not-accessible
     STATUS      current
     DESCRIPTION
         "A table that contains Station-facing Bridge Port (SBP)
          details."
     REFERENCE   "12.26.2"
     ::= { ieee8021BridgeEvbSys 10}

 ieee8021BridgeEvbSbpEntry OBJECT-TYPE
     SYNTAX      Ieee8021BridgeEvbSbpEntry
     MAX-ACCESS  not-accessible
     STATUS      current
     DESCRIPTION
         "A list of objects describing SBP."
     INDEX  { ieee8021BridgeEvbSbpComponentID,
              ieee8021BridgeEvbSbpPortNumber
            }
     ::= { ieee8021BridgeEvbSbpTable 1 }

 Ieee8021BridgeEvbSbpEntry ::=
     SEQUENCE {
      ieee8021BridgeEvbSbpComponentID
                  IEEE8021PbbComponentIdentifier,
      ieee8021BridgeEvbSbpPortNumber
                  IEEE8021BridgePortNumber,
      ieee8021BridgeEvbSbpLldpManual            TruthValue,
      ieee8021BridgeEvbSbpVdpOperRsrcWaitDelay   Unsigned32,
      ieee8021BridgeEvbSbpVdpOperReinitKeepAlive Unsigned32,
      ieee8021BridgeEvbSbpVdpOperToutKeepAlive   Unsigned32
    }

ieee8021BridgeEvbSbpComponentID  OBJECT-TYPE
   SYNTAX        IEEE8021PbbComponentIdentifier
   MAX-ACCESS    not-accessible
   STATUS        current
   DESCRIPTION
   "The SBP component ID"
   REFERENCE   "12.4.1.5"
   ::= { ieee8021BridgeEvbSbpEntry 1 }

ieee8021BridgeEvbSbpPortNumber OBJECT-TYPE
   SYNTAX        IEEE8021BridgePortNumber
   MAX-ACCESS    not-accessible
   STATUS        current
   DESCRIPTION   "The SBP port number."
   REFERENCE     "12.4.2"
```

```
   ::= { ieee8021BridgeEvbSbpEntry 2 }

ieee8021BridgeEvbSbpLldpManual   OBJECT-TYPE
   SYNTAX        TruthValue
   MAX-ACCESS    read-write
   STATUS        current
   DESCRIPTION
   "The evbSbpLldpManual parameter switches EVB TLVs to manual mode.
    In manual mode the running parameters are determined solely from
    the local LLDP database values."
   ::= { ieee8021BridgeEvbSbpEntry 3 }

ieee8021BridgeEvbSbpVdpOperRsrcWaitDelay     OBJECT-TYPE
   SYNTAX       Unsigned32
   UNITS        "micro-seconds"
   MAX-ACCESS   read-only
   STATUS       current
   DESCRIPTION  "The value used to initialize the waitWhile timer
                (41.4.5.7) by the station VDP state machine when
                the state machine is waiting for a response."
   REFERENCE  "D.2.13, 41.5.5.7"
   ::= { ieee8021BridgeEvbSbpEntry 4 }


ieee8021BridgeEvbSbpVdpOperReinitKeepAlive   OBJECT-TYPE
   SYNTAX       Unsigned32
   UNITS        "micro-seconds"
   MAX-ACCESS   read-only
   STATUS       current
   DESCRIPTION  "The value used to initialize the waitWhile timer
                 (41.4.5.5) by the station VDP state machine in
                 order to determine when to transmit a keep alive
                 message."
   REFERENCE  "D.2.13, 41.5.5.5"
   ::= { ieee8021BridgeEvbSbpEntry 5 }

ieee8021BridgeEvbSbpVdpOperToutKeepAlive     OBJECT-TYPE
   SYNTAX       Unsigned32
   UNITS        "micro-seconds"
   MAX-ACCESS   read-only
   STATUS       current
   DESCRIPTION  "The value used to initialize the waitWhile timer
                 (41.4.5.13) by the EVBCB VDP state machine in order to
                 determine when to transmit a keep alive message."
   REFERENCE  "D.2.13, 41.5.5.13"
   ::= { ieee8021BridgeEvbSbpEntry 6 }

-- ============================
-- VSI Database
-- ============================

ieee8021BridgeEvbVSIDBObjects OBJECT IDENTIFIER ::= { ieee8021BridgeEvbObjects 2
}

ieee8021BridgeEvbVSIDBTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021BridgeEvbVSIDBEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
```

```
        "A table that contains database of the active Virtual Station
         Interfaces."
    REFERENCE   "12.26.3"
    ::= { ieee8021BridgeEvbVSIDBObjects 1}

ieee8021BridgeEvbVSIDBEntry OBJECT-TYPE
    SYNTAX      Ieee8021BridgeEvbVSIDBEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A list of objects containing database of the active Virtual Station
         Interfaces."
    INDEX  { ieee8021BridgeEvbVSIComponentID,
             ieee8021BridgeEvbVSIPortNumber,
             ieee8021BridgeEvbVSIIDType,
             ieee8021BridgeEvbVSIID
           }
    ::= { ieee8021BridgeEvbVSIDBTable 1 }

Ieee8021BridgeEvbVSIDBEntry ::=
    SEQUENCE {
      ieee8021BridgeEvbVSIComponentID
                IEEE8021PbbComponentIdentifier,
      ieee8021BridgeEvbVSIPortNumber
                IEEE8021BridgePortNumber,
      ieee8021BridgeEvbVSIIDType           INTEGER,
      ieee8021BridgeEvbVSIID               OCTET STRING,
      ieee8021BridgeEvbVSITimeSinceCreate  Unsigned32,
      ieee8021BridgeEvbVsiVdpOperCmd       INTEGER,
      ieee8021BridgeEvbVsiOperRevert       TruthValue,
      ieee8021BridgeEvbVsiOperHard         TruthValue,
      ieee8021BridgeEvbVsiOperReason       BITS,
      ieee8021BridgeEvbVSIMgrID            OCTET STRING,
      ieee8021BridgeEvbVSIType             Integer32,
      ieee8021BridgeEvbVSITypeVersion      OCTET STRING,
      ieee8021BridgeEvbVSIMvFormat         INTEGER,
      ieee8021BridgeEvbVSINumMACs          Integer32,
      ieee8021BridgeEvbVDPMachineState     INTEGER,
      ieee8021BridgeEvbVDPCommandsSucceeded Counter32,
      ieee8021BridgeEvbVDPCommandsFailed    Counter32,
      ieee8021BridgeEvbVDPCommandReverts    Counter32,
      ieee8021BridgeEvbVDPCounterDiscontinuity TimeTicks
    }

ieee8021BridgeEvbVSIComponentID  OBJECT-TYPE
   SYNTAX      IEEE8021PbbComponentIdentifier
   MAX-ACCESS  not-accessible
   STATUS      current
   DESCRIPTION "The evbVSIComponentID is the ComponentID for the
                C-VLAN component of the EVB Bridge or for the edge
                relay of the EVB station."
   REFERENCE "12.4.1.5"
   ::= { ieee8021BridgeEvbVSIDBEntry 1}

ieee8021BridgeEvbVSIPortNumber  OBJECT-TYPE
   SYNTAX      IEEE8021BridgePortNumber
   MAX-ACCESS  not-accessible
   STATUS      current
   DESCRIPTION "The evbVSIPortNumber is the Port Number for the SBP
```

```
                or URP where the VSI is accessed."
   REFERENCE "12.4.2"
   ::= { ieee8021BridgeEvbVSIDBEntry 2 }

ieee8021BridgeEvbVSIIDType  OBJECT-TYPE
   SYNTAX         INTEGER{
                     vsiidIpv4 (1),
                     vsiidIpv6 (2),
                     vsiidMAC (3),
                     vsiidLocal (4),
                     vsiidUUID (5)
                  }
   MAX-ACCESS     not-accessible
   STATUS         current
   DESCRIPTION
   "This object specifies the VSIID Type for the VSIID in the DCN "
   REFERENCE   "41.2.6"
   ::= { ieee8021BridgeEvbVSIDBEntry 3 }


ieee8021BridgeEvbVSIID  OBJECT-TYPE
   SYNTAX         OCTET STRING (SIZE (16))
   MAX-ACCESS     not-accessible
   STATUS         current
   DESCRIPTION
   "This object specifies the VSIID that uniquely identifies the VSI
    in the DCN "
   REFERENCE   "41.2.7"
   ::= { ieee8021BridgeEvbVSIDBEntry 4 }


ieee8021BridgeEvbVSITimeSinceCreate OBJECT-TYPE
   SYNTAX        Unsigned32
   UNITS        "centi-seconds"
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
      "This object specifies the time since creation "
   REFERENCE   "41"
   ::= { ieee8021BridgeEvbVSIDBEntry 5 }


ieee8021BridgeEvbVsiVdpOperCmd OBJECT-TYPE
   SYNTAX     INTEGER
              {
               preAssociate (1),
               preAssociateWithRsrcReservation (2),
               associate (3),
               deAssociate (4)
              }
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
   "This object identifies the type of TLV."
   REFERENCE   "41.2.1"
   ::= { ieee8021BridgeEvbVSIDBEntry 6 }

ieee8021BridgeEvbVsiOperRevert   OBJECT-TYPE
   SYNTAX       TruthValue
```

```
   MAX-ACCESS   read-only
   STATUS       current
   DESCRIPTION "The evbOperRevert status indicator shows the most
                recent value of the KEEP indicator from the VDP
                protocol exchange."
   REFERENCE     "41.2.3"
   ::= { ieee8021BridgeEvbVSIDBEntry 7 }

ieee8021BridgeEvbVsiOperHard   OBJECT-TYPE
   SYNTAX       TruthValue
   MAX-ACCESS   read-only
   STATUS       current
   DESCRIPTION "The evbVsiHard status indicator shows the most
                recent value of the HARD indicator from the VDP
                protocol exchange."
   REFERENCE     "41.2.3"

::= { ieee8021BridgeEvbVSIDBEntry 8 }

ieee8021BridgeEvbVsiOperReason  OBJECT-TYPE
   SYNTAX        BITS
                 {
                   success (0),
                   invalidFormat (1),
                   insufficientResources (2),
                   otherfailure(3)
                }
   MAX-ACCESS   read-only
   STATUS       current
   DESCRIPTION
   "This object indicates the outcome of a request."
   REFERENCE     "41.2.3"


   ::= { ieee8021BridgeEvbVSIDBEntry 9 }


ieee8021BridgeEvbVSIMgrID      OBJECT-TYPE
   SYNTAX        OCTET STRING (SIZE (1))
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
   "This object identifies the VSI Manager with a database that holds
    the detailed  VSI type and or instance definitions."

   REFERENCE   "41.1.3"
   ::= { ieee8021BridgeEvbVSIDBEntry 10 }

ieee8021BridgeEvbVSIType OBJECT-TYPE
   SYNTAX     Integer32
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION " The VTID is an integer value used to identify
                 a pre-configured set of  controls and attributes
                 that are associated with a set of VSIs."
   REFERENCE  " 41.2.4"
   ::= { ieee8021BridgeEvbVSIDBEntry 11 }
```

```
ieee8021BridgeEvbVSITypeVersion  OBJECT-TYPE
   SYNTAX        OCTET STRING (SIZE (1))
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
   "The VSI Type Version is an integer identifier designating the
   expected/desired VTID version.  The VTID version allows a VSI
   Manager Database to contain multiple versions of a  given VSI
   Type, allowing smooth migration to newer VSI types."

   REFERENCE   "41.2.5"
   ::= { ieee8021BridgeEvbVSIDBEntry 12 }


ieee8021BridgeEvbVSIMvFormat       OBJECT-TYPE
   SYNTAX        INTEGER
                 {
                   basic (1),
                   partial (2),
                   vlanOnly (3)
                 }
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
   "This object specifies the MAC/VLAN format.
    basic   - Basic MAC/VLAN format
    partial - Partial MAC/VLAN format
    vlanOnly - Vlan-only MAC/VLAN format
   "
   REFERENCE   "41.2.8"
   ::= { ieee8021BridgeEvbVSIDBEntry 13 }

ieee8021BridgeEvbVSINumMACs    OBJECT-TYPE
   SYNTAX        Integer32
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
   "This object specifies the the number of MAC address/VLAN ID pairs
   contained in the repeated portion of the MAC/VLANs field in the
   VDP TLV."
   REFERENCE   "41.2.9"
   ::= { ieee8021BridgeEvbVSIDBEntry 14 }


ieee8021BridgeEvbVDPMachineState   OBJECT-TYPE
   SYNTAX      INTEGER
                {
                preAssociate (1),
                preAssociateWithRsrcReservation (2),
                associate (3),
                deAssociate (4)
               }
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
   "This object specifies the VDP state machine. "
   REFERENCE   "41.5.5.14"
   ::= { ieee8021BridgeEvbVSIDBEntry 15 }
```

```
ieee8021BridgeEvbVDPCommandsSucceeded  OBJECT-TYPE
   SYNTAX        Counter32
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
   "This object specifies the VDP number of successful commands since
    creation."
   REFERENCE   "41.5"
   ::= { ieee8021BridgeEvbVSIDBEntry 16 }

ieee8021BridgeEvbVDPCommandsFailed   OBJECT-TYPE
   SYNTAX        Counter32
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
   "This object specifies the VDP number of failed commands since
   creation "
   REFERENCE   "41.5"
   ::= { ieee8021BridgeEvbVSIDBEntry 17 }

ieee8021BridgeEvbVDPCommandReverts   OBJECT-TYPE
   SYNTAX        Counter32
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
   "This object specifies the VDP command reverts since creation  "
   REFERENCE   "41.5"
   ::= { ieee8021BridgeEvbVSIDBEntry 18 }


ieee8021BridgeEvbVDPCounterDiscontinuity OBJECT-TYPE
   SYNTAX        TimeTicks
   UNITS         "hundredths of a second"
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION
             "The time (in hundredths of a second) since the
              last counter discontinuity."
       ::= { ieee8021BridgeEvbVSIDBEntry 19}

-- ================================
-- List of MAC/VLANs
-- ================================

ieee8021BridgeEvbVSIDBMacTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021BridgeEvbVSIDBMacEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table that contains database of the active Virtual Station
        Interfaces."
    REFERENCE   "12.26.3"
    ::= { ieee8021BridgeEvbVSIDBObjects 2 }

ieee8021BridgeEvbVSIDBMacEntry OBJECT-TYPE
    SYNTAX      Ieee8021BridgeEvbVSIDBMacEntry
    MAX-ACCESS  not-accessible
    STATUS      current
```

```
    DESCRIPTION
        "A list of objects containing database of the MAC/VLANs
         associated with Virtual Station Interfaces."

    INDEX  { ieee8021BridgeEvbVSIComponentID,
             ieee8021BridgeEvbVSIPortNumber,
             ieee8021BridgeEvbVSIIDType,
             ieee8021BridgeEvbVSIID,
             ieee8021BridgeEvbGroupID,
             ieee8021BridgeEvbVSIMac,
             ieee8021BridgeEvbVSIVlanId

            }
    ::= { ieee8021BridgeEvbVSIDBMacTable 1 }

Ieee8021BridgeEvbVSIDBMacEntry ::=
    SEQUENCE {
       ieee8021BridgeEvbGroupID              Unsigned32,
       ieee8021BridgeEvbVSIMac               MacAddress,
       ieee8021BridgeEvbVSIVlanId            VlanIndex
   }

ieee8021BridgeEvbGroupID   OBJECT-TYPE
   SYNTAX        Unsigned32
   MAX-ACCESS    not-accessible
   STATUS        current
   DESCRIPTION   "Group ID"
   REFERENCE     "41.2.9"
   ::= { ieee8021BridgeEvbVSIDBMacEntry 1}

ieee8021BridgeEvbVSIMac OBJECT-TYPE
   SYNTAX         MacAddress
   MAX-ACCESS     not-accessible
   STATUS         current
   DESCRIPTION
   "The mac-address part of the MAC/VLANs for a VSI."
   REFERENCE  "41.2.9"
   ::= { ieee8021BridgeEvbVSIDBMacEntry 2}


ieee8021BridgeEvbVSIVlanId OBJECT-TYPE
   SYNTAX         VlanIndex
   MAX-ACCESS     read-only
   STATUS         current
   DESCRIPTION
   "The Vlan ID part of the MAC/VLANs for a VSI."
    REFERENCE  "41.2.9"
   ::= { ieee8021BridgeEvbVSIDBMacEntry 3}


-- ================================================
-- Uplink Access Port table entry managed object
-- ================================================

ieee8021BridgeEvbSChannelObjects OBJECT IDENTIFIER ::=
                              { ieee8021BridgeEvbObjects 3 }


ieee8021BridgeEvbUAPConfigTable OBJECT-TYPE
```

```
     SYNTAX      SEQUENCE OF Ieee8021BridgeEvbUAPConfigEntry
     MAX-ACCESS  not-accessible
     STATUS      current
     DESCRIPTION
         "A table that contains  configuration parameters for UAP."
     REFERENCE   "12.26.4.1 "
     ::= { ieee8021BridgeEvbSChannelObjects 1 }

ieee8021BridgeEvbUAPConfigEntry OBJECT-TYPE
     SYNTAX      Ieee8021BridgeEvbUAPConfigEntry
     MAX-ACCESS  not-accessible
     STATUS      current
     DESCRIPTION
         "A list of objects containing information to configure the
         attributes for UAP."
     INDEX  {
             ieee8021BridgePhyPort
             }
     ::= { ieee8021BridgeEvbUAPConfigTable 1 }

 Ieee8021BridgeEvbUAPConfigEntry ::=
     SEQUENCE {
     ieee8021BridgeEvbUAPComponentId
                         IEEE8021PbbComponentIdentifier,
     ieee8021BridgeEvbUAPPort
                         IEEE8021BridgePortNumber,
     ieee8021BridgeEvbUapConfigIfIndex
                         InterfaceIndexOrZero,
     ieee8021BridgeEvbUAPSchCdcpAdminEnable        INTEGER,
     ieee8021BridgeEvbUAPSchAdminCDCPRole          INTEGER,
     ieee8021BridgeEvbUAPSchAdminCDCPChanCap       Integer32,
     ieee8021BridgeEvbUAPSchOperCDCPChanCap        Integer32,
     ieee8021BridgeEvbUAPSchAdminCDCPSVIDPoolLow   VlanIndex,
     ieee8021BridgeEvbUAPSchAdminCDCPSVIDPoolHigh  VlanIndex,
     ieee8021BridgeEvbUAPSchOperState              INTEGER,
     ieee8021BridgeEvbSchCdcpRemoteEnabled         INTEGER,
     ieee8021BridgeEvbSchCdcpRemoteRole            INTEGER,
     ieee8021BridgeEvbUAPConfigStorageType         StorageType,
     ieee8021BridgeEvbUAPConfigRowStatus           RowStatus
    }

ieee8021BridgeEvbUAPComponentId OBJECT-TYPE
   SYNTAX      IEEE8021PbbComponentIdentifier
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
   "The ComponentID of the port for the UAP."
   ::= { ieee8021BridgeEvbUAPConfigEntry 1 }


ieee8021BridgeEvbUAPPort OBJECT-TYPE
   SYNTAX      IEEE8021BridgePortNumber
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
   "The port number of the port for the UAP."
   ::= { ieee8021BridgeEvbUAPConfigEntry 2 }

ieee8021BridgeEvbUapConfigIfIndex  OBJECT-TYPE
```

```
   SYNTAX      InterfaceIndexOrZero
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
        "The value of the instance of the IfIndex object,
         defined in the IF-MIB, for the interface corresponding
         to this port, or the value 0 if the port has not been
         bound to an underlying frame source and sink."
   ::= { ieee8021BridgeEvbUAPConfigEntry 3 }


ieee8021BridgeEvbUAPSchCdcpAdminEnable  OBJECT-TYPE
   SYNTAX     INTEGER
              {
                 enable (1),
                 disable (2)
              }
   MAX-ACCESS   read-create
   STATUS       current
   DESCRIPTION  "Administrative staus of CDCP."
   REFERENCE   "42.4.2"

   ::= { ieee8021BridgeEvbUAPConfigEntry 4 }


ieee8021BridgeEvbUAPSchAdminCDCPRole  OBJECT-TYPE
   SYNTAX      INTEGER
               {
               cdcpRoleB(1),
               cdcpRoleS (2)
               }
   MAX-ACCESS  read-create
   STATUS      current
   DESCRIPTION  "The administratively configured value for the local
   port's role parameter. The value of AdminRole is not reflected in
   the S-channel TLV. The AdminRole may take the value S or B.
   S indicates  the sender is unwilling to accept S-channels
   configuration  (mode, # channels supported, channel index) from
   its neighbor and that the sender is willing to accept SVID
   assignments from the neighbor.  Stations usually take the S role.
   B indicates the sender is willing to accept  S-channels
   configuration (mode, # channels supported, channel index)
   from its neighbor and that the sender is willing do the best
   it can to fill  the SVID assignments
   from the neighbor. Bridges usually take the B role."

   REFERENCE  "42.4.2"
   DEFVAL  { 1 }

::= { ieee8021BridgeEvbUAPConfigEntry 5 }


ieee8021BridgeEvbUAPSchAdminCDCPChanCap OBJECT-TYPE
   SYNTAX             Integer32 (1 .. 167)
   MAX-ACCESS         read-create
   STATUS             current
   DESCRIPTION       "The administratively configured value for the
                     Number of Channels supported parameter. This
                     value is included as the ChanCap parameter in
```

```
                    the S-channel TLV."
   REFERENCE "42.4.1"

::= { ieee8021BridgeEvbUAPConfigEntry 6 }


ieee8021BridgeEvbUAPSchOperCDCPChanCap  OBJECT-TYPE
   SYNTAX         Integer32 (1 .. 167)
   MAX-ACCESS     read-only
   STATUS         current
   DESCRIPTION    "The operational value for the Number of Channels
                   supported parameter. This value is included
                   as the ChnCap parameter in the S-channel TLV."
   REFERENCE "42.4.8"
   ::= { ieee8021BridgeEvbUAPConfigEntry 7 }

ieee8021BridgeEvbUAPSchAdminCDCPSVIDPoolLow  OBJECT-TYPE
   SYNTAX           VlanIndex
   MAX-ACCESS       read-create
   STATUS           current
   DESCRIPTION      "Determines the lowest S-VIDs available for
                     assignment by CDCP."

   REFERENCE "42.4.7"
   ::= { ieee8021BridgeEvbUAPConfigEntry 8 }


ieee8021BridgeEvbUAPSchAdminCDCPSVIDPoolHigh  OBJECT-TYPE
   SYNTAX           VlanIndex
   MAX-ACCESS       read-create
   STATUS           current
   DESCRIPTION      "Determines the highest S-VIDs available for
                     assignment by CDCP."
   REFERENCE     "42.4.7"

   ::= { ieee8021BridgeEvbUAPConfigEntry 9 }


ieee8021BridgeEvbUAPSchOperState     OBJECT-TYPE
   SYNTAX       INTEGER
                {
                  running (1),
                  notRunning (2)
                 }
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION  "The current runnning state of CDCP."
   REFERENCE "42.4.15"

   ::= { ieee8021BridgeEvbUAPConfigEntry 10 }



ieee8021BridgeEvbSchCdcpRemoteEnabled  OBJECT-TYPE
   SYNTAX        INTEGER
                 {
                   enable (1),
                   disable (2)
                 }
```

```
   MAX-ACCESS    read-only
   STATUS        current
   DESCRIPTION  "CDCP state for the remote S-channel."
   REFERENCE   "42.4.14"
   ::= { ieee8021BridgeEvbUAPConfigEntry 11 }


ieee8021BridgeEvbSchCdcpRemoteRole   OBJECT-TYPE
   SYNTAX      INTEGER
               {
                  cdcpRoleB (1),
                  cdcpRoleS (2)
               }
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION  "The  value for the remote port's role parameter."
   REFERENCE "42.4.12"
   ::= { ieee8021BridgeEvbUAPConfigEntry 12 }

ieee8021BridgeEvbUAPConfigStorageType            OBJECT-TYPE
   SYNTAX      StorageType
   MAX-ACCESS  read-create
   STATUS      current
   DESCRIPTION
        "The storage type for this row. Rows in this table that
         were created through an external process may have a storage
         type of readOnly or permanent.
         For a storage type of permanent, none of the columns have
         to be writable."
   DEFVAL { nonVolatile }
   ::= { ieee8021BridgeEvbUAPConfigEntry 13 }


ieee8021BridgeEvbUAPConfigRowStatus  OBJECT-TYPE
  SYNTAX       RowStatus
  MAX-ACCESS   read-create
  STATUS       current
  DESCRIPTION  "RowStatus for creating a UAP table entry."
  ::= { ieee8021BridgeEvbUAPConfigEntry 14 }



-- ====================================
-- S-Channel Interface Table
-- ====================================

ieee8021BridgeEvbCAPConfigTable OBJECT-TYPE
   SYNTAX      SEQUENCE OF Ieee8021BridgeEvbCAPConfigEntry
   MAX-ACCESS  not-accessible
   STATUS      current
   DESCRIPTION
   "A table that contains  configuration information for
   the S-Channel Access Ports (CAP)."
   REFERENCE   "12.26.4.2 "
   ::= { ieee8021BridgeEvbSChannelObjects 2 }

ieee8021BridgeEvbCAPConfigEntry OBJECT-TYPE
   SYNTAX      Ieee8021BridgeEvbCAPConfigEntry
   MAX-ACCESS  not-accessible
```

```
   STATUS      current
   DESCRIPTION
   "A list of objects containing information for the S-Channel
    Access Ports (CAP)"
   INDEX  { ieee8021BridgePhyPort,
            ieee8021BridgeEvbSchID
   }
   ::= { ieee8021BridgeEvbCAPConfigTable 1 }

Ieee8021BridgeEvbCAPConfigEntry ::=
   SEQUENCE {
   ieee8021BridgeEvbSchID
                     Unsigned32,
   ieee8021BridgeEvbCAPComponentId
                     IEEE8021PbbComponentIdentifier,
   ieee8021BridgeEvbCapConfigIfIndex
                     InterfaceIndexOrZero,
   ieee8021BridgeEvbCAPPort
                     IEEE8021BridgePortNumber,
   ieee8021BridgeEvbCAPSChannelID
                     Unsigned32,
   ieee8021BridgeEvbCAPAssociateSBPOrURPCompID
                     IEEE8021PbbComponentIdentifier,
   ieee8021BridgeEvbCAPAssociateSBPOrURPPort
                     IEEE8021BridgePortNumber,
   ieee8021BridgeEvbCAPRowStatus
                     RowStatus
   }

ieee8021BridgeEvbSchID OBJECT-TYPE
   SYNTAX      Unsigned32 (1..4094)
   MAX-ACCESS  not-accessible
   STATUS      current
   DESCRIPTION
   "This object represents the SVID for a ieee8021BridgeEvbSysType
    of evbBridge and a SCID(S-Channel ID) for a
    ieee8021BridgeEvbSysType of evbStation."
   REFERENCE  "42.4.3"
   ::= { ieee8021BridgeEvbCAPConfigEntry 1 }


ieee8021BridgeEvbCAPComponentId   OBJECT-TYPE
   SYNTAX       IEEE8021PbbComponentIdentifier
   MAX-ACCESS   read-only
   STATUS       current
   DESCRIPTION "Component ID for S-channel Access Port."
   ::= { ieee8021BridgeEvbCAPConfigEntry 2 }

ieee8021BridgeEvbCapConfigIfIndex   OBJECT-TYPE
   SYNTAX       InterfaceIndexOrZero
   MAX-ACCESS   read-only
   STATUS       current
   DESCRIPTION "The value of the instance of the IfIndex object,
         defined in the IF-MIB, for the interface corresponding
         to this port, or the value 0 if the port has not been
         bound to an underlying frame source and sink.
         The underlying IfEntry indexed by this column MUST be persistent
         across reinitializations of the management system."
   ::= { ieee8021BridgeEvbCAPConfigEntry 3 }
```

```
ieee8021BridgeEvbCAPPort   OBJECT-TYPE
   SYNTAX       IEEE8021BridgePortNumber
   MAX-ACCESS   read-only
   STATUS       current
   DESCRIPTION  "Port number for the S-Channel Access Port."
   ::= { ieee8021BridgeEvbCAPConfigEntry 4 }


ieee8021BridgeEvbCAPSChannelID    OBJECT-TYPE
   SYNTAX       Unsigned32
   MAX-ACCESS   read-only
   STATUS       current
   DESCRIPTION  "S-Channel ID (SCID) for this CAP."
   REFERENCE   "42.4.2"
   ::= { ieee8021BridgeEvbCAPConfigEntry 5 }

ieee8021BridgeEvbCAPAssociateSBPOrURPCompID   OBJECT-TYPE
   SYNTAX      IEEE8021PbbComponentIdentifier
   MAX-ACCESS  read-write
   STATUS      current
   DESCRIPTION "Component ID of the Server Edge Port to be
                associated with the CAP."
   REFERENCE    "12.4.1.5"
   ::= { ieee8021BridgeEvbCAPConfigEntry 6 }



ieee8021BridgeEvbCAPAssociateSBPOrURPPort    OBJECT-TYPE
   SYNTAX       IEEE8021BridgePortNumber
   MAX-ACCESS   read-write
   STATUS       current
   DESCRIPTION  "Port number of the Server Edge Port to be
   associated with the CAP."
   REFERENCE    "12.4.2"
::= { ieee8021BridgeEvbCAPConfigEntry 7 }


ieee8021BridgeEvbCAPRowStatus       OBJECT-TYPE
   SYNTAX        RowStatus
   MAX-ACCESS    read-create
   STATUS        current
   DESCRIPTION  "RowStatus to create/destroy this table."

   ::= { ieee8021BridgeEvbCAPConfigEntry 8 }

-- ====================================
-- Uplink Relay Port table entry
-- ====================================

ieee8021BridgeEvbURPTable OBJECT-TYPE
   SYNTAX      SEQUENCE OF Ieee8021BridgeEvbURPEntry
   MAX-ACCESS  not-accessible
   STATUS      current
   DESCRIPTION
   "A table that contains  configuration information for
   the Uplink Relay Ports(URP)."
   REFERENCE   "12.26.5.1 "
```

```
   ::= { ieee8021BridgeEvbSChannelObjects 3 }

ieee8021BridgeEvbURPEntry OBJECT-TYPE
   SYNTAX      Ieee8021BridgeEvbURPEntry
   MAX-ACCESS  not-accessible
   STATUS      current
   DESCRIPTION
   "A list of objects containing information for the Uplink
    Relay Ports(URP)."
   INDEX  { ieee8021BridgeEvbURPComponentId,
            ieee8021BridgeEvbURPPort
   }
   ::= { ieee8021BridgeEvbURPTable 1 }

Ieee8021BridgeEvbURPEntry ::=
   SEQUENCE {
     ieee8021BridgeEvbURPComponentId
                      IEEE8021PbbComponentIdentifier,
     ieee8021BridgeEvbURPPort
                      IEEE8021BridgePortNumber,
     ieee8021BridgeEvbURPIfIndex
                      InterfaceIndexOrZero,
     ieee8021BridgeEvbURPBindToISSPort
                      IEEE8021BridgePortNumber,
     ieee8021BridgeEvbURPLldpManual
                      TruthValue,
     ieee8021BridgeEvbURPVdpOperRsrcWaitDelay
                      Unsigned32,
     ieee8021BridgeEvbURPVdpOperRespWaitDelay
                      Unsigned32,
     ieee8021BridgeEvbURPVdpOperReinitKeepAlive
                      Unsigned32
   }


 ieee8021BridgeEvbURPComponentId        OBJECT-TYPE
    SYNTAX      IEEE8021PbbComponentIdentifier
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "Component ID that the URP belongs to."
    ::= { ieee8021BridgeEvbURPEntry 1 }

 ieee8021BridgeEvbURPPort               OBJECT-TYPE
    SYNTAX      IEEE8021BridgePortNumber
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "port number of the urp."
    ::= { ieee8021BridgeEvbURPEntry 2 }


 ieee8021BridgeEvbURPIfIndex           OBJECT-TYPE
    SYNTAX      InterfaceIndexOrZero
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION "The value of the instance of the IfIndex object,
         defined in the IF-MIB, for the interface corresponding
         to this port, or the value 0 if the port has not been
         bound to an underlying frame source and sink.
```

```
        It is an implementation specific decision as to whether
        this object may be modified if it has been created or
        if 0 is a legal value.

        The underlying IfEntry indexed by this column MUST be
        persistent across reinitializations of the management
        system. "
   ::= { ieee8021BridgeEvbURPEntry 3 }

ieee8021BridgeEvbURPBindToISSPort      OBJECT-TYPE
   SYNTAX      IEEE8021BridgePortNumber
   MAX-ACCESS  read-write
   STATUS      current
   DESCRIPTION "The evbURPBindToISSPort is the ISS Port Number where
                the URP is attached.
                This binding is optional and only required in some
                systems."
   ::= { ieee8021BridgeEvbURPEntry 4 }


ieee8021BridgeEvbURPLldpManual         OBJECT-TYPE
   SYNTAX      TruthValue
   MAX-ACCESS  read-write
   STATUS      current
   DESCRIPTION  "The evbUrpLldpManual parameter control how the EVB
                 TLV determines the operating values for parameters.
                 When set TRUE only the local EVB TLV will be used to
                 determine the parameters."
   ::= { ieee8021BridgeEvbURPEntry 6 }


ieee8021BridgeEvbURPVdpOperRsrcWaitDelay     OBJECT-TYPE
   SYNTAX      Unsigned32
   UNITS       "micro-seconds"
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION  "The parameter evbURPVdpOperRsrcWaitDelay is the
                 exponent of 2 used to set the VDP resourceWaitDelay
                 timer at the EVB Bridge."
   ::= { ieee8021BridgeEvbURPEntry 9 }

ieee8021BridgeEvbURPVdpOperRespWaitDelay     OBJECT-TYPE
   SYNTAX     Unsigned32
   UNITS       "micro-seconds"
   MAX-ACCESS  read-write
   STATUS      current
   DESCRIPTION  "The evbUrpVdpOperRespWaitDelay is how long a
                 EVb station VDP will wait for a response from
                 the EVB Bridge VDP."
   ::= { ieee8021BridgeEvbURPEntry 10 }


ieee8021BridgeEvbURPVdpOperReinitKeepAlive   OBJECT-TYPE
   SYNTAX     Unsigned32
   UNITS      "micro-seconds"
   MAX-ACCESS  read-write
   STATUS      current
   DESCRIPTION  "The evbURPVdpOperReinitKeepAlive is the exponent
                 of 2 used to determine the time interval of Keep
```

```
                  Alives transmitted by the EVB station."
    ::= { ieee8021BridgeEvbURPEntry 11 }


-- ========================================
-- Edge Control Protocol Table
-- ========================================

ieee8021BridgeEvbEcpTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021BridgeEvbEcpEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
    "A table that contains  configuration information for
    the Edge Control Protocol (ECP)."
    REFERENCE   "12.26.4.2 "
    ::= { ieee8021BridgeEvbSChannelObjects 4 }

ieee8021BridgeEvbEcpEntry OBJECT-TYPE
    SYNTAX      Ieee8021BridgeEvbEcpEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
    "A list of objects containing information for theEdge Control
     Protocol (ECP)."
    INDEX  { ieee8021BridgeEvbEcpComponentId,
             ieee8021BridgeEvbEcpPort
    }
    ::= { ieee8021BridgeEvbEcpTable 1 }

    Ieee8021BridgeEvbEcpEntry ::=
       SEQUENCE {
       ieee8021BridgeEvbEcpComponentId
                  IEEE8021PbbComponentIdentifier,
       ieee8021BridgeEvbEcpPort
                  IEEE8021BridgePortNumber,
       ieee8021BridgeEvbEcpOperAckTimerInit    Unsigned32,
       ieee8021BridgeEvbEcpOperMaxRetries      Unsigned32,
       ieee8021BridgeEvbEcpTxFrameCount        Counter32,
       ieee8021BridgeEvbEcpTxRetryCount        Counter32,
       ieee8021BridgeEvbEcpTxFailures          Counter32,
       ieee8021BridgeEvbEcpRxFrameCount        Counter32
       }


  ieee8021BridgeEvbEcpComponentId        OBJECT-TYPE
     SYNTAX      IEEE8021PbbComponentIdentifier
     MAX-ACCESS  not-accessible
     STATUS      current
     DESCRIPTION  "Component ID ."
     ::= { ieee8021BridgeEvbEcpEntry 1 }

  ieee8021BridgeEvbEcpPort               OBJECT-TYPE
     SYNTAX    IEEE8021BridgePortNumber
     MAX-ACCESS  not-accessible
     STATUS      current
     DESCRIPTION "Port number."
     ::= { ieee8021BridgeEvbEcpEntry 2 }
```

```
  ieee8021BridgeEvbEcpOperAckTimerInit       OBJECT-TYPE
     SYNTAX      Unsigned32
     UNITS       "micro-seconds"
     MAX-ACCESS  read-only
     STATUS      current
     DESCRIPTION  "The initial value used to initialize ackTimer
                       (43.3.6.1)."
     ::= { ieee8021BridgeEvbEcpEntry 3 }

  ieee8021BridgeEvbEcpOperMaxRetries            OBJECT-TYPE
     SYNTAX      Unsigned32
     MAX-ACCESS  read-only
     STATUS      current
     DESCRIPTION  "This integer variable defines the maximum number
                   of times that the ECP transmit state machine will
                   retry a transmission if no ACK is received."
     ::= { ieee8021BridgeEvbEcpEntry 4 }

  ieee8021BridgeEvbEcpTxFrameCount          OBJECT-TYPE
     SYNTAX      Counter32
     MAX-ACCESS  read-only
     STATUS      current
     DESCRIPTION "The evbECPTxFrameCount is the number of ECP frame
                  transmitted since ECP was instanciated."
     ::= { ieee8021BridgeEvbEcpEntry 5 }


  ieee8021BridgeEvbEcpTxRetryCount     OBJECT-TYPE
     SYNTAX      Counter32
     MAX-ACCESS  read-only
     STATUS      current
     DESCRIPTION  "The evbECPTxRetryCount is the number of times
                   ECP re-tried transmission since ECP was
                   instanciated."
     ::= { ieee8021BridgeEvbEcpEntry 6 }

  ieee8021BridgeEvbEcpTxFailures     OBJECT-TYPE
     SYNTAX      Counter32
     MAX-ACCESS  read-only
     STATUS      current
     DESCRIPTION  "The evbECPTxFailures is the number of times ECP
                   failed to successfully deliver a frame since ECP
                   was instanciated."
     ::= { ieee8021BridgeEvbEcpEntry 7 }


  ieee8021BridgeEvbEcpRxFrameCount   OBJECT-TYPE
     SYNTAX      Counter32
     MAX-ACCESS  read-only
     STATUS      current
     DESCRIPTION  "The evbECPRxFrameCount is the number
                   of frames received since ECP was instanciated."
     ::= { ieee8021BridgeEvbEcpEntry 8 }



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

```
-- ================================================================

ieee8021BridgeEvbGroups
    OBJECT IDENTIFIER ::= { ieee8021BridgeEvbConformance 1 }

ieee8021BridgeEvbCompliances
    OBJECT IDENTIFIER ::= { ieee8021BridgeEvbConformance 2 }

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

ieee8021BridgeEvbSysGroup OBJECT-GROUP
    OBJECTS {
        ieee8021BridgeEvbSysType,
        ieee8021BridgeEvbSysNumExternalPorts,
        ieee8021BridgeEvbSysEvbLldpTxEnable,
        ieee8021BridgeEvbSysEvbLldpGidCapable,
        ieee8021BridgeEvbSysEvbLldpManual,
        ieee8021BridgeEvbSysEcpAckTimer,
        ieee8021BridgeEvbSysEcpMaxRetries,
        ieee8021BridgeEvbSysVdpDfltRsrcWaitDelay,
        ieee8021BridgeEvbSysVdpDfltReinitKeepAlive


    }
    STATUS      current
    DESCRIPTION
        "The collection of objects used to represent a EVB
         management objects."
    ::= { ieee8021BridgeEvbGroups 1 }


ieee8021BridgeEvbSbpGroup   OBJECT-GROUP
   OBJECTS {
        ieee8021BridgeEvbSbpLldpManual,
        ieee8021BridgeEvbSbpVdpOperRsrcWaitDelay ,
        ieee8021BridgeEvbSbpVdpOperReinitKeepAlive,
        ieee8021BridgeEvbSbpVdpOperToutKeepAlive
   }
   STATUS current
   DESCRIPTION
   "The collection of objects used to represent a SBP
    management objects."
   ::= { ieee8021BridgeEvbGroups 3 }


ieee8021BridgeEvbVSIDBGroup OBJECT-GROUP
    OBJECTS {
           ieee8021BridgeEvbVSITimeSinceCreate ,
           ieee8021BridgeEvbVsiVdpOperCmd,
           ieee8021BridgeEvbVsiOperRevert,
           ieee8021BridgeEvbVsiOperHard,
           ieee8021BridgeEvbVsiOperReason,
           ieee8021BridgeEvbVSIMgrID,
           ieee8021BridgeEvbVSIType,
           ieee8021BridgeEvbVSITypeVersion ,
           ieee8021BridgeEvbVSIMvFormat,
           ieee8021BridgeEvbVSINumMACs ,
```

```
          ieee8021BridgeEvbVDPMachineState ,
          ieee8021BridgeEvbVDPCommandsSucceeded ,
          ieee8021BridgeEvbVDPCommandsFailed ,
          ieee8021BridgeEvbVDPCommandReverts ,
          ieee8021BridgeEvbVDPCounterDiscontinuity,
          ieee8021BridgeEvbVSIVlanId
  }
  STATUS      current
  DESCRIPTION
      "The collection of objects used to represent a EVB VSI
       DB table."
  ::= { ieee8021BridgeEvbGroups 4 }


ieee8021BridgeEvbUAPGroup OBJECT-GROUP
   OBJECTS {
           ieee8021BridgeEvbUAPComponentId,
           ieee8021BridgeEvbUAPPort,
           ieee8021BridgeEvbUapConfigIfIndex,
           ieee8021BridgeEvbUAPSchCdcpAdminEnable,
           ieee8021BridgeEvbUAPSchAdminCDCPRole,
           ieee8021BridgeEvbUAPSchAdminCDCPChanCap,
           ieee8021BridgeEvbUAPSchOperCDCPChanCap,
           ieee8021BridgeEvbUAPSchAdminCDCPSVIDPoolLow,
           ieee8021BridgeEvbUAPSchAdminCDCPSVIDPoolHigh,
           ieee8021BridgeEvbUAPSchOperState,
           ieee8021BridgeEvbSchCdcpRemoteEnabled,
           ieee8021BridgeEvbSchCdcpRemoteRole,
           ieee8021BridgeEvbUAPConfigStorageType ,
           ieee8021BridgeEvbUAPConfigRowStatus
   }
   STATUS      current
   DESCRIPTION
       "The collection of objects used to represent a EVB UAP
        table."
   ::= { ieee8021BridgeEvbGroups 5 }


 ieee8021BridgeEvbCAPConfigGroup OBJECT-GROUP
      OBJECTS {
            ieee8021BridgeEvbCAPComponentId,
            ieee8021BridgeEvbCapConfigIfIndex,
            ieee8021BridgeEvbCAPPort,
            ieee8021BridgeEvbCAPSChannelID,
            ieee8021BridgeEvbCAPAssociateSBPOrURPCompID,
            ieee8021BridgeEvbCAPAssociateSBPOrURPPort,
            ieee8021BridgeEvbCAPRowStatus
      }
      STATUS      current
      DESCRIPTION
          "The collection of objects used to represent a EVB
           CAP management objects."
  ::= { ieee8021BridgeEvbGroups 6 }



  ieee8021BridgeEvbsURPGroup OBJECT-GROUP
    OBJECTS {
        ieee8021BridgeEvbURPIfIndex,
```

```
          ieee8021BridgeEvbURPBindToISSPort ,
          ieee8021BridgeEvbURPLldpManual,
          ieee8021BridgeEvbURPVdpOperRsrcWaitDelay,
          ieee8021BridgeEvbURPVdpOperRespWaitDelay ,
          ieee8021BridgeEvbURPVdpOperReinitKeepAlive
      }
      STATUS      current
      DESCRIPTION
          "The collection of objects used to represent a EVBS URP
           management objects."
  ::= { ieee8021BridgeEvbGroups 7 }


   ieee8021BridgeEvbEcpGroup OBJECT-GROUP
      OBJECTS {
      ieee8021BridgeEvbEcpOperAckTimerInit,
      ieee8021BridgeEvbEcpOperMaxRetries ,
      ieee8021BridgeEvbEcpTxFrameCount,
      ieee8021BridgeEvbEcpTxRetryCount,
      ieee8021BridgeEvbEcpTxFailures ,
      ieee8021BridgeEvbEcpRxFrameCount
      }
      STATUS      current
      DESCRIPTION
      "The collection of objects used to represent a EVB CAP
       management objects."
      ::= { ieee8021BridgeEvbGroups 8 }


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

ieee8021BridgeEvbbCompliance MODULE-COMPLIANCE
    STATUS      current
    DESCRIPTION
        "The compliance statement for devices supporting EVB
        as defined in IEEE 802.1Qbg."
    MODULE
        MANDATORY-GROUPS {
            ieee8021BridgeEvbSysGroup,
            ieee8021BridgeEvbVSIDBGroup,
            ieee8021BridgeEvbSbpGroup,
            ieee8021BridgeEvbEcpGroup
        }

      GROUP  ieee8021BridgeEvbUAPGroup
      DESCRIPTION "This group is mandatory when S-Channels
                   are present."

      GROUP ieee8021BridgeEvbCAPConfigGroup
      DESCRIPTION "This group is mandatory when S-Channels
                   are present."

    ::= { ieee8021BridgeEvbCompliances 1 }


 ieee8021BridgeEvbsCompliance MODULE-COMPLIANCE
    STATUS      current
```

```
    DESCRIPTION
        "The compliance statement for devices supporting EVBS
        as defined in IEEE 802.1Qbg."
    MODULE
        MANDATORY-GROUPS {
            ieee8021BridgeEvbSysGroup,
            ieee8021BridgeEvbVSIDBGroup,
            ieee8021BridgeEvbsURPGroup,
            ieee8021BridgeEvbEcpGroup

        }
        GROUP ieee8021BridgeEvbUAPGroup
        DESCRIPTION "This group is mandatory when S-Channels
                     are present."

        GROUP ieee8021BridgeEvbCAPConfigGroup
        DESCRIPTION "This group is mandatory when S-Channels
                     are present."


    ::= { ieee8021BridgeEvbCompliances 2 }

END
```

Insert the following text, tables, and figures as new Clause 40:

40. Edge Virtual Bridging (EVB)

Figure 40-1 provides an overview of the Edge Virtual Bridging (EVB) architecture. An end station that supports the attachment of one or more virtual stations is said to be an EVB station. Each virtual station has at least one virtual station interface (VSI). Each virtual station communicates with other virtual stations or other stations on the bridged LAN via the edge relay (ERs) to which it is attached (see 3.2).

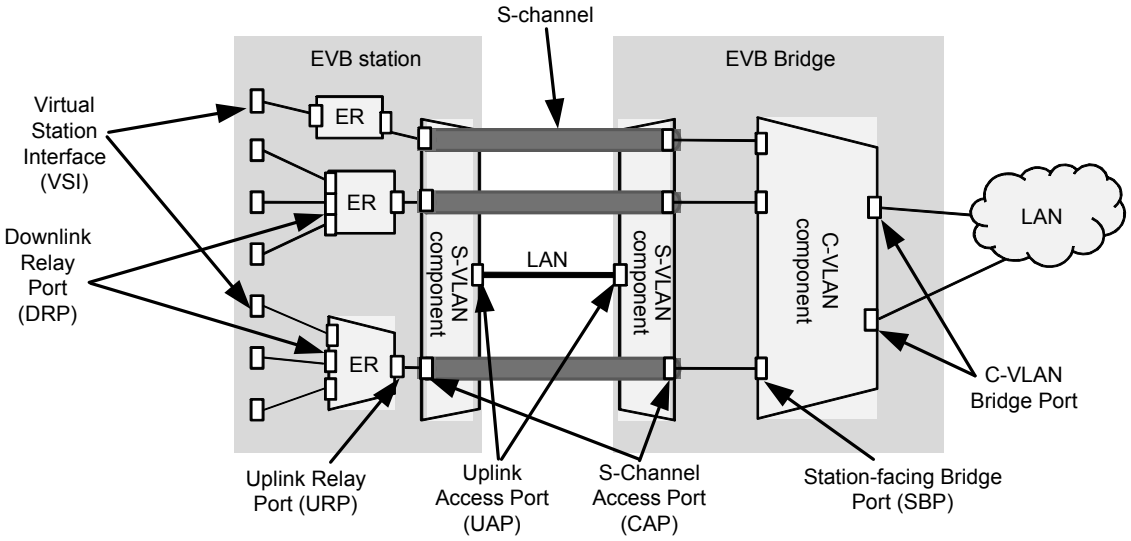


Figure 40-1—EVB architecture overview

An ER supports local relay among virtual stations and/or relay between a virtual station and other stations on the bridged LAN through an EVB Bridge. When forwarding of frames from one DRP to one or more other DRPs associated with the same ER (i.e., local relay) is not supported, then forwarding of traffic from one DRP to one or more other DRPs is performed by the EVB Bridge, utilizing reflective relay (8.6.1).

Connection between a DRP and a virtual station is achieved via a Virtual Station Interface (VSI). Traffic from a VSI traverses an internal LAN connecting the DRP to the virtual station. The operation of an ER does not result in any modifications to relayed frames over and above the normal tagging and un-tagging functions of a VLAN Bridge. ERs do not participate in, or affect, Spanning Tree operation; it is therefore necessary that the logical connectivity maintained within the station is always loop-free (5.23.1).

Figure 40-1 shows a 2-port ER within the EVB station; this illustrates the fact that even where a single VSI is supported by an S-channel, an ER is present in order to provide C-tagging, reserved address filtering (per Table 8-1) and support for the VSI Discovery and Configuration Protocol (VDP, Clause 41), the edge control protocol (ECP), the EVB TLV (D.2.13), and LLDP (IEEE Std 802.1AB).

Each VSI instance is assigned a VSI manager ID, VSI Type ID (VTID) and VSI Instance Identifier (VSIID). VDP associates a VSI instance and its related VLAN Identifier(s), MAC Address(es), GroupID(s), VSI manager ID, VTID,[8] and VSIID with an SBP. Similarly, the VDP protocol de-associates a VSI instance from an SBP.

[8]The meaning of the VTID is decided by local system and network management.

The VDP protocol can also be used to associate a single VTID with, or de-associate a single VTID from the SBP. In this case, the VSI instance does not contain any MAC addresses, VLAN identifiers or Group IDs, and uses the wildcard VID format (41.2.9.1). Only the most recent associate command is used to configure the VTID for the SBP.

An ER supports relaying of frames associated with one or more VSIs. In order to achieve this, an ER can support two types of operation. In the first type, referred to as Virtual Edge Bridge (VEB), traffic transferred from one DRP to another DRP of the same ER is forwarded directly by that ER. In the second type, referred to as VEPA, traffic transferred from one DRP to another DRP of the same ER is forwarded onto a single uplink relay port (URP) beyond the ER to the EVB Bridge. In this case, the EVB Bridge's SBP is enabled with reflective relay (6.6.6, 8.6.1); this allows the frame to be reflected back to the same ER from which it was received by the EVB Bridge. The ER can then forward the frame to the destination. Thus, in the second mode, all traffic transits the EVB Bridge's SBP and is subject to, for example, filtering or policing behavior associated with the EVB Bridge.

NOTE—Connection between an EVB Bridge and an ordinary end station takes place via a C-VLAN Bridge Port, not an SBP.

An S-channel is a point-to-point S-VLAN that spans a pair of Port-mapping S-VLAN components (22.6.4) and can be used to interconnect an ER and the C-VLAN component of an EVB Bridge. Multiple S-channels can share the use of a LAN. The use of multiple S-channels allows the EVB station to support multiple ERs. The end point of an S-channel is known as an S-channel Access Port (CAP); frames are S-tagged on entry to, and are untagged on exit from, the S-VLAN component through a CAP.

EVB TLVs (D.2.13) exchanged via LLDP allow an EVB station and an EVB Bridge to exchange information related to the use of reflective relay and other operational parameters. Each ER has an LLDP database at its URP. Each ER can also have an LLDP database at each DRP.

Each URP and each SBP has an instance of Edge Control Protocol (ECP, Clause 43) used to support the VDP. These instances of ECP use the Nearest Customer Bridge address as the destination for frames exchanged between the URP and SBP. VDP TLVs are packed into PDUs that are handed to ECP for delivery. ECP provides reliable delivery of VDP PDUs.

40.1 EVB architecture without S-channels

Figure 40-2 illustrates the relationship of the EVB entities to the Bridge architecture when no S-channels are supported and no Port-mapping S-VLAN components are implemented. In this configuration, the EVB station and EVB Bridge may exchange an S-channel discovery and configuration protocol (CDCP) TLV over the Nearest non-TMPR Bridge address indicating that the S-VLAN component is not present. If the CDCP TLV managed object does not exist in the LLDP database, then the transmitting station or bridge is assumed to not support S-channels. A URP or an SBP can send a CDCP TLV. If the EVB station supports the CDCP TLV then the nearest non-TPMR LLDP database is located at the URP. If the EVB Bridge supports the CDCP TLV then the nearest non-TPMR LLDP database is located at the SBP.

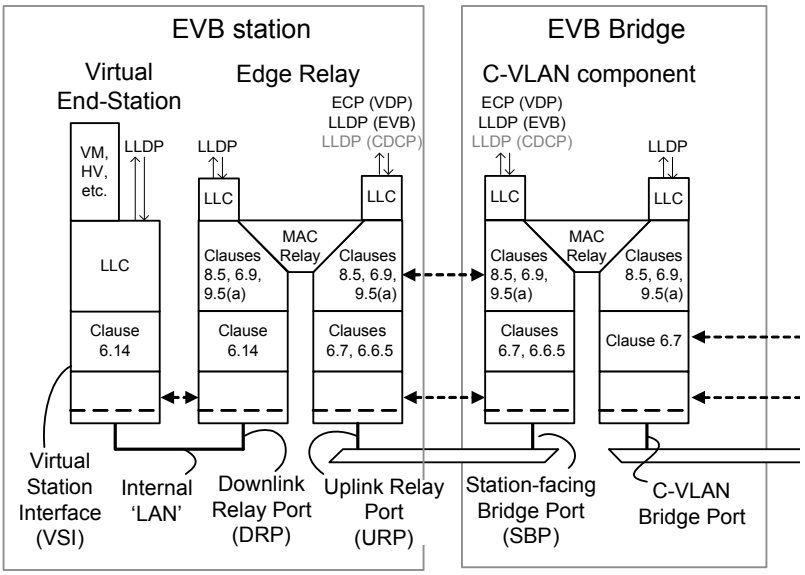


Figure 40-2—EVB architecture without S-channels

40.2 EVB architecture with S-channels

Figure 40-3 shows the relationship of the EVB entities to the Bridge architecture when S-channels are supported. In this configuration, the EVB station and Bridge build nearest non-TPMR LLDP databases at their Uplink Access Ports (UAPs) and use them to exchange CDCP TLVs. Both the EVB station and EVB Bridge set the SComp parameter in the CDCP TLV to TRUE indicating they have an S-VLAN component. The CDCP protocol operating on the CDCP TLVs exchanged by LLDP is used to configure the S-channels.

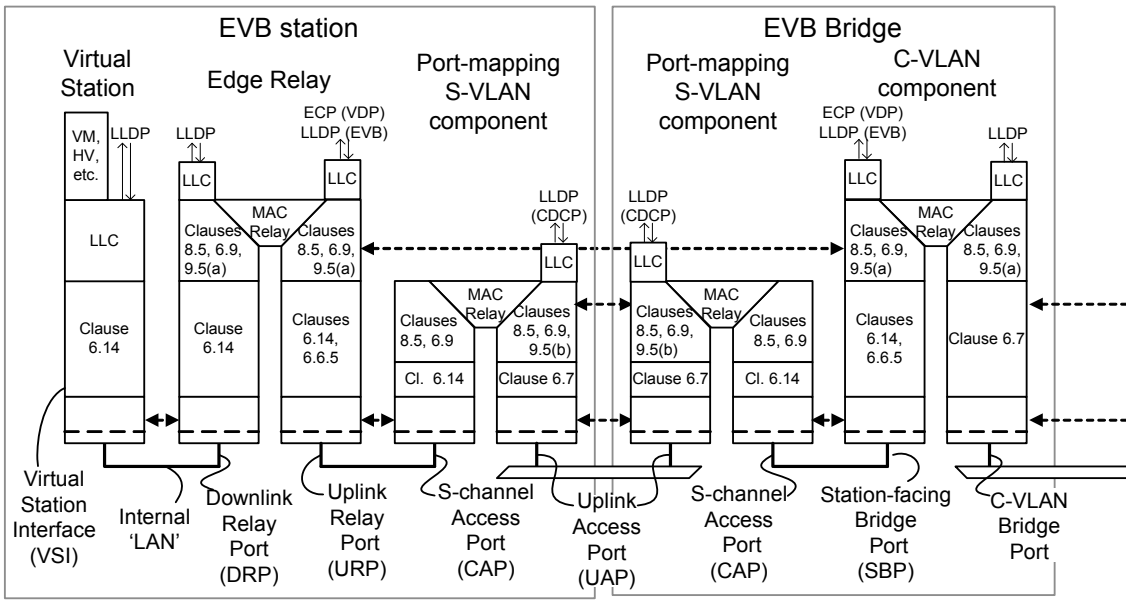


Figure 40-3—EVB archichtecture with S-channel

EVB stations and EVB Bridges use Port-mapping S-VLAN components (5.6, 22.6.4) to instantiate S-channels. Each S-channel connects an S-channel Access Port (CAP) on the EVB station to a CAP on the EVB Bridge. The CAP on an EVB station connects to a single URP on an ER via an internal LAN. The CAP on an EVB Bridge connects to a single SBP on the C-VLAN component via an internal LAN. There is a 1:1 relationship between a CAP and an SBP of the EVB Bridge, and a 1:1 relationship between a CAP and a URP of the EVB station. S-channel support allows the EVB station and EVB Bridge to support multiple ERs on a LAN. Each S-channel is associated with the URP of a distinct ER.

Figure 40-4 shows the relationship between S-channels and S-channel Access Ports (CAPs) and the positioning of a station's internal and external LANs. When S-channels are supported, each physical LAN can be used to support multiple S-channels identified on the LAN by S-tagging. The S-channels are supported by one Port-mapping S-VLAN component for each UAP. Each Port-mapping S-VLAN component within an EVB station can be identified by its single UAP. A CAP is uniquely identified by the combination of the UAP of the S-VLAN component and the S-VID of the S-channel. Each CAP attaches by internal C-tagged LANs to a single URP or SBP. The C-VLANs carried over each S-channel are determined by configuration of the EVB station and of the C-VLAN component within the EVB Bridge.

NOTE—As a result of normal Bridge behaviour described in 6.9, the priority carried in the C-tag is regenerated at the CAP to form the S-tag priority.

The C-VLAN component of the EVB Bridge is a standard C-VLAN component (5.5) that additionally supports reflective relay (6.6.6, 8.6.1), the EVB LLDP TLV, and VDP.

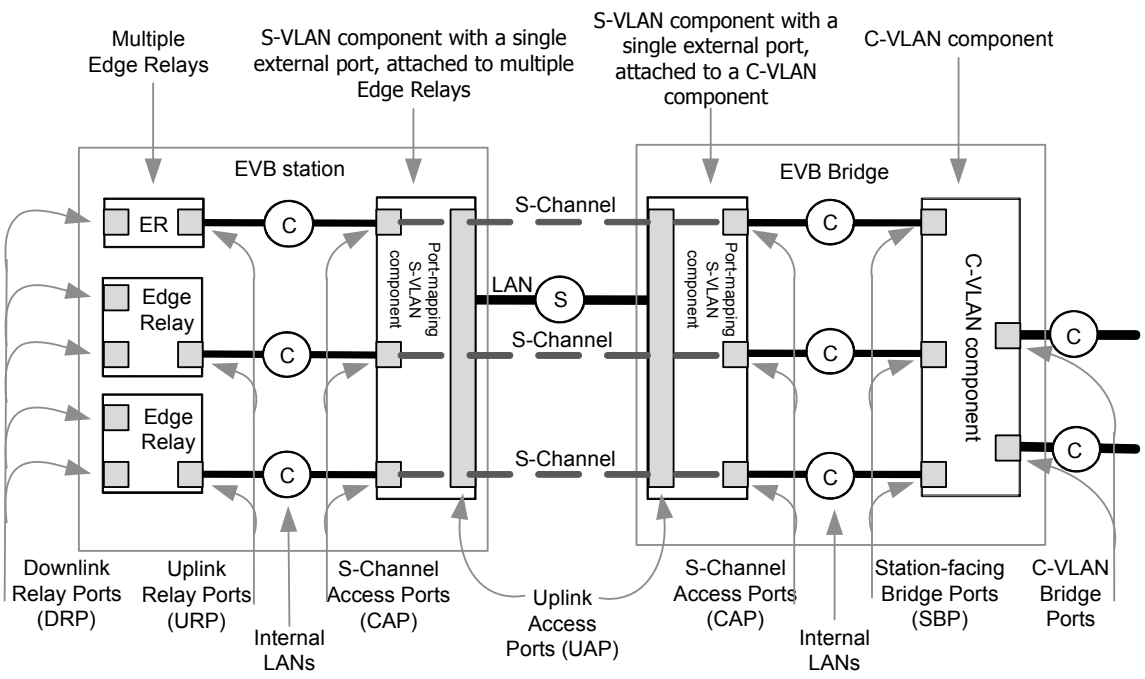


Figure 40-4—EVB components and internal LANs with S-channels

When a UAP table entry (12.26.4.1) is created, a Port-mapping S-VLAN component is instantiated and the following actions are taken automatically:

a) The UAP is configured to
   1) Admit all frames (6.9);
   2) have a PVID parameter equal to the default S-channel S-VID (6.9, 40.3);
   3) be included in the member set for the default S-channel S-VID (8.8.10);
   4) be a member of the untagged set for the default S-channel S-VID (8.8.2);
   5) be included in the member set for all S-VIDs of active S-channels.

b) An S-channel Interface table entry is created if one does not already exist for the default S-channel. This table provides the equivalent functionality of
   1) creating a CAP for the default S-channel;
   2) configuring the CAP to accept only un-S-tagged frames (6.9);
   3) setting the member set for the default S-channel's S-VID to include the CAP;
   4) setting the CAP's PVID to the default S-channel's S-VID;
   5) adding the CAP to the default S-channel S-VID's untagged set;
   6) setting filters on the CAP for the Nearest Bridge and Nearest non-TPMR Bridge group MAC addresses;
   7) in the case of an EVB Bridge allocating (or creating) an SBP on the C-VLAN component attached to the CAP by an internal LAN.

c) An instance of LLDP is started on the UAP transmiting a local database on the Nearest Non-TPMR Bridge Address and including the CDCP TLV.

d) The CDCP protocol is started on the UAP and configured with the parameters specified when the UAP was created.
   1) If the CDCP role is 'B' then CDCP will wait for new S-channel creation requests. As new requests are found CDCP creates new S-channel interface table entries for each new S-channel and deletes entries when S-channels are removed.
   2) If the CDCP role is 'S' then CDCP uses the S-channel interface table to create the list of SCIDs for the S-channels it is requesting from the 'B' side.

40.3 Asymmetric EVB architecture without S-channels

Figure 40-5 and Figure 40-6 illustrate the relationship of the EVB entities to the Bridge architecture when S-channels are supported by only one side at a time; either the EVB Bridge or EVB station, but not both simultaneously. In these configurations, the EVB entity with S-channel support will advertise it has an S-VLAN component by building a nearest non-TPMR LLDP databases at its UAP and including the CDCP TLV with the parameter SComp set to TRUE. The EVB entity without S-channel support may advertise a Nearest non-TMPR Bridge LLDP database with the CDCP TLV indicating an SComp parameter set to FALSE. CDCP is assumed not to be supported by the peer EVB entity until a CDCP TLV has been received.

Each Port-mapping S-VLAN component within an EVB entity supports an internal default S-channel identified by S-VID 1 and uses it to pass untagged frames to its UAP. This default S-channel is always present in the entity supporting an S-VLAN component. In the asymmetric configurations, frames from the system without S-channel support are carried over the default S-channel within the system that has S-channel support.

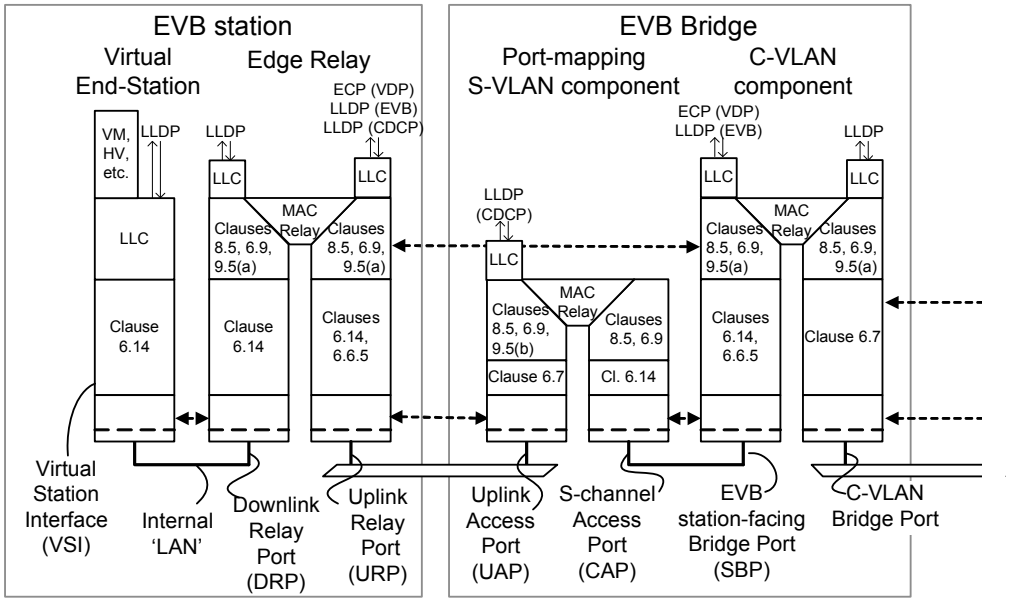


Figure 40-5—EVB architecture without S-channels, with EVB Bridge S-VLAN component

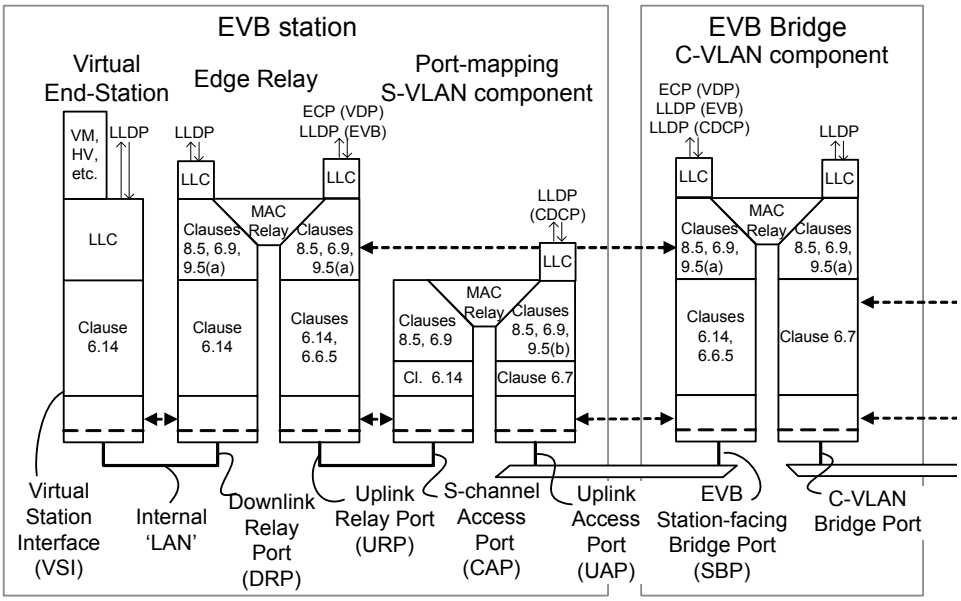


Figure 40-6—EVB architecture without S-channels, with EVB station S-VLAN component

Insert the following text, tables, and figures as new Clause 41:

41. VSI discovery and configuration protocol (VDP)

The VSI discovery and configuration protocol (VDP) associates (registers) a VSI instance with an SBP of an EVB Bridge. VDP simplifies and automates virtual station configuration by enabling 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.

This subclause defines the VDP TLV structure and state machines. VDP uses the Edge Control Protocol (ECP, Clause 43) as a transport protocol for VDP TLV exchanges. Three VDP TLVs are defined as follows:

a) The VSI manager ID TLV (41.1). There is a single instance of this TLV in any ECPDU that carries VDP, and it appears as the first TLV in the ECPDU.
b) The VDP association TLV (41.2). One or more of these TLVs can appear in any ECPDU, following the VSI manager ID TLV.
c) The organizationally defined TLV (41.3).

When ECP is used as a transport protocol for VDP, ECP uses the Nearest Customer Bridge group MAC address (Table 8-1) as the destination address for ECPDUs.

NOTE 1—If there are multiple VSI managers, then their TLVs are transmitted in separate ECPDUs.

NOTE 2—Beyond the requirement stated, that the VSI manager ID TLV appears as the first TLV in ECPDUs carrying VDP, there are no further constraints placed upon how an implementation chooses to pack VDP TLVs into an ECPDU.

NOTE 3—VDP TLVs are not LLDP TLVs, and the TLV type values used in VDP TLVs are assigned from a distinct number space from those used in LLDP TLVs.

41.1 VSI manager ID TLV definition

Figure 41-1 illustrates the format of the VSI manager ID TLV.

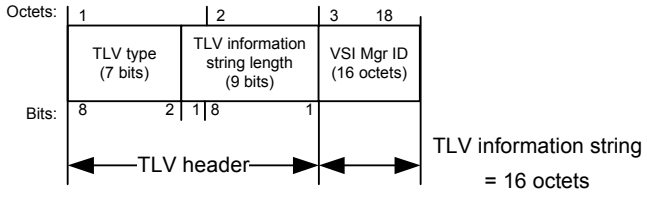


Figure 41-1—VSI manager ID TLV

The VSI manager ID TLV field definitions are contained in 41.1.1 through 41.1.3.

41.1.1 TLV type

The TLV type field takes the value shown in Table 41-1 for VSI manager ID.

41.1.2 TLV information string length

This field contains the length of the TLV information string, which is 16 octets.

Table 41-1—VDP TLV types

TLV type	Value
Pre-Associate	0x01
Pre-Associate with resource reservation	0x02
Associate	0x03
De-associate	0x04
VSI manager ID	0x05
Organizationally defined TLV	0x7F
Reserved for future standardization	0x00, 0x06–0x7E

41.1.3 VSI Manager ID

Identifies the database that should be accessed to get the VSI Type. The value 0 means that the station does not know what VSI Manager ID to use, indicating that the Bridge should select a default value. Any other value is interpreted as an IPv6 address, as defined in IETF RFC 4291.

41.2 VDP association TLV definitions

Figure 41-2 illustrates the format of the VDP association TLV.

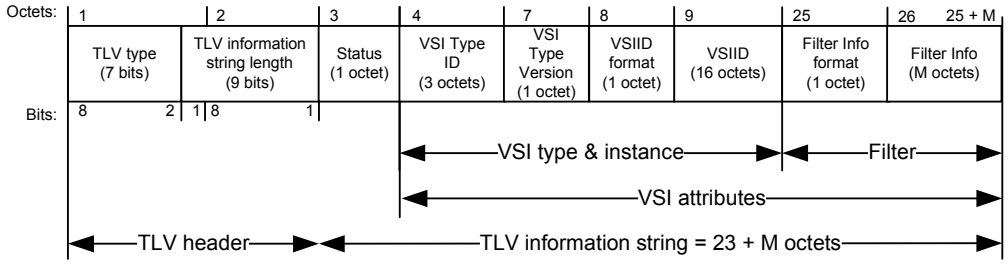


Figure 41-2—VDP association TLV

The VDP association TLV field definitions are contained in 41.2.1 through 41.2.9. The semantics of the VDP association TLV types are defined in 41.2.10.

When the VDP association TLV is sent as a response, the Status field indicates the outcome of the requested operation, and the remaining fields are populated using the information provided in the request or information provided by the EVB Bridge.

41.2.1 TLV type

The TLV type field identifies the type of the VDP association TLV, and can take any of the values shown in Table 41-1 for pre-associate, pre-associate with resource reservation, associate, or de-associate.

41.2.2 TLV information string length

This field contains the length of the TLV information string, calculated as 23 + M octets, where M is the number of octets in the filter info field (41.2.9).

41.2.3 Status

The Status field contains a 4-bit error type, encoded in bits 1–4, and four individual Boolean flags, encoded in bits 5–8.

For all requests, the error type field is reserved for future standardization; it is transmitted as 0x0 and is ignored on receipt.

For all requests, the Boolean flags are interpreted as shown in Table 41-2.

Table 41-2—Flag values in VDP requests

Name	Bit position	Interpretation
M-bit	Bit 5	Indicates that the user of the VSI (e.g., the virtual station) is migrating (M-bit = 1) or provides no guidance on the migration of the user of the VSI (M-bit = 0). The M-bit is used as an indicator relative to the VSI to which the user is migrating.
S-bit	Bit 6	Indicates that the VSI user (e.g., the virtual station) is suspended (S-bit = 1) or provides no guidance as to whether the user of the VSI is suspended (S-bit = 0). A keep-alive Associate request with S-bit = 1 can be sent when the VSI user is suspended. The S-bit is used as an indicator relative to the VSI that the user is migrating from.
Req/Ack	Bit 7	Set to 0 to indicate that the TLV contains a request.
Reserved	Bit 8	Reserved for future standardization.

NOTE—The M-bit is restored to 0 when migration has stopped, either because the migration has succeeded, or it has failed. The S-bit is restored to 0 when the VSI user is no longer suspended.

For all responses, the value of the error type indicates the outcome of the request, as shown in Table 41-3, and the Boolean flags are interpreted as shown in Table 41-4.

41.2.4 VSI Type ID (VTID)

The VTID is an integer value used to identify a VSI Type.

NOTE—One VTID could describe the VSI Type configuration of multiple VSIs. A VTID is only unique per VSI manager ID.

41.2.5 VSI Type Version

The VSI Type Version is an integer identifier that allows a VSI Manager Database to contain multiple versions of a given VSI Type.

41.2.6 VSIID format

The VSIID format field defines the format of the VSIID field that follows it (41.2.7). The possible values of VSIID format are as shown in Table 41-5.

Table 41-3—Error types in VDP responses

Name	Value	Interpretation
Success	0x0	The VDP Request was successfully completed by the bridge.
Invalid Format	0x1	The VDP TLV format is invalid.
Insufficient Resources	0x2	The bridge does not have enough resources to complete the VDP operation successfully.
Unable to contact VSI manager	0x3	The Bridge was unable to contact the VSI manager.
Other failure	0x4	The operation failed for some other reason.
Invalid VID, GroupID, or MAC address	0x5	The operation failed because the VID, GroupID, or MAC address was invalid.
Reserved	0x6–0xF	Reserved for future standardization.

NOTE—"Success" is only interpreted as success by the state machines if all of the flag bits (Table 41-4) are zero.

Table 41-4—Flag values in VDP responses

Name	Bit position	Interpretation
Hard error	Bit 5	Set to 1 to indicate that the operation failed, and if the same operation is re-tried, it is likely to fail in the same way.
Keep	Bit 6	Set to 1 to indicate that the command was rejected and the state prior to the requested command has been kept.
Req/Ack	Bit 7	Set to 1 to indicate that the TLV contains a response.
Reserved	Bit 8	Reserved for future standardization.

41.2.7 VSIID

The VSIID is an identifier for the VSI instance. A VSIID is generated when a VSI instance is created. The VSIID remains constant during virtual station migration. The format of the VSIID is determined by the VSIID format field (41.2.6). In cases where the format uses an identifier value that has fewer than 16 octets, the VSIID field is packed out to 16 octets with leading octets containing zeroes.

41.2.8 Filter Info format

The Filter Info format field determines the format of the Filter Info field (41.2.9). The Filter Info formats defined by this standard are shown in Table 41-6.

Table 41-5—VSIID format values

Name	Description	Value
IPv4	An IPv4 address, encoded as specified in IETF RFC 4291.	0x01
IPv6	An IPv6 address, encoded as specified in IETF RFC 4291.	0x02
MAC	An IEEE 802 MAC address (6 octets), with 10 leading octets containing all zeroes.	0x03
Local	The interpretation of the VSIID is locally defined.	0x04
UUID	A UUID as specified in IETF RFC 4122.	0x05
Reserved	Reserved for future standardization.	0x00, 0x06 through 0xFF

Table 41-6—Filter Info format values

Format	Value
VID (41.2.9.1)	0x01
MAC/VID (41.2.9.2)	0x02
GroupID/VID (41.2.9.3)	0x03
GroupID/MAC/VID (41.2.9.4)	0x04
Reserved for future standardization	0x00, 0x05 through 0xFF

41.2.9 Filter Info field

The Filter Info field contains information from which a filter can be constructed. The filter is a set of VID values or a set of MAC/VID values. The MAC address in a MAC/VID value is an individual MAC address. The filter is applied to traffic transiting ports that do not have direct knowledge of the associated VSI, such as an EVB station-facing Bridge Port, in order to identify the traffic associated with a particular VSI. This allows such ports to apply a VSI Type to the traffic of an individual VSI. Other devices that have direct knowledge of the traffic associated with a VSI, for example devices that form a 1:1 relationship between a port and VSI, simply provide this information via management interfaces.

The Filter Info field can also contain information that is not part of the filter. In particular, the Filter Info field can contain GroupID values. Like the VID, the GroupID identifies a VLAN. When the number of VLANs in the network is less than 4095, each VLAN can be assigned a VID value that is global within the network. When the number of VLANs in the network exceeds 4094, a globally-scoped VID can no longer be

used to uniquely identify each VLAN. Instead, overlapping VIDs may be used in different regions of the network, and a per-region mapping between the global VLAN and the region-specific VID is maintained. In this case, the VLAN is uniquely and globally identified by a GroupID.

When VLANs are identified by GroupID, the station has knowledge of the GroupID but it does not, in general, know the corresponding VID to be used by traffic associated with the VLAN. The Bridge is aware of, or can obtain knowledge of, the VID associated with the specified GroupID. Thus, the station can send GroupID values to the Bridge via the Filter Info field of the VDP Request. The Bridge can map GroupID values to local VID values. The VID is included in the filter constructed by the Bridge and is returned with its corresponding GroupID to the station via the VDP Response.

NOTE 1—The mechanism by which the EVB Bridge determines the GroupID to local VID associations is outside the scope of this standard.

Additionally, the Filter Info field of a VDP TLV in a VDP Response can specify a Priority Code Point (PCP) value associated with any, or all, of the VID values carried by that VDP Response. The PCP value, if specified, is used by the EVB station as the default PCP value associated with the VSI and VID. The Filter Info field contains a PCP Significant (PS) bit associated with each PCP field, indicating whether the PCP field carries a PCP value (binary 1) or does not carry a PCP value (binary 0). If the PCP field carries a PCP value, then the EVB station can adopt that value as the default PCP value associated with the VSI and VID. When sending data frames associated with a given VSI and VID, the EVB station can determine the PCP value associated with each frame by using an algorithm local to the EVB station. For example, the PCP value can be based on the identity of an application associated with the frame as determined by examining higher layer information. For any given frame, it is possible that the algorithm does not provide a specific value of PCP. In such cases, the PCP field is assigned the value of the default PCP associated with the VSI and VID.

NOTE 2—Specification of a PCP value in the VDP Response does not imply that all frames sent by the EVB station, associated with the VSI and VID, carry the specified PCP. It implies only that, if the EVB station has no other information regarding the PCP value that should appear in that particular frame, then the specified default PCP value is used.

41.2.9.1 VID Filter Info format

The VID Filter Info format specifies that the Format Info field contains a set of VID values to be associated with the VSI instance (41.2.7). Figure 41-3 illustrates the VID Filter Info format.

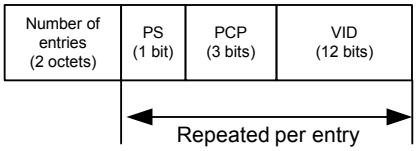


Figure 41-3—VID Filter Info format

The number of VID values in the sequence is specified by the Number of entries field.

The VID field can specify the null VID (see Table 9-2). When the null VID is specified, it is the only VID specified in the Filter Info field (i.e., the Number of entries field is assigned the value 0x0001). Use of the null VID indicates that the set of VID values associated with the VSI is supplied by the Bridge. The Bridge can obtain VID values from the VSI Type whose identity is specified by the VSI Type information in the VDP Request. The set of VID values is returned to the station via the VDP Response.

NOTE—In the case that more than one VID is assigned, the policy that determines how the VIDs are used is outside the scope of this standard.

The Filter Info field can specify the wildcard VID (see Table 9-2). When the wildcard VID is specified, it is the only VID specified in the Filter Info field (i.e., the Number of entries field is assigned the value 0x0001). Use of the wildcard VID value indicates that the VSI Type specified by the VDP Request is designated as the channel VSI Type applied to the EVB station-facing Bridge Port associated with the S-channel.

41.2.9.2 MAC/VID Filter Info format

The MAC/VID Filter Info format indicates that the Format Info field specifies a sequence of MAC/VID value pairs to be associated with the VSI instance (41.2.7). Figure 41-4 illustrates the MAC/VID Filter Info format of the Filter Info field.

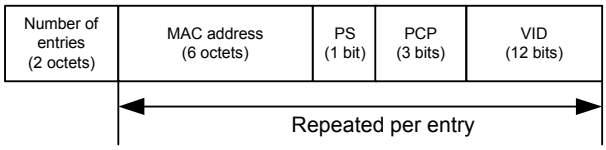


Figure 41-4—MAC/VID filter format

The number of MAC/VID pair values is specified by the field Number of Filter Info entries. Each MAC/VID pair value carries a 6-octet individual MAC address and a 2-octet VID value.

The Filter Info field can specify the null VID for any entry. Use of the null VID indicates that the VID value is supplied by the Bridge.

41.2.9.3 GroupID/VID Filter Info format

The GroupID/VID Filter Info format indicates that the Format Info field specifies a sequence of GroupID/VID pairs to be associated with the VSI instance (41.2.7).

Figure 41-5 illustrates the GroupID/VID Filter Info format of the Filter Info field.

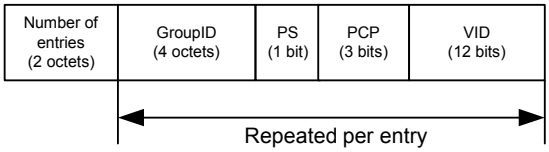


Figure 41-5—GroupID/VID filter format

The number of GroupID/VID pairs is specified by the Number of entries field.

The null VID (see Table 9-2) can be used in a GroupID/VID pair when the GroupID/VID filter format is specified in the VDP Request. In this case, the Bridge is expected to supply the corresponding local VID value in the VDP Response. For this purpose, the Bridge maintains, or has access to, the mapping between GroupID and local VID.

41.2.9.4 GroupID/MAC/VID Filter Info format

The GroupID/MAC/VID Filter Info format indicates that the Filter Info field specifies a sequence of GroupID/MAC/VID triples to be associated with the VSI instance (41.2.7). Figure 41-6 illustrates the GroupID/MAC/VID Filter Info format of the Filter Info field.

The number of GroupID/MAC/VID triples is specified by the value of the Number of entries field. The null VID (see Table 9-2) can be used in a GroupID/MAC/VID triple when the GroupID/MAC/VID filter format is specified in the VDP Request. In this case, the Bridge is expected to supply the corresponding local VID

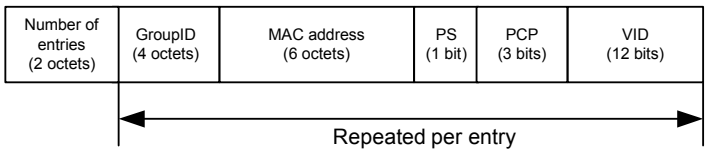


Figure 41-6—GroupID/MAC/VID filter format

value in the VDP Response. For this purpose, the Bridge maintains, or has access to, the mapping between GroupID and local VID.

41.2.10 VDP TLV type and Status semantics

The following subclauses define the semantics associated with each VDP TLV type.

41.2.10.1 Pre-Associate

The Pre-Associate TLV type is used to pre-associate a VSI instance with a bridge port. The Bridge validates the request (see below) and returns a failure Status in case of errors. Successful pre-association does not imply that the VSI Type will be applied to any traffic flowing through the VSI. The pre-associate enables faster response to an associate by allowing the Bridge to obtain the VSI Type prior to an association.

NOTE—If the VSI Type changes without a corresponding change to its version, then inconsistent behavior can result.

41.2.10.2 Pre-Associate with Resource Reservation

Pre-Associate with Resource Reservation involves the same steps as Pre-Associate (41.2.10.1), but on successful pre-association also reserves resources in the Bridge to prepare for a subsequent Associate request.

41.2.10.3 Associate

The Associate TLV Type creates and activates an association between a VSI instance and a bridge port. The Bridge allocates any required bridge resources for the referenced VSI. The Bridge activates the configuration for the VSI Type ID. This association is then applied to the traffic flow to/from the VSI instance.

NOTE—The mechanism used by a Bridge to determine the required resources associated with a VSI Type ID is outside the scope of this standard.

For a given VSIID, a station may issue an Associate without having previously issued a Pre-Associate or Pre-Associate with Resource Reservation. During normal operations a VSI instance is associated on only one port. During network transitions (e.g., virtual station migration) a VSI instance might be associated with more than one port.

If a Pre-Associate or a Pre-Associate with Resource Reservation had previously been received for a given VSI instance, the Bridge establishes the association and allocates resources based only on the information contained in the Associate TLV. Any resources that had been reserved in order to satisfy a previous Pre-Associate with Resource Reservation, and that are not required in order to establish the association as specified in the Associate, are released.

41.2.10.4 De-Associate

The de-associate TLV Type is used to remove an association between a VSI instance and a bridge port. Pre-Associated and Associated VSIs can be de-associated. De-associate releases any resources that were reserved as a result of prior Associate or Pre-Associate operations for that VSI instance.

A de-associate can be initiated either by the station or the Bridge. In the latter case, the Bridge sends a de-associate TLV as if it was a response to a request from the station.

NOTE 1—A Bridge could, for example, issue a de-associate as a consequence of changes in the bridge's status or configuration.

NOTE 2—The result of the above semantics is that a de-associate can be initiated at any time and by either party.

41.3 Organizationally defined TLV definitions

Figure 41-7 illustrates the format of the organizationally defined TLV.

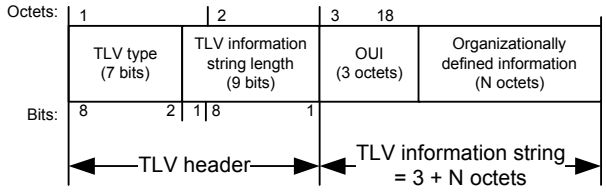


Figure 41-7—Organizationally defined TLV

The organizationally defined TLV field definitions are contained in 41.3.3 and 41.3.4.

41.3.1 TLV type

The TLV type field takes the value shown in Table 41-1 for the organizationally defined TLV.

41.3.2 TLV information string length

This field contains the length of the TLV information string, which is 3 + N octets, where N is the number of octets in the organizationally defined information field (41.3.4).

41.3.3 Organizationally unique identifier (OUI)

Identifies the organization that is responsible for defining the content of the organizationally defined information field (41.3.4). The value of the OUI field is an OUI (see IEEE Std 802) assigned to that organization by the IEEE registration authority.

41.3.4 Organizationally defined information

The content and interpretation of this field is specified by the organization that owns the OUI value contained in the OUI field (41.3.3).

41.4 Validation rules for VDP TLVs

The following rules apply to the validation of received ECPDUs that carry VDP TLVs:

a) If the first TLV in the ECPDU is not a VSI manager ID TLV (41.1), then the entire ECPDU is discarded without further processing.
b) If the ECPDU contains a TLV of a type that is not recognized by the implementation, then that TLV is discarded and is ignored by the VDP state machines.
c) If a TLV extends past the physical end of the ECPDU, then that TLV is discarded.

41.5 VDP state machines

The station VDP state machine is defined in 41.5.3. A station that supports VDP shall support one instance of the station VDP state machine for each active VSI.

The Bridge VDP state machine is defined in 41.5.2. A Bridge that supports VDP shall support one instance of the Bridge VDP state machine for each active VSI.

41.5.1 State machine conventions

The notational conventions used in the specification of VDP are as stated in Annex E.

41.5.2 Bridge VDP state machine

The Bridge VDP state machine shall implement the function defined in Figure 41-8 and the attendant definitions in 41.5.4 through 41.5.7.

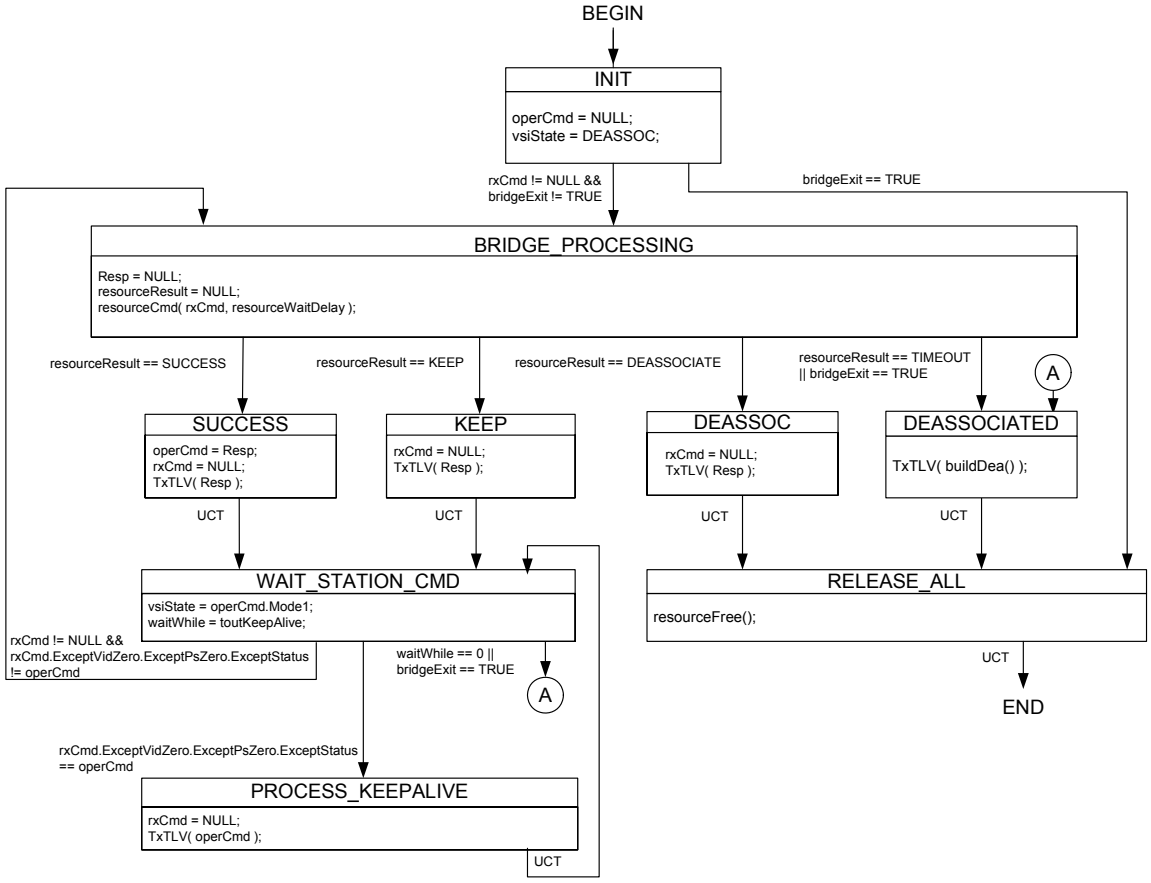
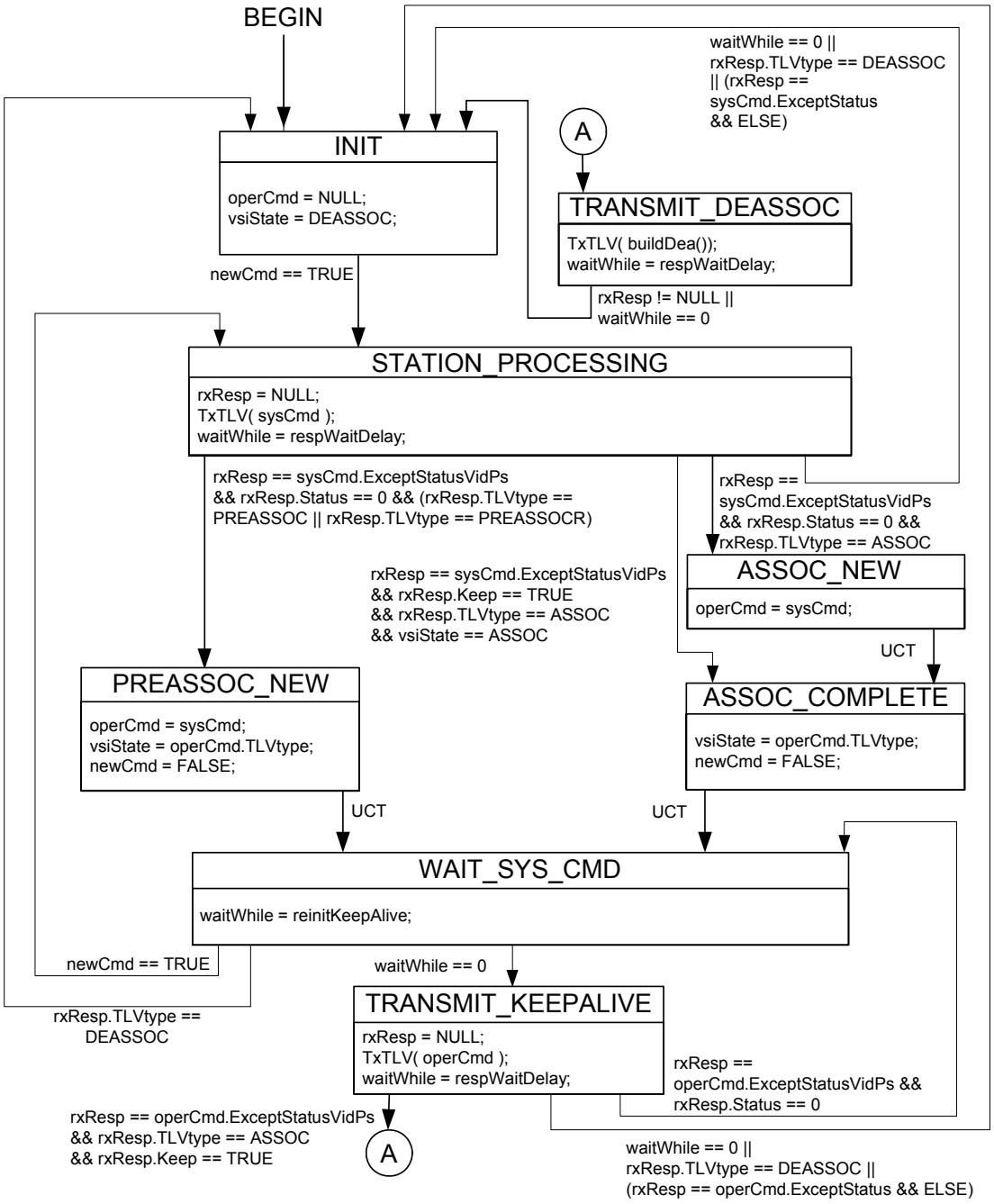


Figure 41-8—Bridge VDP state machine

41.5.3 Station VDP state machine

The station VDP state machine shall implement the function defined in Figure 41-9 and the attendant definitions in 41.5.4 through 41.5.7.



NOTE - The ".Except" notation used in some exit conditions is described in 41.5.6.
In this state machine, "ExceptStatus.ExceptVidZero.ExceptPsZero" has been abbreviated to "ExceptStatusVidPs".

Figure 41-9—Station VDP state machine

41.5.4 VDP state machine timers

A set of timers is used by the VDP state machines; these operate as countdown timers (i.e., they expire when their value reaches zero). These timers are 32-bit countdown timers. They

a) Have a resolution of ten microseconds, with a tolerance of ± 20%.
b) Are started by loading an initial integer value, n, where $0 < n \leq 2^{31}$.
c) Are decremented by one per timer tick, as long as $n > 0$; the interval between timer ticks is the same as the timer resolution.
d) Represent the remaining time in the period.

NOTE—Where timers are used in the VDP state machines to initiate keep-alive messages, it is recommended that a small random component is added to the timer interval in order to avoid the possibility that timers associated with different VSIs become synchronized.

41.5.4.1 waitWhile

An instance of the waitWhile timer exists for each instance of the station VDP state machine (41.5.3) and for each instance of the Bridge VDP state machine (41.5.2).

41.5.5 VDP state machine variables and parameters

41.5.5.1 bridgeExit

A Boolean signal from the Bridge. When TRUE, this variable indicates that the Bridge VDP state machine should exit.

41.5.5.2 newCmd

This variable is set to TRUE by the system to indicate to the state machine that there is a command ready to be transmitted. The state machine sets newCmd FALSE when that command has been processed and is ready to process a further command.

41.5.5.3 NULL

A null value. If NULL is assigned to a TLV variable, it indicates that the variable contains no TLV.

41.5.5.4 operCmd

The command TLV (the TLV that carried the current operating command—associate, pre-associate, or de-associate), at the station or bridge, or NULL if there is no current operating command.

41.5.5.5 reinitKeepAlive

The value used to initialize the waitWhile timer (41.5.4.1) by the station VDP state machine in order to determine when to transmit a keep alive message. This value is derived from the value of the management variable urpVdpOperReinitKeepAlive (12.26.5), which is type timer exp, by using urpVdpOperReinitKeepAlive as an exponent of 2. The variable urpVdpOperReinitKeepAlive is the larger of the values proposed by the station and Bridge.

The default value used by the station for urpVdpOperReinitKeepAlive is an exponent value of 20, representing a timer interval of about 10.5 s.

41.5.5.6 resourceCmdResult

This variable is used to record the result of a resourceCmd() procedure call (41.5.7.2). The possible result values are as follows:

a) Success
b) timeOut
c) Fail (insufficient resources)
d) Fail (invalid format)
e) Fail (other)

41.5.5.7 resourceWaitDelay

The value used to initialize the waitWhile timer (41.5.4.1) by the Bridge VDP state machine when the state machine is waiting for a response. This value is derived from the values of the management variable sbpVdpResourceWaitDelay (12.26.2), which is type timer exp, by using sbpVdpResourceWaitDelay as an exponent of 2. The variable sbpVdpResourceWaitDelay is the larger of the values proposed by the EVB station and EVB Bridge.

The default value used by the station and Bridge for sbpVdpResourceWaitDelay is an exponent value of 20, representing a timer interval of about 10.5 s.

41.5.5.8 Resp

A response TLV returned from the procedure resourceCmd() (41.5.7.2). Resp can be set to NULL prior to issuing resourceCmd(). The variable is always non-NULL when resoureCmd() completes.

41.5.5.9 respWaitDelay

The value used to initialize the waitWhile timer (41.5.4.1) by the station VDP state machine when the state machine is waiting for a response. This value is derived from the values of the management variables urpVdpResourceWaitDelay (12.26.5), ecpOperAckTimerInit (12.27) and ecpOperMaxTries (12.27). The value is expressed by the following equation:

$$\text{respWaitDelay} = 1.5 \times (2^{\text{urpVdpResourceWaitDelay}} + (2 \times \text{ecpOperMaxTries}+1) \times 2^{\text{ecpOperAckTimerInit}})$$

NOTE—The factor of 1.5 allows for a 20% tolerance in the timer values.

The values of urpVdpResourceWaitDelay, ecpOperAckTimerInit, and ecpOperMaxTries that are used are the larger of the values proposed by the station and Bridge.

The default value used by the station and Bridge is about 11.6 s. The default value for urpVdpResourceWaitDelay is an exponent of 20 representing a timer interval of about 10.5 s and for ecpOperAckTimerInit is an exponent of 14 representing a timer interval of about 164 ms. The default value for ecpOperMaxTries is 3.

41.5.5.10 rxCmd

The last received command TLV, or NULL if no command TLV has been received. The rxCmd variable is updated only if it is NULL.

41.5.5.11 rxResp

The last received response TLV. The rxResp variable contains the last received Resp TLV at the station or NULL. RxResp is NULL if no TLVs have been received or if the variable has been cleared by the state machine.

NOTE—It is possible to have a race condition when clearing rxResp since it can be updated asynchronously if the Bridge issues an unsolicited DEASSOC command. If the race condition occurs, the de-associate will occur as a result of the station timer expiring.

41.5.5.12 sysCmd

A command TLV from the system or hypervisor, or NULL if there is no pending command. The VDP state machine is ready to accept a new command when the value of sysCmd is NULL.

41.5.5.13 toutKeepAlive

The value used to initialize the waitWhile timer (41.5.4.1) by the Bridge VDP state machine in order to determine when to expect to receive a keep alive message. This variable is derived from the values of the management variables sbpVdpOperReinitKeepAlive (12.26.2), ecpOperAckTimerInit (12.27), and ecpOperMaxTries (12.27). The value is expressed by the following equation:

$$\text{toutKeepAlive} = 1.5 \times (2^{\text{sbpVdpOperReinitKeepAlive}} + (2 \times \text{ecpOperMaxTries} + 1) \times 2^{\text{ecpOperAckTimerInit}})$$

NOTE—The factor of 1.5 allows for a 20% tolerance in the timer values.

The values of sbpVdpOperReinitKeepAlive, ecpOperAckTimerInit, and ecpOperMaxTries that are used are the larger of the values proposed by the EVB station and EVB Bridge.

The default value used by the station and Bridge is about 11.6 s. The default for sbpVdpOperReinitKeepAlive is an exponent of 20 representing a timer interval of about 10.5 s and for ecpOperAckTimerInit is an exponent of 14 representing a timer interval of about 164 ms. The default value for ecpOperMaxTries is 3.

41.5.5.14 vsiState

The current association state of the VDP state machine. This variable may take the values DEASSOC (de-associated), PREASSOC (pre-associated), PREASSOCR (pre-associated with resource reservation), or ASSOC (associated).

41.5.6 Command-Response TLV field references in state machines

The state machines can make use of the value of individual fields within the value of a TLV by using the following notation:

Tlv-variable-name.Field-name

In practice, only the following two Field-names are used in the state machines:

a) TLVtype, which references the TLV type field of the TLV (41.2.1); and
b) Keep, which references the Keep bit of the Status field in the TLV (41.2.3).

So, for example, a reference in the state diagram to sysCmd.Keep is a reference to the Keep bit of the Status field of the TLV value contained in the sysCmd variable.

The state machines also make use of the ability to compare TLV values for equality or inequality while ignoring a specific field, using the following notation:

Tlv-variable-name.ExceptField-name

This is interpreted as meaning "The value of the TLV contained in the Tlv-variable-name variable, ignoring the value of Field-Name in the comparison."

The reserved Field-name "VidZero" is used to specify that if any VID in the Filter Info field (41.2.9.1) contains zero, then that VID is ignored in the comparison. The reserved Field-name "PsZero" is used to specify that if PS in the Filter Info field contains zero, then the PS and PCP are both ignored in the comparison.

Multiple fields can be excepted by concatenating ExceptField-name items with a separating period.

The value of the Flag bits (Table 41-2 and Table 41-4) are always ignored in field comparison operations.

41.5.7 VDP state machine procedures

41.5.7.1 buildDea()

The buildDea() procedure builds a DEASSOCIATE TLV for the VSI as the return parameter.

41.5.7.2 resourceCmd(rxCmd, delay)

This procedure makes a resource request from the Bridge, waits for a response, builds a response TLV and places it in the variable Resp. The response values reflect the requested resource action (PREASSOC, ASSOC, or DEASSOC), conditioned by return variable resourceResult, which is set to NULL before calling the resourceCmd procedure and set to one of the following values by the procedure:

a) SUCCESS
b) KEEP
c) DEASSOCIATE
d) TIMEOUT

The response constructed by the procedure in the Resp variable can be PREASSOC, ASSOC, or DEASSOC with a Status, keep indicator and hard error indicator. For a successful completion the procedure will copy the rxCmd parameter into the Resp variable. If the Bridge is selecting VIDs based on GroupIDs, then the procedure also replaces zero VIDs with valid VIDs.

The delay parameter specifies how long the procedure should wait for a response. If the delay is exceeded, no response is received, and the VSI is not associated, then the procedure returns a value of TIMEOUT. If the delay is exceeded, no response is received and the VSI is associated, then the procedure returns a value of KEEP along with a Resp equal to the rxCmd parameter. If the delay is not exceeded, then the procedure returns SUCCESS, KEEP, or DEASSOCIATE depending on the response received along with the rxCmd in the Resp and the Status set as follows:

e) **DEASSOC**:
   The procedure returns DEASSOCIATE along with Resp.Status set to Success.
f) **PREASSOC**:
   1) If the request can be satisfied, the procedure returns SUCCESS along with Resp.Status set to Success.

2) If the request cannot be satisfied, the procedure returns DEASSOCIATE along with Resp.Status set to a code other than Success and the Resp.hard set to TRUE if a retry will not change the situation or FALSE if a retry might change the situation.

g) **PREASSOCR**:
   1) If the request can be satisfied, and the resources requested are available and reserved for this VSI, the procedure returns SUCCESS along with the Resp.Status set to Success.
   2) If the request cannot be satisfied, or the resources are unavailable or not reserved for this VSI, the procedure returns DEASSOCIATE along with the Resp.Status code other than Success and the Resp.hard set to TRUE if a retry will not change the situation or FALSE if a retry might change the situation.

h) **ASSOC**:
   1) If the request can be satisfied, and the resources requested are available and enabled for this VSI, the procedure returns SUCCESS along with the Resp.Status set to Success.
   2) If the request cannot be satisfied, or the resources are unavailable or not reserved for this VSI, and the VSI is not currently Associated, the procedure returns DEASSOCIATE along with the Resp.Status code other than Success and the Resp.hard set to TRUE if a retry will not change the situation or FALSE if a retry might change the situation.
   3) If the request cannot be satisfied, or the resources are unavailable or not reserved for this VSI, and the VSI is currently Associated, the procedure returns KEEP along with the Resp.Status code other than Success and the Resp.hard set to TRUE if a retry will not change the situation or FALSE if a retry might change the situation.

NOTE—A deassociate can happen at any time, initiated by either party to an association.

41.5.7.3 resourceFree()

The resourceFree() procedure frees all resources associated with this VSI.

41.5.7.4 TxTlv(tlv)

The TxTlv() procedure causes the TLV passed in the tlv parameter to be transmitted.

Insert the following text, tables, and figures as new Clause 42:

42. S-Channel Discovery and Configuration Protocol (CDCP)

This clause provides an overview, detailed semantics, and state machines for the S-Channel Discovery and Configuration Protocol (CDCP).

42.1 CDCP discovery and configuration

CDCP is used to configure S-channels (see 40.2). S-channels are implemented in stations and bridges using a Port-mapping S-VLAN component (22.6.4). Figure 40-1 illustrates the use of S-channels.

When the Port-mapping S-VLAN components used to create S-channels exist, they can exchange un-S-tagged frames that are assigned to S-VID 1 and are considered to be assigned to the default S-channel, which has an S-channel identifier (SCID) of 1. The default S-channel is always un-S-tagged even when S-channels are enabled.

NOTE 1—SCIDs are locally assigned identifiers.

The S-channel configuration is determined by the bridge's capabilities and by requests made using CDCP described in this clause. The station requests S-channels using CDCP. CDCP in turn uses an LLDP TLV exchange to coordinate the creation and deletion of S-channels. The LLDP database used by CDCP is addressed using the Nearest non-TPMR Bridge address. The Port-mapping S-VLAN component filters both the Nearest Bridge and the Nearest non-TPMR Bridge addresses on all ports and passes the Nearest Customer Bridge address.

The S-channel identifier (SCID) value 1 and S-VID value 1 are always reserved for the exclusive use as the un-S-tagged default S-channel. CDCP reports the default S-channel in the CDCP TLV as the first SCID,S-VID pair (i.e., <1,1>). The Bridge shall not assign this S-VID except to the default S-channel.

NOTE 2—For the default S-channel, any QoS information that is necessary is extracted from the PCP bits of the C-tag.

42.2 CDCP state machine overview

CDCP requires each side of the configuration be assigned a role as a Bridge or a station. This is done by setting the AdminRole variable. In most cases the station or bridge role will not be settable, though the protocol allows for systems that can take either role. For CDCP to configure an S-channel, one side takes the station role and the other side takes the Bridge role. If both sides of the LAN have equipment configured as stations or as bridges the protocol will not configure S-channels.

NOTE—The role adopted by a given system can be fixed if the system is only capable of operating in a given role.

The CDCP state machine (Figure 42-1) operates on data contained in LLDP MIBs that is updated as a result of the reception of CDCP TLVs, exchanged using LLDP operating on the Nearest non-TMPR Bridge address. The structure of CDCP TLVs is defined in D.2.14.

The configuration proceeds by the bridge providing the best match it can to the station's requested channels and configuration. The station makes the resource request, the bridge responds with its best matching resources, the station then goes operational and reports its running configuration to the bridge, and finally the bridge goes operational with the running configuration of the station.

In the event the station wishes to change its configuration it alters the request in its CDCP TLV and then follows the same process as previously stated. If the ‘B’ loses its ability to support the current configuration it can alter the current configuration in its CDCP TLV at which time the station drops down to the resources supplied by the bridge.

42.3 CDCP configuration state machine

The notational conventions used in the specification of CDCP are as stated in Annex E.

In an implementation that supports the station role, the CDCP configuration state machine shall implement the function specified by the state diagram in Figure 42-1 and the attendant definitions in 42.4 and 42.5.

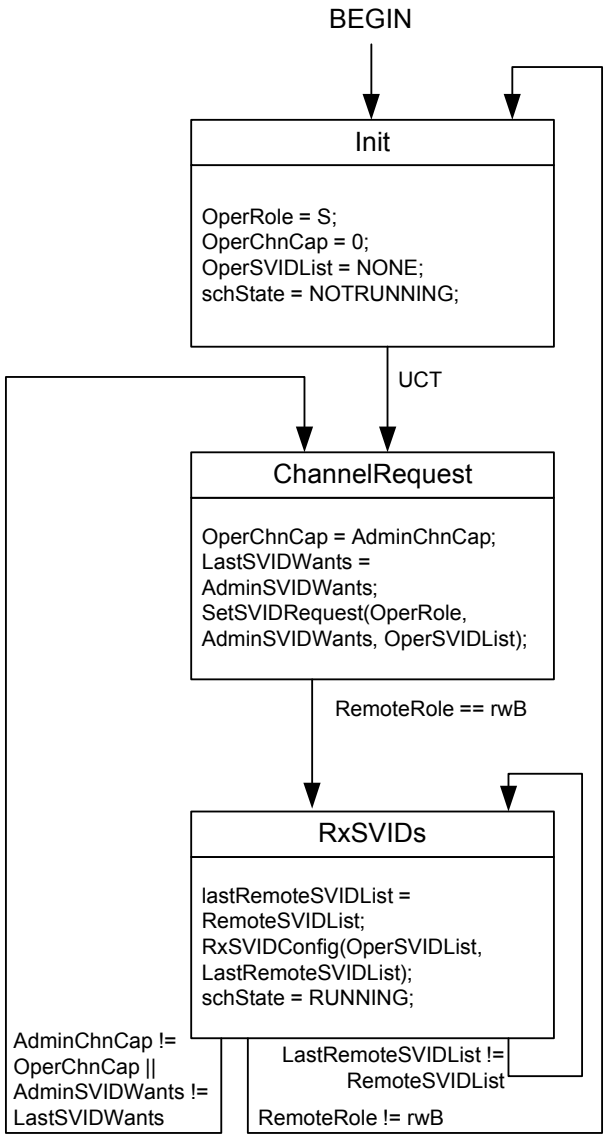


Figure 42-1—CDCP state machine—Station role

In an implementation that supports the Bridge role, the CDCP configuration state machine shall implement the function specified by the state diagram in Figure 42-2 and the attendant definitions in 42.4 and 42.5.

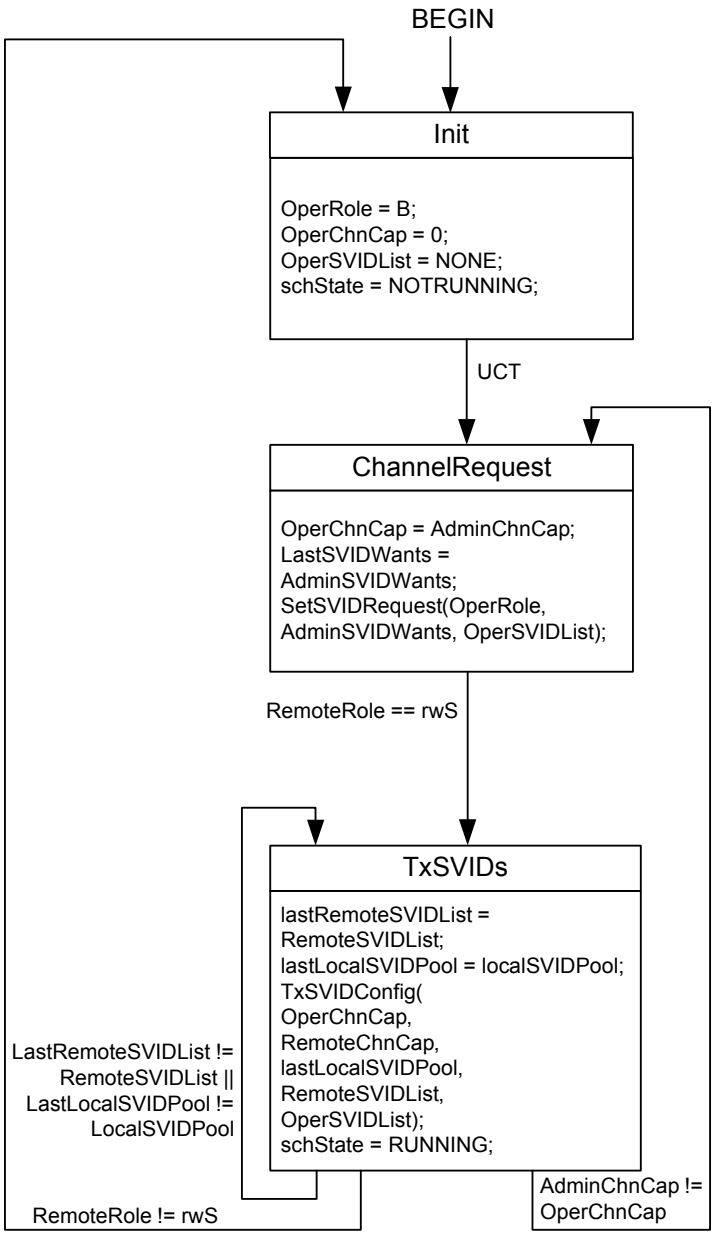


Figure 42-2—CDCP state machine—Bridge role

42.4 CDCP configuration variables

42.4.1 AdminChnCap

The administratively configured value for the Number of Channels supported parameter. This value is included as the ChnCap parameter in the CDCP TLV.

42.4.2 AdminRole

The administratively configured value for the local port's role parameter. The value of AdminRole is not reflected in the CDCP TLV. The AdminRole can take the value S or B; the value determines which of the CDCP state machines is instantiated. If AdminRole is S, the Station role state machine (Figure 42-1) is instantiated. If AdminRole is B, the Bridge role state machine (Figure 42-2) is instantiated. S indicates the sender is unwilling to accept S-channels configuration (mode, number of channels supported, channel index) from its neighbor and that the sender is willing to accept S-VID assignments from the neighbor. Stations usually take the S role. B indicates the sender is willing to accept S-channels configuration (mode, number of channels supported, channel index) from its neighbor and that the sender is willing do the best it can to fill the S-VID requests from the neighbor. Bridges usually take the B role.

42.4.3 AdminSVIDWants

The administratively configured value for (SCID, S-VID) pairs wanted by a station; it is not used by a bridge. The first value is always the pair (1, 1) for the default S-channel assignment. The S-channel numbers may be any valid number in the range 0–167. A 0 S-channel number indicates reserved space in the TLV. If the S-VID value is 0 it means the station is requesting any available S-VID. S-VID value 1 is reserved for exclusive use for the default S-channel S-VID. The AdminSVIDWants parameter is used to form the (SCID, S-VID) pairs in the CDCP TLV. This list is formed from the EVB station's S-channel interface table (12.26.4) and is used to build the EVB Bridge's S-channel interface table.

42.4.4 LastLocalSVIDPool

A temporary copy of the LocalSVIDPool.

42.4.5 LastRemoteSVIDList

Temporary local copy of the RemoteSVIDList. This variable is not included in the CDCP TLV. The LastRemoteSVIDList has the same syntax as RemoteSVIDList.

42.4.6 LastSVIDWants

A local temporary copy of the AdminSVIDWants.

42.4.7 LocalSVIDPool

The set of S-VIDs and bridge ports available for S-channel assignment. These are determined by both administrative resource assignments and by resource availability. The OperSVIDList for a B role is drawn from the LocalSVIDPool.

42.4.8 OperChnCap

The current value for the ChnCap parameter. This value is included as the ChnCap parameter in the local CDCP TLV. The range for this variable is 1–167.

42.4.9 OperRole

The current operational value of the Role parameter in the local port. This value is included as the Role parameter in the CDCP TLV and may take values S or B as described for AdminRole.

42.4.10 OperSVIDList

The current value for (SCID, S-VID) assignments. This is the list of (SCID, S-VID) pairs included in the local CDCP TLV. The total size of the list cannot exceed 167 pairs. The list always includes the default S-channel pair (1,1).The valid range for each S-channel of this list is from 1–167. The valid range for each S-VID in the list is from 0 to 0xffe. For the S role a S-VID of 0 indicates a request for a channel. For the B role an S-VID of 0 indicates a non-configured channel.

42.4.11 RemoteChnCap

The current value for the ChnCap parameter. This value is included as the ChnCap parameter in the remote CDCP TLV. NULL means no remote CDCP TLV exists in the local LLDP database. The range for this variable is 1–167.

42.4.12 RemoteRole

Indicates the value in the remote CDCP TLV role field. rrNull indicates either the TLV was not present in the last LLDP PDU or that no LLDP PDUs have been received. rwS and rwB indicate that the Role field was set in the CDCP TLV received and that it had a value of S or B respectively as described for the AdminRole variable.

42.4.13 RemoteSVIDList

The current value for (SCID, S-VID) assignments. This is the list of (SCID, S-VID) pairs included in the remote CDCP TLV. NULL means no remote CDCP TLV exists in the local LLDP database. If the list is empty but the CDCP TLV is present, its value is NONE. The total size of the list cannot exceed 167 pairs. The valid range for each S-channel of this list is from 1–167. The valid range for each S-VID in the list is from 0 to 0xffe. When the S-VID is value is 0 the S-VID is not configured. For the S role, a S-VID of 0 indicates a request for a channel. For the B role, an S-VID of 0 indicates a non-configured channel. The RemoteSVIDList is reflected within the EVB Bridge in the S-channel interface table (12.26.4).

42.4.14 RemoteVersion

The current value for the remote S-channel Vers parameter. This value is included as the Vers parameter in the remote CDCP TLV. NULL means no remote CDCP TLV exists in the local LLDP database. Setting the value of this variable to VER1=001b enables S-channel setup; setting the value to 000b stops S-channel operation.

42.4.15 schState

The current running state of the S-channel. The values for this variable are NOTRUNNING or RUNNING. This variable can be read using the management functionality defined in Clause 12.

42.5 CDCP configuration procedures

42.5.1 SetSVIDRequest (OperRole, AdminSVIDWants, OperSVIDList)

This function creates the OperSVIDList placed in the Local LLDP database, as follows:

a) If the OperRole for the equipment is B, then the OperSVIDList remains unchanged.
b) If the OperRole for the equipment is S, the function compares the AdminSVIDWants with the OperSVIDList and amends the OperSVIDList, as follows:

1) All active S-channels in the OperSVIDList that are in the AdminSVIDWants are kept active, and in addition, any channels not currently in the OperSVIDList are requested by including them in the OperSVIDList along with a 0 S-VID number. The OperSVIDList S-channel order is changed to match the AdminSVIDWants.
2) Any S-channels in the OperSVIDList that are not in AdminSVIDWants are made inactive and are removed from the OperSVIDList.

42.5.2 RxSVIDConfig (OperSVIDList, LastRemoteSVIDList)

This function creates the OperSVIDList placed in the Local LLDP database for an S role port.

The function compares the AdminSVIDWants with the LastRemoteSVIDList. For each AdminSVIDWants S-channel with an S-VID assignment in the LastRemoteSVIDList, a (SCID, S-VID) pair is generated in the OperSVIDList. For each AdminSVIDWants S-channel without an S-VID assignment in the LastRemoteSVIDList, a (SCID,0) pair is generated in the OperSVIDList. The OperSVIDList S-channel order is set to match the AdminSVIDWants.

42.5.3 TxSVIDConfig (OperChnCap, RemoteChnCap, LastLocalSVIDPool, RemoteSVIDList, OperSVIDList)

This function creates the OperSVIDList placed in the Local LLDP database for a B role port.

First the function takes the smaller of the OperChnCap and RemoteChnCap and truncates the RemoteSVIDList to the smaller of the two.

A new OperSVIDList is created as follows:

a) For each S-channel in the RemoteSVIDList with a (SCID, S-VID) pair in the OperSVIDList, the (SCID, S-VID) remains unchanged unless the S-VID is no longer part of the LastLocalSVIDPool. If the S-VID is no longer in the pool, a new one is selected if available. If no S-VID is available, the (SCID, S-VID) pair will be deleted from the OperSVIDList.
b) For each S-channel in the OperSVIDList without a (SCID, SVID) pair in the RemoteSVIDList, the (SCID, SVID) pair will be deleted from the OperSVIDList.
c) For a (SCID, SVID) pair in the remote list, where the S-VID is zero, an S-VID is assigned if it is available and the pair is inserted in the OperSVIDList. If an S-VID is not available, the pair is not inserted in the OperSVIDList.

Insert the following text, tables, and figures as new Clause 43:

43. Edge Control Protocol (ECP)

This clause provides an overview, detailed semantics, and state machines for the Edge Control Protocol (ECP).

43.1 Edge control protocol operation

Figure 43-1 depicts, at a high level, ECP operation. In step 1, the upper layer protocol (ULP) passes an outgoing ULP Data Unit (ULPDU) to ECP by invoking a transmit request procedure. In step 2, the ULPDU, which for some ULPs (e.g., VDP) may contain a set of ULP TLVs, is transmitted and an ECP low level acknowledgement timer is set. The frame is not yet deleted from the transmit buffer until an acknowledgement (L-ACK in the diagram) is received for that ECPDU. In step 3, the arriving ECP frame is received into a receive buffer, where it is held until it is removed by an ECP indication procedure that passes the ULP Data Unit to the associated upper level protocol. In step 4, when the receive buffer is emptied, a L-ACK is sent to the sender. In step 5, if the L-ACK is received before the L-ACK timer expires, then the transmit buffer is cleared and ECP can process another ULP PDU through the ECP procedure. However, if the L-ACK timer expires before the L-ACK is received, then the frame in the transmit buffer is re-sent and the L-ACK timer is re-initialized. This timeout and re-sending can occur up to a maximum number of retries determined by the value of the *maxRetries* parameter of the transmit state machine. If this number of retries is reached and there is still no response, then the transmit buffer is cleared, a failure counter is incremented, and the transmit state machine is then ready to process another ULP PDU. There is no indication to the ULP that a transmission failure has occurred; it is the ULP's responsibility both to detect the failure condition and to recover from it in an appropriate way.

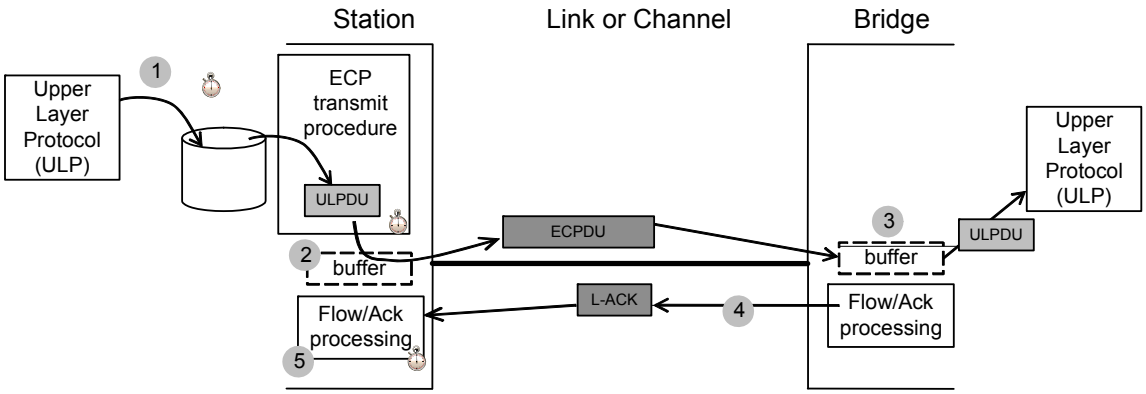


Figure 43-1—Example ECP exchange

ECP is intended to operate between two peers over an IEEE 802 LAN. ECP delivers the following service characteristics:

a) Reliable delivery of ULP PDUs, resilient against frame loss. The value of the *maxRetries* parameter determines the number of sequential lost frames that the protocol can sustain.
b) Delivery of ULP PDUs to the recipient ULP in the order that they were transmitted by the sending ULP.
c) Delivery of a single copy of each ULP PDU to the recipient.
d) Flow control that provides protection against buffer overrun on the receive side.

43.2 Edge Control Sublayer Service (ECSS)

Two service primitives model the hand-off of data units between the ULP and ECP: ECP_UNITDATA.request and ECP_UNITDATA.indication.

ECP_UNITDATA.request (ulptype, ulpdu)

The ECP_UNITDATA.request primitive is invoked by the ULP to notify ECP that a ULPDU is ready to be transmitted. The **ulpdu** parameter is the ULPDU that the ULP wishes to transmit. The **ulptype** parameter identifies the type of the ULP (see 43.3.3).

NOTE—For example, for VDP the ULPDU consists of a set of VDP TLVs passed to ECP for transmission. The maximum size of the ULPDU, and therefore the set of TLVs that it can contain, is determined by the maximum SDU size supported by the underlying MAC (see 6.5.8).

ECP_UNITDATA.indication (ulptype, ulpdu)

The ECP_UNITDATA.indication is invoked by ECP to indicate a ULPDU has been received and is available for ULP processing. The **ulpdu** parameter is the ULPDU that has been received. The **ulptype** parameter identifies the type of the ULP, as indicated in the received ECPDU (see 43.3.3).

43.3 Edge control protocol (ECP) and state machine

43.3.1 State machine conventions

The notational conventions used in the specification of ECP are as stated in Annex E.

43.3.2 Overview

There are two state machines used by each ECP instance: the ECP transmit state machine (43.3.4) and the ECP receive state machine (43.3.5). A Bridge Port that supports ECP shall support one instance of the ECP transmit state machine and one instance of the ECP receive state machine.

Initialization of the transmit and receive state machines occurs when portEnabled (43.3.7.5) is FALSE, or when a BEGIN global event occurs. The transmit state machine transmits an ECPDU in response to an ECP_UNITDATA.request from the ULP that indicates there is a PDU ready to be transmitted. The PDU is transmitted with a sequence number that is used by the (remote) receive state machine in a responding acknowledgement ECPDU. If no acknowledgement with the correct sequence number is received within a defined time period, and if the maximum number of retries has not been reached, the transmit state machine retransmits the ECPDU. If the maximum number of retries is exceeded, or if an acknowledgement is received that matches the last sequence number sent, then the transmit state machine increments the sequence number and waits for the next ECP_UNITDATA.request.

NOTE—The sequence number for the first ECPDU transmitted after a state machine initialization (which occurs when BEGIN is TRUE or portEnabled [43.3.7.5] is FALSE) is an implementation choice; for example, it could be a pre-determined number, a random number, or it could continue the sequence from the last sequence number used.

When the first ECPDU is received following initialization, the receive state machine initializes its local record of the last sequence number received to be one less than the sequence number in the received ECPDU. This record of the last sequence number received allows the state machine to detect whether the received ECPDU has been received already (current and last sequence numbers match) or this is a new ECPDU (current and last sequence numbers differ). In both cases, the receive state machine sends an acknowledgement ECPDU, using the current sequence number. In the case that the received ECPDU is new,

the last received sequence number is updated to reflect the sequence number of the received ECPDU, and an ECP_UNITDATA.indication is sent to the ULP to pass the contents of the ECPDU to the service user.

43.3.3 Edge control protocol data unit (ECPDU)

This subclause specifies the format of a ECPDU, along with the header that is added to and removed from ECP frames by the ECP function. The ECP header allows each ECPDU from the sender to be identified through a sequence number, which the receiver acknowledges by sending a ECP Acknowledgement frame. The format of the ECPDU is illustrated in Figure 43-2.

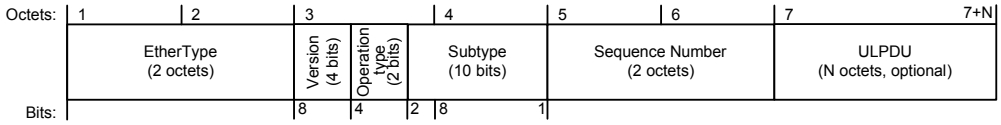


Figure 43-2—ECPDU structure

The destination address of the Ethernet frame that contains a ECPDU is specified by the ULP. The address used is either an individual MAC address or one of the reserved addresses specified in Table 8-1.

The source address shall be the individual MAC address of the sending station or port.

The fields of the ECPDU are defined in the following subclauses.

43.3.3.1 EtherType

A 16-bit field that contains the EtherType assigned for use by ECP (89-40).

43.3.3.2 Version

A 4-bit field that identifies the protocol version. The version shall be 0x01.

43.3.3.3 Operation type

A 2-bit field that identifies the operation type as follows:

a) ECP request (0x0).
b) ECP acknowledgement (0x1).

43.3.3.4 Sub-type

A 10-bit field that defines the ULP type included in the PDU. For ACKs the sub-type is ignored at the station. The sub-type used by VDP is as shown in Table 43-1.

Table 43-1—ECP sub-types

Use	Reference	Sub-type
VDP	Clause 41	0x0001
Port Extender Control and StatusProtocol (PE CSP)	IEEE Std 802.1BR	0x0002
Reserved for future standardization		All other values

43.3.3.5 Sequence number

A 2-octet field that identifies the sequential order of the PDU, with respect to other ECPDUs. The starting sequence number can start anywhere for the first ECPDU, but the sequence number for each subsequent new request ECPDU is incremented by 1 modulo 65536.

NOTE—The sequence numbers used by each instance of the ECP transmit state machine are independent of each other.

43.3.3.6 ULPDU

This field contains an upper layer protocol data unit (ULPDU) if the operation type in the Mode field is ECP request; the field is absent if the operation type is ECP acknowledgement.

43.3.4 ECP transmit state machine

The ECP transmit state machine shall implement the function specified by the state diagram in Figure 43-3 and the attendant definitions in 43.3.6 through 43.3.8.

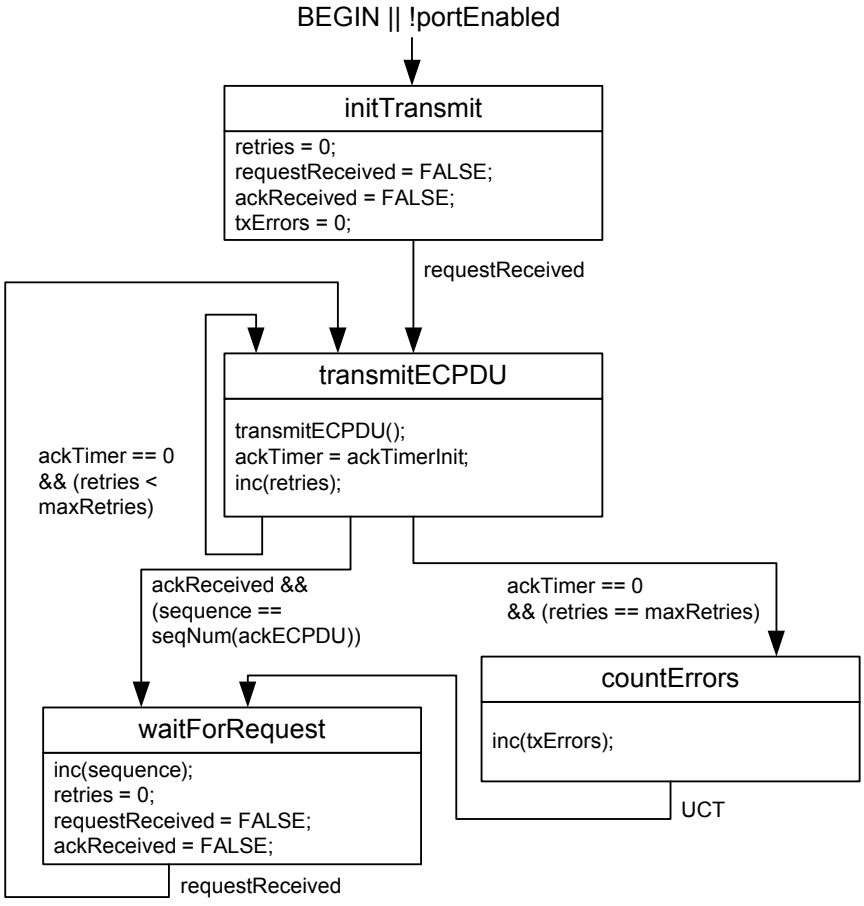


Figure 43-3—ECP transmit state machine

43.3.5 ECP receive state machine

The ECP receive state machine shall implement the function specified by the state diagram in Figure 43-4 and the attendant definitions in 43.3.6 through 43.3.8.

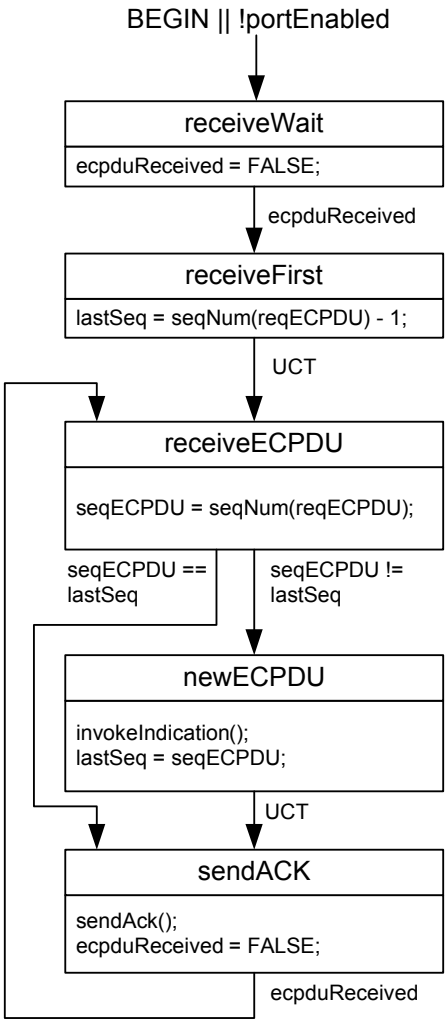


Figure 43-4—ECP receive state machine

43.3.6 ECP state machine timers

A set of timers is used by the ECP state machines; these operate as countdown timers (i.e., they expire when their value reaches zero). These timers are based on the Timer Exp data type (12.3). They

a) Have a resolution of ten microseconds, with a tolerance of ± 20%.
b) Are started by loading an initial integer value, n, where $0 < n \leq 2^{31}$.
c) Are decremented by one per timer tick, as long as $n > 0$; the interval between timer ticks is the same as the timer resolution.
d) Represent the remaining time in the period.

43.3.6.1 ackTimer

The ackTimer is used to determine how long the transmit state machine will wait for an acknowledgement PDU to be received before it either retries a transmission or aborts a transmission due to too many retries. This timer is initialized using the value of ackTimerInit determined as stated in D.2.13.6.

43.3.7 ECP state machine variables and parameters

43.3.7.1 ackReceived

This Boolean variable is set to TRUE when an ECPDU is received with a MODE field indicating that the PDU is an ACK. The variable is set FALSE by the ECP transmit state machine once the ACK has been processed.

43.3.7.2 ecpduReceived

This Boolean variable is set to TRUE when an ECPDU is received with a MODE field indicating that the PDU is a request. The variable is set FALSE by the ECP receive state machine once the request has been processed and the ACK has been sent.

43.3.7.3 lastSeq

This integer variable is used to record the previous received sequence number.

43.3.7.4 maxRetries

This integer variable defines the maximum number of times that the ECP transmit state machine will retry a transmission if no ACK is received. The default value of maxRetries is 3; this variable can be changed by management as documented in 12.26.2. The value is derived from ecpOperMaxTries, Table 12-26.

43.3.7.5 portEnabled

This Boolean variable is set to the value of the MAC_Operational parameter (6.6.2) for the Port.

43.3.7.6 requestReceived

This Boolean variable is set to TRUE when a ULP issues an ECP_UNITDATA.request primitive. The variable is set FALSE by the state machine once the request has been processed.

43.3.7.7 retries

This integer variable counts the number of transmission retries that have been made for the current ECPDU.

43.3.7.8 seqECPDU

This integer variable is used to record the sequence number contained in the most recent received request ECPDU.

43.3.7.9 sequence

This integer variable is used to record the current sequence number that is used in transmitted request ECPDUs.

43.3.7.10 txErrors

This integer variable is used to count the number of times that the ECP transmit state machine has retransmitted an ECPDU.

43.3.8 ECP state machine procedures

43.3.8.1 inc(counter)

This procedure increments the counter variable by 1 modulo 65536.

43.3.8.2 transmitECPDU()

This procedure causes an ECPDU to be transmitted, using the PDU structure defined in 43.3.3. The sequence number field is set to the least significant 16 bits of the current sequence number contained in the sequence variable (43.3.7.9). The mode field is set to ECP request. The ULPDU field is set to the value of the ulpdu parameter of the request primitive. The subtype field is set to the value of the ulptype parameter of the request primitive.

43.3.8.3 invokeIndication()

This procedure causes an ECP_UNITDATA.indication primitive to be invoked in order to pass the contents of an incoming ECPDU to the ECP service user. The ulptype parameter carries the value of the ULP type carried in the ECPDU. The ulpdu parameter carries the value of the ULPDU field of the ECPDU.

43.3.8.4 sendAck()

This procedure causes an ECPDU to be transmitted, using the PDU structure defined in 43.3.3. The sequence number field is set to the least significant 16 bits of the sequence number contained in the seqECPDU variable (43.3.7.8). The mode field is set to ECP acknowledgement. The ULPDU field is absent. The subtype field is set to the value of the ulptype parameter of the received request ECPDU.

43.3.8.5 seqNum(pdu-type)

This procedure returns an integer value equal to the value of the most recently received request ECPDU (pdu-type = reqECPDU) or acknowledgement ECPDU (pdu-type = ackECPDU).

Annex A

(normative)

PICS proforma—Bridge implementations[9]

A.5 Major capabilities

Insert the following rows at the end of the table:

EVB-B	Does the implementation support the functionality of an EVB Bridge?	O	5.22	Yes [ ] No [ ]
EVB-S	Does the implementation support the functionality of an EVB station?	O	5.23	Yes [ ] No [ ]

A.21 MVRP

Change items MVRP1 and MVRP2 in A.21 as follows:

MVRP1	Does the implementation support the exchange of MMRPDUs, using the generic MRPDU format defined in 11.2 to exchange ~~MMRP~~MVRP-specific information, as defined in 10.12?	M	5.4.2, 10.8, 11.2	Yes [ ]
MVRP2	Is the ~~MMRP~~MVRP Application supported as defined in 11.2?	M	5.4.2, 11.2	Yes [ ]

Insert new A.38 through A.42 at the end of the clause, renumbering if necessary, as follows:

A.38 EVB Bridge

Item	Feature	Status	Reference	Support
	If EVB Bridge functionality (EVB-B in Table A.5) is not supported, mark N/A and ignore the remainder of this table.			N/A [ ]
EVB-B-1	Does the implementation comprise a single conformant C-VLAN component?	M	5.5, 5.6, 5.22	Yes [ ]
EVB-B-2	Is each externally accessible port capable of being configured as either a C-VLAN Bridge Port or a Station Facing Bridge Port (SBP)?	M	5.22, 40	Yes [ ]
EVB-B-3	Does the implementation support the functionality of a C-VLAN component?	M	5.5, 5.22	Yes [ ]

[9]*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.38 EVB Bridge *(continued)*

Item	Feature	Status	Reference	Support
EVB-B-4	Does the implementation support at least one SBP on the C-VLAN component?	M	5.22, 40	Yes [ ]
EVB-B-5	Does the implementation support the EVB status parameters for EVBMode = EVB Bridge?	M	5.22, 6.6.6	Yes [ ]
EVB-B-6	Does the implementation support an LLDP nearest Customer Bridge database including the EVB TLV on each SBP?	M	5.22, D.2.13	Yes [ ]
EVB-B-7	Does the implementation support ECP on each SBP?	M	5.22, 43	Yes [ ]
EVB-B-8	Does the implementation support the Bridge role of VDP on each SBP?	M	5.22, 41	Yes [ ]
EVB-B-9	Does the implementation support at least one Port-mapping S-VLAN component and associated UAP configured as specified in 40.2 (a)-(d)?	O	5.22, 22.6.4, 40.2 (a)-(d)	Yes [ ] No [ ]
EVB-B-10	Is each externally accessible port capable of being configured as an Uplink Access Port (UAP)?	O	5.22, 40	Yes [ ] No [ ]
EVB-B-11	Does the implementation support CDCP, as specified in Clause 42, operating in Bridge mode?	EVB-B-9:M	42, 42.3	Yes [ ] N/A [ ]
EVB-B-12	Does the implementation support the enhanced filtering utility criteria and not support the default filtering utility criteria (8.7)?	EVB-B-9:M	8.7	Yes [ ] N/A [ ]
EVB-B-13	Does the implementation support configuration of reflective relay on each SBP of the C-VLAN component?	O	5.22, 6.6.6, 8.6.1	Yes [ ] No [ ]
EVB-B-14	Does the implementation support management for the EVB components?	O	5.22, 12.4-12.12,12.26	Yes [ ] No [ ]
EVB-B-15	Does the implementation support an SNMP management MIB module?	O	5.22, 17.7.20	Yes [ ] No [ ]
EVB-B-16	Does the implementation support assignment of VIDs to GroupIDs?	O	5.22, 41.2.9	Yes [ ] No [ ]
EVB-B-17	Does the implementation support the use of the M and S bits in VDP?	O	5.22, 41.2.3	Yes [ ] No [ ]
EVB-B-18	Does the Bridge reserve the S-channel identifier (SCID) value 1 and S-VID value 1 for the exclusive use as the un-S-tagged default S-channel	M	42.1	Yes [ ]

A.39 EVB station

Item	Feature	Status	Reference	Support
	If EVB station functionality (EVB-S in Table A.5) is not supported, mark N/A and ignore the remainder of this table.			N/A [ ]
EVB-S-1	Does the EVB station comprise one or more conformant ER components?	M	5.6, 5.23.1	Yes [ ]
EVB-S-2	Is each externally accessible port capable of being configured as at least one of: – An Uplink Access Port (UAP); – An Uplink relay port (URP)?	M	5.23, 40	Yes [ ]
EVB-S-3	Is each DRP capable of attaching its ER to one or more VSIs?	M	5.23, 40	Yes [ ]
EVB-S-4	Is each URP capable of attaching its ER to a point-to-point LAN connecting the URP to a CAP, or to the LAN connecting to an EVB Bridge in the case where no Port-mapping S-VLAN component is present?	M	5.23, 40	Yes [ ]
EVB-S-5	Does the implementation support at least one ER?	M	5.23, 40	Yes [ ]
EVB-S-6	Does the implementation support at least one accessible URP?	M	5.23, 40	Yes [ ]
EVB-S-7	Does the implementation support the EVB status parameters for EVBMode = EVB station on each URP?	M	5.23, 6.6.6	Yes [ ]
EVB-S-8	Does the implementation support an LLDP Nearest Customer Bridge database including the EVB TLV on each URP of each ER?	M	5.23, D.2.13	Yes [ ]
EVB-S-9	Does the implementation support ECP on each URP of each ER?	M	5.23, 43	Yes [ ]
EVB-S-10	Does the implementation support the station role of VDP for each URP of each ER?	M	5.23, 41	Yes [ ]
EVB-S-11	Does the implementation support a Port-mapping S-VLAN component on each Port configured as a UAP, configured as specified in 40.2 (a)–(d)?	O	5.23, 22.6.4, 40.2 (a)–(d)	Yes [ ] No [ ]
EVB-S-12	Does the implementation support CDCP, as specified in Clause 42, operating in Station mode?	EVB-S-11:M	42, 42.3	Yes [ ] N/A [ ]
EVB-S-13	Does the implementation support the enhanced filtering utility criteria (8.7.2) and not support the default filtering utility criteria (8.7.1)?	EVB-S-11:M	8.7.1, 8.7.2	Yes [ ] N/A [ ]

A.39 EVB station *(continued)*

Item	Feature	Status	Reference	Support
EVB-S-14	Does the implementation support multiple ERs?	O	5.23, 40	Yes [ ] No [ ]
EVB-S-15	Does the implementation support management for the EVB components?	O	5.23, 12.26	Yes [ ] No [ ]
EVB-S-16	Does the implementation support an EVB station SNMP management MIB module?	O	5.23, 17.7.20	Yes [ ] No [ ]
EVB-S-17	Does the implementation support assignment of VIDs to GroupIDs?	O	5.23, 41.2.9	Yes [ ] No [ ]
EVB-S-18	Does the implementation support Support the use of the M and S bits in VDP?	O	5.23, 41.2.3	Yes [ ] No [ ]

A.40 Edge relay

Item	Feature	Status	Reference	Support
	If EVB station functionality (EVB-S in Table A.5) is not supported, mark N/A and ignore the remainder of this table.			N/A [ ]
ERC-1	Does the ER conform to the relevant standard for the Media Access Control technology implemented at each Port in support of the MAC ISS, as specified in 6.6, 6.7, and 6.14?	M	6.6, 6.7, 6.14	Yes [ ]
ERC-2	Does the ER support the MAC Enhanced Internal Sublayer Service at each Port, as specified in 6.8 and 6.9?	M	6.8, 6.9	Yes [ ]
ERC-3	Does the ER recognize and use C-TAGs?	M	6.9	Yes [ ]
ERC-4	Does the ER relay and filter frames as described in 8.1 and specified in 8.5, 8.6, 8.7, and 8.8?	M	8.5, 8.6, 8.8	Yes [ ]
ERC-5	Does the ER support a PVID value, and configuration of at least one VID whose untagged set includes that Port, on each DRP that supports untagged and priority-tagged frames?	M	6.9, 8.8.2	Yes [ ]
ERC-6	Does the ER support setting the Acceptable Frame Types parameter to *Admit Only VLAN Tagged Frames* on the URP?	M	5.23.1, 6.9	Yes [ ]
ERC-7	Does the ER allow tag headers to be inserted, modified, and removed from relayed frames, as specified in 8.1 and Clause 9, as required by the value(s) of the Acceptable Frame Types parameter supported on each Port, and by the ability of each Port to transmit VLAN-tagged and/or untagged frames?	M	8.1, Clause 9	Yes [ ]
ERC-8	Does the ER support at least one FID?	M	6.6, 8.8.3, 8.8.8, 8.8.9	Yes [ ]
ERC-9	Does the ER allow allocation of at least one VID to each FID that is supported?	M	6.6, 8.8.3, 8.8.8, 8.8.9	Yes [ ]
ERC-10	Does the ER support exactly one URP supporting the parameters of 6.6.6 for EVBMode = EVB station?	M	5.23.1, 6.6.6, 40	Yes [ ]
ERC-11	Does the ER support one or more DRPs each supporting access to VSIs?	M	5.23.1, 40	Yes [ ]
ERC-12	Does the ER filter the Reserved MAC Addresses?	M	5.23.1, Table 8-1	Yes [ ]
ERC-13	Does the ER support more than one DRP?	O	5.23.1	Yes [ ] No [ ]

A.40 Edge relay *(continued)*

Item	Feature	Status	Reference	Support
ERC-14	Does the ER support setting the Enable Ingress Filtering parameter (8.6.2) on each DRP?	ERC-13:M	5.23.1, 8.6.2	Yes [ ] N/A [ ]
ERC-15	Does the ER support setting the Enable Ingress Filtering parameter (8.6.2) on each URP?	ERC-13:M	5.23.1, 8.6.2	Yes [ ] N/A [ ]
ERC-16	Does the ER support the requirements of either a VEB ER or a VEPA ER?	M	5.23.1, 5.23.1.1, 5.23.1.2	Yes [ ]
ERC-17	Does the ER support a PVID value, and configuration of at least one VID whose untagged set includes that Port, if the URP supports untagged and priority-tagged frames?	O	6.9, 8.8.2	Yes [ ]
ERC-18	Does the ER comprise a single conformant C-VLAN component?	O	5.4	Yes [ ] No [ ]
ERC-19	Does the ER support disabling of learning on each DRP?	O	5.23.1, 8.6.1	Yes [ ] No [ ]
ERC-20	Does the ER support discarding frames with unregistered source addresses at each DRP?	O	5.23.1, 8.8.1	Yes [ ] No [ ]
ERC-21	Does the ER support the operation of the learning process?	O	8.7	Yes [ ] No [ ]

A.41 VEB and VEPA edge relay components

Item	Feature	Status	Reference	Support
	If EVB station functionality (EVB-S in Table A.5) is not supported, mark N/A and ignore the remainder of this table.			N/A [ ]
VERC-1	Does the ER component support VEB functionality?	O.6	5.23.1.1	Yes [ ] No [ ]
VERC-2	Does the ER component support VEPA functionality?	O.6	5.23.1.2	Yes [ ] No [ ]
VERC-3	Does the ER component request that reflective relay service not be provided by setting adminReflectiveRelayRequest to FALSE?	VERC-1:M	5.23.1.1	Yes [ ] N/A [ ]
VERC-4	Does the VEPA ER disable learning on the URP?	VERC-2:M	5.23.1.2, 8.6.1	Yes [ ] N/A [ ]
VERC-5	Does the VEPA ER filter frames received at each URP that are destined to a DRP that originated the frame?	VERC-2:M	5.23.1.2, 8.6.1	Yes [ ] N/A [ ]
VERC-6	Does the VEPA ER request reflective relay service by setting adminReflectiveRelayRequest to True?	VERC-2:O	5.23.1.2, 6.6.6	Yes [ ] N/A [ ]
VERC-7	Does the ER filter frames received at each DRP that are destined for the URP until reflective relay is enabled?	VERC-2:O	5.23.1.2, 8.6.1.1	Yes [ ] No [ ] N/A [ ]
VERC-8	Does the ER forward frames as specified in 8.6.3.1?	VERC-2:M	5.23.1.2, 8.6.3.1	Yes [ ] N/A [ ]

A.42 VDP, CDCP, and ECP

Item	Feature	Status	Reference	Support
	If neither EVB station functionality (EVB-S in Table A.5) nor EVB Bridge functionality (EVB-B in Table A.5) is supported, mark N/A and ignore the remainder of this table.			N/A [ ]
VDP-1	Does the implementation support the Bridge VDP state machine as specified in Clause 41?	EVB-B:M	41, 41.5.2	Yes [ ]
VDP-2	Does the implementation support the Station VDP state machine as specified in Clause 41?	EVB-S:M	41, 41.5.3	Yes [ ]
CDCP-1	Does the implementation support the CDCP configuration state machine for the Bridge role, as specified in Clause 42?	EVB-B AND EVB-B-9: M	42, 42.3	Yes [ ] N/A[ ]
CDCP-2	Does the implementation support the CDCP configuration state machine for the station role, as specified in Clause 42?	EVB-S AND EVB-S-11: M	42, 42.3	Yes [ ] N/A[ ]
ECP-1	Does the implementation support the ECP transmit state machine as specified in Clause 43?	M	43, 43.3.4	Yes [ ]
ECP-2	Does the implementation support the ECP receive state machine as specified in Clause 43?	M	43, 43.3.5	Yes [ ]

Annex D

(normative)

IEEE 802.1 Organizationally Specific TLVs

D.1 Requirements of the IEEE 802.1 Organizationally Specific TLV set

Change the third paragraph as follows:

The currently defined IEEE 802.1 Organizationally Specific TLVs specified in this standard are listed in Table D-1. Other standards can also define IEEE 802.1 Organizationally Specific TLVs. The "TLV set name" column identifies the TLV set to which each TLV belongs. Any additions or changes to these TLVs will be included in this annex.

Delete the last row of Table D-1, insert two new rows at the end of the table, change the table title, and add the footnote and NOTE to the table, as follows:

Table D-1— IEEE 802.1 Organizationally Specific TLVs specified in this standard

IEEE 802.1 subtype	TLV name	TLV set name	TLV reference	Feature clause reference
~~08–FF~~	~~Reserved~~		~~—~~	
0D	EVB TLV	evbSet	D.2.13	D.2.13
0E	CDCP TLV	evbSet	D.2.14	D.2.14

D.2 Organizationally Specific TLV definitions

Insert the following paragraph:

In the TLV definitions that follow, any fields that are labelled as "Reserved" are transmitted as zero and ignored on receipt.

Insert new subclauses D.2.13 and D.2.14 as shown, following any existing subclauses that define TLVs, renumbering the subclauses, tables, and figures as necessary.

D.2.13 EVB TLV

The EVB TLV is used to

a) Advertise a station or bridge's EVB capabilities.
b) Negotiate and activate common capabilities.

The EVB TLV is exchanged via LLDP and conforms to the LLDP TLV specification. The LLDP database carrying the EVB TLV is addressed using the Nearest Customer Bridge address. One LLDP database is built at the URP of each ER.

The EVB TLV allows setting the EVB Bridge's C-VLAN component Port to operate in reflective relay. Reflective relay is implemented by changing the active topology enforcement rules described in 8.6.1 to allow forwarding on the reception Bridge Port. When reflective relay is enabled on a given Bridge Port, that port is a potential transmission port for frames received on that port.

The EVB TLV structure is illustrated in Figure D-9.

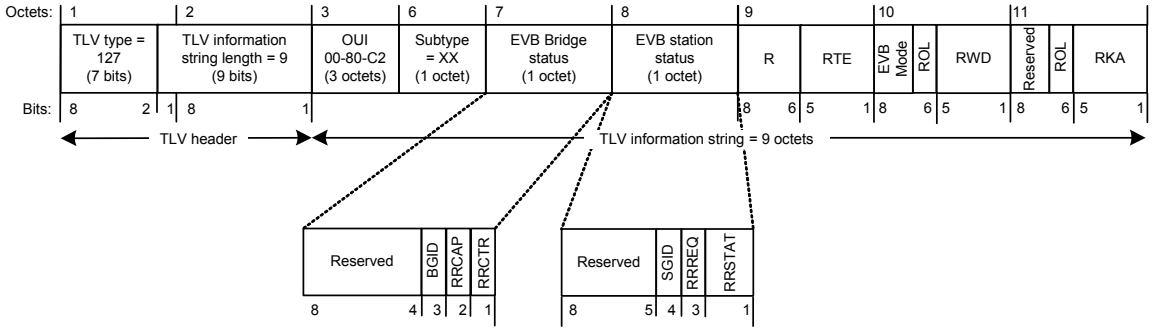


Figure D-9—EVB TLV format

The EVB TLV information string fields are as defined in D.2.13.1 through D.2.13.9.

D.2.13.1 OUI

The OUI used to identify the EVB TLV is the 802.1 OUI 00-80-C2.

D.2.13.2 Subtype

The subtype used to identity the EVB TLV is as shown in Table D-1.

D.2.13.3 EVB Bridge status

The EVB Bridge status field describes EVB capabilities that are supported by the EVB Bridge. If the sender of the TLV is an EVB Bridge (EVB Mode = EVB Bridge—see D.2.13.7), then the field reflects its own capabilities; if the sender of the TLV is an EVB station (EVB Mode = EVB station), then the field reflects the capabilities received from an attached EVB Bridge, or a value of zero if no TLV has been received from an attached EVB Bridge.

Each capability is represented by a single bit flag; a value of TRUE (1) indicates that the capability is supported, a value of FALSE (0) indicates that the capability is not supported.The capabilities are as defined in D.2.13.3.1 through D.2.13.3.3.

D.2.13.3.1 BGID

A value of TRUE indicates that the EVB Bridge wishes to control VID assignments and use the GroupID in VDP exchanges. A value of FALSE indicates that the EVB Bridge does not wish to make use of the Group ID in VDP exchanges.

If the EVB station sets SGID = TRUE, and the EVB Bridge also sets BGID = TRUE, then the EVB Bridge can control VID assignments and use the GroupID in VDP exchanges.

If the EVB station does not set SGID = TRUE, or the EVB Bridge does not set BGID = TRUE, then the EVB Bridge cannot control VID assignments or use the GroupID in VDP exchanges.

D.2.13.3.2 RRCAP

The RRCAP flag indicates the state of the EVB Bridge's reflectiveRelayCapable parameter (6.6.6.2).

If the EVB Bridge's reflectiveRelayCapable parameter is TRUE, and a TLV has been received by the EVB Bridge from an attached EVB station in which the value of RRREQ (D.2.13.4.2) is also TRUE, then the value of the EVB Bridge's operReflectiveRelayControl parameter (6.6.6.2) shall be set to TRUE. Otherwise, the value of the EVB Bridge's operReflectiveRelayControl parameter (6.6.6.2) shall be set to FALSE.

D.2.13.3.3 RRCTR

The RRCTR flag indicates the state of the EVB Bridge's operReflectiveRelayControl parameter (6.6.6.2).

D.2.13.4 EVB station status

The EVB station status field describes EVB capabilities that are supported by the EVB station. If the sender of the TLV is an EVB station (EVB Mode = EVB station—see D.2.13.7), then the field reflects its own capabilities; if the sender of the TLV is an EVB Bridge (EVB Mode = EVB Bridge), then the field reflects the capabilities received from an attached EVB station, or a value of zero if no TLV has been received from an attached EVB station.

Each capability is represented by a single bit flag; a value of TRUE (1) indicates that the capability is supported, a value of FALSE (0) indicates that the capability is not supported.The capabilities are as defined in D.2.13.3.1 through D.2.13.3.3.

D.2.13.4.1 SGID

A value of TRUE indicates that the EVB station can support the use of the GroupID.

If the EVB station sets SGID = TRUE, and the EVB Bridge also sets BGID = TRUE, then the EVB Bridge can control VID assignments and use the GroupID in VDP exchanges.

If the EVB station does not set SGID = TRUE, or the EVB Bridge does not set BGID = TRUE, then the EVB Bridge cannot control VID assignments or use the GroupID in VDP exchanges.

D.2.13.4.2 RRREQ

The RRREQ flag indicates the state of the EVB station's adminReflectiveRelayRequest parameter (6.6.6.3).

D.2.13.4.3 RRSTAT

RRSTAT is a composite flag that indicates the state of the EVB station's operReflectiveRelayStatus parameter (6.6.6.3) as shown in Table D-2.

If a TLV has been received by the EVB station from an attached EVB Bridge in which the value of RRCTR (D.2.13.3.3) is TRUE, then the value of the EVB station's operReflectiveRelayStatus parameter (6.6.6.3) shall be set to TRUE. If a TLV has been received by the EVB station from an attached EVB Bridge in which the value of RRCTR (D.2.13.3.3) is FALSE, then the value of the EVB station's operReflectiveRelayStatus parameter (6.6.6.3) shall be set to FALSE. If no TLV has been received by the EVB station from an attached EVB Bridge, then the value of the EVB station's operReflectiveRelayStatus parameter (6.6.6.3) shall be set to Unknown.

Table D-2—RRSAT flag values and meanings

Bit 1	Bit 2	Meaning
TRUE	FALSE	operReflectiveRelayStatus is TRUE
FALSE	FALSE	operReflectiveRelayStatus is FALSE
TRUE	TRUE	operReflectiveRelayStatus is Unknown
FALSE	TRUE	operReflectiveRelayStatus is Unknown

D.2.13.5 R

This field carries the maxRetries value for the ECP state machine (43.3.7.4). Both sides use the largest of the two values of R. If no remote value is available, then the local value is used.

D.2.13.6 Retransmission Exponent (RTE)

RTE is an EVB link or S-channel attribute used to calculate the minimum ECPDU retransmission time, ackTimerInit. The value of ackTimerInit is calculated as

10×2^{RTE} microseconds

Both sides use the largest of the two values of RTE for this calculation. If no remote value is available, then the greater of 2 ms and local value is used.

D.2.13.7 EVB Mode

The EVB Mode field represents the value of the EVBMode parameter (6.6.6) for the sender of the TLV, as shown in Table D-4.

Table D-4—EVB Mode values

EVBMode (6.6.6)	Field value
Not Supported	0
EVB Bridge	1
EVB station	2
Reserved for future standardization	3

D.2.13.8 Remote or Local (ROL) and Resource Wait Delay (RWD)

The RWD values transmitted by the EVB Bridge and EVB station indicate the exponent value that each device proposes for determining the value of the resourceWaitDelay variable (41.5.5.7). The value of resourceWaitDelay is calculated as

10×2^{RWD} microseconds

Both sides use the largest of the local and remote values of RWD for this calculation; if there is no remote value available, the local (proposed) value is used. The Remote or Local (ROL) flag is used by the EVB station to indicate whether the remote RWD value is in use (TRUE) or the local value is in use (FALSE).

D.2.13.9 Remote or Local (ROL) and Reinit Keep Alive (RKA)

The RKA value transmitted by the EVB station indicates the exponent value in use by the EVB station for determining the value of the reinitKeepAlive variable (41.5.5.5). The value of reinitKeepAlive is calculated as

$$10 \times 2^{RKA} \text{ microseconds}$$

Both sides use the largest of the two values of RKA for this calculation; if there is no remote value available, the local value is used. The Remote or Local (ROL) flag is used by the EVB Bridge to indicate whether the remote RKA value is in use (TRUE) or the local value is in use (FALSE). In both cases, the EVB Bridge transmits the exponent value being used for its toutKeepAlive variable.

D.2.14 CDCP TLV

The EVB station and Bridge both use the same LLDP TLV to configure S-channels (see Figure D-10). This TLV is in LLDP OUI format (8.6 of IEEE Std 802.1AB). The S-channel's capabilities, requests and running configuration is encoded in the information string of this TLV as defined in D.2.14.1 through D.2.14.8.

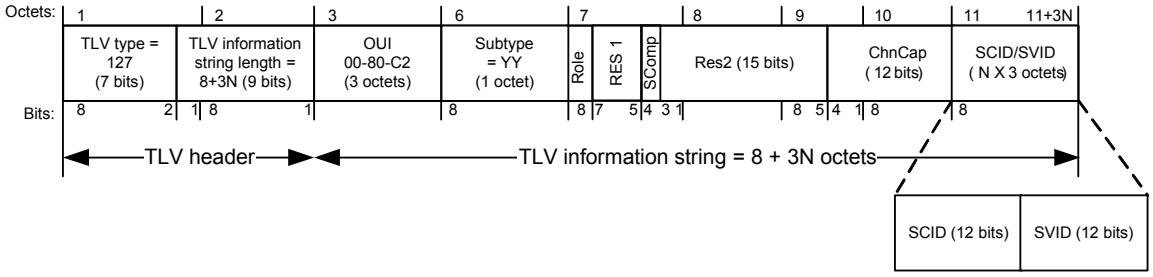


Figure D-10—CDCP TLV structure

D.2.14.1 OUI

The OUI used to identify the CDCP TLV is the 802.1 OUI 00-80-C2.

D.2.14.2 Subtype

The subtype used to identity the CDCP TLV is as shown in Table D-1.

D.2.14.3 Role

Role is a 1-bit field, defined as follows:

a) S(1)—Indicates the sender is operating in the station role, assigns channels numbers and a default S-VID for the default channel 1, and requests S-VID assignments from the neighboring 'B'.
b) B(0)—Indicates the sender is operating in the bridge role, accepts S-channel configuration requests from its neighboring 'S' and that the sender will do the best it can to fill the S-VID assignment requests from the neighboring 'S'.

D.2.14.4 RES1

RES1 is a 3-bit field, reserved for future standardization. This field is transmitted as zero and ignored on receipt.

D.2.14.5 SComp

SComp is a 1-bit field that indicates the presence or absence of an S-VLAN component for S-channel support. A value of 1 indicates TRUE, zero indicates FALSE.

NOTE—If this bit is zero, then the sender does not have a CDCP state machine, and the other fields in the TLV are not valid.

D.2.14.6 Res2

RES2 is a 15-bit field, reserved for future standardization. This field is transmitted as zero and ignored on receipt.

D.2.14.7 ChnCap

Channel capacity. Identifies the total number of S-channels, both assigned and available to be assigned, that the sender has.

D.2.14.8 SCID/S-VID

An SCID/S-VID pair exists for each S-channel that is currently supported by the sender. Each SCID/S-VID pair consists of two 12-bit values, as follows:

a) SCID—indicates the index number of the S-channel. The station assigns S-channel numbers in the range 0-167. Zero is reserved. The S-channel index should be between 1 and the maximum number of S-channels supported by the port.
b) S-VID—The VLAN ID assigned to the S-channel. The Bridge assigns SVIDs to channels in the range 1-0xffe. A station uses the 0 S-VID to request an S-VID assignment from the Bridge.

After the station receives the S-VID assignment from the Bridge, it uses the S-VID assigned value in all subsequent exchanges for that specific SCID (SCID/S-VID pair).

NOTE 1—The first entry in the list of SCID/S-VID pairs contains the default S-channel. (i.e., the first channel pair is <1,1>).

NOTE 2—A maximum of 167 S-channels can be supported. Other formats (assuming sequential SVIDs) could be defined to allow support for 4K+ S-channels.

NOTE 3—This listing can be sparse (in order to indicate arrival and removal of S-channels). The S-channel going away is indicated by removing the SCID/S-VID pair.

NOTE 4—The order of the list determines the priority of S-VID assignments. If the Bridge does not have resources for all channels, it assigns the first channels in the list.

D.3 IEEE 802.1 Organizationally Specific TLV management

D.3.2 IEEE 802.1 managed objects—TLV variables

Insert new subclauses as follows:

D.3.2.9 EVB TLV managed objects

a) **EVB TLV:** see D.2.13.

D.3.2.10 CDCP TLV managed objects

a) **CDCP TLV:** see D.2.14.

D.4 IEEE 802.1/LLDP extension MIB

Insert the following rows in existing Table D-5, at the end of the Configuration group section, as shown:

lldpXdot1EvbConfigEvbTable		Augments lldpV2Xdot1LocManVidEntry
	lldpXdot1EvbConfigEvbTxEnable	Normal LLPDUs, 9.1.2.1 of IEEE Std 802.1AB
lldpXdot1EvbConfigCdcpTable		Augments lldpV2Xdot1LocManVidEntry
	lldpXdot1EvbConfigCdcpTxEnable	Normal LLPDUs, 9.1.2.1 of IEEE Std 802.1AB

Insert the following rows in existing Table D-5, at the end of the Local system information section, as follows:

lldpV2Xdot1LocEvbTlvTable		D.2.13
	lldpV2LocPortIfIndex	(Table index)
	lldpV2Xdot1LocEvbTlvString	EVB TLV string, D.2.13
lldpV2Xdot1LocCDCPTlvTable		D.2.14
	lldpV2LocPortIfIndex	(Table index)
	lldpV2Xdot1LocCDCPTlvString	CDCP TLV string, D.2.14

Insert the following rows in existing Table D-5, at the end of the Remote system information, as follows:

lldpV2Xdot1RemEvbTlvTable		D.2.13
	lldpV2RemTimeMark	(Table index)
	lldpV2RemLocalIfIndex	(Table index)
	lldpV2RemLocalDestMACAddress	(Table index)
	lldpV2RemIndex	(Table index)
	lldpV2Xdot1RemEvbTlvString	EVB TLV string, D.2.13
lldpV2Xdot1RemCDCPTlvTable		D.2.14
	lldpV2RemTimeMark	(Table index)
	lldpV2RemLocalIfIndex	(Table index)
	lldpV2RemLocalDestMACAddress	(Table index)
	lldpV2RemIndex	(Table index)
	lldpV2Xdot1RemCDCPTlvString	CDCP TLV string, D.2.14

Insert new subclause D.4.6 as shown:

D.4.6 EVB extensions to the IEEE 802.1 LLDP extension MIB module

In the following MIB definition, should any discrepancy between the DESCRIPTION text and the corresponding definition in D.2.1 through D.4 occur, the definition in D.2.1 through D.4 shall take precedence.

```
LLDP-EXT-DOT1-EVB-EXTENSIONS-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY,
    OBJECT-TYPE
        FROM SNMPv2-SMI
    TruthValue
        FROM SNMPv2-TC
    MODULE-COMPLIANCE,
    OBJECT-GROUP
        FROM SNMPv2-CONF
    ifGeneralInformationGroup
        FROM IF-MIB
    lldpV2LocPortIfIndex,
    lldpV2RemTimeMark,
    lldpV2RemLocalIfIndex,
    lldpV2RemLocalDestMACAddress,
    lldpV2RemIndex,
    lldpV2PortConfigEntry
        FROM LLDP-V2-MIB
    lldpV2Xdot1MIB
        FROM LLDP-EXT-DOT1-V2-MIB;

-- Define the MIB module
        lldpXDot1EvbExtensions MODULE-IDENTITY
    LAST-UPDATED "201202150000Z" -- February 15, 2012
```

```
    ORGANIZATION "IEEE 802.1 Working Group"
    CONTACT-INFO
           "WG-URL: http://www.ieee802.org/1
            WG-EMail: STDS-802-1-L@LISTSERV.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@LISTSERV.IEEE.ORG"
    DESCRIPTION
            "The LLDP Management Information Base extension module for
            IEEE 802.1 organizationally defined discovery information
            for the EVB extension objects.

            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
            MIB module are to IEEE Std 802.1Qbg-20XX.

            Copyright (C) IEEE (2011).  This version of this MIB module
            is published as D.4.6 of IEEE Std 802.1Qbg-20XX;
            see the standard itself for full legal notices."

    REVISION "201202150000Z" -- February 15, 2012

    DESCRIPTION
            "Initial version published as part of IEEE Std. 802.1Qbg"

-- Hang this MIB module under the stand-alone extension MIBs arc:
   ::= { lldpXdot1StandAloneExtensions 1 }

-- Define the root arc for stand-alone extension MIBs in 802.1
lldpXdot1StandAloneExtensions OBJECT IDENTIFIER ::= { lldpV2Xdot1MIB 7 }

-----------------------------------------------------------------------
-----------------------------------------------------------------------
--
-- Organizationally Defined Information Extension - IEEE 802.1
-- Definitions to support the evbSet TLV set (Table D-1)
-- for Edge Virtual Bridging
--
-----------------------------------------------------------------------
-----------------------------------------------------------------------

lldpXdot1EvbMIB OBJECT IDENTIFIER ::= { lldpXDot1EvbExtensions 1 }
lldpXdot1EvbObjects OBJECT IDENTIFIER ::= { lldpXdot1EvbMIB 1 }

-- EVB 802.1 MIB Extension groups

lldpXdot1EvbConfig OBJECT IDENTIFIER ::= { lldpXdot1EvbObjects 1 }
lldpXdot1EvbLocalData OBJECT IDENTIFIER ::= { lldpXdot1EvbObjects 2 }
lldpXdot1EvbRemoteData OBJECT IDENTIFIER ::= { lldpXdot1EvbObjects 3 }


-----------------------------------------------------------------------
-- IEEE 802.1 - EVB Configuration
-----------------------------------------------------------------------
```

```
--
--  lldpXdot1EvbConfigEvbTable : configure the
--  transmission of the EVB TLV on a set of ports
--

lldpXdot1EvbConfigEvbTable OBJECT-TYPE
    SYNTAX        SEQUENCE OF LldpXdot1EvbConfigEvbEntry
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
        "A table that controls selection of EVB
         TLVs to be transmitted on individual ports."
    ::= { lldpXdot1EvbConfig 1 }

lldpXdot1EvbConfigEvbEntry OBJECT-TYPE
    SYNTAX        LldpXdot1EvbConfigEvbEntry
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
        "LLDP configuration information that controls the
        transmission of IEEE 802.1 organizationally defined
        EVB TLV on LLDP transmission capable ports.

        This configuration object augments the lldpV2PortConfigEntry of
        the LLDP-MIB, therefore it is only present along with the port
        configuration defined by the associated lldpV2PortConfigEntry
        entry.

        Each active lldpConfigEntry is restored from non-volatile
        storage (along with the corresponding lldpV2PortConfigEntry)
        after a re-initialization of the management system."
    AUGMENTS      { lldpV2PortConfigEntry }
    ::= { lldpXdot1EvbConfigEvbTable 1 }

LldpXdot1EvbConfigEvbEntry ::= SEQUENCE {
    lldpXdot1EvbConfigEvbTxEnable TruthValue
}

lldpXdot1EvbConfigEvbTxEnable OBJECT-TYPE
    SYNTAX        TruthValue
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "The lldpXdot1EvbConfigEvbTxEnable, which is
        defined as a truth value and configured by the network
        management, determines whether the IEEE 802.1 organizationally
        defined EVB TLV transmission is allowed
        on a given LLDP transmission capable port.

        The value of this object is restored from non-volatile
        storage after a re-initialization of the management system."
    REFERENCE
        "D.2.13"
    DEFVAL         { false }
    ::= { lldpXdot1EvbConfigEvbEntry 1 }

--
--  lldpXdot1EvbConfigCdcpTable : configure the
--  transmission of the CDCP TLV on a set of ports
--

lldpXdot1EvbConfigCdcpTable OBJECT-TYPE
    SYNTAX        SEQUENCE OF LldpXdot1EvbConfigCdcpEntry
    MAX-ACCESS    not-accessible
```

```
    STATUS        current
    DESCRIPTION
        "A table that controls selection of EVB
         TLVs to be transmitted on individual ports."
    ::= { lldpXdot1EvbConfig 2 }

lldpXdot1EvbConfigCdcpEntry OBJECT-TYPE
    SYNTAX        LldpXdot1EvbConfigCdcpEntry
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
        "LLDP configuration information that controls the
        transmission of IEEE 802.1 organizationally defined
        CDCP TLV on LLDP transmission capable ports.

        This configuration object augments the lldpV2PortConfigEntry of
        the LLDP-MIB, therefore it is only present along with the port
        configuration defined by the associated lldpV2PortConfigEntry
        entry.

        Each active lldpConfigEntry is restored from non-volatile
        storage (along with the corresponding lldpV2PortConfigEntry)
        after a re-initialization of the management system."
    AUGMENTS      { lldpV2PortConfigEntry }
    ::= { lldpXdot1EvbConfigCdcpTable 1 }

LldpXdot1EvbConfigCdcpEntry ::= SEQUENCE {
    lldpXdot1EvbConfigCdcpTxEnable TruthValue
}

lldpXdot1EvbConfigCdcpTxEnable OBJECT-TYPE
    SYNTAX        TruthValue
    MAX-ACCESS    read-write
    STATUS        current
    DESCRIPTION
        "The lldpXdot1EvbConfigCdcpTxEnable, which is
        defined as a truth value and configured by the network
        management, determines whether the IEEE 802.1 organizationally
        defined CDCP TLV transmission is allowed
        on a given LLDP transmission capable port.

        The value of this object is restored from non-volatile
        storage after a re-initialization of the management system."
    REFERENCE
        "D.2.14"
    DEFVAL         { false }
    ::= { lldpXdot1EvbConfigCdcpEntry 1 }

-----------------------------------------------------------------------
-- IEEE 802.1 - EVB Local System Information
-----------------------------------------------------------------------

---
---
--- lldpV2Xdot1LocEvbTlvTable: EVB TLV Information Table
---
---

lldpV2Xdot1LocEvbTlvTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF LldpV2Xdot1LocEvbTlvEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "This table contains one row per port of EVB
```

```
            TLV information (as a part of the LLDP
            802.1 organizational extension) on the local system
            known to this agent."
    ::= { lldpXdot1EvbLocalData 1 }

lldpV2Xdot1LocEvbTlvEntry OBJECT-TYPE
    SYNTAX      LldpV2Xdot1LocEvbTlvEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "EVB TLV information about a
            particular port component."
    INDEX   { lldpV2LocPortIfIndex }
    ::= { lldpV2Xdot1LocEvbTlvTable 1 }

LldpV2Xdot1LocEvbTlvEntry ::= SEQUENCE {
      lldpV2Xdot1LocEvbTlvString     OCTET STRING
      }

lldpV2Xdot1LocEvbTlvString OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE (0..514))
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "This object contains the EVB TLV information string
            for the Port, as defined in D.2.13.
            As the elements within the string are not individually
            manipulated via SNMP (they are of concern only to the
            state machines), the sub-structure of the string
            is not visible as separate objects within the
            local database."
    REFERENCE
            "D.2.13"
    ::= { lldpV2Xdot1LocEvbTlvEntry 1 }


---
---
--- lldpV2Xdot1LocCdcpTlvTable: CDCP TLV Information Table
---
---

lldpV2Xdot1LocCdcpTlvTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF LldpV2Xdot1LocCdcpTlvEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "This table contains one row per port of CDCP
            TLV information (as a part of the LLDP
            802.1 organizational extension) on the local system
            known to this agent."
    ::= { lldpXdot1EvbLocalData 2 }

lldpV2Xdot1LocCdcpTlvEntry OBJECT-TYPE
    SYNTAX      LldpV2Xdot1LocCdcpTlvEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "CDCP TLV information about a
            particular port component."
    INDEX   { lldpV2LocPortIfIndex }
    ::= { lldpV2Xdot1LocCdcpTlvTable 1 }

LldpV2Xdot1LocCdcpTlvEntry ::= SEQUENCE {
```

```
      lldpV2Xdot1LocCdcpTlvString     OCTET STRING
      }

lldpV2Xdot1LocCdcpTlvString OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE(0..514))
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "This object contains the CDCP TLV information string
            for the Port, as defined in D.2.14.
            As the elements within the string are not individually
            manipulated via SNMP (they are of concern only to the
            state machines), the sub-structure of the string
            is not visible as separate objects within the
            local database."
    REFERENCE
            "D.2.14"
    ::= { lldpV2Xdot1LocCdcpTlvEntry 1 }

-- ------------------------------------------------------------------------
-- IEEE 802.1 - EVB Remote System Information
-- ------------------------------------------------------------------------

---
---
--- lldpV2Xdot1RemEvbTlvTable: EVB TLV Information Table
---
---

lldpV2Xdot1RemEvbTlvTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF LldpV2Xdot1RemEvbTlvEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "This table contains one row per port of EVB
            TLV information (as a part of the LLDP
            802.1 organizational extension) on the remote system
            known to this agent."
    ::= { lldpXdot1EvbRemoteData 1 }

lldpV2Xdot1RemEvbTlvEntry OBJECT-TYPE
    SYNTAX      LldpV2Xdot1RemEvbTlvEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "EVB TLV information about a
            particular port component."
    INDEX   { lldpV2RemTimeMark,
              lldpV2RemLocalIfIndex,
              lldpV2RemLocalDestMACAddress,
              lldpV2RemIndex }
    ::= { lldpV2Xdot1RemEvbTlvTable 1 }

LldpV2Xdot1RemEvbTlvEntry ::= SEQUENCE {
      lldpV2Xdot1RemEvbTlvString     OCTET STRING
      }

lldpV2Xdot1RemEvbTlvString OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE (0..514))
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "This object contains the EVB TLV information string
            for the Port, as defined in D.2.13.
```

```
            As the elements within the string are not individually
            manipulated via SNMP (they are of concern only to the
            state machines), the sub-structure of the string
            is not visible as separate objects within the
            local database."
    REFERENCE
            "D.2.13"
    ::= { lldpV2Xdot1RemEvbTlvEntry 1 }


---
---
--- lldpV2Xdot1RemCdcpTlvTable: CDCP TLV Information Table
---
---

lldpV2Xdot1RemCdcpTlvTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF LldpV2Xdot1RemCdcpTlvEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "This table contains one row per port of CDCP
            TLV information (as a part of the LLDP
            802.1 organizational extension) on the remote system
            known to this agent."
    ::= { lldpXdot1EvbRemoteData 2 }

lldpV2Xdot1RemCdcpTlvEntry OBJECT-TYPE
    SYNTAX      LldpV2Xdot1RemCdcpTlvEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "CDCP TLV information about a
            particular port component."
    INDEX   { lldpV2RemTimeMark,
              lldpV2RemLocalIfIndex,
              lldpV2RemLocalDestMACAddress,
              lldpV2RemIndex }
    ::= { lldpV2Xdot1RemCdcpTlvTable 1 }

LldpV2Xdot1RemCdcpTlvEntry ::= SEQUENCE {
      lldpV2Xdot1RemCdcpTlvString     OCTET STRING
      }

lldpV2Xdot1RemCdcpTlvString OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE (0..514))
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "This object contains the CDCP TLV information string
            for the Port, as defined in D.2.14.
            As the elements within the string are not individually
            manipulated via SNMP (they are of concern only to the
            state machines), the sub-structure of the string
            is not visible as separate objects within the
            local database."
    REFERENCE
            "D.2.14"
    ::= { lldpV2Xdot1RemCdcpTlvEntry 1 }

-------------------------------------------------------------------------
-- IEEE 802.1 - EVB Conformance Information
-------------------------------------------------------------------------
```

```
lldpXdot1EvbConformance OBJECT IDENTIFIER ::= { lldpXDot1EvbExtensions 2 }

lldpXdot1EvbCompliances
    OBJECT IDENTIFIER ::= { lldpXdot1EvbConformance 1 }
lldpXdot1EvbGroups
    OBJECT IDENTIFIER ::= { lldpXdot1EvbConformance 2 }

--
-- EVB - Compliance Statements
--

lldpXdot1EvbCompliance MODULE-COMPLIANCE
    STATUS        current
    DESCRIPTION
         "A compliance statement for SNMP entities that implement
         the IEEE 802.1 organizationally defined Congestion
         Notification LLDP extension MIB.

         This group is mandatory for agents that implement the
         EVB evbSet TLV set."
    MODULE        -- this module
        MANDATORY-GROUPS  { lldpXdot1EvbGroup,
                            ifGeneralInformationGroup }
    ::= { lldpXdot1EvbCompliances 1 }


--
-- EVB - MIB groupings
--

lldpXdot1EvbGroup  OBJECT-GROUP
    OBJECTS {
      lldpXdot1EvbConfigEvbTxEnable,
      lldpXdot1EvbConfigCdcpTxEnable,
      lldpV2Xdot1LocEvbTlvString,
      lldpV2Xdot1LocCdcpTlvString,
      lldpV2Xdot1RemEvbTlvString,
      lldpV2Xdot1RemCdcpTlvString
      }
    STATUS  current
    DESCRIPTION
        "The collection of objects that support the
        EVB evbSet TLV set."
    ::= { lldpXdot1EvbGroups 1 }

END
```

D.5 PICS proforma for IEEE 802.1 Organizationally Specific TLV extensions

Insert the following rows immediately after row "dot1cntlvt"of Table D.5.3:

dot1evbSet	Is the IEEE 802.1 Organizationally Specific TLV evbSet implemented?	O.1	D.2.13, D.2.14, Table D-1	Yes [ ] No [ ]
dot1evbTlv	Is each TLV in the IEEE 802.1 Organization-ally Specific TLV evbSet implemented?	dot1evbSet:M	D.2.13, D.2.14, Table D-1	Yes [ ] N/A[ ]