IEEE STANDARDS ASSOCIATION

IEEE Standard for
Local and metropolitan area networks—

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

Amendment 20: Shortest Path Bridging

IEEE Computer Society

Sponsored by the
LAN/MAN Standards Committee

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

29 June 2012

IEEE Std 802.1aq™-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, and
IEEE Std 802.1Qbf™-2011)

IEEE Std 802.1aq™-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, and
IEEE Std 802.1Qbf™-2011)

IEEE Standard for
Local and metropolitan area networks—

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

Amendment 20: Shortest Path Bridging

Sponsor

LAN/MAN Standards Committee
of the
IEEE Computer Society

Approved 29 March 2012

IEEE-SA Standards Board

Abstract: This amendment to IEEE Std 802.1Q-2011 specifies shortest path bridging of unicast and multicast frames, including protocols to calculate multiple active topologies that can share learned station information, and support of a virtual local area network (VLAN) by multiple, per topology VLAN identifiers (VIDs).

Keywords: Bridged Local Area Networks, IEEE 802.1aq, LANs, local area networks, MAC Bridges, metropolitan area networks, MSTP, Multiple Spanning Tree Protocol, Provider Bridged Local Area Networks, Shortest Path Bridging Protocol, 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 978-0-7381-7262-0** **STD97237**
Print: **ISBN 978-0-7381-7368-9** **STDPD97237**

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

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

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

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

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

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

Official Statements: A statement, written or oral, that is not processed in accordance with the IEEE-SA Standards Board Operations Manual shall not be considered the official position of IEEE or any of its committees and shall not be considered to be, nor be relied upon as, a formal position of IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position of IEEE.

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

Comments on standards should be submitted to the following address:

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

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

Introduction

This introduction is not part of 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 20: Shortest Path Bridging.

This amendment to IEEE Std 802.1Q-2011 specifies shortest path bridging of unicast and multicast frames, including protocols to calculate multiple active topologies that can share learned station information, and support of a virtual local area network (VLAN) by multiple, per topology VLAN identifiers (VIDs).

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

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

Updating of IEEE documents

Users of IEEE Standards documents should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect. In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit the IEEE-SA website 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*
Glenn Parsons, *Vice Chair*
Donald Fedyk and **Michael Seaman**, *Editors*
Stephen Haddock, *Chair, Interworking Task Group*

Zehavit Alon
Yafan An
Ting Ao
Peter Ashwood-Smith
Christian Boiger
Paul Bottorff
Rudolf Brandner
Craig Carlson
Paul Congdon
Rodney Cummings
Claudio DeSanti
Zhemin Ding
Donald Eastlake
Janos Farkas
Norman Finn
Ilango Ganga
Geoffrey Garner
Anoop Ghanwani
Mark Gravel
Eric Gray
Yingjie Gu
Craig Gunther
Hitoshi Hayakawa
Hal Keen
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
David Olsen
Donald Pannell
Mark Pearson
Joseph Pelissier
Rene Raeber
Karen T. Randall
Josef Roese
Dan Romascanu
Jessy Rouyer
Ali Sajassi
Panagiotis Saltsidis
Rakesh Sharma
Kevin Stanton
Robert Sultan
Michael Johas Teener
Patricia Thaler
Chait Tumuluri
Maarten Vissers

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

Thomas Alexander
Butch Anton
Danilo Antonelli
Michael Bahr
Nancy Bravin
William Byrd
Keith Chow
Charles Cook
Patrick Diamond
Thomas Dineen
Donald Eastlake
Richard Eckard
Richard Edgar
John Egan
Jose Dominic Espejo
Donald Fedyk
Yukihiro Fujimoto
Devon Gayle
Gregory Gillooly
Joel Goergen
Randall Groves
Michael Gundlach
C. Guy
Stephen Haddock
David Hunter
Tetsushi Ikegami
Noriyuki Ikeuchi
Paul Isaacs
Atsushi Ito
Raj Jain
Tony Jeffree
Vincent Jones
Shinkyo Kaku
Piotr Karocki
Michael Kelsen
Stuart J. Kerry
Yongbum Kim
Bruce Kraemer
Paul Lambert
Juan L. Lazaro
Michael Lerer
William Lumpkins
Greg Luri
Thomas Mack-Crane
Elvis Maculuba
Wayne W. Manges
David Martin
Gary Michel
Jose Morales
Michael S. Newman
Satoshi Obara
Maximilian Riegel
Benjamin Rolfe
Jessy Rouyer
Randall Safier
Panagiotis Saltsidis
Peter Saunderson
Bartien Sayogo
Benson Schliesser
Rich Seifert
Gil Shultz
Kapil Sood
Matthew Squire
Thomas Starai
Walter Struppler
Robert Sultan
Joseph Tardo
William Taylor
Michael Johas Teener
Mark-Rene Uchida
Dmitri Varsanofiev
Prabodh Varshney
John Vergis
Ludwig Winkel
Oren Yuen

When the IEEE-SA Standards Board approved this amendment on 29 March 2012, it had the following membership:

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

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 20: Shortest Path Bridging

This amendment to IEEE Std 802.1Q™-2011 specifies changes to allow bridged frames to travel on the shortest path between their source and destination(s). Changes are applied to the base text of 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, and IEEE Std 802.1Qbf™-2011.

IMPORTANT NOTICE: This standard is not intended to ensure safety, security, health, or environmental protection. Implementers of the standard are responsible for determining appropriate safety, security, environmental, and health practices or regulatory requirements.

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

NOTE—The editing instructions contained in this amendment define how to merge the material contained therein into the existing base standard and its amendments to form the comprehensive standard. Text shown in bold italics in this amendment defines the editing instructions necessary to changes to this base text. Three editing instructions are used: ***change***, ***delete***, and ***insert***. ***Change*** is used to make a change to existing material. The editing instruction specifies the location of the change and describes what is being changed. Changes to existing text may be clarified using ~~strikeout~~ markings to indicate removal of old material and underscore markings to indicate addition of new material. ***Delete*** removes existing material. ***Insert*** adds new material without changing the existing material. Insertions may require renumbering. If so, renumbering instructions are given in the editing instruction. Editorial notes will not be carried over into future editions of IEEE Std 802.1Q.[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 after item bh) in 1.3:

This standard specifies shortest path bridging of unicast and multicast frames, specifying protocols to calculate multiple active topologies that can share learnt station information, and support of a VLAN by multiple, per topology shortest path VLAN identifiers (SPVIDs). To this end it:

bi) Describes the use of shortest paths to increase throughput and minimize transit delay, while introducing a negligible rate of frame misordering.

bj) Requires that active topologies calculated by spanning tree protocols and Shortest Path Tree (SPT) protocols be stable, predictable, reproducible, reverse path congruent and unicast-multicast congruent to: permit learning of station location from the source addresses of all frames, simplify the detection and management of faults, and maintain the characteristics of the MAC Service provided.

bk) Specifies the calculation of symmetric sets of shortest path trees (SPTs), each rooted at a bridge within an SPT Region comprising bridges operating compatible protocols and configurations.

bl) Specifies the use of Bridge Protocol Data Units (BPDUs) to identify and bound SPT Regions, and to ensure loop-free interoperability with regions using the Rapid Spanning Tree Protocol (RSTP) and Multiple Spanning Tree Protocol (MSTP).

bm) Specifies both Shortest Path Bridging VID (SPBV) and Shortest Path Bridging MAC (SPBM):

    1) for SPBV, identifying each SPT by SPVID and locating end stations by source MAC address learning;
    2) for SPBM, identifying each SPT by VID and source MAC address and distributing end station location information explicitly.

bn) Supports management selection of the Common Spanning Tree (CST), a Multiple Spanning Tree Instance (MSTI) or SPB for support of any given VLAN within an SPT Region.

bo) Specifies a protocol that automatically assigns SPVIDs for each VLAN supported by SPBV.

bp) Supports load sharing by equal cost trees (ECT) through the calculation of multiple SPT Sets, with each shortest path VLAN being assigned to one SPT Set.

bq) Specifies ISIS-SPB: the use of and extensions to the Intermediate System to Intermediate System protocol (IS-IS) to calculate SPTs for both SPBV and SPBM.

br) Describes the addressing of ISIS-SPB entities and specifies the group MAC Addresses they use.

bs) Specifies the use of loop prevention (for SPBV and for multicast frames for SPBM) and loop mitigation (for unicast frames for SPBM).

bt) Specifies an Agreement Protocol that prevents loops, specifying the necessary state information and computation as part of ISIS-SPB, and communicating agreement information for the CIST and (as a compact Digest) for SPTs in each BPDU.

2. Normative references

Insert the following references into Clause 2 in alphanumeric order:

IETF RFC 5120 (Feb. 2008) M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-ISs).[2]

IETF RFC 5303 (Oct. 2008), Three-Way Handshake for IS-IS Point-to-Point Adjacencies.

IETF RFC 5305 (Oct. 2008), IS-IS Extensions for Traffic Engineering.

IETF RFC 6165 (Apr. 2011), Extensions to IS-IS for Layer-2 Systems.

ISO/IEC 9577:1999, Information technology — Protocol identification in the network layer.[3]

ISO/IEC 10589:2002, Information technology — Telecommunications and information exchange between systems — Intermediate System to Intermediate System intra-domain routeing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode network service.

[2] Internet RFCs are retrievable by FTP at ds.internic.net/rfc/rfcnnnn.txt (where nnnn is a standard's publication number such as 1042), or call InterNIC at 1-800-444-4345 for information about receiving copies through the mail.

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

3. Definitions

Insert the following definitions into Clause 3 in alphabetic order, number them appropriately, and renumber the subsequent definitions accordingly:

3.x Adjacency: A neighbor in IS-IS. From Clause 3.6 of ISO 10589.

3.x Agreement Protocol: A protocol that uses control plane synchronisation between adjacent bridges to guarantee loop prevention, so enabling rapid transition to forwarding after network topology changes (13.16, 13.17).

3.x Base VID: The VID used to identify a VLAN in management operations in both SPBV and SPBM.

3.x ECT Algorithm: A Shortest Path Tree algorithm yielding a consistent result for a given root bridge when run by all bridges within an SPT Region.

3.x Equal Cost Tree (ECT): A unique instantiation of a Shortest Path Tree that uses a specific method to pick from equal cost alternatives. In networks that have more than one diverse path between nodes, different Shortest Path Trees from the same root node that have equal cost paths from the root node to leaf nodes are considered ECTs.

3.x Multi-Topology (MT): One of a set of independent topologies that IS-IS supports within an IS-IS domain. This MT extension can be used for a variety of purposes, such as an in-band management network “on top” of the original IS-IS topology, maintaining separate IS-IS routing domains for isolated topology islands within the network, or forcing a subset of an address space to follow a different topology.

3.x Multi-Topology Identifier (MTID): The identifier of an IS-IS MT.

3.x Port State: One of a number of possible combinations of per tree *forwarding* and *learning* variables, used by a spanning tree protocol or shortest path tree protocols for each Bridge Port, to enforce the active topology of that tree. There is port state for every tree and there could be many trees per port.

3.x Rapid Spanning Tree (RST) Bridge: A Bridge capable of supporting only a single spanning tree, the CST, by the Rapid Spanning Tree Algorithm and Protocol (RSTP), defined in Clause 13.

3.x Shortest Path Bridging (SPB): The method of bridging using shortest path trees for forwarding from each bridge. SPB is used for references that are applicable to all modes of Shortest Path Bridging, SPBM and SPBV.

3.x Shortest Path Bridging MAC Mode (SPBM): The Shortest Path Bridging method where all SPTs are rooted at a Backbone Edge Bridge (BEB). In frames with a group destination address the SPT is identified by a destination MAC address derived from the SPSourceID; in frames with an individual address the SPT is identified by a MAC address of the source BEB. This method does not use learning.

3.x Shortest Path Bridging VID Mode (SPBV): The Shortest Path Bridging method where VIDs (SPVIDs) are used to identify the SPTs for individual and group addresses. This method is learning capable.

3.x Shortest Path First (SPF): Abbreviation for the Dijkstra Shortest Path First Algorithm that is used in Link state protocols such as IS-IS.

3.x Shortest Path Source Identifier (SPSourceID): A 20-bit identifier of a bridge that is unique within a SPT Domain. SPSourceID is used to auto-create SPT Domain significant Group MAC Addresses utilizing the local address bit in SPBM (27.10).

3.x Shortest Path Tree (SPT) Bridge: A VLAN-aware Bridge capable of operating Shortest Path Bridging (SPB) protocols.

3.x Shortest Path Tree (SPT) Domain: A set of SPT Bridges that are interconnected by LANs and have the same configured IS-IS area address. An SPT Domain contains one or more SPT Regions each comprising some or all of the bridges in the Domain.

3.x Shortest Path Tree (SPT) Region: A set of LANs and SPT Bridges interconnected via Ports on those SPT Bridges, where each LAN's CIST Designated Bridge is an SPT Bridge, and each Port is either the Designated Port on one of the LANs, or else a non-Designated Port of an SPT Bridge that is connected to one of the LANs, whose MCID matches exactly the MST Configuration Identifier (MCID) or the Auxiliary MCID (27.4.1) of the Designated Bridge of that LAN.

3.x Shortest Path Tree (SPT) Set: A set of shortest path trees determined by one ECT Algorithm and capable of supporting one or more VLANs within an SPT Region (8.9.4, 13.6). The set is the union of all the shortest path trees from all sources to all other bridges using that single ECT Algorithm. A single SPT set is created for each ECT Algorithm. Active SPT Sets are identified by their {ECT Algorithm, FID} tuples.

3.x Shortest Path VID (SPVID): The identifier of both the VLAN and the SPT that is used for the transmission of a tagged frame in SPBV.

3.x Spanning Tree Algorithm and Protocol (STP) Bridge: A Bridge capable of supporting only a single spanning tree, the CST, by the Spanning Tree Algorithm and Protocol (STP) defined in Clause 8 of IEEE Std 802.1D, 1998 Edition.

3.x SPB System Identifier: A unique 6-byte value for each ISIS-SPB capable bridge and is equivalent to the IS-IS System ID. The System Identifier is used to identify unique bridges in the topology. For the purpose of this document this ID is the 6-byte Bridge Address (8.13.8).

4. Abbreviations

Insert the following abbreviations into Clause 4 in alphabetic order:

ECT	Equal Cost Tree
ISIS-SPB	IS-IS for SPBV and SPBM
LSP	Link State PDU
MT	Multiple Topology
PATHID	Path Identifier
SPB	Shortest Path Bridging
SPT	Shortest Path Tree
SPVID	Shortest Path VLAN Identifier
SPBM	Shortest Path Bridging MAC Mode
SPBV	Shortest Path Bridging VID Mode

5. Conformance

5.4 VLAN-aware Bridge component requirements

5.4.1 VLAN-aware Bridge component options

Change 5.4.1 as follows:

An implementation of a VLAN-aware Bridge component may

a) Support MST operation (5.4.1.1);

b) Support Port-and-Protocol-based VLAN classification (5.4.1.2), including multiple VID values per port, administrative control of the values of the multiple VIDs, and a Protocol Group Database.

c) Support CFM operation (5.4.1.4);

d) Support Shortest Path Bridging operation (5.4.5);

e) ~~d)~~ Support congestion notification (5.4.3);

f) ~~e)~~ Support Priority-based Flow Control (5.11);

g) ~~f)~~ Support Extended Filtering Services (6.16.5) and the operation of Multiple MAC Registration Protocol (MMRP, see 10.9) to support automatic configuration and management of MAC address information on all Ports (5.4.1.3);

h) ~~g)~~ Allow the Filtering Database to contain Static and Dynamic VLAN Registration Entries (8.8) for more than one VID, up to a maximum of 4094 VIDs;

i) Allow translation of VIDs through support of the VID Translation Table or through support of both the VID Translation Table and Egress VID translation table on one or more Bridge Ports (6.9);

NOTE 1—The maximum number of VIDs that can be supported is 4094 rather than 4096, as the VID values 0 and FFF are reserved, as indicated in Table 9-2. As conformance to this standard is only with regard to externally visible protocol behavior, this limit on the number of VIDs that can be supported does not imply any such limitation with regard to the internal architecture of a Bridge.

j) ~~h)~~ On each Port, support all of the permissible values for the Acceptable Frame Types parameter, as defined in 8.3, and support configuration of the parameter value via management;

k) ~~i)~~ Support enabling and disabling of Ingress Filtering (6.9);

l) ~~j)~~ Allow configuration of more than one VID whose untagged set includes that Port (8.8.2);

m) ~~k)~~ Support the management functionality defined in Clause 12;

n) ~~l)~~ Support more than one FID (8.8);

o) ~~m)~~ Allow allocation of more than one VID to each supported FID (8.8, 8.8.8);

p) ~~n)~~ Allow configuration of VLAN Learning Constraints (8.8.8, 12.10.3);

q) ~~o)~~ Allow configuration of fixed VID to FID allocations (8.8.8, 12.10.3);

r) ~~p)~~ Allow configuration of the Restricted_MAC_Address_Registration parameter (10.12.2.3) for each port of the Bridge;

s) ~~q)~~ Support the ability to configure the value of the Restricted_VLAN_Registration parameter (11.2.3.2.3) for each Port of the Bridge;

t) ~~r)~~ Support forwarding and queuing for time-sensitive streams (5.4.1.5);

u) ~~s)~~ Support SMIv2 MIB modules for the management of VLAN-aware Bridge capabilities (Clause 17);

v) ~~t)~~ Support Multiple Stream Reservation Protocol (MSRP; Clause 35);

w) ~~u)~~ Support the operation of MVRP (5.4.2, 11.2) as a New-only Participant;

x) ~~v)~~ Support the MVRP Extension Management Information Base (MIB) module defined in 17.7.15;

y) ~~w)~~ Support enhanced transmission selection (ETS) (see 5.4.1.6);

z) ~~x)~~ Support DCBX (see 5.4.1.7).

NOTE 2—In previous versions of IEEE Std 802.1Q, the MIB modules were maintained by IETF (e.g., IETF RFC 4363 for the Q-BRIDGE MIB). For this standard, the relevant set of MIB modules required to manage a Bridge is contained in Clause 17.

Insert the following subclause, 5.4.5, after 5.4.4:

5.4.5 Shortest Path Bridging operation (optional)

A VLAN-aware Bridge implementation that conforms to the provisions of this standard for SPB in Clause 27 shall:

a) Support SPB, in either VID mode (SPBV) or MAC mode (SPBM), as specified in Clause 27;
b) Support the IS-IS Link State Protocol with procedures to ensure Loop Prevention as specified in Clause 28 to provide SPT computation for shortest path bridging;
c) Encode, decode, and validate SPT BPDUs for the Agreement Protocol (13.17) and support Agreement Protocol logic in IS-IS as specified in Clause 28;
d) Support at least three FIDs;
e) Support the management functionality specified in Clause 12.

A VLAN-aware Bridge implementation in conformance to the provisions of this standard for SPB in Clause 27 may:

f) Support both SPB VID mode (SPBV) and SPB MAC mode (SPBM);
g) Support the SPB Management Information Base (MIB) module defined in 17.7.19;
h) Support both the VID Translation Table and Egress VID translation table (6.9) on SPBV Boundary Ports.

5.5 C-VLAN component conformance

Change 5.5 as follows:

A C-VLAN component comprises a VLAN-aware Bridge component with the EISS on all Ports supported by the use of a Customer VLAN tag (C-TAG) (6.8, 9.5).

A conformant implementation of a C-VLAN component shall

a) Comprise a single conformant VLAN-aware bridge component;
b) Recognize and use C-TAGs;
c) Filter the Reserved MAC Addresses specified in Table 8-1;
d) Use the Customer Bridge MVRP Address specified in Table 10-1; and

shall not

~~e) Use a VID Translation Table (6.5.6) on any Port;~~
e) ~~f)~~ Use Service VLAN tags (S-TAG) except in support of the functionality specified in 6.13.

5.6 S-VLAN component conformance

5.6.1 S-VLAN component options

Change 5.6.1 as follows:

A conformant S-VLAN component may implement any of the options specified for a VLAN-aware Bridge component (5.4.1). ~~and may~~

~~Allow translation of VIDs through support of a VID Translation Table (6.9) on each Port.~~

6. Support of the MAC Service

6.5 Quality of service maintenance

6.5.1 Service availability

In 6.5.1, insert the following note after list item c), and number the existing notes at the end of the subclause as "NOTE 2" and "NOTE 3":

NOTE 1—Each reference to spanning tree algorithms and protocols throughout Clause 6 includes any whose purpose is to verify that each transmitted frame gives rise to at most one service indication at any of the MSAPs for which frames for that LAN or VLAN are to be received. Such protocols include RSTP (13.4), MSTP (13.5), and SPB (Clause 27).

6.5.3 Frame misordering

Change Note 2 in 6.5.3 as follows:

NOTE 2—Where STP is in use (see Clause 8 of IEEE Std 802.1D, 1998 Edition), frame misordering cannot occur during normal operation. ~~Where RSTP is in use (see Clause 13), ~~When a protocol that is capable of rapid reconfiguration after network component failure, such as RSTP, MSTP, or SPB is used, there is an increased probability that frames that are in transit through the network will be misordered, because a Bridge can buffer frames awaiting transmission through its Ports. The probability of misordering occurring as a result of such an event is dependent on implementation choices and is associated with ~~Spanning Tree~~ reconfiguration events. Some known LAN protocols, for example, LLC Type 2, are sensitive to frame misordering and duplication; in order to allow Bridges that support RSTP to be used in environments where sensitive protocols are in use, the forceVersion parameter (13.7.2) can be used to force a Bridge that supports RSTP to operate in an STP-compatible manner. A more detailed discussion of misordering in RSTP can be found in K.3 of IEEE Std 802.1D-2004.

6.5.4 Frame duplication

Change 6.5.4 as follows:

The MAC Service (9.2 of ISO/IEC 15802-1) permits a negligible rate of duplication of frames. The operation of Bridges introduces a negligible rate of duplication of user data frames.

~~The potential for frame duplication in a network arises through the possibility of duplication of received frames on subsequent transmission within a Bridge, or through the possibility of multiple paths between source and destination end stations.~~

The potential for frame duplication in a bridged network arises through the possibility of the following:

a) Repeated transmission, through a given Bridge Port, of a frame received on another Bridge Port;
b) Multiple paths between source and destination end stations;
c) A loop in a path between source and destination stations.

A bridge does not transmit any given received frame more than once through any Port (8.6.7).

~~Where Bridges in a network are capable of connecting the individual MACs in such a way that multiple paths between any source station–destination station pairs exist, the operation of a protocol is required to ensure that a single path is used.~~

When bridges in a network connect individual LANs in such a way that physical topology is capable of providing multiple paths between any source and destination, a protocol is required to ensure that the active topology comprises a single path. SST bridges assign all frames to a single active topology, other bridges assign frames to an active topology using the VID or VID and destination address. For frames assigned to

ESPs by PBB-TE, the protocol comprises the operation of an external agent that configures static filtering entries in the Filtering Database (see 25.10); for all other frames RSTP, MSTP, or ISIS-SPB is used to configure the controls used for active topology enforcement (8.6.1). Reception of duplicate copies of a single transmitted frame can be avoided by loop prevention (6.5.4.1) or loop mitigation (6.5.4.2).

NOTE—Where RSTP, MSTP, or SPB is ~~in~~ used (see Clause 13), there is an increased probability that a ~~Spanning Tree~~ reconfiguration event can cause frames that are in transit through the network to be duplicated, because a Bridge can buffer frames awaiting transmission through its Ports. As the probability of duplication occurring as a result of such an event is small, and the frequency of ~~Spanning Tree~~ reconfiguration events is also small, the degradation of the properties of the MAC service caused by this source of frame duplication is considered to be negligible. A more detailed discussion of frame duplication in RSTP can be found in K.2 of IEEE Std 802.1D-2004.

Insert the following subclauses, 6.5.4.1 and 6.5.4.2, after 6.5.4:

6.5.4.1 Loop prevention

Each bridge that participates in RSTP, MSTP, or ISIS-SPB calculates a spanning tree for each active topology, and prevents loops by exchanging information between neighboring bridges to ensure either that their topology calculations are consistent and each frame is consistently allocated to the same active topology, or that frames are not relayed.

6.5.4.2 Loop mitigation

Each bridge that participates in SPBM may mitigate loops. The properties of loop mitigation are listed below. Allowing temporary loops to form for unicast traffic and using loop mitigation provides for faster response to topology changes. Loop mitigation is accomplished if a unicast frame can only be forwarded by the bridges as follows:

a) From at most one reception Port to at most one transmission Port; and
b) To a different transmission Port from the reception Port;

and

c) Each bridge does not relay, from any reception Port, a frame whose source address and VID are the same as that used by any entity within the bridge; and
d) Each bridge does not transmit, through any Port, a frame whose destination address and VID are the same as that used by any entity within the bridge;

and

e) All LANs in the network provide only point-to-point connectivity.

Loop mitigation limits the effect of loops since any loop in the active topology cannot include either the source or destination of the frame, nor can the loop cause multiple copies of the initial frame to be delivered to the destination, or to or from any LAN in the network that is not part of the loop.

Loop mitigation does not require that neighboring bridges assign a given frame to the same active topology; or even that the reception and transmission Ports for that frame have been derived from the same topology calculation within a given bridge. While this standard specifies the calculation of paths that are reverse path congruent (Clause 3) to support bi-directional communication, that property has no bearing on loop mitigation. As a consequence, the dynamic filtering entries that are used for active topology enforcement (8.6.1) and loop mitigation with SPBM (27.4.1) can be updated one at a time after ISIS-SPB has completed a link state calculation or during the course of that calculation.

6.9 Support of the EISS

Change 6.9 as follows:

The EISS is supported by tagging and detagging functions that in turn use the ISS (6.6, 6.7). Any given instance of the EISS shall be supported by using one but not both of the following VLAN tag types:

a) Customer VLAN tag (C-TAG); or
b) Service VLAN tag (S-TAG);

selected as specified in 9.5.

Each Bridge Port shall support the following parameters for use by these functions:

c) An Acceptable Frame Types parameter with at least one of the following values:
   1) *Admit Only VLAN-tagged frames*;
   2) *Admit Only Untagged and Priority-tagged frames*;
   3) *Admit All frames*.
d) A PVID parameter for port-based VLAN classification;

and may support the following parameters:

e) A VID Set for port-and-protocol-based classification (6.12);

~~An instance of the EISS supported by using the S-VLAN tag type may also support the following parameter:~~

f) A VID translation table;
g) An Egress VID translation table.

All three values for Acceptable Frame Types may be supported; if so, they shall be configurable using the management operations defined in Clause 12, and the default shall be *Admit All Frames*. A frame is treated as Untagged if the initial octets of the mac_service_data_unit parameter do not contain a VLAN tag of the type used to support the EISS (9.5), as Priority-tagged if the value contained in the VID field of the VLAN tag is zero, and as VLAN-tagged if the value is non-zero.

NOTE 1—The VID translation tables are for use only at the edge of administrative or control protocol domains, e.g., Network-Network Interfaces, Shortest Path Bridging domains, or Root and Leaf interfaces to a rooted multipoint service, as described in this standard. Control protocols that are designed for operation through ports where VID translation is permitted (e.g., MSTP, MVRP, CFM) explicitly compensate for the translation of VID values carried within their PDUs. Successful operation and maintenance of Bridged Local Area Networks depends on the fact that VID values are not translated at other ports or at ports within the operational span of control protocols that do not recognize translation.

The PVID and VID Set shall contain valid VID values (Table 9-2) and may be configured by management. If they have not been explicitly configured, the PVID shall assume the value of the default PVID defined in Table 9-2 and the VID Set shall be empty.

NOTE 2—The default behavior of a Bridge that supports port-and-protocol-based classification is the same as that of a Bridge that supports only port-based classification, since all the Protocol Group Identifiers in the VID Set for each Port assign the same VID as the PVID.

~~The VID translation table, when supported, shall contain a one-to-one bidirectional mapping of VID values included in the S-TAG of frames transmitted and received on the port (local VID) and VID values passed in the parameters of the EISS service primitives (relay VID). The table is configurable by management, and the default table configuration maps each value of the local VID to the same value for the relay VID.~~

The VID translation table and Egress VID translation table provide a mapping between “local VID” values and “relay VID” values. The local VID is the value encoded in a VLAN tag header in the initial octets of the mac_service_data_unit parameter of an ISS service primitive (M_UNITDATA.indication or M_UNITDATA.request). The relay VID is the value contained in the vlan_identifier parameter of an EISS service primitive (EM_UNITDATA.indication or EM_UNITDATA.request). The VID translation table may be supported by itself, or in combination with the Egress VID translation table. When only the VID translation table is supported it is used for bidirectional mapping between the local VID and the relay VID in both indication and request service primitives. This enforces a symmetric one-to-one translation (i.e., if local VID A maps to relay VID B then relay VID B will map to local VID A). When both tables are supported the VID translation table is used only for mapping the local VID to a relay VID in indication service primitives, and the Egress VID translation table is used for mapping the relay VID to a local VID in request service primitives. This allows asymmetric and many-to-one translations (e.g., for translating to and from SPVIDs at the edge of an SPT Region). Both tables are configurable by management, and the default configuration maps each value of the local VID to the same value for the relay VID, and vice versa. For Bridge Ports at the boundary of an SPBV SPT Region the tables are dynamically modified by the ISIS-SPB.

6.9.1 Data indications

Change the first paragraph in 6.9.1 as follows:

On receipt of an M_UNITDATA.indication primitive from the Internal Sublayer Service, the received frame is discarded if:

a) The initial octets of the mac_service_data_unit do not contain a valid VLAN tag header (9.3) of the type used to support the EISS (9.5), and the Acceptable Frame Types is *Admit Only VLAN-tagged frames*; or,

b) The initial octets of the mac_service_data_unit contain a valid VLAN tag header (9.3) of the type used to support the EISS (9.5), and
   1) The VID value is FFF, reserved in Table 9-2 for future implementation use; or
   2) The VID translation table is supported and the translated VID value (relay VID) is FFF (indicating local discard); or
   3) ~~2)~~ The VID value is 000 (indicating priority-tagged), and the Acceptable Frame Types is *Admit Only VLAN-tagged frames*; or
   4) ~~3)~~ The VID value is in the range 001-FFE, and the Acceptable Frame Types is *Admit Only Untagged and Priority-tagged frames*.

Change the fifth paragraph in 6.9.1 as follows:

The value of the **vlan_identifier** parameter is as follows:

e) The value contained in the VID field, optionally translated using the VID translation table, if the frame is VLAN-tagged;

f) The value of the PVID for the Port, if ~~the frame is untagged or Priority-tagged and~~ port-and-protocol VLAN classification is not ~~implemented;~~supported and the frame is untagged, Priority-tagged, or the VID translation table is supported and the translated VID value (relay VID) is zero;

g) As determined by port-and-protocol-based VLAN classification (6.12) if that capability is implemented and the frame is untagged or Priority-tagged~~.~~ or the VID translation table is supported and the translated VID value (relay VID) is zero.

6.9.2 Data requests

Change the first and third paragraphs of 6.9.2 as follows:

On invocation of an EM_UNITDATA.request primitive by a user of the E-ISS, the frame is discarded if the Egress VID translation table is supported and the translated VID value (local VID) is FFF (reserved in Table 9-2). Otherwise, an M-UNITDATA.request primitive is invoked, with parameter values as follows:

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

Unless the Bridge Port is a member of the untagged set (8.8.2) for the VID identified by the **vlan_identifier** parameter, then a tag header, formatted as necessary for the destination MAC type, is inserted as the initial octets of the mac_service_data_unit parameter. The values of the **vlan_identifier**, (optionally translated using the VID translation table or Egress VID translation table), **priority**, and **drop_eligible** parameters are used to determine the contents of the tag header, in accordance with Clause 9.

7. Principles of network operation

Change the introductory text of Clause 7 as follows:

This clause establishes the principles and a model of Virtual Bridged Local Area Network operation. It defines the context necessary for:

a) The operation of individual VLAN-aware Bridges (Clause 8);
b) Their participation in the Spanning Tree (Clause 8 of IEEE Std 802.1D, 1998 Edition), Rapid Spanning Tree (Clause 13), or Multiple Spanning Tree Protocol (Clause 13), and Shortest Path Bridging protocols (SPB) (Clause 27);
c) The management of individual Bridges (Clause 12); and
d) The management of VLAN Topology (Clause 11)

to support, preserve, and maintain the quality of the MAC Service as discussed in Clause 6.

7.1 Network overview

Replace Figure 7.1 with the following figure (which includes references to SPB):

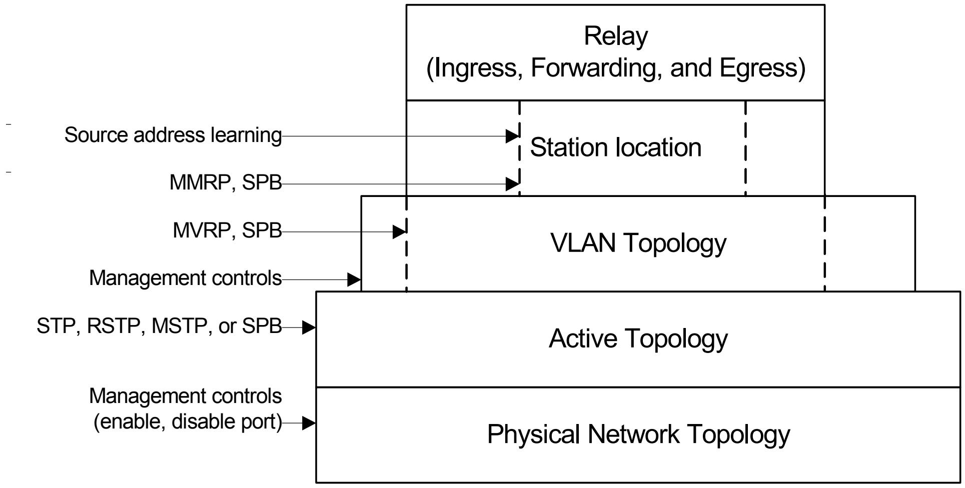


Figure 7-1—VLAN Bridging overview

7.2 Use of VLANs

Change the following list item in 7.2 as follows:

d) Identify ~~the~~ configuration parameters that partition ~~or restrict access from one part of the~~ a physical network.

Insert the following subclause, 7.3, after 7.2, and renumber the subsequent subclauses in Clause 7 accordingly:

7.3 Active topology

Bridges cooperate to calculate one or more loop-free and fully connected active topologies, i.e., spanning trees. The algorithms and protocols—RSTP (13.4), MSTP (13.5), SPB (Clause 27)—that support that calculation provide rapid recovery from network component failure (6.5.1) by using alternate physical connectivity, without requiring management intervention.

The bridge's forwarding processes constrain the potential path for each user data frame to a single spanning tree. A bridge that uses RSTP or STP allocates all frames to a single Common Spanning Tree (CST). An MST or SPT Region (13.8, 27.4.1) comprises the transitive closure of bridges that use MSTP or SPB and that agree on the allocation of frames with a given VLAN Identifier (VID) to a given subtree within the region. That subtree can be a Multiple Spanning Tree Instance (MSTI), a Shortest Path Tree (SPT), or the Internal Spanning Tree (IST) that is the logical continuation of the CST through the region. Interoperability between MST and SPT Bridges and Single Spanning Tree (SST) Bridges, and between differently configured MST/SPT Bridges, is achieved by allocating all frames to the CST at region boundaries. Thus the spanning tree for a given frame can comprise subtrees of the CST together with an MSTI or SPT within each region. Frames with different VIDs can be assigned to the same or to different trees and subtrees within a region.

NOTE—In a stable network, the MSTP and SPB algorithms verify that each MST/SPT Region is fully connected internally. However, if the CST Root lies outside the region, the region can be partitioned temporarily as necessary to prevent external connectivity between Bridge Ports at the region boundary from creating a loop. A given region of the network can be an MST Region, an SPT Region, or both, but the protocol mechanisms that verify connectivity internal to the region require that regions do not overlap or subset other regions.

Change the former 7.3 as follows:

7.4 ~~7.3~~ VLAN topology

~~Each Bridge cooperates with others to operate a spanning tree protocol to calculate one or more loop-free, fully connected, active topologies. This calculation supports the quality of the MAC Service (Clause 6) and provides rapid recovery from network component failure, by using alternate physical connectivity, without requiring management intervention.~~

~~All user data frames classified as belonging to a given VLAN are constrained by the forwarding process of each Bridge to a single active topology. Each and every VLAN is thus associated with a spanning tree, although more than one VLAN can be associated with any given tree. Each VLAN may occupy the full extent of the active topology of its associated spanning tree or a connected subset of that active topology. The maximum extent of the connected subset may be bounded by management by explicitly excluding certain Bridge Ports from a VLAN's connectivity.~~

Typically each of the VIDs for frames assigned to the CST, and to the IST or an MSTI within an MST Region, supports a different VLAN, and each VLAN is supported by only one VID and only one subtree within the region. However, frames for VLANs supported by shortest path bridging are assigned to the SPT rooted at the bridge where they enter an SPT Region, and SPBV requires that each source bridge for a given VLAN uses a unique SPVID. To provide interoperability with other regions and end systems that support a VLAN with a single VID, the VIDs for SPBV VLANs are translated at the SPT Region boundary (6.9, 27.12). SPBM uses a single VID for all the shortest path trees for a given VLAN.

NOTE 1—Support of a network comprising a single SPT Region with directly attached VLAN-unaware or priority tagging end stations does not require VID translation. This includes, for example, a PBN or PBBN where the edge of the network corresponds with the boundary of the SPT Region.

Within a given region, the SPTs that support any given shortest path VLAN provide full and symmetric connectivity (6.3). To allow frames in different VLANs to make use of equal cost shortest paths, multiple SPT sets can be used, each providing full symmetric connectivity between all bridges in the region.

Each VLAN may occupy the full extent of the active topology of its associated spanning tree(s) or a continuously connected subset of that active topology. The connected subset must be continuously connected otherwise the VLAN is split. The maximum extent of the connected subset may be bounded by management by explicitly excluding certain Bridge Ports from a VLAN's connectivity.

At any time the current extent of a VLAN can be further reduced from the maximum to include only those LANs that provide communication between attached devices, by the use of protocol that allows end stations to request and release services that use the VLAN. Such a protocol is specified in Clause 11. The dynamic determination of VLAN extent provides flexibility and bandwidth conservation, at the cost of network management complexity.

NOTE ~~1~~2—Dynamic determination of VLAN extent is generally preferable to static configuration for bandwidth conservation, as the latter is error prone and can defeat potential alternate connectivity-requiring active management intervention to recover from network component failure.

NOTE ~~2~~3—To accommodate end stations that do not participate in the MVRP protocol specified in Clause 11, management controls associated with each Bridge Port allow the Port to identify the attached LAN as connecting end stations that require services using specified VLANs.

NOTE 4—Where a given VLAN is supported by multiple VIDs, as for shortest path bridging, management controls identify the VLAN by a Base VID (that used when frames are allocated to the CST). Use of this single VID reduces the configuration burden, particularly when bridges are added to the SPT Region and further SPVIDs are allocated.

Change the former 7.4 as shown:

7.5 ~~7.4~~ Locating end stations

Functioning as a distributed system, ~~Bridges~~ bridges within the current extent of a VLAN can, through explicit and or implicit cooperation, locate those LANs where an attached end station or end stations are intended to receive frames addressed to a specified individual address or group address. Bridges can thus reduce traffic by confining frames to the LANs where their transmission is necessary.

NOTE 1—Individual~~ly Bridges~~ bridges do not determine the precise location of end stations but merely determine which of their Bridge Ports need to forward frames toward the destination(s). For the system of Bridges this is sufficient to restrict frames to the paths necessary to reach the destination LANs.

The multiple MAC registration protocol (MMRP), specified in Clause 10, allows end stations to advertise their presence and their desire to join (or leave) a multicast group, or to register an individual MAC address, in the context of a VLAN. The protocol communicates this information to other ~~Bridges~~ bridges using the VLAN and its active topology.

NOTE 2—To accommodate end stations that do not participate in MMRP, management controls associated with each Bridge Port allow the Port to identify the attached LAN ~~LANs~~ as connecting end stations that are intended to receive specified group addresses. The continuous operation of MMRP and the propagation of location information through ~~Bridges~~ bridges using the current active topology for the VLAN support multicast traffic reduction, while ensuring rapid restoration of multicast connectivity without management intervention if alternate connectivity is selected following network component failure.

Each end station implicitly advertises its attachment to an individual LAN and its individual MAC address whenever it transmits a frame. Bridges learn from the source address as they forward the frame along the active topology to its destination or destinations – or throughout the Bridged Local Area Network or VLAN if the location of the destination or destinations is unknown. The learned information is stored in the Filtering Database (8.8) and used to filter frames on the basis of their destination addresses.

Address information learned from a frame with a given VID can be used to filter frames with other VIDs. For example, information learned from a frame with any of the VIDs that support shortest path bridging for a given VLAN (i.e., VIDs associated with the same Base VID) is used to filter frames with any of the VIDs for that VLAN. In some network configurations, it is also necessary to use address information learned in any one of a set of VLANs to filter frames with VIDs that identify any VLAN in the set (Shared VLAN Learning, Clause 3). In other configurations it is necessary or desirable not to share learned information between certain VLANs (Independent VLAN Learning, Clause 3) while in some cases it is immaterial whether address information is shared between VLANs or not. Annex F discusses the network configuration requirements, and some of the related interoperability issues.

~~The Filtering Database architecture defined in this standard recognizes that~~

~~a) For some configurations, it is necessary to allow address information learned in one VLAN to be shared among a number of VLANs. This is known as Shared VLAN Learning (3.159);~~
~~b) For some configurations, it is desirable to ensure that address information learned in one VLAN is not shared with other VLANs. This is known as Independent VLAN Learning (3.73);~~
~~c) For some configurations, it is immaterial as to whether learned information is shared between VLANs.~~

~~NOTE 1—Annex B discusses the need for Shared and Independent VLAN Learning and some related interoperability issues.~~

~~Shared VLAN Learning is achieved by including learned information from a number of VLANs in the same Filtering Database; Independent VLAN Learning is achieved by including information from each VLAN in distinct Filtering Databases.~~

~~NOTE 2—The actual Filtering Database specification specifies a single Filtering Database that, through the inclusion of VLAN identification information in each database entry, can model the existence of one or more distinct Filtering Databases.~~

~~Within a given network, there may be a combination of configuration requirements, so that individual Bridges may be called on to share learned information, or not share it, according to the requirements of particular VLANs or groups of VLANs. The Filtering Database structure that is defined in this standard allows both Shared and Independent VLAN Learning to be implemented within the same Bridge; i.e., allows learned information to be shared between those VLANs for which Shared VLAN Learning is necessary, while also allowing learned information not to be shared between those VLANs for which Independent VLAN Learning is necessary. The precise requirements for each VLAN with respect to sharing or independence of learned information (if any) are made known to Bridges by means of a set of VLAN Learning Constraints (8.8.8.2) and fixed allocations of VIDs to filtering databases (8.8.8.1), which may be configured into the Bridges by means of management operations. By analyzing the set of learning constraints and fixed allocations for the VIDs that are currently active, the Bridge can determine~~

~~d) How many independent Filtering Databases are required in order to meet the constraints;~~
~~e) For each VID, which Filtering Database it will feed any learned information into (and use learned information from).~~

~~The manner in which this mapping of VIDs onto Filtering Databases is achieved is defined in 8.8.8; the result is that each VID is associated with exactly one Filtering Database.~~

~~The most general application of the Filtering Database specification in this standard is a Bridge that can support M independent Filtering Databases and can map N VIDs onto each Filtering Database. Such a Bridge is known as an SVL/IVL Bridge (3.161).~~

~~The conformance requirements in this standard (5.4) recognize that Bridges will be implemented with differing capabilities in order to meet a wide range of application needs, and that the full generality of the SVL/IVL approach is not always either necessary or desirable, as observed in the discussion in Annex B. In a given conformant implementation, there may be restrictions placed on the number of Filtering Databases that can be supported and/or the number of VIDs that can be mapped onto each Filtering Database. The full spectrum of conformant Filtering Database implementations is therefore as follows:~~

~~f)~~ ~~The SVL/IVL Bridge, as described above. Such Bridges provide support for M Filtering Databases, with the ability to map N VIDs onto each one;~~
~~g)~~ ~~Support for a single Filtering Database only. MAC Address information that is learned in association with one VID can be used in filtering decisions taken relative to all other VIDs supported by the Bridge. Bridges that support a single Filtering Database are referred to as SVL Bridges;~~
~~h)~~ ~~Support for multiple Filtering Databases, but only a single VID can be mapped onto each Filtering Database. MAC Address information that is learned in association with one VID cannot be used in filtering decisions taken relative to any other VID. Bridges that support this mode of operation are referred to as IVL Bridges.~~

Clause 15 and Clause 16 describe how the mapping of Customer VLANs (C-VLANs) into Service VLANs (S-VLANs) is accomplished within Provider Bridged Networks and Provider Backbone Bridged Networks. These networks provide additional mapping facilities to support hierarchies of VLANs allowing a provider a separate filtering database from customers. The conformance definitions of 5.3 have been extended to support S-VLANs. Clause 25 and Clause 26 describe how the mapping of S-VLANs into backbone service instances is accomplished within Provider Backbone Bridged Networks.

Change the former 7.5 as shown:

7.6 ~~7.5~~ Ingress, forwarding, and egress rules

The relay function provided by each Bridge controls:

a) Classification of each received frame as belonging to one and only one VLAN, and discard or acceptance of the frame for further processing on the basis of that classification and the received frame format, which can be one of three possible types:
   1) Untagged, and not explicitly identifying the frame as being associated with a particular VID;
   2) Priority-tagged, i.e., including a tag header conveying explicit priority information but not identifying the frames as being associated with a specific VID;
   3) VLAN-tagged, i.e., explicitly associating the frames with a particular VID.

   This aspect of relay implements the *ingress* rules.
b) Implementation of the decisions governing where each frame is to be forwarded as determined by the current extent of the VLAN topology (7.4), station location information (7.5), and the additional management controls specified in Clause 8. This aspect of relay implements the *forwarding* rules.
c) Queueing of frames for transmission through the selected Bridge Ports, management of the queued frames, selection of frames for transmission, and determination of the appropriate frame format type, VLAN-tagged or untagged. This aspect of relay implements the *egress* rules.

The structuring of the relay functionality into the implementation of ingress, forwarding, and egress rules constitutes a generic approach to the provision of VLAN functionality. All VLAN-aware Bridges can correctly forward received frames that are already VLAN-tagged. These are classified as belonging to the

VLAN identified by the VID in the tag header. All VLAN-aware Bridges can also classify untagged and priority-tagged frames received on any given port as belonging to a specified VLAN. In addition to this default Port-based ingress classification, this standard specifies an optional Port-and-Protocol-based classification.

The classification of untagged and priority-tagged frames, and the addition or removal of tag headers, is performed in support of the EISS (6.9).

Frames that carry control information to determine the active topology and current extent of each VLAN, i.e., spanning tree or SPB and MVRP PDUs, and frames from other link constrained protocols, such as EAPOL and LLDP, are not forwarded. Permanently configured static entries in the filtering database (8.2, 8.3, and 8.12) ensure that such frames are discarded by the Forwarding Process (8.6).

NOTE—MRPDUs destined for any MRP application are forwarded or filtered depending on whether the application concerned is supported by the bridge, as specified in 8.12.

The forwarding rules specified for VLAN-tagged frames facilitate the interoperation of bridges conformant to this standard with end stations that directly support attachment of MAC service users to VLANs by transmitting VLAN-tagged frames, and with Bridges that are capable of additional proprietary ingress classification methods.

Frames transmitted on a given LAN by a VLAN-aware Bridge for a given VLAN shall be either:

d) All untagged; or
e) All VLAN tagged with the same VID~~.~~; or
f) All VLAN tagged, each with an SPVID allocated to the Base VID for that VLAN.

8. Principles of bridge operation

Change the introductory text of Clause 8 as indicated:

This clause

a) Explains the principal elements of the operation of VLAN-aware ~~Bridge~~bridges, and the operation of VLAN-aware bridge components within systems, and lists the supporting functions.
b) Establishes a ~~Bridge~~ bridge architecture that governs the provision of these functions.
c) Provides a model of ~~Bridge~~ bridge operation in terms of processes and entities that support the functions.
d) Details the addressing requirements in a ~~Bridged Local Area Network~~bridged network.
e) Specifies the addressing of ~~Entities~~ entities in a ~~Bridge~~bridge.

NOTE—The provisions of this clause subsume the provisions of Clause 7 of IEEE Std 802.1D-2004.

8.1 Bridge operation

8.1.1 Relay

Change the first paragraph in 8.1.1 as follows:

A MAC Bridge or VLAN-aware Bridge relays individual MAC user data frames between the separate MACs of the ~~bridged~~ individual LANs connected to its Ports. The functions that support relaying of frames and maintain the Quality of Service are

a) Frame reception (8.5).
b) Discard on received frame in error (6.5.2).
c) Discard of frames that do not carry user data (6.7).
d) Priority and drop eligibility decoding from a VLAN TAG, if present, and regeneration of priority, if required (6.9).
e) ~~Classification of each received frame to a particular VLAN~~ Assignment of a VLAN Identifier to each received frame (6.9).
f) Assure that each frame is confined to an active topology (8.6.1).~~g) Frame discard to suppress loops in the physical topology of the network (8.6.1).~~
g) ~~f)~~ Frame discard to support management control over the active topology of each VLAN (6.9, 8.6.2).
h) Frame discard following the application of filtering information (8.6.3).
i) Metering of frames, potentially marking as drop eligible or discarding frames exceeding bandwidth limits (8.6.5).
j) Forwarding of received frames to other Bridge Ports (8.6.4).
k) Selection of traffic class and queuing of frames by traffic class (8.6.6).
l) Frame discard to ensure that a maximum bridge transit delay is not exceeded (6.5.6, 8.6.7).
m) Preferential discard of drop eligible frames to preserve QoS for other frames (8.6.7).
n) Selection of queued frames for transmission (8.6.8).
o) Mapping of service data units and Frame Check Sequence recalculation, if required (6.5.7).
p) Frame discard on transmittable service data unit size exceeded (6.5.8).
q) Selection of outbound access priority (6.5.9).
r) Frame transmission (8.5).

8.1.2 Filtering and relaying information

Change the following list item in 8.1.2 as follows:

a) Duplicate frame prevention: to maintain a loop-free active topology for ~~each VLAN~~every frame;

8.1.3 Duplicate frame prevention

Change 8.1.3 as follows:

A Bridge filters frames, i.e., does not relay frames received by a Bridge Port to other Ports on that Bridge, in order to prevent the duplication of frames (6.5.4). The functions that support the use and maintenance of information for this purpose are

a) Configuration and calculation of one or more spanning tree active topologies.
b) In MST Bridges, explicit configuration to assign ~~of the relationship between~~ VIDs ~~and spanning trees~~ to MSTIs (8.9).
c) In SPT Bridges: explicit or default selection of shortest path, CIST, or MSTI support for specified VLANs; and automatic assignment of VIDs to SPTs for shortest path VLANs.

8.2 Bridge architecture

Change the third and fourth paragraphs and their accompanying notes in 8.2 as shown:

Each Bridge Port also functions as an end station and shall provide ~~providing one or more instances of~~ the MAC Service to an LLC Entity that operates LLC Type 1 procedures to support protocol identification, multiplexing, and demultiplexing, for PDU transmission and reception by the Spanning Tree Protocol Entity and other higher layer entities. Other types of LLC procedures can also be supported for use by other protocols. The Bridge Port provides additional MAC service access points if those are explicitly required by the specifications of the attached higher layer entities. Each instance of the MAC Service is provided to a distinct LLC Entity that supports protocol identification, multiplexing, and demultiplexing, for PDU transmission and reception by one or more higher layer entities.

~~NOTE 1—In most cases, each Port provides a single instance of the MAC Service, to an LLC Entity that supports all Higher Layer Entities that require a point of attachment to the Port. Further instances are only provided when the specifications of the Higher Layer Entities require the use of different instances of the MAC service or of different source addresses.~~

~~An LLC Entity for each Bridge Port shall use an instance of the MAC Service provided for that Port to support the operation of LLC Type 1 procedures in order to support the operation of the Spanning Tree Protocol Entity. Bridge Ports may support other types of LLC procedures for use by other protocols, such as protocols providing Bridge Management (8.12).~~

NOTE ~~2~~—For simplicity of specification, this standard refers to a single LLC Entity that can provide both the procedures specified by ISO/IEC 8802.2 and Ethernet Type protocol discrimination in the cases where the media access method for the attached LAN supports the latter.

8.3 Model of operation

Change the following list item in 8.3 as follows:

d) The Forwarding Process (8.6), that:
   1) Enforces ~~a~~ loop-free active topolog~~y~~ies for all frames ~~for all VLANs~~ (8.1.3, 8.4, 8.6.1);

Change paragraphs 7 to 9 and the accompanying note in 8.3 as follows (note that Figure 8-6 and Figure 8-7 remain unchanged):

Figure 8-6 illustrates the operation of the Multiple Registration Protocol (MRP) Entity (8.11).

Higher Layer Entities that require only one point of attachment for the ~~Bridge~~ bridge as a whole may attach to an LLC Entity that uses an instance of the MAC Service provided by a Management Port. A Management Port does not use an instance of the ISS to attach to a network but uses the Bridge Port Transmit and Receive Process and the MAC Relay Entity to provide connectivity to other Bridge Ports and the attached LANs.

NOTE—Management port functionality may also be provided by an end station connected to an IEEE 802 LAN that is wholly contained within the system that incorporates the ~~Bridge~~bridge. The absence of external connectivity to the LAN ~~ensures~~ means that the management port is always forwarding~~access to the management port through the bridge's relay functionality can be assured at all times~~.

Figure 8-7 illustrates the reception and transmission by an ~~Entity~~ entity attached to a Management Port.

8.4 Active topologies, learning, and forwarding

Change 8.4 as follows:

An *active topology* is the connectivity provided, for frames with a specified set of VID, destination address, and source address values, by interconnecting the LANs and bridges in a bridged network with *forwarding* Bridge Ports.

The distributed spanning tree algorithms and protocols, i.e., the Rapid Spanning Tree Protocol (RSTP, Clause 13), ~~and~~ the Multiple Spanning Tree Protocol (MSTP, Clause 13), and ISIS-SPB (Clause 27) construct one or more active topologies, each simply and fully connected for frames being associated with any given ~~VID~~VLAN and transmitted from any end station to any other. The *forwarding* and *learning* performed by each Bridge Port for each spanning tree is dynamically managed by RSTP, ~~or~~ MSTP, or ISIS-SPB to prevent temporary loops and reduce excessive traffic in the network while minimizing denial of service following any change in the *physical topology* of the network.

Provider Backbone Bridge Traffic Engineering enables construction of active topologies by the external agent that is responsible for setting up Ethernet Switched Paths (ESPs).

Any port that is not enabled, i.e., has MAC_Operational (6.6.2) False or has been excluded from the active topology by management setting of the Administrative Bridge Port State to Disabled, has both forwarding and learning disabled for all spanning trees and ESPs.

If the Bridge Port is enabled, PBB-TE disables learning and enables forwarding for all frames allocated to each ESP-VID. An external agent manages the Filtering Database to control the forwarding of frames with particular values of ESP-VID and destination MAC address.

The term Port State summarizes per tree combinations of forwarding and learning, and any additional per tree variables used by a given spanning tree protocol to enforce the active topologies it has calculated, and is used by RSTP and MSTP as follows. Any port that has learning and forwarding disabled is assigned the Port State Discarding. A Port that has learning enabled but forwarding disabled has the Port State *Learning*, and a Port that both learns and forwards frames has the Port State *Forwarding*. However the RSTP and MSTP state machines (Clause 13) do not control the Port State directly but use independent forwarding and learning variables for each tree.

ISIS-SPB controls both forwarding and learning variables for each SPT and Bridge Port directly when used to support Shortest Path Bridging VID (SPBV, Clause 27) just like RSTP and MSTP. ISIS-SPB disables learning and sets forwarding to true when supporting Shortest Path Bridging MAC (SPBM, Clause 27).

RSTP constructs a single spanning tree, the Common Spanning Tree (CST), and maintains a single Port State for each Bridge Port. SST Bridges allocate all frames to that single spanning tree irrespective of their VLAN classification or source and destination MAC addresses.

MSTP constructs multiple spanning trees, the Common and Internal Spanning Tree (CIST) and additional Multiple Spanning Tree Instances (MSTIs), and maintains a Port State for each spanning tree for each Bridge Port. An MST Bridge allocates all frames classified as belonging to a given VLAN to the CIST or to one of the MSTIs using the MST Configuration Table. An MSTID of 0 identifies the CIST. A reserved MSTID value (TE-MSTID) is used to identify VIDs for use by PBB-TE with ESPs. Further MSTID values are used to identify the use of SPBV or SPBM. A single VID is used to identify frames assigned to any given VLAN that is supported by the CIST or an MSTI. That VID is used by the Forwarding Process (8.6) to identify the spanning tree for the relayed frame, and thus identifies the applicable Port State.

ISIS-SPB constructs symmetric (Clause 3) shortest path trees (SPTs) rooted at each bridge within an SPT Region, and maintains independent forwarding and learning variables for each SPT for each Bridge Port.

Each frame classified as belonging to a VLAN supported by SPBV, is allocated to an SPT rooted at the bridge that first relays the frame into the Region. If a C-VLAN, S-VLAN or B-VLAN is supported by SPBV (Clause 27) the frame is assigned a Shortest Path VLAN Identifier (SPVID) that is used by the Forwarding Process (8.6) to identify both the VLAN and the SPT, and thus identifies the applicable learning and forwarding values.

If a B-VLAN is supported by SPBM, (Clause 27) the frame is assigned a B-VID value that identifies the frame as subject to SPBM and identifies the B-VLAN and the SPT Set (Clause 3). The SPT is identified (from amongst those in that set) by the source address of the frame (B-SA) which identifies the SPT root. When ISIS-SPB sets forwarding True for that SPT, a Dynamic Filtering Entry for that B-VID, source address tuple is included in the Filtering Database so that it passes the active topology enforcement check (8.6.1). If a B-VLAN is supported by SPBM, learning is disabled for all frames for that B-VID.

Figure 8-5 illustrates the operation of the Spanning Tree Protocol Entity, which operates the Spanning Tree algorithm and its related protocols, and its modification of Port state information as part of determining the active topology of the network.

Figure 8-3 illustrates the Forwarding Process's use of the Port State: first, for a Port receiving a frame, to determine whether the received frame is to be relayed through any other Ports; and second, for another Port in order to determine whether the relayed frame is to be forwarded through that particular Port.

Figure 8-4 illustrates the use of the Port state information for a Port receiving a frame, in order to determine whether the station location information is to be incorporated in the Filtering Database.

8.6 The Forwarding Process

Replace Figure 8-11 with the following figure (which connects Active topology enforcement to the Filtering Database):

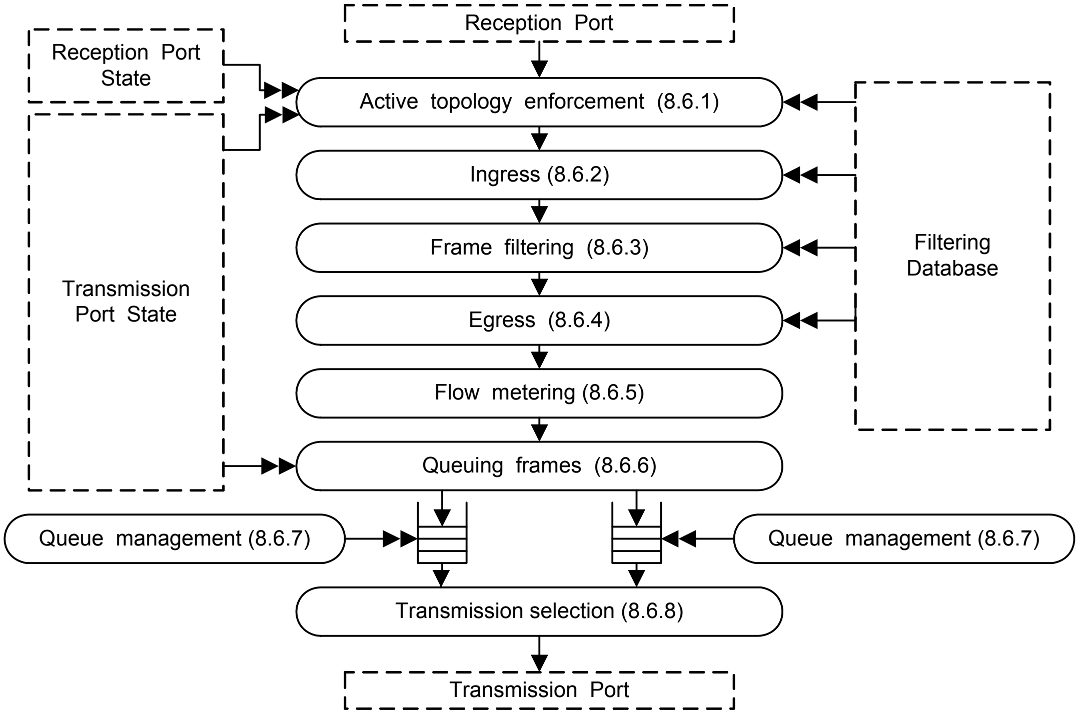


Figure 8-11—Forwarding Process functions

8.6.1 Active topology enforcement

Change 8.6.1 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 for the reception 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 for the reception Port, each Bridge Port, other than the reception Port, with forwarding true is identified as a potential transmission Port.

An SST Bridge supports a single active topology, the Common Spanning Tree (CST). For each Bridge Port, RSTP determines a single value for forwarding and a single value for learning (8.4) for all frames.

Bridges with MST, ~~or~~ PBB-TE, or SPB capabilities use the VID of the received frame to determine forwarding and learning for that frame for each Bridge Port that is enabled, i.e., has MAC_Operational True and an Administrative Bridge Port State of Enabled, as follows:

If the bridge supports PBB-TE, and the VID is an ESP-VID, forwarding is true and learning is false. Control over the active topology of each ESP is provided by configuration of static filtering entries in the Filtering Database (8.8.1, 25.10.1).

If the bridge uses MSTP to configure the CIST, and the VID identifies a VLAN assigned to the CIST, forwarding and learning are as determined by MSTP (Clause 13) for the CIST.

If the bridge uses MSTP to configure MSTIs, and the VID identifies a VLAN assigned to an MSTI calculated by MSTP, forwarding and learning are as determined by MSTP for that MSTI.

If the bridge uses ISIS-SPB to configure the CIST, and the VID identifies a VLAN assigned to the CIST, forwarding and learning take the value of forwarding determined by ISIS-SPB (Clause 28) for the CIST.

If the bridge uses ISIS-SPB, and the VID is used by SPBV as an SPVID, both forwarding and learning take the value of forwarding determined by ISIS-SPB for the SPT identified by that SPVID.

If the bridge uses ISIS-SPB and the VID identifies a VLAN supported by SPBM, then for that VID:

- learning is False for all Bridge Ports;
- forwarding is True for the reception Port if and only if a Dynamic Filtering Entry for the frame’s source address exists and specifies Forward for that Port; and
- forwarding is True for all other Bridge Ports.

NOTE 1—Effective control over the egress component of an SPT for VLANs supported by SPBM is provided by ISIS-SPB configuration of (a) a Dynamic Filtering Entry for the VLAN’s VID and Root MAC Address (for unicast frames); and (b) MAC Address Registration Entries for the VID and SPT specific Group MAC Addresses (for multicast frames); and (c) a MAC Address Registration Entry for the VID to filter “All Unregistered Group Addresses” at all Ports.

All bridges other than SST Bridges implement the MST Configuration Table (8.9.1). The use of VIDs to determine learning and forwarding, as required by this clause shall be consistent with that table as follows. VIDs allocated by the MST Configuration Table (8.9.1) of value: ~~All VIDs allocated by the MST Configuration Table to the CIST-MSTID (0x000) are assigned to the CIST, all VIDs allocated to the TE-MSTID (0xFFE) are ESP-VIDs, and all VIDs allocated to other values of MSTID are assigned to the MSTI identified by that MSTID.~~

- CIST-MSTID (0x000) are CIST VIDs.
- SPBM-MSTID (0xFFC) are Base VIDs supported by SPBM.
- SPBV-MSTID (0xFFD) are Base VIDs supported by SPBV.
- TE-MSTID (0xFFE) are ESP-VIDs supported by PBB-TE.
- SPVID-MSTID (0xFFF) are VIDs reserved for use by ISIS-SPB as SPVIDs.
- All VIDs allocated to other values of MSTID are assigned to the MSTI identified by that MSTID.

NOTE 2—While MST Configuration Table is capable of detailing the configuration of each SPVID it captures the allocation of Base VIDs only. Individual allocation of SPVIDs are captured in the ISIS-SPB database, the SPB ECT Static Entry (12.25.4) and in the port VID translation tables. By specifying only the type of VID and its allocation to SPB, backwards compatibility with STP, RSTP and MSTP is maintained.

8.7 The Learning Process

Change the first paragraph of 8.7 as follows:

The Learning Process receives the source MAC Addresses and VIDs of received frames from the Forwarding Process, subject to active topology enforcement (8.6.1) and the application of ingress filtering (8.6.2). The Learning Process is not invoked (see 8.6.1) for frames whose VID is an ESP-VID or identifies a VLAN supported by SPBM.

Insert the following subclause, 8.7.3 (including Table 8-6), after 8.7.2, and renumber the subsequent tables in Clause 8 accordingly:

8.7.3 Ageing of Dynamic Filtering Entries

Dynamic Filtering Entries shall be automatically removed after a specified time, the Ageing Time, has elapsed since the entry was created or last updated by the Learning Process.

The ageing out of Dynamic Filtering Entries ensures that end stations that have been moved to a different part of the network will not be permanently prevented from receiving frames. It also takes account of changes in the active topology of the network that can cause end stations to appear to move from the point of view of the bridge; i.e., the path to those end stations subsequently lies through a different Bridge Port.

The Ageing Time may be set by management (Clause 12). A range of applicable values and a recommended default is specified in Table 8-6; this is suggested to remove the need for explicit configuration in most cases. If the value of Ageing Time can be set by management, the Bridge shall have the capability to use values in the range specified, with a granularity of 1 s.

Table 8-6—Ageing time parameter value

Parameter	Recommended default value	Range
Ageing time	300.0 s	10.0–1 000 000.0 s

NOTE—The granularity is specified in order to establish a common basis for the granularity expressed in the management operations defined in Clause 12, not to constrain the granularity of the actual timer supported by a conformant implementation. If the implementation supports a granularity other than 1 s, then it is possible that the value read back by management following a Set operation will not match the actual value expressed in the Set.

The Spanning Tree Algorithm and Protocol specified in Clause 8 of IEEE Std 802.1D, 1998 Edition, includes a procedure for notifying all Bridges in the network of topology change. It specifies a short value for the Ageing Timer, to be enforced for a period after any topology change (8.3.5 of IEEE Std 802.1D, 1998 Edition). While the topology is not changing, this procedure allows normal ageing to accommodate extended periods during which addressed end stations do not generate frames themselves, perhaps through being powered down, without sacrificing the ability of the network to continue to provide service after automatic configuration.

8.8 The Filtering Database

Change the first paragraph of 8.8 as follows:

The Filtering Database supports queries by the Forwarding Process to determine whether received frames, with given values of destination MAC Address and VID, are to be forwarded through a given potential transmission Port (8.6.1, 8.6.3, 8.6.4). It can also be used, by SPBM (8.6.1), to determine whether received frames with given values of source B-MAC Address and B-VID, are to be forwarded from a given reception Port. The Filtering Database contains filtering information in the form of filtering entries that are either

a) Static, and explicitly configured by management action; or
b) Dynamic, and automatically entered into the Filtering Database by the normal operation of the bridge and the protocols it supports.

In 8.8, change the following list items (including the insertion of Note 1) as shown, and renumber the subsequent notes in 8.8 accordingly:

h) Each Dynamic Filtering Entry specifies the Port through which a frame with a given individual MAC Address and VID has been or can be received. Dynamic Filtering Entries can be created and updated by the Learning Process (8.7) and by ISIS-SPB. Entries that are updated by the Learning Process are subject to ageing and removal by the Filtering Database.

NOTE 1—ISIS-SPB has to create the Dynamic Filtering Entries for VLANs supported by SPBM, since learning is not used. Initial creation of these entries does no harm for VLANs supported by SPBV (when learning is used) as it can reduce flooding of frames to unknown destinations.

i) MAC Address Registration Entries support the registration of MAC Addresses. They are created, updated, and removed by MMRP in support of Extended Filtering Services (8.8.4, and Clause10) and by ISIS-SPB. If the Restricted_MAC_Address_Registration management control (10.12.2.3) is TRUE, then the creation of a MAC Address Registration Entry by MMRP is not permitted unless a Static Filtering Entry exists that permits dynamic registration for the Group concerned.

j) Dynamic VLAN Registration Entries are used to specify the Ports on which VLAN membership has been dynamically registered. They are created, updated, and removed by MVRP, ~~and~~ MIRP, and ISIS-SPB, in support of automatic VLAN membership configuration (Clause 11 and Clause 39)~~;~~. The MVRP controlled Dynamic VLAN Registration Entries are subject to the state of the Restricted_VLAN_Registration management control (11.2.3.2.3). If the value of this control is TRUE, then the creation of a Dynamic VLAN Registration Entry by MVRP is not permitted unless a Static VLAN Registration Entry exists that permits dynamic registration for the VID concerned.

In 8.8, change paragraph 15, and insert a new paragraph after paragraph 15 as follows:

The Filtering Database shall support the creation, updating, and removal of Dynamic Filtering Entries by the Learning Process (8.7). In Bridges that support Extended Filtering Services, the Filtering Database shall support the creation, updating, and removal of MAC Address Registration Entries by MMRP (Clause 10). In bridges that support SPBM, Dynamic Filtering Entries are populated by ISIS-SPB (Clause 28). In bridges that support SPBV the Dynamic Filtering Entries are created and updated by the Learning Process (8.7) and the use of ISIS-SPB for this operation is optional (28.12.9).

Shortest Path Bridging (SPB), (27.10) allocates SPVIDs dynamically to support shortest path bridging of frames assigned to one or more VLANs, each identified by a Base VID (Clause 3). A Static Filtering Entry should not be created for an SPVID, irrespective of whether that SPVID is currently used or not, and SPVIDs should not be used in registration protocols that create MAC Address Registration Entries or Dynamic VLAN Registration Entries. In an SPT Bridge, configuration or registration of a filtering database entry with the Base VID of a VLAN supported by SPBV shall cause the bridge to filter frames as if that entry was also present for every one of the SPVIDs for that Base VID.

Change the following note in 8.8 as shown:

NOTE ~~2~~3—The ability to create VLAN-dependent Filtering Database entries allows a Bridge to support ~~—M~~multiple end stations with the same individual MAC Address residing on different VLANs. It also supports ~~;—E~~end stations with multiple interfaces, each using the same individual MAC Address, as long as not more than one end station or interface that uses a given MAC Address resides in a given VLAN.

8.8.1 Static Filtering Entries

Change the following list item in 8.8.1 as follows:

a) A MAC Address specification, comprising
   3) All Individual Addresses, for which no more specific ~~Static~~ Filtering Entry exists; or

Change the second paragraph of 8.8.1 as follows:

All Bridges shall have the capability to support the first two values for the MAC Address specification, both values of the VLAN identifier specification, and all three values for each control element for all Static Filtering Entries [i.e., shall have the capability to support item a1), item a2), item b1), item b2), item c1), item c2), and item c3)]. All Bridges supporting PBB-TE or SPT Bridges shall, in addition, support items a3) and a4) for ESP-VIDs (8.9) or VIDs identifying VLANs supported by SPBM respectively and shall always have in their filtering databases Static Filtering Entries specifying items a3) and a4), and a Port Map specifying filtering for each outbound port, for every ESP-VID or SPBM VID respectively.

8.8.2 Static VLAN Registration Entries

Insert the following note at the end of 8.8.2:

NOTE 3—For VLANs supported by SPBV the Filtering specified by a static VLAN registration entry for the Base VID is applied for all SPVIDs associated with that Base VID.

8.8.3 Dynamic Filtering Entries

Change 8.8.3 as follows:

A Dynamic Filtering Entry specifies

a) An individual MAC Address;
b) The FID, an identifier assigned by the MAC Bridge (8.8.8) to identify a set of VIDs for which no more than one Dynamic Filtering Entry can exist for any individual MAC Address;

NOTE 1—An FID identifies a set of VIDs among which *Shared VLAN Learning* (Clause 3) takes place. Any pair of FIDs identifies two sets of VIDs between which *Independent VLAN Learning* (Clause 3) takes place. The allocation of FIDs by a Bridge is described in 8.8.8.

c) A Port Map specifying forwarding for the destination MAC Address and FID to a single Port.

NOTE 2—This is equivalent to specifying a single port number; hence, this specification is directly equivalent to the specification of dynamic entries in IEEE Std 802.1D, 1993 Edition [B9].

The Port Map may contain a **connection_identifier** (8.8.12) for each outbound Port.

Dynamic Filtering Entries ~~are~~ can be created, ~~and~~ updated, and removed by the Learning Process (8.7). They shall be automatically removed after a specified time, the Ageing Time (8.7.3), has elapsed since the entry was created or last updated. ~~No more than one Dynamic Filtering Entry shall be created in the Filtering Database for a given combination of MAC Address and FID.~~

~~Dynamic Filtering Entries cannot be created or updated by management.~~

Dynamic Filtering Entries can also be created and updated by SPB (Clause 27). ISIS-SPB (Clause 28) refreshes these entries, so they are not aged out by the normal operation of the Learning Process. These entries are used by SPBM (Clause 27) to support both unicast loop mitigation (6.5.4.2) and multicast loop prevention (6.5.4.1) for active topology enforcement (8.6.1), as well as to locate end stations. These entries are updated by the normal Learning Process for SPBV, but could also be controlled by ISIS-SPB. This allows entries for the higher layer entities in a bridge (see Figure 8-11) to be populated rapidly after any topology change.

No more than one Dynamic Filtering Entry shall be created in the Filtering Database for a given combination of MAC Address and FID. Dynamic Filtering Entries cannot be created or updated by management.

NOTE 3—Dynamic Filtering Entries may be read by management (Clause 12). The FID is represented in the management Read operation by any one of the VIDs that it represents. For a given VID, the set of VIDs that share the same FID may also be determined by management.

~~The ageing out of Dynamic Filtering Entries ensures that end stations that have been moved to a different part of the network will not be permanently prevented from receiving frames. It also takes account of changes in the active topology of the network that can cause end stations to appear to move from the point of view of the bridge; i.e., the path to those end stations subsequently lies through a different Bridge Port.~~

~~The Ageing Time may be set by management (Clause 12). A range of applicable values and a recommended default is specified in Table 8-6; this is suggested to remove the need for explicit configuration in most cases. If the value of Ageing Time can be set by management, the Bridge shall have the capability to use values in the range specified, with a granularity of 1 s.~~

~~Table 8-6—Ageing time parameter value~~

~~Parameter~~	~~Recommended default value~~	~~Range~~
~~Ageing time~~	~~300.0 s~~	~~10.0–1 000 000.0 s~~

~~NOTE 4—The granularity is specified in order to establish a common basis for the granularity expressed in the management operations defined in Clause 12, not to constrain the granularity of the actual timer supported by a conformant implementation. If the implementation supports a granularity other than 1 s, then it is possible that the value read back by management following a Set operation will not match the actual value expressed in the Set.~~

~~The Spanning Tree Algorithm and Protocol specified in Clause 8 of IEEE Std 802.1D, 1998 Edition includes a procedure for notifying all Bridges in the network of topology change. It specifies a short value for the Ageing Timer, to be enforced for a period after any topology change (8.3.5 of IEEE Std 802.1D, 1998 Edition). While the topology is not changing, this procedure allows normal ageing to accommodate extended periods during which addressed end stations do not generate frames themselves, perhaps through being powered down, without sacrificing the ability of the network to continue to provide service after automatic configuration.~~

8.8.4 MAC Address Registration Entries

Change the second and third paragraphs in 8.8.4 as follows:

MAC Address Registration Entries are created, modified, and deleted by the operation of MMRP (Clause 10) or by ISIS-SPB. No more than one MAC Address Registration Entry shall be created in the Filtering Database for a given combination of MAC Address specification and VID.

NOTE—It is possible to have a Static Filtering Entry that has values of Forward or Filter on some or all Ports that mask the dynamic values held in a corresponding MAC Address Registration Entry. The values in the MAC Address Registration Entry will continue to be updated by MMRP; hence, subsequent modification of that entry to allow the use of dynamic filtering information on one or more Ports immediately activates the true MMRP registration state that was hitherto masked by the static information.

The creation of MAC Address Registration Entries by MMRP is subject to the Restricted_MAC_Address_Registration management control (10.12.2.3). If the value of this control is TRUE, a dynamic entry for a given MAC address may only be created if a Static Filtering Entry already exists for that MAC address, in which the Registrar Administrative Control value is Normal Registration.

8.8.8 Allocation of VIDs to FIDs

Change 8.8.8 and 8.8.8.1 as follows:

~~The allocation of VIDs to FIDs within a Bridge determines how learned individual MAC Address information is used in forwarding/filtering decisions within a Bridge; whether such learned information is confined to individual VIDs, shared among all VIDs, or confined to specific sets of VIDs.~~

~~The allocation of VIDs to FIDs is determined on the basis of~~

~~a)~~ ~~The set of VLAN Learning Constraints that have been configured into the Bridge (by means of the management operations defined in Clause 12);~~

~~b)~~ ~~Any fixed mappings of VIDs to FIDs that may have been configured into the Bridge (by means of the management operations defined in Clause 12);~~

~~c)~~ ~~The set of active VIDs (i.e., those VIDs on whose behalf the Bridge may be called on to forward frames). A VID is active if either of the following is true:~~

   ~~1)~~ ~~The VID's member set (8.8.10) contains one Port that is in a forwarding state, and at least one other Port of the Bridge is both in a forwarding state and has Ingress Filtering (8.6.2) disabled;~~

   ~~2)~~ ~~The VID's member set contains two or more Ports that are in a forwarding state.~~

~~d)~~ ~~The capabilities of the Bridge with respect to the number of FIDs that it can support, and the number of VIDs that can be allocated to each FID.~~

~~A Bridge shall support a minimum of one FID and may support up to 4094 FIDs. For the purposes of the management operations, FIDs are numbered from 1 through N, where N is the maximum number of FIDs supported by the implementation.~~

~~A Bridge shall support the ability to allocate at least one VID to each FID and may support the ability to allocate more than one VID to each FID.~~

~~The number of VLAN Learning Constraints supported by a Bridge is an implementation option.~~

~~NOTE—In an SVL/IVL Bridge (3.167), a number of FIDs are supported, and one or more VIDs can be mapped to each FID. In an SVL Bridge (3.166), a single FID is supported, and all VIDs are mapped to that FID. In an IVL Bridge (3.76), a number of FIDs are supported, and only one VID can be mapped to each FID.~~

~~An MST Bridge shall support the ability to allocate at least one FID to each spanning tree and may support the ability to allocate more than one FID to each spanning tree.~~

~~NOTE—In other words, the number of FIDs supported by the Bridge must be greater than or equal to the number of spanning trees supported by the Bridge.~~

~~An MST Bridge shall ensure that the maximum supported numbers of FIDs and VIDs can be associated unambiguously. This requires either 1) a number of fixed VID to FID allocations at least equal to the maximum number of VIDs supported; or 2) one I Constraint entry per FID supported and one S Constraint entry per MSTI supported, or both (8.8.8.2).~~

The allocation of VIDs to FIDs within a VLAN-aware Bridge determines how learned individual MAC Address information is used in forwarding and filtering decisions. If two VIDs are allocated to the same FID learned information is shared, i.e., an individual MAC address learned following receipt of a frame affects the forwarding or filtering of a subsequent frame with the same address as its destination address if either of the two VIDs have been assigned to each of the two frames. If two VIDs are allocated to different FIDs learned information is independent, i.e., an individual MAC address for frames that have one of the VIDs assigned does not affect forwarding or filtering of frames with the other VID. Independent VLAN Learning can allow two separate stations to use the same MAC address, provided they use different VLANs, and also allows frames transmitted by a single station to be assigned to independent active topologies. Shared VLAN Learning allows support of a single VLAN by multiple VIDs, enabling further control over the filtering of frames for that VLAN, and allowing shortest path bridging to support the VLAN with a set of symmetric trees while still using learning. Further considerations for the use of Independent and or Shared VLAN Learning are described in Annex F (informative).

A VLAN-aware Bridge shall support either Shared or Independent VLAN Learning, and may support both. If Independent VLAN Learning is not supported the bridge shall support a single FID. If Independent VLAN Learning is supported, the number of FIDs shall be the same as the maximum number of VLANs supported by the implementation. For the purposes of the management operations, FIDs are numbered from 1 through N, where N is the maximum number of FIDs supported by the implementation. A bridge that supports both Shared and Independent VLAN Learning shall be capable of allocating any VID to any FID, through configuration of the VID to FID allocation table. Following any set of management operations designed to change the allocation of VIDs to FIDs, whether initiated by an administrative request or as a consequence of other aspects of the bridge’s operation, the value of any FID that is in use should be the same as that of one of the VIDs allocated to that FID. One exception to this rule is the allocation of SPVIDs to FIDs for SPBV (Clause 27). SPVIDs that are allocated are associated with a Base VID’s FID. The learning behavior of SPVIDs assigned to a FID is determined by the learning behavior of the associated Base VID’s FID (27.11)

~~8.8.8.1 Fixed and dynamic VID to FID allocations~~

~~A Bridge may support the ability to define fixed allocations of specific VIDs to specific FIDs, via an~~ The VID to FID allocation table ~~that~~ may be read and modified by means of the management operations defined in Clause 12. For each VID supported by the implementation, the allocation table indicates one of the following:

a) The VID is currently not allocated to any FID; or
b) A fixed allocation has been defined (via management), allocating this VID to a specific FID; or
c) A dynamic allocation has been defined ~~(as a result of applying the VLAN Learning Constraints)~~, allocating this VID to a specific FID.

~~For any VID that has no fixed allocation defined, the Bridge can dynamically allocate that VID to an appropriate FID, in accordance with the current set of VLAN Learning Constraints.~~

An SPT Bridge can dynamically allocate VIDs, that have been reserved for use as SPVIDs, to FIDs.

An MST Bridge shall support Independent VLAN Learning and may support Shared VLAN Learning. An MST Bridge shall be capable of allocating any FID to any MSTID through configuration of the FID to MSTID Allocation Table (12.12.2). An SPT Bridge supports both Independent and Shared VLAN Learning.

Delete 8.8.8.2 (VLAN Learning Constraints) and 8.8.8.3 (VLAN Leaning Constraint inconsistencies and violations) in their entirety.

Change 8.9 through 8.9.3 as follows:

8.9 MST, SPB, and ESP configuration information

In order to support multiple spanning trees, and shortest path bridging, all the bridges in an MST or SPT Region have ~~an MST Bridge has~~ to be configured with ~~an unambiguous~~ assignments of VIDs to spanning trees. ~~This is achieved by:~~

~~a) Ensuring that the allocation of VIDs to FIDs (8.8.8) is unambiguous; and~~
~~b) Ensuring that each FID supported by the Bridge is allocated to exactly one Spanning Tree.~~

~~The first of these requirements is met by configuring a set of VLAN learning constraints and/or fixed VID to FID mappings that are self-consistent, and which define an I Constraint, an S Constraint, or a fixed VID to FID allocation for all VIDs supported by the Bridge.~~

~~The second requirement is met by means of the FID to MSTI Allocation Table (8.9.3).~~

The combination of the VID to FID allocations (8.8.8) and the FID to MSTI allocations (8.9.3) defines a mapping of VIDs to spanning trees, MSTIDs, represented by the MST Configuration Table (8.9.1). If an MSTID is included in the MSTI List (12.12.1), frames with VIDs allocated to that MSTID will be supported by an MSTI calculated by MSTP (Clause 13). If the MSTID is not included in the MSTI List, and is not a reserved MSTID, those frames will be supported by the IST.

A ~~Bridge~~ bridge supporting PBB-TE can assign specific VID values to ESPs but can also support any of the spanning tree protocols. This is achieved by allocating the VIDs associated with the PBB-TE ESPs to a special MSTID, the TE-MSTID.

An SPT Bridge can allocate any given VLAN to the CIST, or to the reserved MSTIs, as well as supporting shortest path bridging for other VLANs within an SPT Region. The assignment of VLANs and VIDs to MSTIs is subject to similar considerations and constraints as for MST Bridges. In addition to the Base VID used to identify frames for the VLAN when transmitted on the CST outside the region, each VLAN supported by SPBV uses a number of dynamically allocated shortest path VIDs (SPVIDs), one for each SPT Bridge participating in the VLAN within the region.

The network administrator allocates enough SPVIDs for SPBV to avoid unintentionally excluding bridges from the topology. The details of SPVID allocation, including mitigation of the consequences of not allocating enough SPVIDs, are specified in 27.10.

Administratively, by not translating from Base VID to SPVID, an SPBV VLAN can be forced to the IST. When there is no shortage of SPVIDs and the MCIDs match but the Base VID algorithms disagree the algorithm defaults to the Spanning tree. This behavior allows SPBV to change the ECT Algorithms.

Dynamic SPVID allocation is supported by ISIS-SPB and allows the addition of bridges to existing SPT Regions without disruptive configuration changes, as well as allowing supporting auto configuration of shortest path bridging in simple networks where all user data is assigned to a VLAN identified by the default PVID of 1 (Table 9-2). However, prior to SPVID allocation, all bridges in an SPT Region have to agree which VLANs are to be shortest path bridged, and which SPT Set (ECT -Algorithm) is to be used for each of those VLANs. By default, all VLANs whose Base VID is allocated to MSTID 0xFFD will be supported by SPBV using the SPB default ECT Algorithm. The SPT configuration information can also specify that frames for SPBV and SPBM be supported by another specified SPB ECT Algorithm as specified by configuration (12.25.4, Table 28-1).

NOTE—The use of MSTIDs to support allocation of Base VIDs to SPB allows the per VLAN components of the necessary information to be conveyed in the MST Configuration Digest, thus avoiding any need for a separate digest. However the digest is not specific about how the VIDs are mapped to specific SPTs, just that they are allocated to SPBV, SPBM, or SPVIDs.

8.9.1 MST Configuration Table

The MST Configuration Table specifies an MSTID for each possible VID.

In an MST Bridge, ~~defines, for each VID, the~~ each MSTID identifies ~~of~~ the MSTI ~~spanning tree instance~~ to which the VID is allocated and an MSTID of zero identifies the CIST.

~~In a Bridge that supports PBB-TE an MSTID of 0xFFE, the TE-MSTID, identifies VIDs that can be used by ESPs. Each of these VIDs, the ESP-VIDs, shall be allocated to an individual FID.~~

The combination of the VID to FID allocations (8.8.8) and the FID to MSTID allocations (8.9.3) define a mapping of VIDs to MSTIDs summarized in the MST Configuration Table (8.9.1).~~The MST Configuration Table cannot be configured directly; configuration of the table occurs as a consequence of configuring the relationships between VIDs and FIDs (8.8.8) and between FIDs and spanning trees (8.9.3)~~.

8.9.2 MST configuration identification

For two or more MST Bridges to be members of the same MST Region (Clause 3), it is necessary for those Bridges to ~~be directly connected together (i.e.,~~ support the same MST Region configuration and to be interconnected only by means of LANs, ~~without intervening Bridges that are not members of the region), and for them to support the same MST Region configuration~~ with all intervening Bridges being members of the region or being transparent to the BPDUs of the members of the region. Two MST Region configurations are considered to be the same if the correspondence between VIDs and ~~spanning trees~~ MSTIDs is identical in both configurations and they use the same information to identify the configuration, including the same Configuration Name.

NOTE 1—If two adjacent MST Bridges consider themselves to be in the same MST Region despite having different mappings of VIDs to ~~spanning trees~~MSTIDs, then the possibility exists of undetectable loops arising within the MST Region.

In order to ensure that adjacent MST Bridges are able to determine whether they are part of the same MST Region, the MST BPDU supports the communication of an MST Configuration Identifier (13.8).

NOTE 2—As the MST Configuration Identifier is smaller than the mapping information that it summarizes, there is a small but finite possibility that two MST Bridges will assume that they have the same MST Region Configuration when this is not actually the case. However, given the size of the identifier, this standard assumes that this possibility is sufficiently small that it can safely be ignored. Appropriate use of the Configuration Name and Revision Level portions of the identifier can remove the possibility of an accidental match between MST Configuration Identifiers that are derived from different configurations within a single administrative domain (see 13.8).

8.9.3 FID to MSTID Allocation Table

The FID to MSTID Allocation Table defines, for all FIDs that the Bridge supports, the MSTID ~~of the spanning tree instance~~ to which the FID is allocated. ~~An MSTID of zero is used to identify the CIST.~~

a) The IST is identified by the reserved MSTID value 0.
b) The use of PBB-TE is identified by the reserved MSTID value TE-MSTID (0xFFE).
c) Each MSTID in the MSTI List identifies an MSTI.
The reserved MSTID values 0 and TE-MSTID, SPBV-MSTID, SPBM-MSTID and 0xFFF are never used in the MSTI List.

d) The following MSTID values identify the method used to support the VLANs identified by the Base VIDs allocated to those MSTIDs:
  1) 0xFFC—SPBM-MSTID
  2) 0xFFD—SPBV-MSTID
  3) 0xFFF—Allocated to FIDs that are not used to filter frames including SPVIDs for SPBV

NOTE—MSTIDs that are present in the MSTI List (12.12) identify spanning tree instances supported by MSTP. MSTIDs identify (indirectly) VLANs that are supported by shortest path bridging.

~~NOTE—The management operations defined in 12.12 make use of the concept of an MSTI List to instantiate/de-instantiate MST instances and will only permit the allocation of FIDs to MSTIDs that are present in the MSTI List.~~

Insert the following subclause, 8.9.4, after 8.9.3:

8.9.4 SPT Configuration Identification

For two or more SPT Bridges to be members of the same SPT Region (Clause 3), it is necessary for those Bridges to support the same SPT Region configuration and to be interconnected only by means of LANs, with all intervening Bridges being members of the region or being transparent to the BPDUs and/or SPB Hello PDUs of the members of the region. This requirement is identical to that for MST Regions: a given SPT Region can support selected MSTIs as well as shortest path trees. MCIDs only specify the VIDs that are configured.

Two SPT Region configurations are considered to be the same if they have the same MST Region configuration or alternate MST Region configuration, they agree on the VLANs that are to be shortest path bridged and they use the same ISIS-SPB. Furthermore by using ISIS-SPB they agree on which SPT Set is to be used for each of these VLANs (Clause 27).

The MST Region information captures the VIDs for SPB but it does not capture the operational state of the VIDs for SPB. This allows allocation of SPVIDs and upgrade of multiple ECT Algorithms without change to the MST Region when sufficient pre allocation of VIDs has been configured. However in situations where the network grows beyond the current allocation of VIDs the options are to temporarily create a region boundary as the new configuration is deployed or allow in-service upgrades by making changes that are compatible with the current MST region configuration.

To support in-service upgrade from one set of VID to MSTID allocations to another, SPT Bridges support a second (auxiliary) MCID (28.9). Two such bridges consider themselves to be in the same MST Region provided that at least one of their MCIDs matches with one of their neighbors. In this way, a new configuration can be incrementally introduced without time constraints, and put into operation, all the while maintaining operation as a single SPT Region.

8.10 Spanning Tree Protocol Entity

Insert the following paragraph at the end of 8.10:

The ISIS-SPB Entity comprises an instance of IS-IS with shortest path bridging extensions as specified in Clause 28. The ISIS-SPB Entity is described, for the purposes of management, as distinct from the Spanning Tree Protocol Entity. Nonetheless, the ISIS-SPB and Spanning Tree Protocol Entities that calculate active topologies for the same Bridge Port use the same ISS service access point (8.5, 27.2), and the same individual MAC address. However, ISIS-SPB uses one of the group MAC Addresses in Table 8-13 and does not use the group MAC Addresses used by the Spanning Tree Protocol specified in Table 8-1. The ISIS-SPB Entity uses the state machines and state machine variables supported by the Spanning Tree Protocol Entity (Clause 27, 13.6) and calculates the Agreement Digest that the Spanning Tree Protocol Entity transmits in SPT BPDUs and/or SPB Hello PDUs.

8.13 Addressing

Change 8.13 as follows:

All MAC Entities communicating across a ~~Bridged Local Area Network~~ bridged network use EIU-48 addresses. These can be Universally Administered Addresses or a combination of Universally Administered and Locally Administered Addresses.

8.13.3 Use of LLC by Spanning Tree Protocol Entities

Change 8.13.3 as follows:

Spanning Tree Protocol Entities uses the DL_UNITDATA.request and DL_UNITDATA.indication primitives (ISO/IEC 8802-2) provided by individual LLC Entities associated with each Bridge Port to transmit and receive ~~frames~~BPDUs (Clause 14). The source_address and destination_address parameters of the DL_UNITDATA.request shall both denote the standard LLC address assigned to the Bridge Spanning Tree Protocol (Table 8-13~~Table 8-12~~). Each DL_UNITDATA request primitive gives rise to the transmission of an LLC UI command PDU, which conveys the BPDU in its information field.[4]

Table 8-13—~~Table 8-12—~~Standard LLC address assignment

Assignment	Value
Bridge spanning tree protocol	01000010
IS-IS protocol PDUs	11111110

Code Representation: The least significant bit (LSB) of the value shown is the right-most. The bits increase in significance from right to left. It should be noted that the code representation used here has been chosen in order to maintain consistency with the representation used elsewhere in this standard, and with the representation used in IEEE Std 802.1D.

~~IEEE Std 802.1D defines a~~ A Protocol Identifier field, present in all BPDUs (Clause 14), ~~which~~ serves to identify different protocols supported within the scope of the LLC address assignment. Further values of this field are reserved for future standardization. A Spanning Tree Protocol Entity that receives a BPDU with an unknown Protocol Identifier shall discard that PDU.

IS-IS protocol PDUs are LLC (encoded with a DSAP and SSAP of 0xFE (section 7.5 of ISO 10589:2002), then a single byte indicating Control 0x03 (5.2.3 and 5.3.1 of IEEE Std 802.2, 1998 Edition) immediately followed by a IS-IS PDU starting with an Interdomain Routeing Protocol Discriminator equal to 0x83 (section 5.3 of ISO 9577:1999).

Change 8.13.5 as follows (including creating a new subclause, 8.13.5.1):

8.13.5 Group MAC Addresses for spanning tree ~~protocols~~entity

A Spanning Tree Protocol Entity uses an instance of the MAC Service provided by each of the Bridge's Ports to transmit frames addressed to the Spanning Tree Protocol Entities of all the other Bridges attached to the same LAN as that Port.

[4] Administration of LLC address assignments is the responsibility of the IEEE Registration Authority (IEEE RA). A full list of standard LLC address assignments, and the criteria for assignment, can be found on the IEEE RA website at http://standards.iee.org/regauth/index.html.

8.13.5.1 Group MAC Addresses for spanning tree protocols

Spanning Tree protocols use a reserved address for peer to peer communication between entities of the same type.

An EUI-48 universally administered Group Address, known as the Bridge Group Address, has been assigned (Table 8-1) for use by C-VLAN components for this purpose.

It is essential to the operation of the spanning tree protocols that frames with this destination address both reach all peer protocol entities attached to the same LAN and do not reach protocol entities attached to other LANs. The Bridge Group Address is therefore included in the block of C-VLAN components Reserved MAC Addresses and is always filtered by Customer Bridges and C-VLAN components of Provider Edge Bridges (8.6.3).

As a network of Provider Bridges needs to appear to attached Customer Bridges as if it was a single LAN, frames addressed to the Bridge Group Address are forwarded by S-VLAN components. Provider Bridges also use a Spanning Tree Protocol to provide one or more loop-free active topologies, so a distinct EUI-48 universally administered Group Address, the Provider Bridge Group Address, which can be confined to the LANs that their Bridge Ports attach, has been assigned (Table 8-2). The Provider Bridge Group Address is included in both the C-VLAN component and the S-VLAN component Reserved MAC Addresses and is always filtered by all Bridges (8.6.3).

The source MAC address field of frames transmitted by Spanning Tree Protocol Entities contains the individual MAC Address for the Bridge Port used to transmit the frame.

Insert the following subclause, 8.13.5.2, after 8.13.5.1:

8.13.5.2 Group MAC Addresses for Shortest Path Bridging

ISIS-SPB use a group address for ISIS-SPB peer to peer communication. ISIS-SPB is configured to use only one of the addresses listed in Table 8-14. All other addresses in this table will be ignored by IS-IS and will not be filtered by the MAC Relay. The choice of which address is used depends on the environment.

Table 8-14 lists the various addresses that may be configured for ISIS-SPB. Table 8-15 lists the deployment environments for SPB. The usage of ISIS-SPB is described in Clause 28. These addresses are from a reserved address space.These addresses have the property that:

a) Where a given address in the set is used by a bridge component to support ISIS-SPB, frames destined for that address shall not be forwarded by that bridge component; i.e., a static filtering entry for that address is maintained for that group MAC address in order to prevent the forwarding of frames destined for that address.
b) Where a given address in the set is not used by a bridge component to support ISIS-SPB, frames destined for that address received on any Port that is part of a given active topology shall be forwarded by that bridge component on all other Ports that are part of that active topology.

An SPT Bridge, whether it is running SPBV, or SPBM, or both, uses a single Group MAC address. Table 8-15 provides the reference for the possible environments.

In Provider Bridge only environments, for SPBV the address should be the allocated All Provider Bridge Intermediate Systems address.

In Customer Bridge only environments where only SPBV is supported the address should be the All Customer Bridge Intermediate Systems address.

In SPBM or SPBV deployments where only Provider backbone components are configured the use of addresses is flexible as indicated in Table 8-15. To allow peering with other systems capable of using IS-IS for IP, the existing IS-IS addresses may be used.

Table 8-14—ISIS-SPB reserved addresses

Assignment	Value
All Level 1 Intermediate Systems	01-80-C2-00-00-14[a]
All Level 2 Intermediate Systems	01-80-C2-00-00-15[a]
All Intermediate Systems	09-00-2B-00-00-05[a]
All Provider Bridge Intermediate Systems	01-80-C2-00-00-2E
All Customer Bridge Intermediate Systems	01-80-C2-00-00-2F

[a] The first three addresses are existing IS-IS addresses (Table 9 of ISO 10589:2002).

Table 8-15—ISIS-SPB Recommended Address Usage

Network Deployment	Level 1	Level 2	All Levels
SPT Bridges Peering with other IS-IS capable systems	01-80-C2-00-00-14	01-80-C2-00-00-15	09-00-2B-00-00-05
SPT Bridges Provider Backbone Bridges only	01-80-C2-00-00-14	01-80-C2-00-00-15	09-00-2B-00-00-05 or 01-80-C2-00-00-2E
SPT Bridges in SPBV mode for S-VLANs Provider Bridge	01-80-C2-00-00-2E		
SPT Bridges in SPBV mode for C-VLANs Customer Bridge	01-80-C2-00-00-2F		

8.13.9 Points of attachment and connectivity for Higher Layer Entities

Change the first paragraph of 8.13.9 as follows:

The Higher Layer Entities in a ~~Bridge~~bridge, such as the Spanning Tree Protocol Entity (8.10), MRP entities (8.11), the ISIS-SPB Entity (Clause 28), and Bridge Management (8.12), are modeled as attaching directly to one or more individual LANs connected by the Bridge's Ports, in the same way that any distinct end station is attached to the network. While these entities and the relay function of the ~~Bridge~~ bridge use the same individual MAC entities to transmit and receive frames, the addressing and connectivity to and from these entities is the same as if they were attached as separate end stations "outside" the Port or Ports where they are actually attached. Figure 8-12 is functionally equivalent to Figure 8-2 but illustrates this logical separation between the points of attachment used by the Higher Layer Entities and those used by the MAC Relay Entity.

9. Tagged frame format

9.7 Backbone Service Instance Tag Control Information

Change Table 9-3 as follows:

Table 9-3—Reserved I-SID values

I-SID value (hexadecimal)	Meaning/Use
0	Reserved for ~~use by future amendments to this standard.~~ implementation use. This I-SID value shall not be configured as an identifier for a backbone service instance or transmitted in a Backbone Service Instance Tag header. This I-SID value may be used in management and control operations, e.g., to indicate a deleted or null I-SID in ISIS-SPB TLVs (see Clause 28).
1	Default value—Unassigned ISID on a VIP.
2 through F~~F~~E	Reserved for use by future amendments to this standard.
FF	SPBM default I-SID
FFFFFF	Reserved for implementation use. This I-SID value shall not be configured as an identifier for a backbone service instance or transmitted in a Backbone Service Instance Tag header. This I-SID value may be used to indicate a wildcard match for the I-SID in management operations.

12. Bridge management

12.1 Management functions

12.1.1 Configuration management

Insert the following list item at the end of 12.1.1:

k) The ability to control the operation of SPB.

12.2 VLAN-aware bridge objects

Insert the following list item after item p) in 12.2:

q) The Shortest Path Bridging managed objects.

12.3 Data Types

Insert the following list items at the end of 12.3:

q) ECT-ALGORITHM. A 4-byte unsigned identifier. Used as a worldwide unique definition of an ECT-ALGORITHM, the first 3 bytes are expected to be taken from the OUI space for the organization that has defined the algorithm. The last byte is allocated by that organization.

r) SPSourceID. A 20-bit Unsigned identifier. Used to represent a node uniquely within an SPT Domain (27.10).

12.7 Filtering Database

12.7.7 General Filtering Database operations

12.7.7.1 Create Filtering Entry

12.7.7.1.3 Outputs

Change 12.7.7.1.3 as follows:

a) Operation rejected because a Port identified by the Port Map includes a port already in the member set of a VID of a different type than the currently registered VID;
b) Operation rejected because the specified Static Filtering Entry is associated with an Infrastructure Protection Group and is administered by IPS Control;
c) Operation rejected because the VID identifies an SPVID;
d) ~~c)~~ Operation accepted.

12.10 Bridge VLAN managed objects

12.10.1 Bridge VLAN Configuration managed object

Insert the following list items at the end of 12.10.1:

i) Configure VID Translation Table (12.10.1.8);
j) Configure Egress VID Translation Table (12.10.1.9).

12.10.1.1 Read Bridge VLAN Configuration

12.10.1.1.3 Outputs

Insert the following list items in 12.10.1.1.3 c) ("For each Port:"):

8) Whether the Bridge supports a VID Translation Table on that Port; whether the Bridge supports an Egress VID Translation Table on that Port;
9) For a Bridge that supports a VID Translation Table on that Port, a set of (Local VID, Relay VID) bindings (6.9);
10) For a Bridge that supports an Egress VID Translation Table on that Port, a set of (Relay VID, Local VID) bindings (6.9).

Insert the following subclauses, 12.10.1.8 to 12.10.1.9.2.3, after 12.10.1.7.2:

12.10.1.8 VID Translation Configuration managed object

To configure the VID Translation Table (6.9) associated with a Port. This object is not applicable to Ports that do not support a VID Translation Table. The default configuration of the table has the value of the Relay VID equal to the value of the Local VID for all Local VID values. If the port supports an Egress VID translation table, the VID Translation Configuration object configures the Local VID to Relay VID mapping on ingress only. If an Egress VID translation is not supported, the VID Translation Configuration object defines a single bidirectional mapping.

The management operations that can be performed are as follows:

a) Read VID Translation Table Entry (12.10.1.8.1).
b) Configure VID Translation Table Entry (12.10.1.8.2).

12.10.1.8.1 Read VID Translation Table Entry

12.10.1.8.1.1 Purpose

To read the relay VID associated with the local VID.

12.10.1.8.1.2 Inputs

a) Port Number: the number of the port that is capable of performing VID translation.
b) Local VLAN Identifier: a 12-bit VID.

12.10.1.8.1.3 Outputs

a) Relay VLAN Identifier: a 12-bit VID.

12.10.1.8.2 Configure VID Translation Table Entry

12.10.1.8.2.1 Purpose

To modify an entry in the VID Translation Table.

12.10.1.8.2.2 Inputs

a) Port Number: the number of the port that is capable of performing VID translation. This port should be at the edge of an administrative or a control protocol domain (6.9).

b) Local VLAN Identifier: a 12-bit VID.
c) Relay VLAN Identifier: a 12-bit VID.

12.10.1.8.2.3 Outputs

a) Operation status. This takes one of the following values:
   1) Operation rejected because the port does not have Egress VID translation capabilities.
   2) Operation accepted.

12.10.1.9 Egress VID Translation Configuration managed object

To configure the Egress VID Translation Table (6.9) associated with a Port. This object is not applicable to Ports that do not support an Egress VID Translation Table. The default configuration of the table has the value of the Local VID equal to the value of the Relay VID for all Relay VID values.

The management operations that can be performed are as follows:

a) Read Egress VID Translation Table Entry (12.10.1.9.1).
b) Configure Egress VID Translation Table Entry (12.10.1.9.2).

12.10.1.9.1 Read Egress VID Translation Table Entry

12.10.1.9.1.1 Purpose

To read the local VID associated with the relay VID.

12.10.1.9.1.2 Inputs

a) Port Number: the number of the port that is capable of performing Egress VID translation.
b) Relay VLAN Identifier: a 12-bit VID.

12.10.1.9.1.3 Outputs

a) Local VLAN Identifier: a 12-bit VID.

12.10.1.9.2 Configure Egress VID Translation Table Entry

12.10.1.9.2.1 Purpose

To modify an entry in the Egress VID Translation Table.

12.10.1.9.2.2 Inputs

a) Port Number: the number of the port that is capable of performing Egress VID translation. This port should be at the edge of an administrative or a control protocol domain (6.9).
b) Relay VLAN Identifier: a 12-bit VID.
c) Local VLAN Identifier: a 12-bit VID.

12.10.1.9.2.3 Outputs

a) Operation rejected because the port does not have Egress VID translation capabilities.
b) Operation accepted.

12.10.3 The VLAN Learning Constraints managed object

In 12.10.3, change the "8.8.8.1" and "8.8.8.2" cross references to "8.8.8."

12.10.3.3 Set VLAN Learning Constraint

12.10.3.3.3 Outputs

In 12.10.3.3.3, change the "8.8.8.3" cross references to "8.8.8."

12.10.3.8 Set VID to FID allocation

12.10.3.8.3 Outputs

In 12.10.3.8.3, change the "8.8.8.3" cross reference to "8.8.8."

12.12 MST configuration entities

12.12.1 The MSTI List

12.12.1.2 Create MSTI

12.12.1.2.3 Outputs

Change 12.12.1.2.3 as follows:

a) Operation status. This takes one of the following values:
   1) Operation rejected due to the number of MSTIs currently supported by the Bridge being equal to the maximum number of MSTIs that the Bridge is able to support.
   2) Operation rejected as the MSTID value supplied in the input parameters is already present in the MSTI List.
   3) Operation rejected as the MSTID value supplied in the input parameters is one of the reserved MSTID values (8.9.3).
   4) ~~3)~~ Operation accepted.

12.12.3 The MST Configuration Table

In 12.12.3, change the "8.8.8.1" cross reference to "8.8.8."

12.12.3.2 Read VIDs assigned to MSTID

Change 12.12.3.2.2 and 12.12.3.2.3 as follows:

12.12.3.2.2 Inputs

a) An MSTID value, in the range 0 through ~~4094~~4095.

12.12.3.2.3 Outputs

a) A MSTID value, in the range 0 through ~~4094~~4095;
b) A 4096-bit vector in which bit N is set TRUE if VID N is assigned to the given MSTID and is otherwise set FALSE.

12.13 Provider Bridge management

Change the last paragraph in 12.13 as follows:

The following managed objects define the semantics of the management operations specific to Provider Bridges:

c) The Provider Bridge Port Type managed object (12.13.1);
~~d)~~ ~~The Network Port Configuration managed object (12.13.2);~~
d) ~~e)~~ The Customer Edge Port Configuration managed object (12.13.2);
e) ~~f)~~ The Remote Customer Access Port Configuration managed object (12.3.3).

Delete the 12.13.2 ("Network Port Configuration managed object") to 12.13.2.2.3 ("Outputs") in their entirety, and renumber the subsequent subclauses in 12.13 accordingly.

Insert the following subclauses, 12.25 to 12.25.12.1.3 (including Figure 12-3), after 12.24.2.3.3:

12.25 Shortest Path Bridging managed objects

The Shortest Path Bridging managed objects model operations that create, modify, delete, or enquire about the configuration and the operation of an SPT Bridge and the region in which it is operating. These managed objects model only behaviors unique to Shortest Path Bridging. Figure 12-3 illustrates the hierarchical relationships among the various SPB managed objects.

IS-IS provides a logical topology instance and within each instance the SPT Domain defines a set of managed objects. The following are the managed objects in an SPT Bridge:

a) The SPB System managed object (12.25.1).
b) The SPB MTID (Clause 3) Static managed object (12.25.2).
c) The SPB Topology Instance Dynamic managed object (12.25.3).
d) The SPB ECT Static Entry managed object (12.25.4).
e) The SPB ECT Dynamic Entry managed object (12.25.5).
f) The SPB Adjacency Static Entry managed object (12.25.6).
g) The SPB Adjacency Dynamic Entry managed object (12.25.7).
h) The SPB Topology Node Table managed object (12.25.8).
i) The SPB Topology ECT Table managed object (12.25.9).
j) The SPB Topology Edge Table managed object (12.25.10).
k) The SPBM Topology Service Table managed object (12.25.11).
l) The SPBV Topology Service Table managed object (12.25.12).

12.25.1 The SPB System managed object

There is one SPB System managed object per ISIS-SPB area supported by an SPT Bridge. A single area only is supported in this version of the specification. It contains the fundamental SPB configuration parameters for the bridge. It is persistent over reboot. The management operations that can be performed on the SPB System managed object are as follows:

a) Create SPB System managed object (12.25.1.1)
b) Write SPB System managed object (12.25.1.2)
c) Read SPB System managed object (12.25.1.3)
d) Delete SPB System managed object (12.25.1.4)

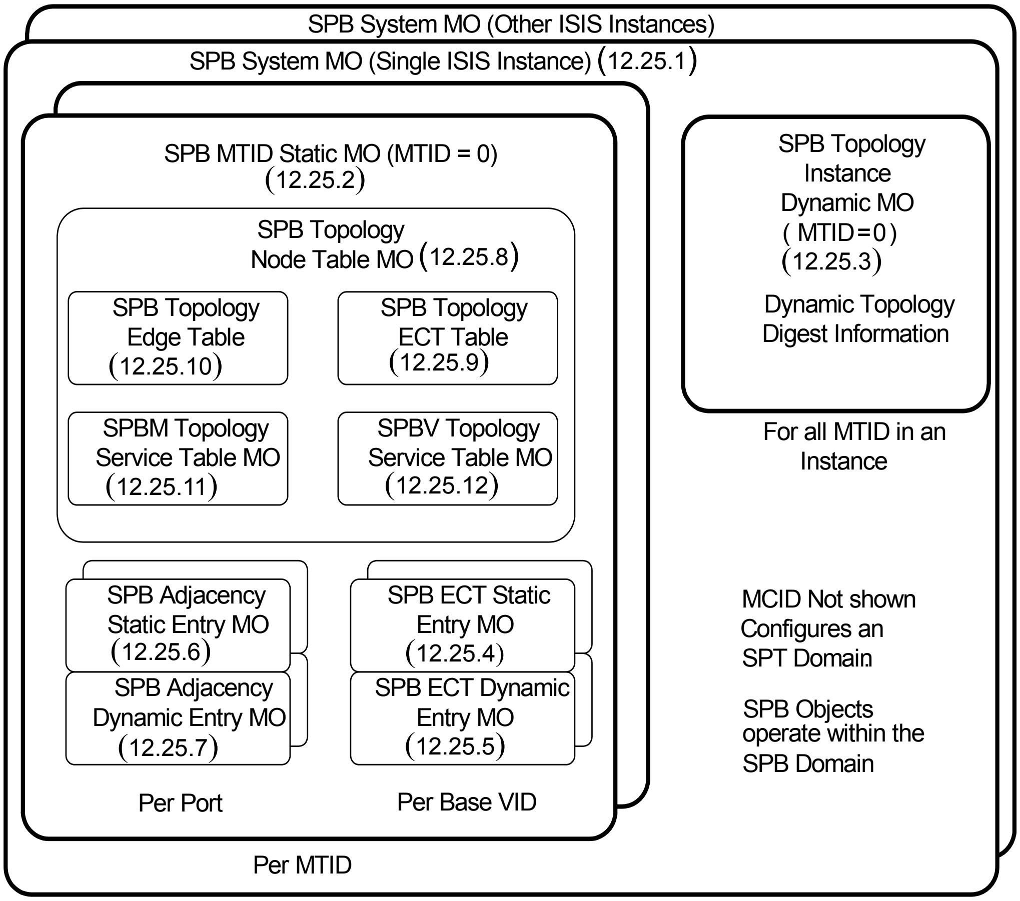


Figure 12-3—SPB Management objects

12.25.1.1 Create SPB System Managed object

12.25.1.1.1 Purpose

To create and configure the SPB System managed object on a bridge.

12.25.1.1.2 Inputs

a) The part of the IS-IS Area Address which is domain specific (see Figure 4, ISO/IEC 10589). This shall be set to zero in this version of this specification whenever SPB is the only protocol being supported by this IS-IS instance.
b) The MAC address to be used as the SPB System Identifier by all instances of ISIS-SPB on this bridge. The System ID defaults to the Bridge Address if this input is not provided. The Bridge Address is the value returned by item a) of Discover Bridge (12.4.1.1.3) and Read Bridge (12.4.1.2.3).
c) The Group address used to address this ISIS-SPB instance (see Table 8-14). This address is configured to manage the transparency to ISIS-SPB of neighboring network elements, and also to specify the IS-IS Level at which this SPB instance is operating, which shall be Level 1 in this version of this specification.

12.25.1.1.3 Outputs

a) Operation status. This takes one of the following values:
   1) Operation rejected because the specified IS-IS Area Address is already in use.
   2) Operation rejected because the provided Group address does not correspond to an allowed address (Table 8-14).
   3) Operation rejected because the specified SPB System Identifier value is not valid.
   4) Operation rejected due to the bridge's inability to support the requested IS-IS Level.
   5) Operation accepted.

12.25.1.2 Write SPB System managed object

12.25.1.2.1 Purpose

To configure an SPB System managed object on a bridge.

12.25.1.2.2 Inputs

a) The part of the IS-IS Area Address that is domain specific (see Figure 4, ISO/IEC 10589). This shall be zero in this version of this specification.
b) The SPB System Shortest Path Source Identifier (SPSourceID) (27.10).
c) If SPBV mode is used, whether the SPVIDs associated with this bridge are assigned, or are to be auto-allocated. The use of SPBV mode is inferred from the assignment of at least one VID to the reserved MSTID value 0xFFD (27.4) in the FID to MSTID Allocation Managed Object (12.12.2.2).
d) If SPBM mode is used, whether the SPsourceID associated with this bridge is assigned, or is to be auto-allocated. The use of SPBM mode is inferred from the assignment of at least one VID to the reserved MSTID value 0xFFC (see 27.4) in the FID to MSTID Allocation Managed Object (12.12.2.2).
e) The agreement digest Convention identifier (28.4.3), specifying the agreement rules being used. It can take the values 1 to 3 as specified in 28.4.3.

12.25.1.2.3 Outputs

a) Operation status. This takes one of the following values:
   1) Operation rejected due to the bridge's inability to support the requested SPB capability.
   2) Operation accepted.

12.25.1.3 Read SPB System managed object

12.25.1.3.1 Purpose

To obtain information about SPB System managed object on a bridge.

12.25.1.3.2 Inputs

a) The part of the IS-IS Area Address that is domain specific (see Figure 4, ISO/IEC 10589).

12.25.1.3.3 Outputs

a) The part of the IS-IS Area Address that is domain specific (see Figure 4, ISO/IEC 10589).
b) The SPB System Identifier used by all instances of ISIS-SPB on this bridge.
c) The Group address used to address this ISIS-SPB instance (Table 8-14). This address also identifies the IS-IS Level at which this SPB instance is operating.

d) The SPB System Name used by all instances of ISIS-SPB. This takes the value configured by Set Bridge Name (12.4.1.3) and which can also be read as item b) of Read Bridge (12.4.1.2).
e) The Bridge Priority (13.26.3). The value used by SPB is set in the CIST Bridge Protocol Parameters (12.8.1.3).
f) The SPB System SPSourceID (27.10).
g) The SPBV Mode, identifying whether, if SPBV mode is used, the SPVIDs associated with this bridge are assigned, or are to be auto-allocated.
h) The SPBM Mode, identifying whether, if SPBM mode is used, the SPsourceID associated with this bridge is assigned, or is to be auto-allocated.
i) The agreement digest Convention identifier (28.4.3), specifying the agreement rules being used.

12.25.1.4 Delete SPB System managed object

12.25.1.4.1 Purpose

To delete the SPB System managed object on a bridge, so disabling it as an SPT bridge.

12.25.1.4.2 Inputs

a) The part of the IS-IS Area Address that is domain specific (see Figure 4, ISO/IEC 10589).

12.25.1.5 Outputs

a) Operation status. This takes one of the following values:
   1) Operation rejected because one or more SPB MTID Static managed objects still exist and have not been deleted for this SPB instance.
   2) Operation accepted.

12.25.2 The SPB MTID Static managed object

There is one SPB MTID Static managed object per IS-IS Multi-Topology Identifier used for SPB operation. It contains the SPB MTID value and the ISIS-SPB Overload flag for this MTID. It is persistent over reboot. The management operations that can be performed on the SPB MTID Static managed object are as follows:

a) Create SPB MTID Static managed object (12.25.2.1).
b) Write SPB MTID Static managed object (12.25.2.2).
c) Read SPB MTID Static managed object (12.25.2.3).
d) Delete SPB MTID Static managed object (12.25.2.4).

12.25.2.1 Create SPB MTID Static managed object

12.25.2.1.1 Purpose

To create and configure an SPB MTID Static managed object on a bridge.

12.25.2.1.2 Inputs

a) A Multi-Topology Identifier to support IS-IS MT operation.

12.25.2.1.3 Outputs

a) Operation status. This takes one of the following values:
   1) Operation rejected because the Multi-Topology Identifier is already in use.
   2) Operation accepted.

b) A Topology Index value that uniquely identifies this MTID Static managed object.

12.25.2.2 Write SPB MTID Static managed object

12.25.2.2.1 Purpose

To configure an SPB MTID Static managed object.

12.25.2.2.2 Inputs

a) A Multi-Topology Identifier to support IS-IS MT operation.
b) The Overload flag value, which has ISIS “no transit” semantics, applied only in this SPB topology.

12.25.2.2.3 Outputs

a) The Multi-Topology Identifier value.
b) The Overload flag value, which has ISIS “no transit” semantics, but applied only in this SPB topology.

12.25.2.3 Read SPB MTID Static managed object

12.25.2.3.1 Purpose

To obtain information about an SPB MTID Static managed object.

12.25.2.3.2 Inputs

a) A Multi-Topology Identifier to support IS-IS MT operation.

12.25.2.3.3 Outputs

a) The Multi-Topology Identifier value.
b) The Overload flag value, which has ISIS “no transit” semantics, but applied only in this SPB topology.
c) A Topology Index value that uniquely identifies this MTID Static managed object.

12.25.2.4 Delete SPB MTID Static managed object

12.25.2.4.1 Purpose

To delete an SPB MTID Static managed object.

12.25.2.4.2 Inputs

a) A Multi-Topology Identifier to support IS-IS MT operation.

12.25.2.4.3 Outputs

a) Operation status. This takes one of the following values:
    1) Operation rejected because not all peer links in the SPB Adjacency Static Entry managed object for this Multi-Topology Identifier have an Administrative State of Disabled.
    2) Operation accepted.

12.25.3 The SPB Topology Instance Dynamic managed object

There is one SPB Topology Instance Dynamic managed object for each SPB MTID Static managed object. It contains the dynamic parameters computed for this bridge as a consequence of its participation in SPB in the ISIS-SPB topology defined by the specified MTID within this instance of SPB. This object is automatically created as a consequence of the creation of the SPB MTID Static managed object associated with the topology. It is deleted with the deletion of the SPB MTID Static managed object associated with the topology. The management operations that can be performed on the SPB Topology Instance Dynamic managed object are as follows:

a) Read SPB Topology Instance Dynamic managed object.

12.25.3.1 Read SPB Topology Instance Dynamic managed object

12.25.3.1.1 Purpose

To obtain information about the dynamic parameters computed by this bridge as a consequence of its participation in SPB in the ISIS-SPB topology defined by the specified MTID within this instance of SPB.

12.25.3.1.2 Inputs

a) A Topology Index value that uniquely identifies a topology instance.

12.25.3.1.3 Outputs

a) The Topology Index value that uniquely identifies this topology instance.
b) The value of the topology Agreement Digest (13.17.1) being advertised by this bridge to its peers in this ISIS-SPB Topology. This includes all parameters defined in the Agreement Digest (28.4).
c) The value of the MCID (8.9, 13.5, 13.8) being advertised by this bridge to its peers.
d) The value of the Auxiliary MCID (27.4.1, 28.12.2) being advertised by this bridge to its peers.

12.25.4 The SPB ECT Static Entry managed object

There is one SPB ECT Static Entry managed object per Base VID allocated to SPB operation. A Base VID exists in the context of a single ISIS-SPB MTID. The static Entry contains the SPB configuration parameters for the forwarding plane associated with the Base VID. It is persistent over reboot. The management operations that can be performed on the ECT Static Entry managed object are as follows:

a) Create an SPB ECT Static Entry managed object (12.25.4.1).
b) Read an SPB ECT Static Entry managed object (12.25.4.2).
c) Delete an SPB ECT Static Entry managed object (12.25.4.3).

12.25.4.1 Create SPB ECT Static Entry managed object

12.25.4.1.1 Purpose

To create and configure and interrogate an SPB ECT Static Entry managed object on a bridge.

12.25.4.1.2 Inputs

a) A Topology Index value that uniquely identifies a topology instance.
b) A Base VID. This VID must have been assigned either to the reserved MSTID value 0xFFC for SPT bridges in SPBM operation, or to the reserved MSTID value 0xFFD for SPBV (27.4) in the FID to MSTI Allocation table (12.12.2.2).

c) The identifier of the Equal Cost Tree algorithm (ECT-ALGORITHM) to be used for this Base VID. The Default is the LowPATHID 00-80-C2-01 (28.8).
d) The pre-configured value of the SPVID assigned to this bridge if operating in SPBV mode. This input is ignored if auto-allocated is selected [item c) in 12.25.1.2.2]. This VID value must have been assigned to the reserved MSTID value 0xFFF (27.4) in the FID to MSTI Allocation Managed Object (12.12.2.2).

12.25.4.1.3 Outputs

a) Operation status. This takes one of the following values:
   1) Operation rejected because the Topology Index value is not defined.
   2) Operation rejected because the Base VID is already in use.
   3) Operation rejected due to requested Base VID not assigned to specified SPB use.
   4) Operation rejected due to requested SPVID not assigned to the SPBV pool.
   5) Operation accepted.

12.25.4.2 Read SPB ECT Static Entry managed object

12.25.4.2.1 Purpose

To obtain information about the SPB ECT Static Entry managed object for a Base VID.

12.25.4.2.2 Inputs

a) A Topology Index value that uniquely identifies a topology instance.
b) A Base VID. The value 4095 is a wildcard indicating that information, from any Base VID assigned to SPB operation, is requested.

12.25.4.2.3 Outputs

a) Operation status. This takes one of the following values:
   1) Operation rejected because the Base VID is not assigned to SPB use.
   2) Operation accepted.
b) A Topology Index value that uniquely identifies this topology instance.
c) The Base VID.
d) The identifier of the Equal Cost Tree algorithm (ECT-ALGORITHM) to be used for this Base VID.
e) The value of the SPVID to be used by this bridge for this Base VID if operating in SPBV mode.

12.25.4.3 Delete SPB ECT Static Entry managed object

12.25.4.3.1 Purpose

To delete the SPB ECT Static Entry managed object for a Base VID.

12.25.4.3.2 Inputs

a) A Topology Index value that uniquely identifies a topology instance.
b) A Base VID.

12.25.4.3.3 Outputs

a) Operation status. This takes one of the following values:
   1) Operation rejected because the bridge has services currently assigned to this Base VID.
   2) Operation rejected because the SPT Region has services currently using this Base VID.

3) Operation accepted.

12.25.5 The SPB ECT Dynamic Entry managed object

There is one SPB ECT Dynamic Entry managed object per Base VID allocated to SPB operation. It contains the SPB dynamic parameters associated with the Base VID. This object is automatically created as a consequence of the creation of the SPB ECT Static Entry managed object. It is deleted with the deletion of the SPB ECT Static Entry managed object. The management operations that can be performed on the ECT Dynamic Entry managed object are as follows:

a) Read SPB ECT Dynamic Entry managed object.

12.25.5.1 Read SPB ECT Dynamic Entry managed object

12.25.5.1.1 Purpose

To obtain information about the SPB ECT Dynamic Entry managed object for a Base VID.

12.25.5.1.2 Inputs

a) A Topology Index value that uniquely identifies a topology instance.
b) A Base VID. The value 4095 is a wildcard indicating that information, from any Base VID assigned to SPB operation, is requested.

12.25.5.1.3 Outputs

a) The Topology Index value that uniquely identifies this topology instance.
b) The Base VID.
c) A flag identifying whether the operational mode of this Base VID is SPBV or SPBM.
d) A flag indicating that this bridge has services currently assigned to this Base VID.
e) A flag indicating that the SPT Region has services currently using this Base VID.
f) A count of the number of ingress check failures (an error condition) on this Base VID since creation of this SPB ECT Dynamic Entry managed object.

12.25.6 The SPB Adjacency Static Entry managed object

There is one SPB Adjacency Static Entry managed object per port per ISIS-SPB topology. It contains the SPB configuration parameters for the port in a topology. It is persistent over reboot. The management operations that can be performed on the SPB Adjacency Static Entry managed object are as follows:

a) Write SPB Adjacency Static Entry managed object (12.25.6.1).
b) Read SPB Adjacency Static Entry managed object (12.25.6.2).

12.25.6.1 Write SPB Adjacency Static Entry managed object

12.25.6.1.1 Purpose

To configure an SPB Adjacency Static Entry managed object on a port for an ISIS-SPB topology.

12.25.6.1.2 Inputs

a) A Topology Index value that uniquely identifies a topology instance.
b) The system interface index of a port.

c) The value of the ieee802.1 SPB Link State metric for the link on this port. For MTID = 0, or when a single MTID only is in use, this value should be set equal to the Path Cost set in the CIST Port Parameters Managed Object (see 12.8.2.1 and 12.8.2.3).
d) The SPB administrative state of a port.

NOTE—Item c) - The primary function of IS-IS MT is to allow different link metrics to be associated with each topology, so that different routes are preferentially favoured in different topologies. However, care must be taken when configuring IS-IS MT that CIST path costs are assigned to one topology, and that this topology is the one used to compute the CIST, in order that SPB calculates its part of the CIST consistently.

12.25.6.1.3 Outputs

a) The Topology Index value that uniquely identifies this topology instance.
b) The system interface index of the port.
c) The value of the ieee802.1 SPB Link State metric for the link on this port.
d) The SPB administrative state of this port.

12.25.6.2 Read SPB Adjacency Static Entry managed object

12.25.6.2.1 Purpose

To read back the parameters of an SPB Adjacency Static Entry managed object on a port for an ISIS-SPB topology.

12.25.6.2.2 Inputs

a) A Topology Index value that uniquely identifies a topology instance.
b) The system interface index of a port. The value of zero is a wildcard indicating information, for any interface on which SPB operation is enabled, is requested.

12.25.6.2.3 Outputs

a) The Topology Index value that uniquely identifies this topology instance.
b) The system interface index of the port.
c) The value of the ieee802.1 SPB Link State metric for the link on this port.
d) The SPB administrative state of this port.

12.25.7 The SPB Adjacency Dynamic Entry managed object

There is one SPB Adjacency Dynamic Entry managed object per port per ISIS-SPB topology. It contains the dynamically discovered SPB parameters for the port in that topology. The management operations that can be performed on the SPB Adjacency Dynamic Entry managed object are as follows:

a) Read SPB Adjacency Dynamic Entry managed object in a topology.

12.25.7.1 Read SPB Adjacency Dynamic Entry managed object

12.25.7.1.1 Purpose

To obtain information about an SPB Adjacency Dynamic Entry managed object on a port.

12.25.7.1.2 Inputs

a) A Topology Index value that uniquely identifies a topology instance.

b) The system interface index of a port. The value of zero is a wildcard indicating information, for any interface on which SPB operation is enabled, is requested.
c) The SPB System Identifier of a peer attached to the port.

12.25.7.1.3 Outputs

a) The Topology Index value that uniquely identifies this topology instance.
b) The system interface index of the port.
c) The SPB System Identifier of the peer attached to the port.
d) The Bridge Port Number (same as item d)1), 12.4.1.2.3).
e) The SPB operational state of this port.
f) The IS-IS System name of the peer attached to the port.
g) The value of the topology Agreement Digest (13.17.1) being advertised by the peer. This includes all parameters defined in the Agreement Digest (28.4).
h) The value of the MCID (8.9, 13.5, 13.8) being advertised by the peer.
i) The value of the Auxiliary MCID (27.4.1, 28.12.2) being advertised by the peer.
j) The value of the Local Circuit ID of this node, used by ISIS to select a single adjacency if more than one link connects a pair of SPT Bridges.
k) The value of the Local Circuit ID of the peer node, used by ISIS to select a single adjacency if more than one link connects a pair of SPT Bridges.
l) The value of the Port Identifier on this node for the link selected by ISIS to form the adjacency, equal to the value read as item c) from the CIST Port Parameters Managed Object (see 12.8.2.1) on this node.
m) The value of the Port Identifier on the peer node for the link selected by ISIS to form the adjacency with this node.
n) An index pointing into the isisCircTable in the IS-IS MIB for cross-referencing.

12.25.8 The SPB Topology Node Table managed object

There is one SPB Topology Node Table managed object per bridge per ISIS-SPB topology. It is populated by ISIS-SPB with the nodal SPB parameters discovered for all SPT Bridges participating in that topology within the SPT Region. This object is automatically created as a consequence of the creation of the SPB MTID Static managed object for that topology. It is deleted with the deletion of that SPB MTID Static managed object. The management operations that can be performed on the SPB Topology Node Table managed object are as follows:

a) Read SPB Topology Node Table managed object.

12.25.8.1 Read SPB Topology Node Table managed object

12.25.8.1.1 Purpose

To obtain nodal information about all SPT Bridges in an SPT Region for an ISIS-SPB topology.

12.25.8.1.2 Inputs

a) A Topology Index value that uniquely identifies a topology instance.
b) The SPB System Identifier of ISIS-SPB on a remote bridge. The value zero is a wildcard indicating that information, from all remote bridges in the SPT region, is requested.

12.25.8.1.3 Outputs

A list, which may be empty, comprising one entry for each bridge within the SPT Region of which the local IS-IS instance is aware. Each entry contains the following information about a remote bridge:

a) The Topology Index value that uniquely identifies this topology instance.
b) The SPB System Identifier of ISIS-SPB on the remote bridge.
c) The Bridge Priority [the value set in CIST Protocol Parameters (12.8.1.3) for the remote bridge].
d) The SPSourceID (27.10) of the remote bridge.
e) The SPB System Name of the remote bridge.

If b) identifies a specific remote bridge then the list above will comprise one entry per Base VID, as specified in item c), containing only information about the specific remote bridge.

12.25.9 The SPB Topology ECT Table managed object

There is one SPB Topology ECT Table managed object per bridge per ISIS-SPB topology. It is populated by ISIS-SPB with the nodal ECT parameters discovered for all SPT Bridges participating in that topology within the SPT Region. This object is automatically created as a consequence of the creation of the SPB MTID Static managed object. It is deleted with the deletion of that SPB MTID Static managed object. The management operations that can be performed on the SPB Topology ECT Table managed object are as follows:

a) Read SPB Topology ECT Table managed object.

12.25.9.1 Read SPB Topology ECT Table managed object

12.25.9.1.1 Purpose

To obtain nodal ECT information, including the binding of ECT Algorithms to Base VIDs, about all SPT Bridges in an SPT Region for an ISIS-SPB topology.

12.25.9.1.2 Inputs

a) A Topology Index value that uniquely identifies a topology instance.
b) The SPB System Identifier of ISIS-SPB on a remote bridge. The value zero is a wildcard indicating information, for all System identifiers, is requested.
c) The Base VID. The value 4095 is a wildcard indicating that information, from any Base VID assigned for SPB operation, is requested.

12.25.9.1.3 Outputs

A list, which may be empty, comprising one entry for each bridge within the SPT Region of which the local ISIS-SPB instance is aware. Each entry contains the following information about a remote bridge:

a) The Topology Index value that uniquely identifies this topology instance.
b) The SPB System Identifier of ISIS-SPB on the remote bridge.
c) The Base VID.
d) The identifier of the Equal Cost Tree algorithm being used.
e) The SPB operational mode (SPBV or SPBM).
f) The SPVID allocation mode (manual or automatic).
g) The value of the SPVID when operating in SPBV mode.
h) A flag indicating that this bridge has services currently assigned to this Base VID.

If item b) identifies a specific remote bridge then the list above will comprise one entry per Base VID as specified in item c), containing only information about the specific remote bridge.

12.25.10 The SPB Topology Edge Table managed object

There is one SPB Topology Edge Table managed object per bridge per ISIS-SPB topology. It is populated by ISIS-SPB with the discovered Edge (link) parameters for that topology within the SPT Region, referenced by the System Identifiers of the pair of SPT Bridges forming the end-points. This object is automatically created as a consequence of the creation of the SPB MTID Static managed object for that topology. It is deleted with the deletion of that SPB MTID Static managed object. The management operations that can be performed on the SPB Topology Edge Table managed object are as follows:

a) Read SPB Topology Edge Table managed object.

12.25.10.1 Read SPB Topology Edge Table managed object

12.25.10.1.1 Purpose

To obtain information about all Edges interconnecting SPT Bridges within an SPT Region for an ISIS-SPB topology.

NOTE—The use of the ISIS Local Circuit ID for tie breaking enables a single Edge (adjacency) per node pair and a unique Port Identifier on each node for Spanning Tree compatibility; refer to 28.11 and the Read operation on SPB Adjacency Dynamic Entry managed object.

12.25.10.1.2 Inputs

a) A Topology Index value that uniquely identifies a topology instance.
b) The SPB System Identifier used by the near bridge on an Edge. The value zero is a wildcard indicating information, for all System identifiers, is requested.
c) The SPB System Identifier used by the far bridge on an Edge. The value zero is a wildcard indicating information, for all System identifiers, is requested.

12.25.10.1.3 Outputs

A list, which may be empty, comprising one entry for each Edge within the SPT Region of which the local IS-IS instance is aware. Each entry contains the following information about an Edge:

a) The Topology Index value that uniquely identifies this topology instance.
b) The SPB System Identifier used by the near bridge on the Edge.
c) The SPB System Identifier used by the far bridge on the Edge.
d) The value of the ieee802.1 SPB Link State metric for this Edge, which is being advertised by the near bridge.
e) The value of the ieee802.1 SPB Link State metric for this Edge, which is being advertised by the far bridge.

NOTE—The terms “near” and “far” are used locally solely to distinguish the two ends of the Edge and associated metric values, and do not signify any relationship between the SPT Bridges advertising the Edge and the local bridge supporting the SPB Topology Edge Table managed object. As a consequence of this definition, in a stable topology two Edges are created per physical link, one advertised in an LSP by each bridge.

12.25.11 The SPBM Topology Service Table managed object

There is one SPBM Topology Service Table managed object per bridge per ISIS-SPB topology. It is populated by ISIS-SPB with the mapping of Service instances to SPT bridges operating in SPBM mode participating in that topology discovered within the SPT Region. This object is automatically created as a consequence of the creation of the SPB MTID Static managed object for that topology. It is deleted with the deletion of that SPB MTID Static managed object. The SPBM Topology Service Table managed object only exists to provide Region-wide visibility of I-SID instance locations. Configuration of local I-SID instances is performed using the BEB Management (12.16) and the mapping of VIPs to PIPs (and hence to the B-MAC address associated with an I-SID) by PIP Management (12.16.4).

The management operations that can be performed on the SPBM Topology Service Table managed object are as follows:

a) Read SPBM Topology Service Table managed object.

12.25.11.1 Read SPBM Topology Service Table managed object

12.25.11.1.1 Purpose

To obtain information about the location of all I-SID instances within an SPT Region in an ISIS-SPB topology.

12.25.11.1.2 Inputs

a) A Topology Index value that uniquely identifies a topology instance.
b) The SPB System Identifier of an SPT bridges operating in SPBM mode hosting an I-SID. The value zero is a wildcard indicating information, from all remote bridges in the SPT region hosting any I-SIDs, is requested.
c) An I-SID value. The value 0xFFFFFF is a wildcard indicating that information, from the remote bridge(s) specified in item b) above hosting any I-SID, is requested.
d) The Base VID associated with this I-SID instance. The value 4095 is a wildcard indicating that information, from any Base VID assigned to SPB operation, is requested.

12.25.11.1.3 Outputs

A list, which may be empty, comprising one entry for each I-SID assigned to a Base VID within the SPT Region of which the local IS-IS instance is aware. Each entry contains the following information about the I-SID:

a) The Topology Index value that uniquely identifies this topology instance.
b) The SPB System Identifier of the SPT bridges operating in SPBM mode hosting an I-SID instance.
c) The I-SID value.
d) The Base VID associated with this I-SID (and hence the ECT-ALGORITHM applied to this service).
e) The B-MAC address associated with this I-SID. If multiple I-SID instances are configured on a single bridge but are reachable by different B-MAC addresses, a separate record is returned for each B-MAC address associated with that I-SID.
f) The I-SID instance service characteristic; transmit-only, receive-only, or both.

NOTE—In general, an I-SID will be assigned to a single Base VID. During the administratively mediated migration of traffic from one ECT-ALGORITHM to a new one, an I-SID may be assigned to two Base VIDs for the duration of the transition in order to achieve loss-less migration (28.9.2).

12.25.12 The SPBV Topology Service Table managed object

There is one SPBV Topology Service Table managed object per bridge per ISIS-SPB topology. It is populated by ISIS-SPB with the mapping of Group MAC addresses to SPT bridges operating in SPBV mode participating in that topology discovered within the SPT Region. This object is automatically created as a consequence of the creation of the SPB MTID Static managed object. It is deleted with the deletion of that SPB MTID Static managed object. The SPBV Topology Service Table managed object only exists to provide Region-wide visibility of Group MAC address instance locations. Local configuration of Group MAC address instances is performed using the procedures for management of Static Filtering Entries (12.7.2), or MMRP procedures (10.9).

The management operations that can be performed on the SPBV Topology Service Table managed object are as follows:

a) Read SPBV Topology Service Table managed object.

12.25.12.1 Read SPBV Topology Service Table managed object

12.25.12.1.1 Purpose

To obtain information about all Group MAC address instances within an SPT Region in an ISIS-SPB topology.

12.25.12.1.2 Inputs

a) A Topology Index value that uniquely identifies a topology instance.
b) The SPB System Identifier of an SPT bridges operating in SPBV mode hosting a Group MAC address. The value zero is a wildcard indicating information, from all remote bridges in the SPT region hosting Group MAC addresses, is requested.
c) The Group MAC address value. The value zero is a wildcard indicating information, about all Group MACs for the specific bridge(s) with the System Identifier as specified by item b), is requested.

12.25.12.1.3 Outputs

A list, which may be empty, and comprising one entry for each Group MAC address instance within the SPT Region of which the local IS-IS instance is aware. Each entry contains the following information about the Group MAC address:

a) The Topology Index value that uniquely identifies this topology instance.
b) The SPB System Identifier of the SPT bridges operating in SPBV mode hosting a Group MAC address instance.
c) The Group MAC address value.
d) The Base VID (and hence ECT-ALGORITHM applied to this service).
e) The Group MAC address instance service characteristic; transmit-only, receive-only, or both.

If item b) identifies a specific remote bridge hosting an Group Address then the list above will comprise only information about the specific remote bridge.

13. Spanning Tree Protocols

Change Clause 13 as follows:

The spanning tree algorithms and protocols specified by this standard provide simple and full connectivity throughout a Bridged Local Area Network comprising arbitrarily interconnected bridges. Each bridge can use the Rapid Spanning Tree Protocol (RSTP), ~~or the~~ Multiple Spanning Tree Protocol (MSTP), or SPB (Shortest Path Bridging) protocols.

NOTE 1—The spanning tree protocols specified by this standard supersede the Spanning Tree Protocol (STP) specified in IEEE Std 802.1D revisions prior to 2004, but facilitate migration by interoperating with the latter without configuration restrictions beyond those previously imposed by STP. However networks that include bridges using STP can reconfigure slowly and constrain active topologies.

NOTE 2—Although the active topologies determined by the spanning tree protocols connect all the components of a Bridged Local Area Network, filtering (MVRP, etc.) can restrict frames to a subset of each active topology.

RSTP assigns all frames to a Common Spanning Tree (CST), without being aware of the active topology assignments made by MSTP or SPB that allow frames to follow separate paths within Multiple Spanning Tree (MST) or Shortest Path Tree (SPT) Regions. Each of these regions comprises MST or SPT Bridges that consistently assign any given frame to the same active topology (see 8.4) and the LANs that interconnect those bridges. These regions and other bridges and LANs are connected into the CST, to provide loop-free network wide connectivity even if active topology assignments or spanning tree protocols differ locally.

MSTP and SPB connect~~s~~ all bridges and LANs with a single Common and Internal Spanning Tree (CIST) that supports the automatic determination of each region, choosing its maximum possible extent. The connectivity calculated for the CIST provides the CST for interconnecting the regions, and an Internal Spanning Tree (IST) within each region. MSTP calculates a number of independent Multiple Spanning Tree Instances (MSTIs) within each region, and ensures that frames with a given VID are assigned to one and only one of the MSTIs or the IST within the region (or reserved for use by SPB or PBB-TE), that the assignment is consistent among all bridges within the region, and that the stable connectivity of each MSTI and the IST at the boundary of the region matches that of the CST. SPB calculates symmetric sets of Shortest Path Trees (SPTs), each rooted at a bridge within a region, and ensures that frames for any given VLAN are assigned to the same symmetric SPT set within the region (or to the IST, an MSTI, or PBB-TE). Within an SPT Region, all bridges that receive a given shortest path bridged frame assign that frame to the same SPT. The assignment process is specified in Clause 27, and can be done in one of two ways: by using additional VIDs (SPBV, Clause 3), or by using the frame's MAC addresses (SPBM, Clause 3). SPT Bridges use the VID of each received frame to decide whether to use the IST, MSTI, PBB-TE, SPBV, or SPBM. ISIS-SPB can dynamically allocate the additional VIDs (SPVIDs) used by SPBV mode, and ensure that their allocation is consistent within a region.

Spanning tree protocol entities transmit and receive BPDUs (Clause 14) to convey parameters used by RSTP and MSTP to calculate CST, CIST, and MSTI spanning trees. BPDUs also convey parameters that all the spanning tree protocols use to interoperate with each other, that determine the extent of MST and SPT Regions, and that ensure that temporary loops are not created when neighboring bridges are acting on different topology information. SPB uses ISIS-SPB (see Clause 28) to share information used to calculate the IST and SPTs, and to perform that calculation. SPT BPDU information that ensures loop-free connectivity even if neighboring bridges' views of the topology differ (13.17) is also carried in SPB Hello PDUs (28.12.2).

This clause

a) Specifies protocol design and support requirements (13.1, 13.2) and design goals (13.3).
b) Provides an overview of RSTP (13.4), ~~and~~ MSTP (13.5), SPBV and SPBM (13.6) operation.
c) Describes how the spanning tree protocols interoperate and coexist (13.7).
d) Specifies how spanning tree priority vectors (13.9) are calculated (13.10, 13.11) and used to assign the Port Roles (13.12) that determine the Port States, i.e., forwarding and learning (8.4), for each tree.
e) Shows that RSTP, ~~and the~~ MSTP, and ISIS-SPB provide stable connectivity (13.13).
f) Describes how spanning tree priority vectors are communicated (13.14) and changed (13.15).
g) Describes how Port Roles are used to change Port States without introducing loops (13.16).
h) Recommends defaults and ranges for the parameters that determine each tree's topology (13.18).
i) Describes the updating of learned station location information when a tree reconfigures (13.19).
j) Specifies additional controls that can speed reconfiguration or prevent unwanted outcomes (13.20).
k) Describes how loops are prevented when a LAN is only providing one-way connectivity (13.21), and can be prevented when the network includes bridges whose protocol operation can fail (13.23).
l) Describes how a bridge's protocol processing can be 'hot upgraded' in an active network (13.22).
m) Specifies RSTP, ~~and~~ MSTP, and support for SPB using state machines (13.24–13.40).
n) Specifies the use and configuration of the spanning tree protocols for the special cases of a Provider Edge Bridge's Customer Edge Ports (13.41), a Backbone Edge Bridge's Virtual Instance Ports (13.42), and an L2 Gateway Port connecting a customer to a provider (13.20, 13.40).

NOTE 3—Readers of this specification are urged to begin by familiarizing themselves with RSTP.

Clause 14 specifies the format of BPDUs. Clause 27 describes the uses of Shortest Path Bridging. Clause 28 specifies ISIS-SPB. The text of this clause (Clause 13) takes precedence should any conflict be apparent between it and the text in other parts of this standard (in particular, Clause 12, Clause 14, and Annex A). Within this clause (Clause 13) the state machine specifications (13.24–13.40) take precedence over the general description (13.1–13.23). A distinctive font is used to highlight references to state machine variables, procedures, and STATES in the general description.

13.1 Protocol design requirements

The Spanning Tree Algorithm and its associated protocols operate in Bridged Local Area Networks of arbitrary physical topology comprising MST or SST Bridges connecting shared media or point-to-point LANs, so as to support, preserve, and maintain the quality of the MAC Service in all its aspects as specified by Clause 6.

RSTP configures the Port State (8.4) of each Bridge Port. MSTP configures the Port State for the CIST and each MSTI, and verifies the allocation of VIDs to FIDs and FIDs to trees. ISIS-SPB configures the Port State for the CIST and each SPT, configures the VID Translation Table for each Bridge Port, allocates FIDs to SPT sets, and assigns frames to SPTs using SPVIDs or MAC addresses (for SPBV and SPBM respectively, see Clause 27). Operating both independently and together RSTP, ~~and~~ MSTP, and ISIS-SPB meet the following requirements:

a) They configure one or more active topologies that fully connect all physically connected LANs and bridges, and stabilize (with high probability) within a short, known bounded interval after any change in the physical topology, maximising service availability (6.5.1).
b) The active topology for any given frame remains simply connected at all times (6.5.3, 6.5.4), and will (with high probability) continue to provide simple and full connectivity for frames even in the presence of administrative errors (e.g., in the allocation of VIDs to MSTIs).
c) The configured stable active topologies are unicast multicast congruent, downstream congruent, and reverse path congruent (symmetric) (Clause 3, 6.3).

d) The same symmetric active topology is used, in a stable network, for all frames using the same FID, i.e., between any two LANs all such frames are forwarded through the same Bridge Ports (6.3).

e) The active topology for a given VID can be chosen by the network administrator to be a common spanning tree, ~~or~~ one of multiple spanning trees (if MSTP is implemented), or shortest path (if ISIS-SPB is implemented).

f) Each active topology is predictable, reproducible, and manageable, allowing Configuration Management (following traffic analysis) to meet Performance Management goals (6.5 and 6.5.10).

g) The configured network can support VLAN-unaware end stations, such that they are unaware of their attachment to a single LAN or a Bridged Local Area Network, or their use of a VID (6.2).

h) The communications bandwidth on any particular LAN is always a small fraction of the total available bandwidth (6.5.10).

NOTE—The spanning tree protocols cannot protect against temporary loops caused by the interconnection of LANs by devices other than bridges (e.g., LAN repeaters) that operate invisibly with respect to support of the MAC Internal Sublayer Service and the MAC_Operational status parameter (6.6.2).

13.2 Protocol support requirements

In order for the spanning tree protocols to operate, the following are required:

a) A unique Group MAC Address used by the Spanning Tree Protocol Entities (8.10) of participating bridges or bridge components (5.2), and recognized by all the bridges attached to a LAN.

b) An identifier for each bridge or bridge component, unique within the Bridged Local Area Network.

c) An identifier for each Bridge Port, unique within a bridge or bridge component.

Values for each of these parameters shall be provided by each bridge. The unique MAC Address that identifies the Spanning Tree Protocol Entities of MAC Bridges, VLAN Bridges (5.9), and C-VLAN components (5.5) is the Bridge Group Address (Table 8-1). The unique MAC Address that identifies the Spanning Tree Protocol Entities of S-VLAN components is the Provider Bridge Group Address (Table 8-2).

To allow management of active topology (for RSTP, ~~or~~ MSTP, or SPB) means of assigning values to the following are required:

d) The relative Bridge Priority of each bridge in the network.

e) A Port Path Cost for each Bridge Port.

f) The relative Port Priority of each Bridge Port.

13.2.1 MSTP support requirements

MSTP does not require any additional configuration, provided that communication between end stations is supported by a number of VLANs. However, to realize the improved throughput and associated frame loss and transit delay performance improvements made possible by the use of multiple spanning trees, the following are required:

a) Assessment of the probable distribution of traffic between VLANs and between sets of communicating end stations using those VLANs.

b) Per MSTI assignment of Bridge Priority and Internal Port Path Costs to configure the MSTIs.

c) Consistent assignment of VIDs to MSTIDs within each potential MST Region.

d) Administrative agreement on the Configuration Name and Revision Level used to represent the assignments of VIDs to MSTIDs.

13.2.2 SPB support requirements

In order for the ISIS-SPB protocol to operate, the following are required:

a) A Multi-Topology Identifier (MTID) indicating the topology information to be used for route calculation.
b) An IS-IS Area Address for the IS-IS routing area.
c) An IS-IS system ID for the SPT Bridge, normally the Bridge Identifier.
d) A group MAC address for use by IS-IS SPB (Table 8-14).
e) A Base VID for each SPB VLAN.
f) An ECT-ALGORITHM identifying the algorithm for calculating shortest path trees and selecting from among equal cost trees.

SPBV can provide both common spanning tree and shortest path support for VLANs without further configuration. To support default SPB operation this standard specifies a default configuration for SPT Bridges (13.8).

13.3 Protocol design goals

All the spanning tree protocols meet the following goal, which simplifies operational practice:

a) Bridges do not have to be individually configured before being added to the network, other than having their MAC Addresses assigned through normal procedures.
b) In normal operation, the time taken to configure the active topology of a network comprising point-to-point LANs is independent of the timer values of the protocol.

RSTP and MSTP meet the following goal, which limits the complexity of bridges and their configuration:

c) The memory requirements associated with each Bridge Port are independent of the number of bridges and LANs in the network.

It is highly desirable that the operation of ISIS-SPB supports updating of the SPB configuration, so that:

d) SPT Bridges can be added to a region without disrupting communication for existing VLANs.
e) Additional VLANs can be supported by SPB without disrupting communication for the VLANs that are already supported by SPB, or for those VLANs that are supported by RSTP or MSTP.
f) Shortest path bridging support can be enabled or disabled for individual VLANs that are being used to support user communication, with the minimum of frame loss on those VLANs.

13.4 RSTP overview

The Rapid Spanning Tree Protocol (RSTP) configures the Port State (8.4) of each Bridge Port in the Bridge Local Area Network. RSTP ensures that the stable connectivity provided by each bridge between its ports and by the individual LANs attached to those ports is predictable, manageable, full, simple, and symmetric. RSTP further ensures that temporary loops in the active topology do not occur if the network has to reconfigure in response to the failure, removal, or addition of a network component, and that erroneous station location information is removed from the Filtering Database after reconfiguration.

Each of the bridges in the network transmits Configuration Messages (13.14). Each message contains spanning tree priority vector (13.9) information that identifies one bridge as the Root Bridge of the network, and allows each bridge to compute its own lowest path cost to that Root Bridge before transmitting its own Configuration Messages. A Port Role (13.12) of Root Port is assigned to the one port on each bridge that provides that lowest cost path to the Root Bridge, and a Port Role of Designated Port to the one Bridge Port that provides the lowest cost path from the attached LAN to the Root Bridge. Alternate Port and Backup Port roles are assigned to Bridge Ports that can provide connectivity if other network components fail.

State machines associated with the Port Roles maintain and change the Port States that control forwarding (8.6) and learning (8.7) of frames. In a stable network, Root Ports and Designated Ports are Forwarding, while Alternate, Backup, and Disabled Ports are Discarding. Each Port's role can change if a Bridge, Bridge Port, or LAN fails, is added to, or removed from network. Port state transitions to Learning and Forwarding are delayed, and ports can temporarily transition to the Discarding state to prevent loops and to ensure that misordering (6.5.3) and duplication (6.5.4) rates remain negligible.

RSTP provides rapid recovery of connectivity to minimize frame loss (6.5.2). A new Root Port, and Designated Ports attached to point-to-point LANs, can transition to Forwarding without waiting for protocol timers to expire. A Root Port can transition to Forwarding without transmitting or receiving messages from other bridges, while a Designated Port attached to a point-to-point LAN can transition when it receives an explicit agreement transmitted by the other bridge attached to that LAN. The forwarding transition delay used by a Designated Port attached to a shared media LAN is long enough for other bridges attached to that LAN to receive and act on transmitted messages, but is independent of the overall network size. If all the LANs in a network are point-to-point, RSTP timers define worst-case delays that only occur if protocol messages are lost or rate transmission limits are exceeded.

A Bridge Port attached to a LAN that has no other bridges attached to it may be administratively configured as an Edge Port. RSTP monitors the LAN to ensure that no other bridges are connected, and may be configured to automatically detect an Edge Port. Each Edge Port transitions directly to the Forwarding Port State, since there is no possibility of it participating in a loop.

13.4.1 Computation of the active topology

The bridge with the best Bridge Identifier is selected as the Root Bridge. The unique Bridge Identifier for each bridge is derived, in part, from the Bridge Address (8.13.8) and, in part, from a manageable priority component (13.26). The relative priority of bridges is determined by the numerical comparison of the unique identifiers, with the lower numerical value indicating the better identifier.

Every bridge has a Root Path Cost associated with it. For the Root Bridge this is zero. For all other bridges, it is the sum of the Port Path Costs on the least cost path to the Root Bridge. Each port's Path Cost may be managed, 13.18 recommends default values for ports attached to LANs of various speeds.

The Bridge Port on each bridge with the lowest Root Path Cost is assigned the role of Root Port for that bridge (the Root Bridge does not have a Root Port). If a bridge has two or more ports with the same Root Path Cost, then the port with the best Port Identifier is selected as the Root Port. Part of the Port Identifier is fixed and different for each port on a bridge, and part is a manageable priority component (13.26). The relative priority of Bridge Ports is determined by the numerical comparison of the unique identifiers, with the lower numerical value indicating the better identifier.

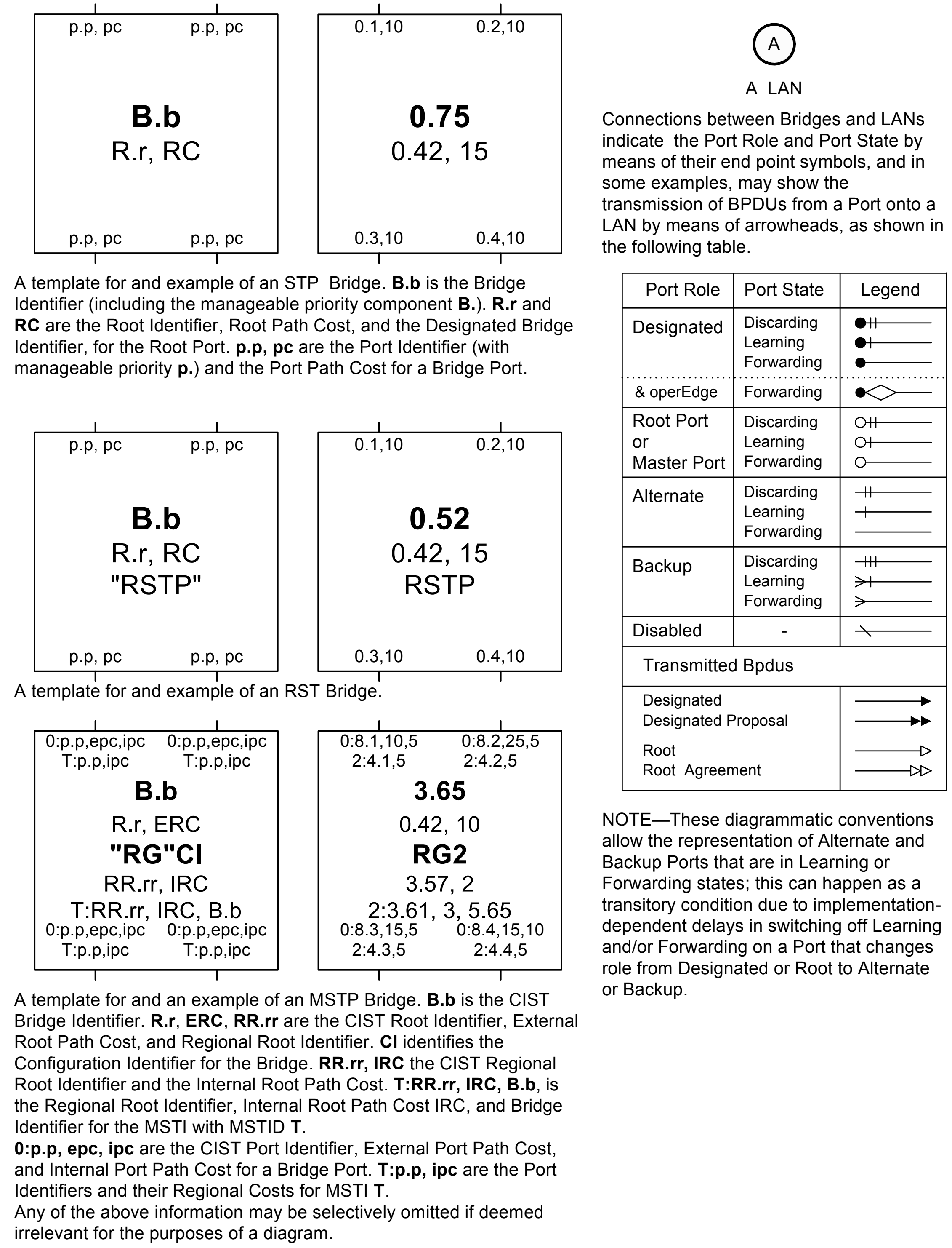
Each LAN in the Bridged Local Area Network also has an associated Root Path Cost. This is the Root Path Cost of the lowest cost bridge with a Bridge Port connected to that LAN. This bridge is selected as the Designated Bridge for that LAN. If there are two or more bridges with the same Root Path Cost, then the bridge with the best priority (least numerical value) is selected as the Designated Bridge. The Bridge Port on the Designated Bridge that is connected to the LAN is assigned the role of Designated Port for that LAN. If the Designated Bridge has two or more ports connected to the LAN, then the Bridge Port with the best priority Port Identifier (least numerical value) is selected as the Designated Port.

In a Bridged Local Area Network whose physical topology is stable, i.e., RSTP has communicated consistent information throughout the network, every LAN has one and only one Designated Port, and every bridge with the exception of the Root Bridge has a single Root Port connected to a LAN. Since each bridge provides connectivity between its Root Port and its Designated Ports, the resulting active topology connects all LANs (is “spanning”) and will be loop-free (is a “tree”).

Any Bridge Port that is enabled, but not a Root or Designated Port, is a Backup Port if that bridge is the Designated Bridge for the attached LAN, and an Alternate Port otherwise. An Alternate Port offers an alternate path in the direction of the Root Bridge to that provided by the bridge’s own Root Port, whereas a Backup Port acts as a backup for the path provided by a Designated Port in the direction of the leaves of the Spanning Tree. Backup Ports exist only where there are two or more connections from a given bridge to a given LAN; hence, they (and the Designated Ports that they back up) can only exist where the bridge has two or more ports attached to a shared media LAN, or directly connected by a point-to-point LAN.

13.4.2 Example topologies

The spanning tree examples in this clause use the conventions of Figure 13-1.



A template for and example of an STP Bridge. **B.b** is the Bridge Identifier (including the manageable priority component **B.**). **R.r** and **RC** are the Root Identifier, Root Path Cost, and the Designated Bridge Identifier, for the Root Port. **p.p, pc** are the Port Identifier (with manageable priority **p.**) and the Port Path Cost for a Bridge Port.

A template for and example of an RST Bridge.

A template for and an example of an MSTP Bridge. **B.b** is the CIST Bridge Identifier. **R.r**, **ERC**, **RR.rr** are the CIST Root Identifier, External Root Path Cost, and Regional Root Identifier. **CI** identifies the Configuration Identifier for the Bridge. **RR.rr, IRC** the CIST Regional Root Identifier and the Internal Root Path Cost. **T:RR.rr, IRC, B.b**, is the Regional Root Identifier, Internal Root Path Cost IRC, and Bridge Identifier for the MSTI with MSTID **T**.
0:p.p, epc, ipc are the CIST Port Identifier, External Port Path Cost, and Internal Port Path Cost for a Bridge Port. **T:p.p, ipc** are the Port Identifiers and their Regional Costs for MSTI **T**.
Any of the above information may be selectively omitted if deemed irrelevant for the purposes of a diagram.

Connections between Bridges and LANs indicate the Port Role and Port State by means of their end point symbols, and in some examples, may show the transmission of BPDUs from a Port onto a LAN by means of arrowheads, as shown in the following table.

Port Role	Port State	Legend
Designated	Discarding Learning Forwarding	●╫——— ●┼——— ●————
& operEdge	Forwarding	●◇———
Root Port or Master Port	Discarding Learning Forwarding	○╫——— ○┼——— ○————
Alternate	Discarding Learning Forwarding	—╫——— —┼——— —————
Backup	Discarding Learning Forwarding	—╫┼—— >┼——— >————
Disabled	-	—⟋———
Transmitted Bpdus		
Designated Designated Proposal Root Root Agreement		———▶ ———▶▶ ———▷ ———▷▷

NOTE—These diagrammatic conventions allow the representation of Alternate and Backup Ports that are in Learning or Forwarding states; this can happen as a transitory condition due to implementation-dependent delays in switching off Learning and/or Forwarding on a Port that changes role from Designated or Root to Alternate or Backup.

Figure 13-1—Diagrammatic conventions for spanning tree topologies

Figure 13-2 shows a simple, redundantly connected, structured wiring configuration, with bridges connected by point-to-point LANs A through N, and a possible spanning tree active topology of the same network. Bridge 111 has been selected as the Root (though one cannot tell simply by looking at the active topology which bridge is the Root).

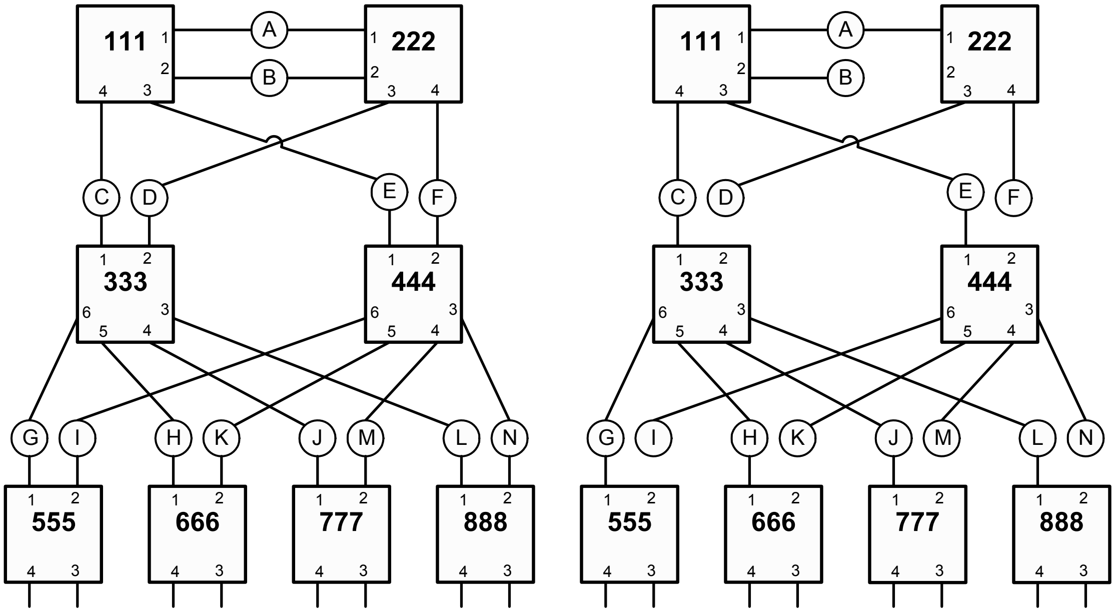


Figure 13-2—Physical topology and active topology

Figure 13-3 shows the Port Roles and Port States of each Bridge Port. It can be seen that bridge 111 is the Root, as its Ports are all Designated Ports, each of the remaining bridges have one Root Port.

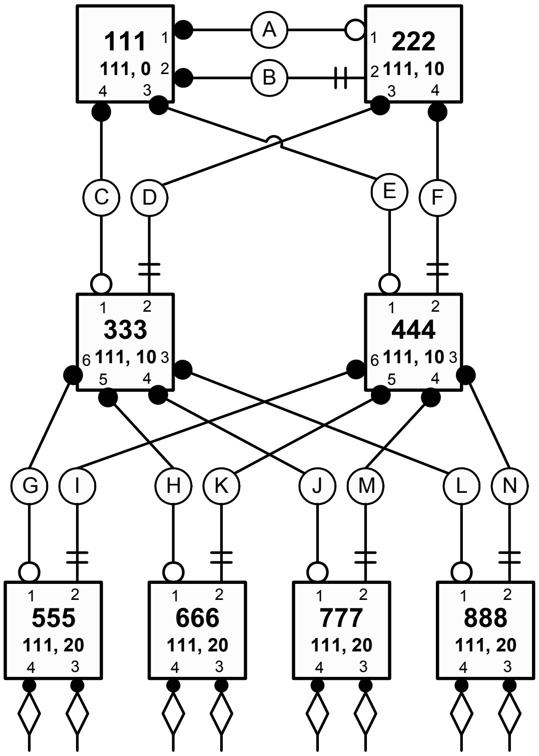


Figure 13-3—Port Roles and Port States

Figure 13-4 shows the result of connecting two of the ports of bridge 888 to the same LAN. As port 4 of bridge 888 has worse priority than port 3 and both offer the same Root Path Cost, port 4 will be assigned the Backup Port Role and will therefore be in the Discarding Port State. Should port 3 or its connection to LAN O fail, port 4 will be assigned the Designated Port Role and will transition to the Forwarding Port State.

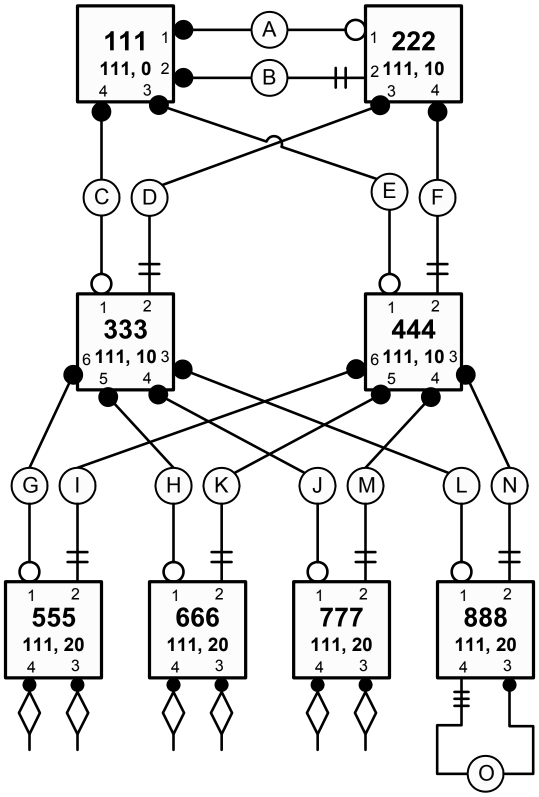


Figure 13-4—A Backup Port

Figure 13-5 shows a “ring” topology constructed from point-to-point links, as in some resilient backbone configurations. Bridge 111 is the Root, as in previous examples.

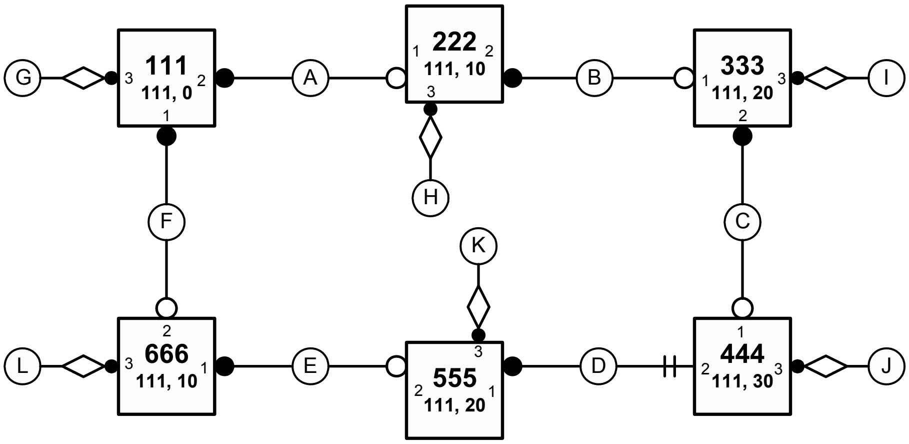


Figure 13-5—“Ring Backbone” example

13.5 MSTP overview

The Multiple Spanning Tree Protocol specifies:

a) An MST Configuration Identifier (13.8) that allows each bridge to advertise its assignment, to a specified MSTI or to the IST, of frames with any given VID.

b) A priority vector (13.9) that comprises bridge identifier and path cost information for constructing a deterministic and manageable single spanning tree active topology, the CIST, that:
   1) Fully and simply connects all bridges and LANs in a Bridged Local Area Network.
   2) Permits the construction and identification of MST Regions of bridges and LANs that are guaranteed fully connected by the bridges and LANs within each region.
   3) Ensures that paths within each MST Region are always preferred to paths outside the region.

c) An MSTI priority vector (13.9), comprising information for constructing a deterministic and independently manageable active topology for any given MSTI within each region.

d) Comparisons and calculations performed by each bridge in support of the distributed spanning tree algorithm (13.10). These select a CIST priority vector for each Bridge Port, based on the priority vectors and MST Configuration Identifiers received from other bridges and on an incremental Path Cost associated with each reception port. The resulting priority vectors are such that in a stable network:
   1) One bridge is selected to be the CIST Root of the Bridged Local Area Network as a whole.
   2) A minimum cost path to the CIST Root is selected for each bridge and LAN, thus preventing loops while ensuring full connectivity.
   3) The one bridge in each MST Region whose minimum cost path to the Root is not through another bridge using the same MST Configuration Identifier is identified as its region's CIST Regional Root.
   4) Conversely, each bridge whose minimum cost path to the Root is through a bridge using the same MST Configuration Identifier is identified as being in the same region as that bridge.

e) Priority vector comparisons and calculations performed by each bridge for each MSTI (13.11). In a stable network:
   1) One bridge is independently selected for each MSTI to be the MSTI Regional Root.
   2) A minimum cost path to the MSTI Regional Root that lies wholly within the region is selected for each bridge and LAN.

f) CIST Port Roles (13.12) that identify the role in the CIST active topology played by each port on a bridge:
   1) The Root Port provides the minimum cost path from the bridge to the CIST Root (if the bridge is not the CIST Root) through the Regional Root (if the bridge is not a Regional Root).
   2) A Designated Port provides the least cost path from the attached LAN through the bridge to the CIST Root.
   3) Alternate or Backup Ports provide connectivity if other bridges, Bridge Ports, or LANs fail or are removed.

g) MSTI and SPT Port Roles (13.12) that identify the role played by each port on a bridge for each MSTI's or SPT's active topology within and at the boundaries of a region.
   1) The Root Port provides the minimum cost path from the bridge to the Regional Root (if the bridge is not the Regional Root for the tree).
   2) A Designated Port provides the least cost path from the attached LAN though the bridge to the Regional Root.
   3) A Master Port provides connectivity from the region to a CIST Root that lies outside the region. The Bridge Port that is the CIST Root Port for the CIST Regional Root is the Master Port for all MSTIs and SPTs.
   4) Alternate or Backup Ports provide connectivity if other bridges, Bridge Ports, or LANs fail or are removed.

h) State machines and state variables associated with each spanning tree (CIST, ~~or~~ MSTI, or SPT), port, and port role, to select and change the Port State (8.4, 13.24) that controls the processing and forwarding of frames assigned to that tree by a MAC Relay Entity (8.3).

13.5.1 Example topologies

Figure 13-6 is an example Bridged Local Area Network, using the conventions of Figure 13-1, and chosen to illustrate MSTP calculations rather than as an example of a common or desirable physical topology.

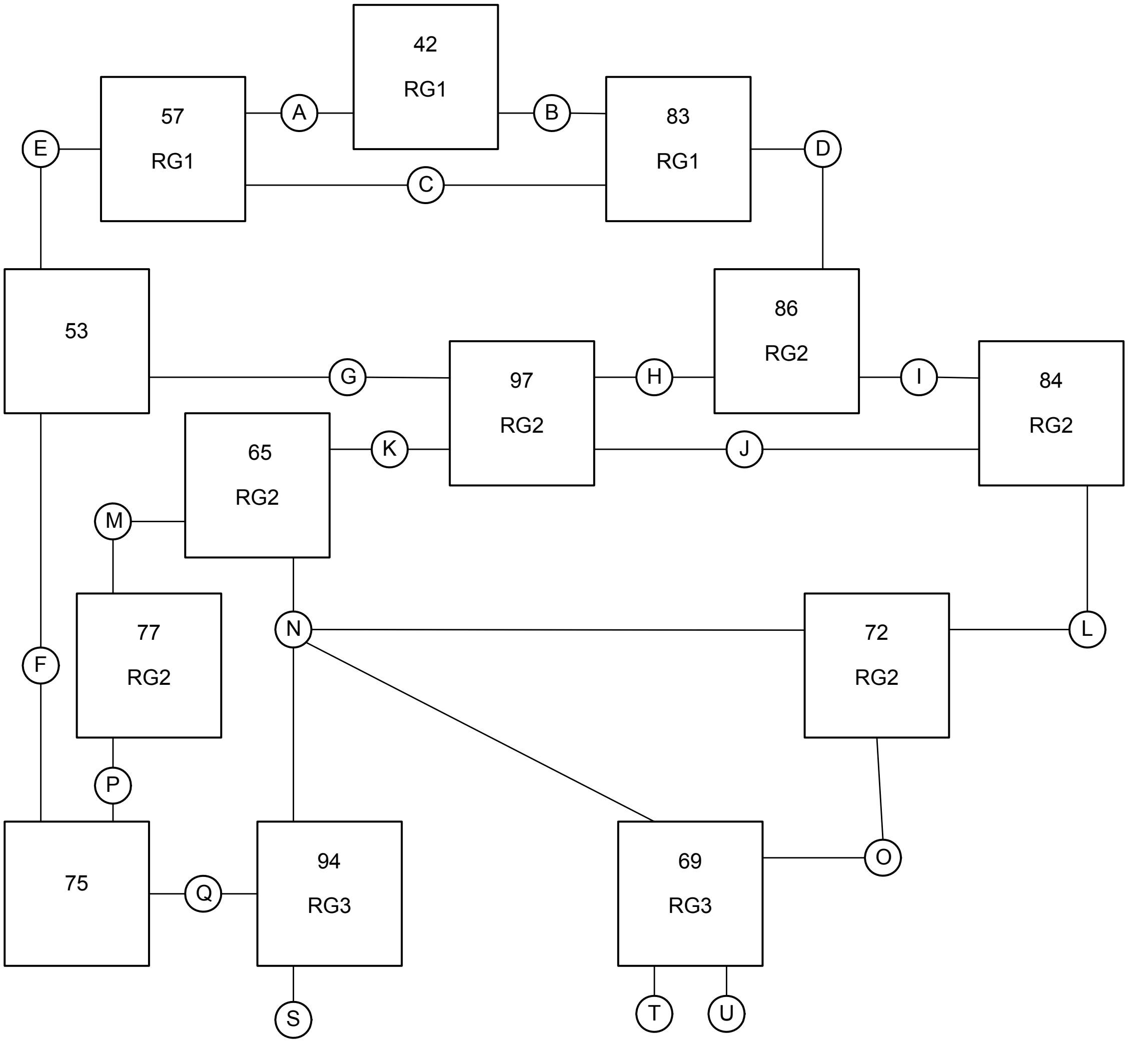


Figure 13-6—An MST Bridge network

Figure 13-7 is the same network showing bridges and LANs with better CIST spanning tree priorities higher on the page, and including CIST priority vectors, port roles, and MST Regions. In this example:

a) Bridge 0.42 is the CIST Root because it has the best (numerically lowest) Bridge Identifier.
b) Bridges 0.57 and 2.83 are in the same MST Region (1) as 0.42, because they have the same MST Configuration Identifier as the latter. Because they are in the same MST Region as the CIST Root, their External Root Path Cost is 0, and their CIST Regional Root is the CIST Root.
c) LANs A, B, C, and D are in Region 1 because their CIST Designated Bridge is a Region 1 MST Bridge, and no STP bridges are attached to those LANs. LAN E is not in an MST Region (or in its own region—an equivalent view) because it is attached to bridge 0.53, which is not an MST Bridge.
d) Bridges 0.77, 0.65, 0.97, 0.86, 3.84, and 3.72 are in the same MST Region (2) since they have the same MST Configuration Identifier and are interconnected by LANs for which one of them is the CIST Designated Bridge.

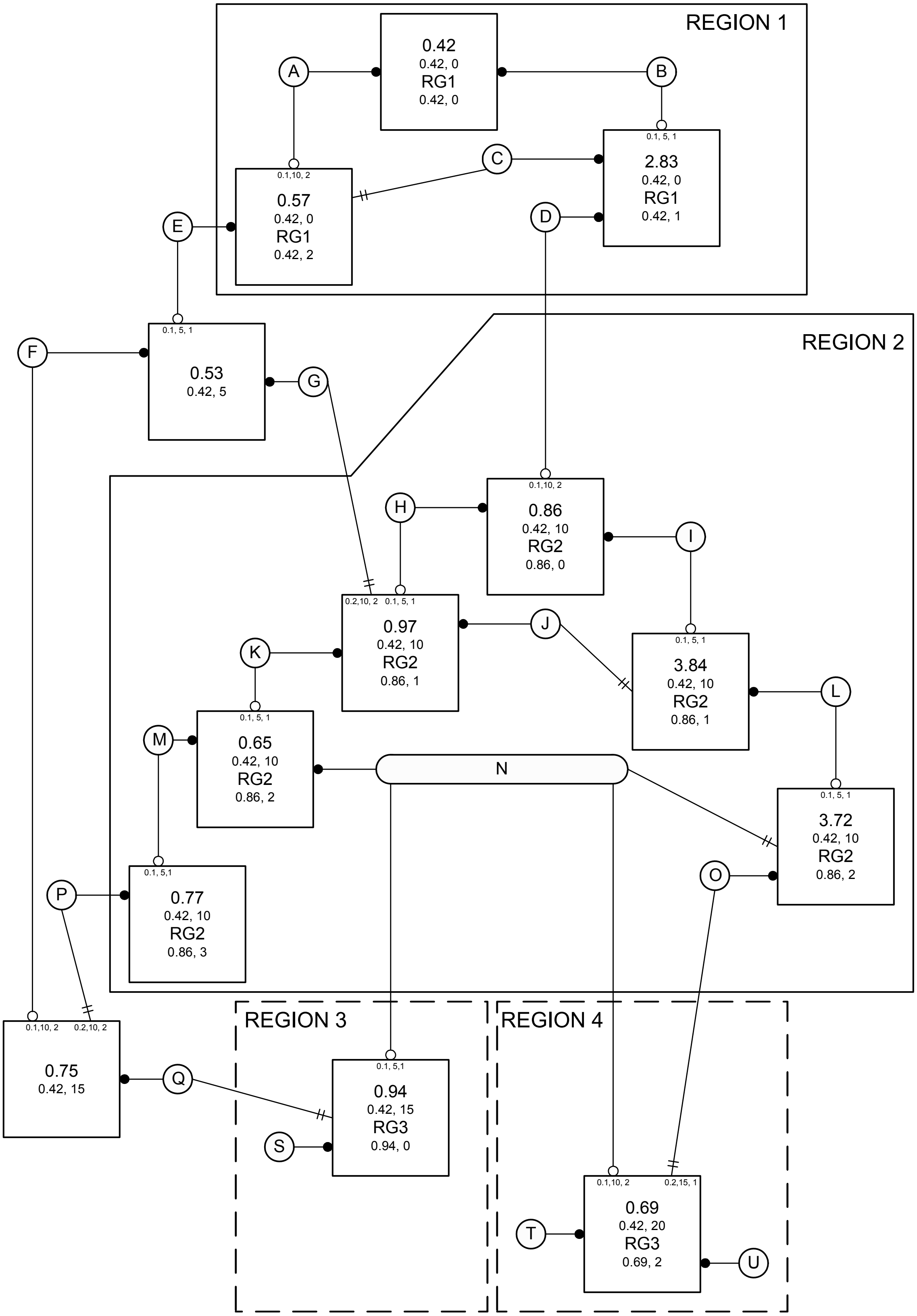


Figure 13-7—CIST Priority Vectors, Port Roles, and MST Regions

e) Bridge 0.86 is the CIST Regional Root for Region 2 because it is has the lowest External Root Path Cost through a Boundary Port.

f) LAN N is in Region 2 because its CIST Designated Bridge is in Region 2. Frames assigned to different MSTIDs may reach N from bridge 0.86 (for example) by either bridge 0.65 or bridge 3.72, even though bridges 0.94 and 0.69 with MST Configuration Identifiers that differ from those for bridges in Region 2 are attached to this shared LAN.

g) Bridges 0.94 and 0.69 are in different regions, even though they have the same MST Configuration Identifier, because the LAN that connects them (N) is in a different region.

Figure 13-8 shows a possible active topology of MSTI 2 within Region 2.

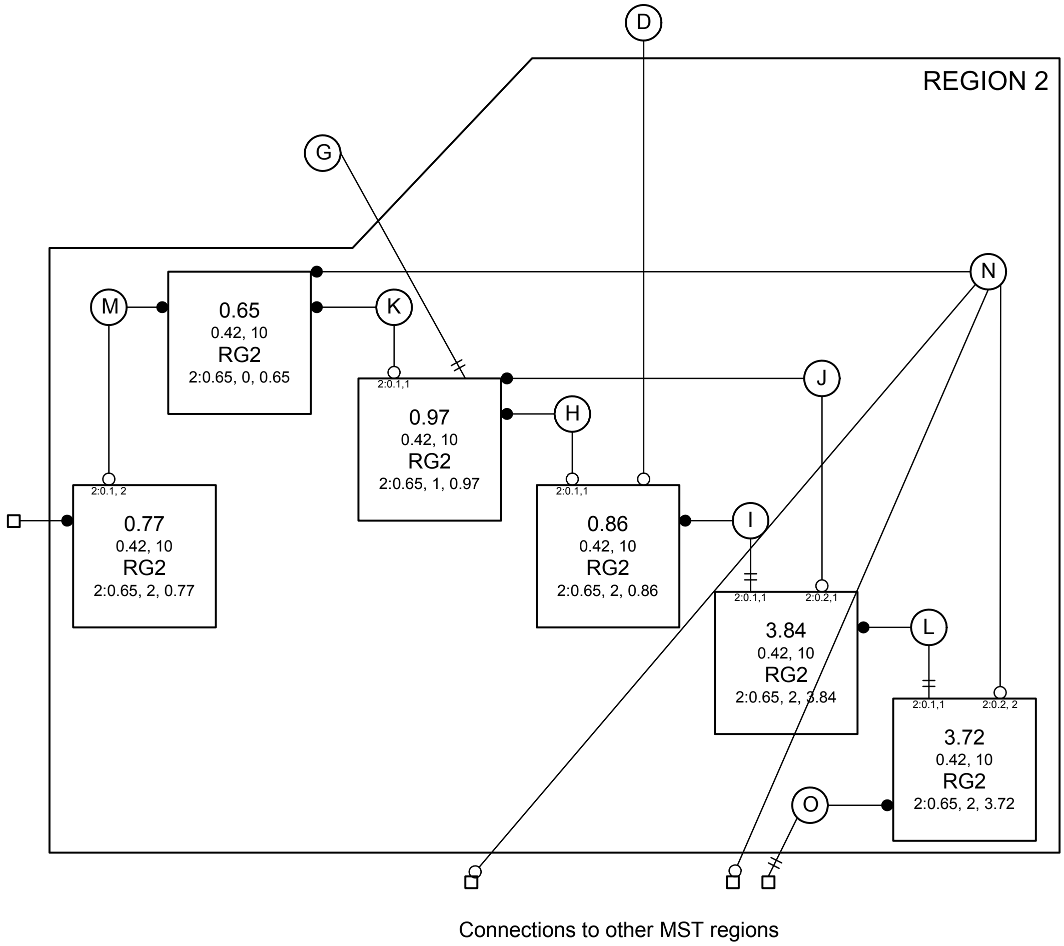


Figure 13-8—MSTI Active Topology in Region 2

h) Bridge 0.65 has been chosen as the MSTI Regional Root because it has the best (numerically the lowest) Bridge Identifier of all bridges in the region for this MSTI.

i) The connectivity between the whole of Region 2 and Region 1 is provided through a single Bridge Port, the Master Port on bridge 0.86. This port was selected for this role because it is the CIST Root Port on the CIST Regional Root for the region (see Figure 13-7).

j) The connectivity between the whole of Region 2 and LANs and bridges outside the region for the MSTI is the same as that for the CIST. This connectivity is similar to that which might result by replacing the entire region by a single SST Bridge. The region has a single Root Port (this port is the Master Port for each MSTI) and a number of Designated Ports.

13.5.2 Relationship of MSTP to RSTP

MSTP is based on RSTP, extended so frames for different VLANs can follow different trees within regions.

a) The same fundamental spanning tree algorithm selects the CIST Root Bridge and Port Roles, but extended priority vector components are used within (13.9, 13.10) in each region. As a result each region resembles a single bridge from the point of view of the CST as calculated by RSTP.
b) Each MSTI's Regional Root Bridge and Port Roles are also computed using the same fundamental spanning tree algorithm with modified priority vector components (13.11).
c) Different bridges may be selected as the Regional Root for different MSTIs by modifying the manageable priority component of the Bridge Identifier differently for the MSTIs.
d) MST Configuration Identification is not applicable to RSTP.~~ specific to MSTP.~~
e) The Port Roles used by the CIST (Root, Designated, Alternate, Backup or Disabled Port) are the same as those of RSTP. The MSTIs use the additional port role Master Port. The Port States associated with each spanning tree and port are the same as those of RSTP.
f) The state variables for each Bridge Port for each tree and for the bridge itself are those specified for RSTP as per port and per bridge with a few exceptions, additions, and enhancements.
g) The performance parameters specified for RSTP apply to the CIST, with a few exceptions, additions, and enhancements. A simplified set of performance parameters apply to the MSTIs.
h) This standard specifies RSTP state machines and procedures as a subset of MSTP.

13.5.3 Modeling an MST or SPT Region as a single bridge

The nominal replacement of an entire region by a single RST~~P~~ Bridge leads to little impact on the remainder of the Bridged Local Area Network. This design is intended to assist those familiar with RSTP to comprehend and verify MSTP, and to administer networks using MSTP. Treating the MST Regions as single bridges provides the network administrator with a natural hierarchy. The internal management of MST Regions can be largely separated from the management of the active topology of the network as a whole.

The portion of the active topology of the network that connects any two bridges in the same MST Region traverses only MST Bridges and LANs in that region and never bridges of any kind outside the region; in other words, connectivity within the region is independent of external connectivity. This is because the protocol parameters that determine the active topology of the network as a whole, the Root Identifier and Root Path Cost (known in the MSTP specification as the CIST Root Identifier and CIST External Root Path Cost) are carried unchanged throughout and across the MST Region, so bridges within the region will always prefer spanning tree information that has been propagated within the region to information that has exited the region and is attempting to reenter it.

NOTE 1—No LAN can be in more than one MST Region at a time, so two bridges (0.11 and 0.22 say) that would otherwise be in the same region by virtue of having the same MST Configuration and of being directly connected by a LAN, may be in distinct regions if that is a shared LAN with other bridges attached (having a different MST Configuration) and no other connectivity between 0.11 and 0.22 and lying wholly within their region is available. The region that the shared LAN belongs to may be dynamically determined. No such dynamic partitioning concerns arise with single bridges. Obviously the sharing of LANs between administrative regions militates against the partitioning of concerns and should only be done following careful analysis.

The Port Path Cost (MSTP's External Port Path Cost) is added to the Root Path Cost just once at the Root Port of the CIST Regional Root, the closest bridge in the region to the Root Bridge of the entire network. The Message Age used by STP and RSTP is also only incremented at this port. If the CIST Root is within a region, it also acts as the Regional Root, and the Root Path Cost and Message Age advertised are zero, just as for a single bridge.

Within an MST Region, each MSTI operates in much the same way as an independent instance of RSTP with dedicated Regional Root Identifier, Internal Root Path Cost, and Internal Port Path Cost parameters.

Moreover, the overall spanning tree (the CIST) includes a fragment (the IST) within each MST Region that can be viewed as operating in the same way as an MSTI with the Regional Root as its root.

NOTE 2—Since an MST Region behaves like a single bridge and does not partition (except in the unusual configuration involving shared LANs noted above), it has a single Root Port in the CST active topology. Partitioning a network into two or more regions can therefore force nonoptimal blocking of Bridge Ports at the boundaries of those regions.

13.6 SPB overview

Clause 27 provides a comprehensive overview of SPT Bridges using SPBV and SPBM mode operation. This clause (13.6) summarizes aspects of SPB operation that relate to the transmission and reception of BPDUs, and their role in providing interoperability with RSTP and MSTP, and in carrying the agreements that ensure that each Shortest Path Tree (SPT) provides loop-free connectivity throughout an SPT Region. The details of agreements are considered in 13.17, the assignment of frames to SPTs in 8.4 and Clause 27, and protocols and procedures to allow plug-and-play generation of SPVIDs in Clause 28.

Considerations of backward and forward compatibility and interoperability figure largely in this clause (Clause 13) and to some extent these consideration revolve around the notion of MST or SPT Region, with each region capable of using different protocols or different configurations. These considerations should not be allowed to obscure the fact that the ideal network configuration (from the point of view of connectivity and bandwidth efficiency) comprises a single region, or possibly one core region with RST bridges attached, and that separate regions are most likely to arise when continued connectivity is being provided by the CST as configuration changes are made to bridges in the network. If an SPT Region is bounded by Backbone Edge Bridges or LAN with no other bridges attached, B-VLANs can be supported by SPBM mode, with frames assigned to SPTs using their MAC addresses, while other B-VLANs and S-VLANs can be supported by SPBV mode, with frames assigned to each SPT by SPVID. In both cases, however, SPTs are calculated in the same way and the effects on this clause (Clause 13) are limited to allowing loop mitigation (6.5.4.2) for unicast frames supported by SPBM mode as well as loop prevention (6.5.4.1). The need for unicast multicast congruence (Clause 3), the simplification possible by not introducing differences between SPBM mode unicast forwarding and the other uses of SPTs, and the need for source address lookup to support loop mitigation means that all shortest path bridged frames are assigned to an SPT rooted at the source bridge and follow the standard bridging paradigm of multicast distribution with filtering, even when the frame has a single destination and the path to the destination is known through the use of routing protocol (ISIS-SPB).

NOTE—When loop mitigation is used for unicast frames, the Port State used to prevent loops (as expressed by the state machine variables forwarding and learning) need only apply to frames with group destination addresses.

SPT Bridges send and receive BPDUs in order to advertise their presence to, and recognize the presence of, other bridges. The requirement for loop-free SPT connectivity also means that SPT Bridges have to communicate with their nearest neighbors, when calculations that follow the reception of ISIS-SPB messages have been completed rather than as a part of the IS-IS protocol itself. The agreements (13.17) that satisfy this requirement could be separated conceptually from other uses of BPDUs, but the opportunity to share common information elements, synchronization of neighbor state, and an overall reduction in the number of protocol frames sent make it convenient to use BPDUs to integrate its specification with the other protocol variables, procedures, and state machines in this clause (13.24–13.40). Agreement protocol variables with exactly the same semantics are also carried in SPB Hello PDUs. The protocol handles duplicated and misordered agreement information, so that updated information can be received from SPB Hello PDUs or both SPB Hello PDUs and BPDUs.

SPT Bridges use the configuration and active topology management parameters (Bridge Identifiers, Port Path Costs) already required for the CIST. They also retain MSTP capabilities as a subset of their operation, though it is always possible to configure zero MSTIs.

13.7 ~~13.6~~ Compatibility and interoperability

RSTP, ~~and~~ MSTP, and the ISIS-SPB designed to interoperate with each other and with STP. This clause (13.7) reviews aspects of their design that are important to meeting that requirement. ISIS-SPB and SPT BPDUs include the functionality provided by MSTP and MST BPDUs, so the compatibility with RSTP and STP provided by the latter extends to SPB.

13.7.1 ~~13.6.1~~ Designated Port selection

Correct operation of the spanning tree protocols requires that all Bridge Ports attached to any given LAN agree on a single CIST Designated Port after a short interval sufficient for any Bridge Port to receive a configuration message from that Designated Port.

A unique spanning tree priority (13.9) is required for each Bridge Port for STP, which has no other way of communicating port roles. Since port numbers on different bridges are not guaranteed to be unique, this necessitates the inclusion of the transmitting bridge's Bridge Identifier in the STP BPDU. RSTP and MSTP's Port Protocol Migration state machines (13.32) ensure that all bridges attached to any LAN with an attached STP bridge send and receive STP BPDUs exclusively.

NOTE 1—This behavior satisfies the requirement for unique, agreed Designated Port for LANs with attached STP bridges, but means that an MST Region cannot completely emulate a single bridge since the transmitted Designated Bridge Identifier can differ on Bridge Ports at the region's boundary.

MSTP transmits and receives the Regional Root Identifier and not the Designated Bridge Identifier in the BPDU fields recognized by RSTP (14.4) to allow both the MST~~P~~ and the RST~~P~~ Bridges potentially connected to a single LAN to perform comparisons (13.9, 13.10) between all spanning tree priority vectors transmitted that yield a single conclusion as to which RST~~P~~ Bridge or MST Region includes the Designated Port. MST and RST BPDUs convey the transmitting port's CIST Port Role. This is checked on receipt by RSTP when receiving messages from a Designated Bridge, thus ensuring that an RST~~P~~ Bridge does not incorrectly identify one MST Bridge Port as being Designated rather than another, even while omitting the competing Bridge Ports' Designated Bridge Identifiers from comparisons.

NOTE 2—This ability of MST~~P~~ Bridges to communicate the full set of MSTP information on shared LANs to which RST~~P~~ Bridges are attached avoids the need for the Port Protocol Migration machines to detect RST~~P~~ Bridges. Two or more MST~~P~~ and one or more RST~~P~~ Bridges may be connected to a shared LAN, with full MSTP operation. This includes the possibility of different MSTI Designated Ports (see 13.5.1).

13.7.2 ~~13.6.2~~ Force Protocol Version

A Force Protocol Version parameter, controlled by management, permits emulation of aspects of the behavior of earlier versions of spanning tree protocol that are not strictly required for interoperability. The value of this parameter applies to all Bridge Ports.

a) STP BPDUs, rather than MST BPDUs, are transmitted if Force Protocol Version is 0. ~~RST BPDUs omit the MST Configuration Identifier and all MSTI Information.~~
b) RST BPDUs, rather than MST BPDUs, are transmitted if Force Protocol Version is 2. RST BPDUs omit the MST Configuration Identifier and all MSTI Information.
c) All received BPDUs are treated as being from a different MST Region if Force Protocol Version is 0 or 2.
d) Rapid transitions are disabled if Force Protocol Version is 0. This allows MST~~P~~ Bridges to support applications and protocols that can be sensitive to the increased rates of frame duplication and misordering that can arise under some circumstances, as discussed in Annex K of IEEE Std 802.1D-2004.
e) The MSTP state machines allow full MSTP behavior if Force Protocol Version is 3 or more.
f) SPT BPDUs are transmitted if Force Protocol Version is 4 or more.

NOTE—Force Protocol Version does not support multiple spanning trees with rapid transitions disabled.

13.8 ~~13.7~~ MST Configuration Identifier

It is essential that all bridges within an MST or SPT Region agree on the allocation of VIDs to spanning trees. If the allocation differs, frames for some VIDs may be duplicated or not delivered to some LANs at all. MST and SPT Bridges check that they are allocating VIDs to the same spanning trees as their neighbors in the same region by transmitting and receiving MST Configuration Identifiers in BPDUs. Each MST Configuration Identifier includes a Configuration Digest that is compact but designed so that two matching identifiers have a very high probability of denoting the same allocation of VIDs to MSTIDs (Clause 3, 8.4) even if the identifiers are not explicitly managed. Suitable management practices for equipment deployment and for choosing Configuration Names and Revision Levels (see below) can guarantee that the identifiers will differ if the VID to tree allocation differs within a single administrative domain.

An MST or SPT Region comprises one or more MST or SPT Bridges with the same MST Configuration Identifiers, interconnected by and including LANs for which one of those bridges is the Designated Bridge for the CIST and which have no bridges attached that cannot receive and transmit RST BPDUs.

SPT BPDUs are a superset of MST BPDUs received and validated by MST Bridges as if they were MST BPDUs, so MSTP operates within an SPT Region just as if it were an MST Region. However, each shortest path VLAN is represented by a reserved MSTID value that is included in the Configuration Digest, so an SPT Region contains only SPT Bridges.

Each MST Configuration Identifier (MCID) contains the following components:

1) A Configuration Identifier Format Selector, the value 0 encoded in a fixed field of one octet to indicate the use of the following components as specified in this standard.
2) The Configuration Name, a variable length text string encoded within a fixed field of 32 octets, conforming to IETF RFC 2271's definition of SnmpAdminString. If the Configuration Name is less than 32 characters, the text string should be terminated by the NUL character, with the remainder of the 32-octet field filled with NUL characters. Otherwise the text string is encoded with no terminating NUL character.
3) The Revision Level, an unsigned integer encoded within a fixed field of 2 octets.
4) The Configuration Digest, a 16-octet signature of type HMAC-MD5 (see IETF RFC 2104 (1997)) created from the MST Configuration Table (Clause 3, 8.9). To calculate the digest, the table is considered to contain 4096 consecutive two octet elements, where each element of the table (with the exception of the first and last) contains an MSTID value encoded as a binary number, with the first octet being most significant. The first element of the table contains the value 0, the second element the MSTID value corresponding to VID 1, the third element the MSTID value corresponding to VID 2, and so on, with the next to last element of the table containing the MSTID value corresponding to VID 4094, and the last element containing the value 0. The key used to generate the signature consists of the 16-octet string specified in Table 13-1.

Table 13-1—Configuration Digest Signature Key

Parameter	Mandatory value
Configuration Digest Signature Key	0x13AC06A62E47FD51F95D2BA243CD0346

NOTE—The formulation of the signature as described above does not imply that a separate VID to MSTID translation table has to be maintained by the implementation; rather that it should be possible for the implementation to derive the logical contents of such a table, and the signature value as specified above, from the other configuration information maintained by the implementation, as described in Clause 12.

The Configuration Digests of some VID to MSTID translations are shown in Table 13-2 to help verify implementations of this specification.

Table 13-2—Sample Configuration Digest Signature Keys

VID to MSTID translation	Configuration Digest
All VIDs map to the CIST, no VID mapped to any MSTI	0xAC36177F50283CD4B83821D8AB26DE62
All VIDs map to MSTID 1	0xE13A80F11ED0856ACD4EE3476941C73B
Every VID maps to the MSTID equal to (VID modulo 32) + 1	0x9D145C267DBE9FB5D893441BE3BA08CE

It is recommended that MST Bridge implementations provide an easily selectable or default configuration comprising a Configuration Name of the Bridge Address as a text string using the Hexadecimal Representation specified in IEEE Std 802, a Revision Level of 0, and a Configuration Digest representing a VID to MSTID translation table containing the value 0 for every element. Such a table represents the mapping of all VIDs to the CIST. Since the Bridge Address is unique to each bridge, no two bridges using this default configuration will be identified as belonging to the same region.

It is recommended that SPT Bridge implementations provide an easily selectable or default configuration comprising a Configuration Name of 'IEEE802.1 SPB Default', a Revision Level of 0, and a Configuration Digest representing an MST Configuration Table containing the value 0xFFD for VID 1, the value 0xFFF for VIDs 3600 to 3999, and the value 0 for every other element. Such a table represents the mapping of the default PVID to SPBV, an SPVID pool containing VIDs 3600 to 3999, and mapping all other VLANs to the CIST. This will allow SPT Bridges using the default configuration to form an SPT Region and operate the default VLAN in SPBV mode.

An Auxiliary MCID (14.4, 27.4.1, 28.12.2) is specified for SPB configuration and is conveyed in SPT BPDUs and SPB Hello PDUs. When a configuration change that cannot introduce loops is made, the previous MCID can be retained in the Auxiliary MCID, allowing a neighboring bridge whose configuration has not yet been changed to maintain communication without creating a region boundary. This allows an operator to increase the number of SPVIDs available to SPBV mode or the number of Base VIDs for SPBV or SPBM modes (for example) without disrupting communication. To avoid creating a region boundary all the bridges have to be configured so they are all using the same new MCID before any other change can be made.

13.9 ~~13.8~~ Spanning Tree Priority Vectors

Priority vectors permit concise specification of each protocol's computation of the active topology, both in terms of the entire network and of the operation of individual bridges in support of the distributed algorithm. MST, RST, and STP bridges use *spanning tree priority vector* information in Configuration Messages (13.14), sent and received from neighboring bridges, to assign Port Roles that determine each port's participation in a fully and simply connected active topology based on one or more spanning trees. SPT Bridges use ISIS-SPB to disseminate the information necessary to calculate Port Roles throughout SPT Regions, and to perform those calculations, but also use this Configuration Message information for the CIST to ensure that neighboring bridges agree on that active topology, and to receive the information if the CIST Root lies outside their SPT Region.

CIST priority vectors comprise the following components:

a) CIST Root Identifier, the Bridge Identifier of the CIST Root;
b) CIST External Root Path Cost, the inter-regional cost from the transmitting bridge to the CIST Root;
c) CIST Regional Root Identifier, the Bridge Identifier of the single bridge in a region whose CIST Root Port connects to a LAN in a different region, or of the CIST Root if that is within the region;
d) CIST Internal Root Path Cost, the cost to the CIST Regional Root;
e) CIST Designated Bridge Identifier, the Bridge Identifier for the transmitting bridge for the CIST;
f) CIST Designated Port Identifier, the Port Identifier for the transmitting port for the CIST;
g) CIST Receiving Port Identifier (not conveyed in Configuration Messages, used as tie-breaker between otherwise equal priority vectors within a receiving bridge).

The first two components of the CIST priority vector are significant throughout the network. The CIST External Root Path Cost transmitted by a bridge is propagated along each path from the CIST Root, is added to at Bridge Ports that receive the priority vector from a bridge in a different region, and thus accumulates costs at the Root Ports of bridges that are not MST or SPT Bridges or are CIST Regional Roots and is constant within a region. The CIST Internal Root Path Cost is only significant and defined within a region. The last three components are used as locally significant tie breakers, not propagated within or between regions. The set of all CIST spanning tree priority vectors is thus totally ordered.

Since RSTP is not aware of regions, RSTP specifications also refer to the CIST Root Identifier and CIST External Root Path Cost simply as the Root Bridge Identifier and Root Path Cost, respectively, and omit the CIST Internal Root Path Cost (as does STP). MSTP encodes the CIST Regional Root Identifier in the BPDU field used by RSTP to convey the Designated Bridge Identifier (14.1.1), so an entire region appears to an RSTP capable bridge as a single bridge. RSTP's CST use of CIST priority vectors can be conveniently specified by the use of the zero for the Internal Root Path Cost and the same values for both the Regional Root Identifier and Designated Bridge Identifier.

NOTE 1—The path to the CIST Root from a bridge with a CIST Root Port within a region always goes to or through the CIST Regional Root.

NOTE 2—STP lacks the fields necessary for MST Bridges to communicate the Designated Bridge Identifier to resolve a potential priority vector tie, and MSTP BPDUs are not sent on a LAN to which an STP bridge is attached.

Each MSTI priority vector comprises the following components for the particular MSTI in a given region:

h) MSTI Regional Root Identifier, the Bridge Identifier of the MSTI Regional Root;
i) MSTI Internal Root Path Cost, the path cost to the MSTI Regional Root;
j) MSTI Designated Bridge Identifier, the Bridge Identifier for the transmitting bridge for this MSTI;
k) MSTI Designated Port Identifier, the Port Identifier for the transmitting port for this MSTI;
l) MSTI Receiving Port Identifier (not conveyed in Configuration Messages).

The set of priority vectors for a given MSTI is only defined within a region. Within each region they are totally and uniquely ordered. A CIST Root Identifier, CIST External Root Path Cost, and CIST Regional Root Identifier tuple defines the connection of the region to the external CST and is required to be associated with the source of the MSTI priority vector information when assessing the agreement of information for rapid transitions to forwarding, but plays no part in priority vector calculations.

As each bridge and Bridge Port receives priority vector information from bridges and ports closer to the Root, calculations and comparisons are made to decide which priority vectors to record, and what information to pass on. Decisions about a given port's role are made by comparing the priority vector components that could be transmitted with that received by the port. For all components, a lesser numerical value is better, and earlier components in the above lists are more significant. As each Bridge Port receives information from ports closer to the Root, additions are made to one or more priority vector components to yield a worse priority vector for potential transmission through other ports of the same bridge.

NOTE 3—The consistent use of lower numerical values to indicate better information is deliberate as the Designated Port that is closest to the Root Bridge, i.e., has a numerically lowest path cost component, is selected from among potential alternatives for any given LAN (13.9). Adopting the conventions that lower numerical values indicate better information, that where possible more significant priority components are encoded earlier in the octet sequence of a BPDU (14.1), and that earlier octets in the encoding of individual components are more significant (14.2) allow concatenated octets that compose a priority vector to be compared as if they were a multiple octet encoding of a single number, without regard to the boundaries between the encoded components. To reduce the confusion that naturally arises from having the lesser of two numerical values represent the better of the two, i.e., that chosen all other factors being equal, this clause uses the following consistent terminology. Relative numeric values are described as "least," "lesser," "equal," and "greater," and their comparisons as "less than," "equal to," or "greater than," while relative Spanning Tree priorities are described as "best," "better," "the same," "different," and "worse" and their comparisons as "better than," "the same as," "different from," and "worse than." The operators "<" and "=" represent less than and equal to, respectively. The terms "superior" and "inferior" are used for comparisons that are not simply based on priority but include the fact that a priority vector can replace an earlier vector transmitted by the same Bridge Port. All of these terms are defined for priority vectors in terms of the numeric comparison of components below (13.10, 13.11).

NOTE 4—To ensure that the CIST and each MSTI's view of the boundaries of each region remain in synchronization at all times, each BPDU carries priority vector information for the CIST as well as for MSTIs. Associating the CIST Root Identifier, External Path Cost, and Regional Root Identifier with the priority vector information for each MSTI does not therefore raise a requirement to transmit these components separately. A single bit per MSTI vector, the Agreement flag, satisfies the requirement to indicate that the vector beginning with the MSTI Regional Root Identifier for that specific MSTI has always been associated with the single CIST Root Identifier, etc. transmitted in the BPDU.

To allow the active topology to be managed for each tree through adjusting the relative priority of different bridges and Bridge Ports for selection as the CIST Root, a CIST or MSTI Regional Root, Designated Bridge, or Designated Port, the priority component of the bridge's Bridge Identifier can be independently chosen for the CIST and for each MSTI. The priority component used by the CIST for its CIST Regional Root Identifier can also be chosen independently of that used for the CIST Root Identifier. Independent configuration of Port Path Cost and Port Priority values for the CIST and for each MSTI can also be used to control selection of the various roles for the CIST and for each MSTI.

In principle an SPT priority vector could be defined within an SPT Region, comprising a Regional Root Identifier and Internal Root Path Cost, and would reflect the construction of each SPT (lowest Internal Root Path Cost from each bridge and LAN to the Regional Root). However the set of such vectors cannot be totally ordered by the addition of purely local tie-breaker components: as each SPT Set has to be symmetric. Clause 28 specifies ISIS-SPB's calculation of SPTs.

13.10 ~~13.9~~ CIST Priority Vector calculations

The *port priority vector* is the priority vector held for the port when the reception of BPDUs and any pending update of information has been completed:

$$\begin{aligned} \textit{port priority vector} = \quad & \{RootID : ExtRootPathCost : \\ & RRootID : IntRootPathCost : \\ & DesignatedBridgeID : DesignatedPortID : RcvPortID\} \end{aligned}$$

The *message priority vector* is the priority vector conveyed in a received Configuration Message. For a bridge with Bridge Identifier B receiving a Configuration Message on a port P_B from a Designated Port P_D on bridge D claiming a CIST Root Identifier of $R_{D,}$ a CIST External Root Path Cost of ERC_D, a CIST Regional Root Identifier of $RR_{D,}$ and a CIST Internal Root Path Cost of IRC_D:

$$\textit{message priority vector} = \{R_D : ERC_D : RR_D : IRC_D : D : P_D : P_B\}$$

If B is not in the same region as D, the Internal Root Path Cost has no meaning to B and is set to 0.

NOTE—If a Configuration Message is received in an RST or STP BPDU, both the Regional Root Identifier and the Designated Bridge Identifier are decoded from the single BPDU field used for the Designated Bridge Parameter (the

MST BPDU field in this position encodes the CIST Regional Root Identifier). An STP or RST~~P~~ bridge is always treated by MSTP as being in an region of its own, so the Internal Root Path Cost is decoded as zero.

The received CIST message priority vector is the same as B’s port priority vector if:

$$\begin{aligned}
&(R_D == \mathit{RootID})\ \&\&\ (\mathit{ERC}_D == \mathit{ExtRootPathCost})\ \&\&\ (\mathit{RR}_D == \mathit{RRootID})\ \&\& \\
&(\mathit{IRC}_D == \mathit{IntRootPathCost})\ \&\&\ (D == \mathit{DesignatedBridgeID})\ \&\&\ (P_D == \mathit{DesignatedPortID})
\end{aligned}$$

and is better if:

$$\begin{aligned}
&((R_D < \mathit{RootID}))\ \|\\
&((R_D == \mathit{RootID})\ \&\&\ (\mathit{ERC}_D\ < \mathit{ExtRootPathCost}))\ \|\\
&((R_D == \mathit{RootID})\ \&\&\ (\mathit{ERC}_D == \mathit{ExtRootPathCost})\ \&\&\ (\mathit{RR}_D < \mathit{RRootID}))\ \|\\
&((R_D == \mathit{RootID})\ \&\&\ (\mathit{ERC}_D == \mathit{ExtRootPathCost})\ \&\&\ (\mathit{RR}_D == \mathit{RRootID}) \\
&\qquad \&\&\ (\mathit{IRC}_D < \mathit{IntRootPathCost}))\ \|\\
&((R_D == \mathit{RootID})\ \&\&\ (\mathit{ERC}_D == \mathit{ExtRootPathCost})\ \&\&\ (\mathit{RR}_D == \mathit{RRootID}) \\
&\qquad \&\&\ (\mathit{IRC}_D == \mathit{IntRootPathCost})\ \&\&\ (D < \mathit{DesignatedBridgeID}))\ \|\\
&((R_D == \mathit{RootID})\ \&\&\ (\mathit{ERC}_D == \mathit{ExtRootPathCost})\ \&\&\ (\mathit{RR}_D == \mathit{RRootID}) \\
&\qquad \&\&\ (\mathit{IRC}_D == \mathit{IntRootPathCost})\ \&\&\ (D == \mathit{DesignatedBridgeID}) \\
&\qquad \&\&\ (P_D < \mathit{DesignatedPortID}))
\end{aligned}$$

A received CIST message priority vector is superior to the port priority vector if, and only if, the message priority vector is better than the port priority vector, or the Designated Bridge Identifier Bridge Address and Designated Port Identifier Port Number components are the same; in which case, the message has been transmitted from the same Designated Port as a previously received superior message, i.e., if:

$$\begin{aligned}
&(\{R_D\ :\mathit{ERC}_D : \mathit{RR}_D : \mathit{IRC}_D : D : P_D : P_B\} \\
&\qquad \textit{is better than} \\
&\ \{\mathit{RootID} : \mathit{ExtRootPathCost} : \mathit{RRootID} : \mathit{IntRootPathCost} : \\
&\qquad \mathit{DesignatedBridgeID} : \mathit{DesignatedPortID} : \mathit{RcvPortID}\} \\
&)\|\ ((D.\mathit{BridgeAddress} == \mathit{DesignatedBridgeID.BridgeAddress})\ \&\& \\
&\qquad (P_D.\mathit{PortNumber} == \mathit{DesignatedPortID.PortNumber}))
\end{aligned}$$

If the message priority vector received in a Configuration Message from a Designated Port is superior, it will replace the current port priority vector.

A *root path priority vector* for a port can be calculated from a port priority vector that contains information from a message priority vector, as follows:

If the port priority vector was received from a bridge in a different region (13.29.8), the External Port Path Cost EPC_{PB} is added to the External Root Path Cost component, and the Regional Root Identifier is set to the value of the Bridge Identifier for the receiving bridge. The Internal Root Path Cost component will have been set to zero on reception.

$$\textit{root path priority vector} = \{R_D\ : \mathit{ERC}_D + \mathit{EPC}_{PB} : B : 0 : D : P_D : P_B\}$$

If the port priority vector was received from a bridge in the same region (13.29.8), the Internal Port Path Cost IPC_{PB} is added to the Internal Root Path Cost component.

$$\textit{root path priority vector} = \{R_D\ : \mathit{ERC}_D : \mathit{RR}_D : \mathit{IRC}_D + \mathit{IPC}_{PB} : D : P_D : P_B\}$$

The *bridge priority vector* for a bridge *B* is the priority vector that would, with the Designated Port Identifier set equal to the transmitting Port Identifier, be used as the message priority vector in Configuration Messages transmitted on bridge *B*’s Designated Ports if *B* was selected as the Root Bridge of the CIST.

$$bridge\ priority\ vector = \{B : 0 : B : 0 : B : 0 : 0\}$$

The *root priority vector* for bridge B is the best priority vector of the set of priority vectors comprising:

a) The bridge priority vector; plus
b) All root path priority vectors that:
   1) Have a Designated Bridge Identifier D that is not equal to B, and
   2) Were received from a Bridge Port attached to a LAN that is not in the same SPT Region as B; plus
c) The root path priority vector calculated by ISIS-SPB (if SPB is enabled, and the attached LAN is within the bridge's SPT Region).

NOTE—The BPDUs sent and received by all bridges attached to a LAN allow MST and SPT Bridges to determine whether each attached LAN is within their region independently of priority vector values. SPT Bridges take advantage of this fact by using ISIS-SPB to communicate the CST priority vector for each of SPT Region's potential Master Ports throughout the region, at the same time as ISIS-SPB calculates the Port Roles for each SPT (see Clause 28).

If the bridge priority vector is better than the port priority vector, Bridge B has been selected as the CIST Root. Otherwise the root priority vector will only be that calculated by ISIS-SPB if the Root Bridge or Regional Root is within the bridge's SPT Region.

The *designated priority vector* for a port Q on bridge B is the root priority vector with B's Bridge Identifier B substituted for the *DesignatedBridgeID* and Q's Port Identifier Q_B substituted for the *DesignatedPortID* and *RcvPortID* components. If Q is attached to a LAN that has one or more STP bridges attached (as determined by the Port Protocol Migration state machine), B's Bridge Identifier B is also substituted for the *RRootID* component.

If the designated priority vector is better than the port priority vector and the LAN attached to the port is not within the bridge's SPT Region (possibly because SPB is not enabled), the port will be the Designated Port for that LAN and the current port priority vector will be updated. If the attached LAN is within the bridge's SPT Region, then the Port Role is as calculated by ISIS-SPB and the port priority vector will be updated with the designated priority vector if and only if the port is a Designated Port. The message priority vector in Configuration Messages transmitted by a port always comprises the components of the designated priority vector for the port, even if the port is a Root Port.

13.11 ~~13.10~~ MST Priority Vector calculations

The *port priority vector* for a given MSTI is the priority vector held for the port per MSTI when the reception of BPDUs and any pending update of information has been completed:

$$port\ priority\ vector = \{RRootID : IntRootPathCost : DesignatedBridgeID : DesignatedPortID : RcvPortID\}$$

The *message priority vector* for a given MSTI is the MSTI priority vector conveyed in a received Configuration Message. For a bridge with Bridge Identifier B receiving a Configuration Message on a Regional Port P_B from a Designated Port P_D on bridge D belonging to the same MST Region and claiming an Internal Root Path Cost of IRC_D:

$$message\ priority\ vector = \{RR_D : IRC_D : D : P_D : P_B\}$$

An MSTI message priority vector received from a bridge not in the same MST Region is discarded.

An MSTI message priority vector received from a Bridge Port internal to the region is the same as the port priority vector if:

$((RR_D == RRootID)\ \&\&\ (IRC_D == IntRootPathCost)\ \&\&\ (D == DesignatedBridgeID)$
$\quad\&\&\ (P_D == DesignatedPortID))$

and is better if:

$((RR_D < RRootID))\ ||$
$((RR_D == RRootID)\ \&\&\ (IRC_D < IntRootPathCost))\ ||$
$((RR_D == RRootID)\ \&\&\ (IRC_D == IntRootPathCost)\ \&\&\ (D < DesignatedBridgeID))\ ||$
$((RR_D == RRootID)\ \&\&\ (IRC_D == IntRootPathCost)\ \&\&\ (D == DesignatedBridgeID)$
$\quad\&\&\ (P_D < DesignatedPortID))$

An MSTI message priority vector is superior to the port priority vector if, and only if, the message priority vector is better than the port priority vector, or the Designated Bridge Identifier Bridge Address and Designated Port Identifier Port Number components are the same; in which case, the message has been transmitted from the same Designated Port as a previously received superior message, i.e., if:

$(\{RR_D : IRC_D : D : P_D : P_B\}$
$\quad$ *is better than*
$\{RRootID : IntRootPathCost : DesignatedBridgeID : DesignatedPortID : RcvPortID\}$
$)\ ||\ ((D.BridgeAddress == DesignatedBridgeID.BridgeAddress)\ \&\&$
$\quad (P_D.PortNumber == DesignatedPortID.PortNumber))$

If the message priority vector received in a Configuration Message from a Designated Port for the MSTI is superior, it will replace the current port priority vector.

NOTE 1—The agree flag (13.27.3) for the port and this MSTI will be cleared if the CIST Root Identifier, CIST External Root Path Cost, and CIST Regional Root Identifier in the received BPDU are not better than or the same as those for the CIST designated priority vector for the port following processing of the received BPDU.

A *root path priority vector* for a given MSTI can be calculated for a port that has received a port priority vector from a bridge in the same region by adding the Internal Port Path Cost IPC_{PB} to the Internal Root Path Cost component.

$$root\ path\ priority\ vector = \{RR_D : IRC_D + IPC_{PB} : D : P_D : P_B)$$

NOTE 2—Internal Port Path Costs are independently manageable for each MSTI, as are the priority components of the Bridge and Port Identifiers. The ability to independently manage the topology of each MSTI without transmitting individual Port Path Costs is a key reason for retaining the use of a Distance Vector protocol for constructing MSTIs. A simple Link State Protocol requires transmission (or *a priori* sharing) of all Port Costs for all links.

The *bridge priority vector* for a bridge *B* for a given MSTI is the priority vector that would, with the Designated Port Identifier set equal to the transmitting Port Identifier, be used as the message priority vector in Configuration Messages transmitted on bridge *B*'s Designated Ports if *B* was selected as the Root Bridge of a given tree.

$$bridge\ priority\ vector = \{B : 0 : B : 0\}$$

The *root priority vector* for bridge *B* is the best priority vector of the set of priority vectors comprising the bridge priority vector plus all root path priority vectors whose Designated Bridge Identifier *D* is not equal to *B*. If the bridge priority vector is the best of this set of priority vectors, Bridge *B* has been selected as the Root of the tree.

The *designated priority vector* for a port *Q* on bridge *B* is the root priority vector with *B*'s Bridge Identifier *B* substituted for the *DesignatedBridgeID* and *Q*'s Port Identifier Q_B substituted for the *DesignatedPortID* and *RcvPortID* components.

If the designated priority vector is better than the port priority vector, the port will be the Designated Port for the attached LAN and the current port priority vector will be updated. The message priority vector in MSTP BPDUs transmitted by a port always comprises the components of the designated priority vector of the port, even if the port is a Root Port.

Figure 13-8 shows the priority vectors and the active topology calculated for an MSTI in a region of the example network of Figure 13-6.

13.12 ~~13.11~~ Port Role assignments

Each bridge assigns CIST Port Roles (when new information becomes available as specified in this clause, 13.12) before assigning MSTI or SPT Port Roles. The calculations specified in 13.10 are used to assign a role to each Bridge Port that is enabled as follows:

a) If the bridge is not the CIST Root, the source of the root priority vector is the Root Port.
b) Each port whose port priority vector is the designated priority vector is a Designated Port.
c) Each port, other than the Root Port, with a port priority vector received from another bridge is a Alternate Port.
d) Each port with a port priority vector received from another port on this bridge is a Backup Port.

If the port is not enabled, i.e., its MAC_Operational status is FALSE or its Administrative Bridge Port state is Disabled (*8.4*), it is assigned the Disabled Port role for the CIST, ~~and~~ all MSTIs, and all SPTs, to identify it as having no part in the operation of any of the spanning trees or the active topology of the network.

If the bridge is an MST or SPT Bridge, the calculations specified in 13.11 are used to assign a role to each enabled Bridge Port for each MSTI as follows:

e) If the port is the CIST Root Port and the CIST port priority vector was received from a bridge in another MST or SPT Region, the port is the Master Port.
f) If the bridge is not the MSTI Regional Root, the port that is the source of the MSTI root priority vector is the Root Port.
g) Each port whose port priority vector is the designated priority vector derived from the root priority vector is a Designated Port.
h) Each port, other than the Master Port or the Root Port, with a port priority vector received from another bridge or a CIST port priority vector from a bridge in another region, is an Alternate Port.
i) Each port that has a port priority vector that has been received from another port on this bridge is a Backup Port.

Independently of priority vector values and active topology calculations, each SPT Bridge Port determines from received BPDUs whether all the bridges attached to its LAN are in the same SPT Region. If not, the port is a Boundary Port, and its role for each SPT is determined by its CIST Port Role as follows:

j) If the port is the CIST Root Port, the port is the Master Port for all SPTs.
k) If the port is not the CIST Root Port, the port's role is the same as that for the CIST.

By excluding Boundary Ports from the physical topology used to calculate SPTs, and adopting CIST connectivity at those ports, ISIS-SPB ensures that the use of SPVIDs (see Clause 27) is not required on shared media LANs attached to bridges in other regions.

SPT Bridges use ISIS-SPB to assign Port Roles for each SPT to non-Boundary Ports as follows:

l) If the bridge is not the SPT Root Bridge, the port that ISIS-SPB has calculated as providing the path for frames assigned to the SPT and forwarded to the bridge from that SPT Root is the Root Port.

m) Each port, other than the Root Port, that ISIS-SPB has calculated as providing a path for frames forwarded from the SPT Root to the attached LAN is a Designated Port.

n) Each port that is attached to the same LAN as another port on that same bridge that is a Designated Port for the SPT, is a Backup Port.

o) Each port not assigned a Root, Designated, or Backup Port role is an Alternate Port.

13.13 ~~13.12~~ Stable connectivity

This clause provides an analysis to show that RSTP, ~~and~~ MSTP, and ISIS-SPB meet the goal of providing full and simple connectivity for frames assigned to any given VID in a stable network, i.e., where the physical topology has remained constant for long enough that the spanning tree information communicated and processed by bridges is not changing.

NOTE 1—The FDB can be configured to prevent connectivity, in particular this analysis assumes that every Bridge Port is a member of every VID's Member Set (8.8.10). Spanning tree protocol controls can also be used to prevent new connectivity (to allow for upgrades), or to disallow certain topologies (restricting the location of the CIST Root, for example). This analysis assumes that those controls are not being used, that all the bridges are using conformant protocol implementations and that the LANs are providing omnidirectional connectivity.

Every LAN provides connectivity for all frames between all attached Bridge Ports. Every bridge provides connectivity between and only between its CIST Root and Designated Ports for frames assigned to the CIST; ~~and~~ between the Root, Designated, and Master Ports for a given MSTI for frames assigned to that MSTI; and between Root and Designated Ports for a given SPT for frames assigned to that SPT. Any given bridge does not assign frames to more than one tree and has one Root Port per tree, unless it is the Root of that tree.

Every LAN has one and only one CIST Designated Port, and every bridge apart from the CIST Root has one and only one CIST Root Port. The CIST spanning tree priority vector of the Designated Port attached to the LAN that is connected to a bridge's Root Port is better than of any Designated Port of that bridge. The CIST thus connects all bridges and LANs (is "spanning") and loop-free (is a "tree").

Each MST or SPT Region is bounded by CST Root and Alternate Ports. At the CST Root Ports connectivity for frames assigned to MSTIs or SPTs within the connected regions is the same as that for the CIST. Every region apart from that containing the CIST Root has a single CST Root Port, identified as the Master Port for each MSTI. The CIST spanning tree priority vector of the LAN attached to the region's CST Root Port is better than that of any CST Designated Port of a bridge in the region attached to a LAN also attached to the CST Root Port of another region. The CST thus provides loop-free connectivity between all regions.

NOTE 2—The term "Common Spanning Tree (CST)" refers to the CIST connectivity between regions, and the term "Internal Spanning Tree (IST)" to the CIST connectivity within each region. An RST~~P~~ bridge and the LANs for which it is the Designated Bridge are conveniently considered as forming an MST region of limited extent.

Within each region each frame is consistently assigned to the CIST, ~~or~~ an MSTI, or an SPT, and each of these spanning trees provides full loop-free connectivity to each of the bridges within the region, just as the CIST does for the network as a whole, including connectivity between the CST Root Port (Master Port) and the CST Designated Ports. Since each bridge or LAN is in one and only one region, and it has already been shown that loop-free connectivity is provided between regions, loop-free connectivity is thus provided between all the bridges and LANs in the network.

Figure 13-9 illustrates the above connectivity with the simple example of Region 1 from the example network of Figure 13-6 and Figure 13-8. Bridge 0.42 has been selected as the CIST Root and Regional Root, bridge 0.57 as the Regional Root for MSTI 1, and bridge 2.83 for MSTI 2 by management of the per MSTI Bridge Identifier priority component. The potential loop through the three bridges in the region is blocked at different Bridge Ports for the CIST, and each MSTI, but the connectivity across the region and from each LAN and bridge in the region through the boundaries of the region is the same in all cases.

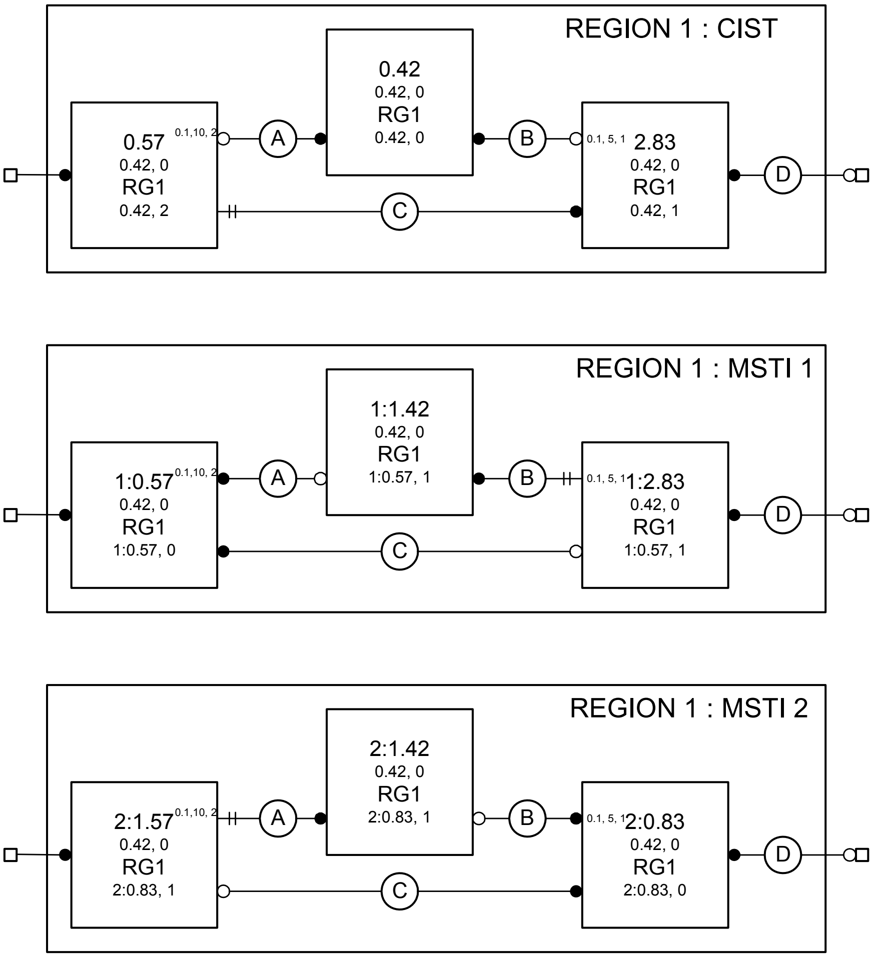


Figure 13-9—CIST and MSTI active topologies in Region 1 of the example network

13.14 ~~13.13~~ Communicating Spanning Tree information

A Spanning Tree Protocol Entity transmits and receives group addressed BPDUs (Clause 14, 8.13.4) through each of its Bridge Ports to communicate with the Spanning Tree Protocol Entities of the other bridges attached to the same LAN. The group address used is one of a small number of addresses that identify frames that are not directly forwarded by bridges (8.6.3), but the information in the BPDU can be used by a bridge in calculating its own BPDUs to transmit and can stimulate that transmission.

BPDUs are used to convey the following:

a) Configuration Messages
b) Topology Change Notification (TCN) Messages
c) MST Configuration Identifiers
d) Agreement parameters to support SPB

Designated Ports also transmit BPDUs at intervals to guard against loss and to assist in the detection of failed components (LANs, bridges, or Bridge Ports), so all messages are designed to be idempotent.

A Configuration Message for the CIST can be encoded in an STP Configuration BPDU, an RST BPDU,~~ or~~ an MST BPDU, or an SPT BPDU (14.1, 14.5). A TCN Message for the CIST can be encoded in an STP Topology Change Notification BPDU (14.2), or an RST, MST, or SPT BPDU with the TC flag set. Configuration and TCN Messages for the CIST and for all MSTIs in an MST Region are encoded in a single MST or SPT BPDU, as is the MST Configuration Identifier. No more than 64 MSTI Configuration Messages shall be encoded in an MST BPDU, and no more than 64 MSTIs shall be supported by an MST Bridge.

When SPB is enabled, ISIS-SPB is used to communicate CST priority vectors, IST topology information, and CST topology change information to and from the other bridges in the SPT Region and to calculate IST and SPT Port Roles and designated priority vectors. However the full CIST priority information is still conveyed in SPT BPDUs, partly to encode agreement information for the IST, but principally to support interoperability at the boundaries of each region without having to assess whether the transmitting port is at a boundary before deciding what to encode in each BPDU.

Configuration and Topology Change Notification BPDUs are distinguished from each other and from RST and MST BPDUs by their BPDU Type (Clause 14). RST and MST BPDUs share the same BPDU Type and are distinguished by their version identifiers.

Bridges implementing STP (Clause 8 of IEEE Std 802.1D, 1998 Edition) transmit and decode Configuration and Topology Change Notification BPDUs, and ignore RST and MST BPDUs on receipt. This ensures that connection of a Bridge Port of such a bridge to a LAN that is also attached to a bridge implementing RSTP or MSTP is detected, as transmission of RSTP or MSTP BPDUs does not suppress regular transmissions by the STP bridge. This functionality is provided by the Port Protocol Migration state machine for RSTP (13.32). The Port Protocol Migration state machines select the BPDU types used to encode Spanning Tree messages so that all bridges attached to the same LAN participate in a spanning tree protocol, while maximizing the available functionality. If one or more attached bridges only implement STP, only Configuration and Topology Change Notification BPDUs will be used and the functionality provided by the protocol will be constrained.

13.15 ~~13.14~~ Changing Spanning Tree information

Addition, removal, failure, or management of the parameters of bridges and LAN connectivity can change spanning tree information and require Port Role changes in all or part of the network (for the CIST) or all or part of an MST or SPT Region (for an MSTI or SPT Set). A CIST or MSTI configuration message received in a BPDU is considered superior to, and will replace, that recorded in the reception Port's port priority vector if its message priority vector is better, or if it was transmitted by the same Designated Bridge and Designated Port and the message priority vector, timer, or hop count information differ from those recorded.

RSTP and MSTP propagate new information rapidly from bridge to bridge, superseding prior information and stimulating further transmissions until it reaches either Designated Ports that have already received the new information through redundant paths in the network or the leaves of the Spanning Tree, as defined by the new configuration. Configuration Message transmissions will then once more occur at regular intervals from ports selected as Designated Ports.

To ensure that old information does not endlessly circulate through redundant paths in the network, preventing the effective propagation of the new information, MSTP associates a hop count with the information for each spanning tree. The hop count is assigned by the CIST Regional Root or the MSTI Regional Root and decremented by each reception Port. Received information is discarded and the port made a Designated Port if the hop count reaches zero.

RSTP and MSTP's CST processing do not use an explicit hop count (for reasons of STP compatibility), but detect circulating aged information by treating the BPDU Message Age parameter as an incrementing hop count with Max Age as its maximum value. MSTP increments Message Age for information received at the boundary of an MST Region, discarding the information if necessary.

If a Bridge Port's MAC_Operational parameter becomes FALSE, the port becomes a Disabled Port and received information is discarded. Spanning tree information for the tree can be recomputed, the bridge's Port Roles changed, and new spanning tree information transmitted if necessary. Not all component failure conditions can be detected in this way, so each Designated Port transmits BPDUs at regular intervals and a reception Port will discard information and become a Designated Port if two transmissions are missed.

NOTE—Use of a separate hop count and message loss detection timer provides superior reconfiguration performance compared with the original use of Message Age and Max Age by STP. Connectivity loss detection is not compromised by the need to allow for the overall diameter of the network, nor does the time allowed extend the number of hops permitted to aged recirculating information. Management calculation of the necessary parameters for custom topologies is also facilitated, as no allowance needs to be made for relative timer jitter and accuracy in different bridges.

ISIS-SPB communicates CST, IST, and SPT information throughout an SPT Region using link state procedures specified in Clause 28. In addition to the normal hop-by-hop distribution of this information, which is essential to guarantee its dissemination, each link state PDU is also broadcast on the SPT rooted at the originating bridge, so new information can reach bridges in the region with the same delay as data.

13.16 ~~13.15~~ Changing Port States with RSTP or MSTP

The Port State for the CIST and each MSTI for each Bridge Port is controlled by state machines whose goal is to maximize connectivity without introducing temporary loops in each of these active topologies. Root Ports, Master Ports, and Designated Ports are transitioned to the Forwarding Port State, and Alternate Ports and Backup Ports to the Discarding Port State, as rapidly as possible. Transitions to the Discarding Port State can be simply effected without the risk of data loops. This clause (13.16) describes the conditions that RSTP and MSTP use to transition a Port State for a given spanning tree to Forwarding.

Starting with the assumption that any connected fragment of a network is composed of bridges, Bridge Ports, and connected LANs that form a subtree of a spanning tree, ports with Root Port, Master Port, or Designated Port roles are transitioned using conditions that ensure that the newly enlarged fragment continues to form either a subtree or the whole of the spanning tree. Since the conditions are used every time a fragment is enlarged, it is possible to trace the growth of a fragment from a single bridge—a consistent, if small, subtree of a spanning tree—to any sized fragment, thus justifying the initial assumption.

Port States in two subtrees, each bounded by ports that are not forwarding or are attached to LANs not attached to any other bridge, can be made consistent by waiting for any changes in the priority vector information used to assign Port Roles to reach all bridges in the network, thus ensuring that the subtrees are not, and are not about to be, joined by other Forwarding Ports. However, it can be shown that a newly selected Root Port can forward frames as soon as prior recent root ports on the same bridge cease to do so, without further communication from other bridges. Rapid transitions of Designated Ports and Master Ports do require an explicit Agreement from the bridges in the subtrees to be connected. The Agreement mechanism is described, together with a Proposal mechanism that forces satisfaction of the conditions if they have not already been met by blocking Designated Ports connecting lower subtrees that are not yet in agreement. The same Agreement mechanism is then used to transition the newly blocked ports back to forwarding, advancing any temporary cut in the active topology toward the edge of the network.

13.16.1 ~~13.15.1~~ Subtree connectivity and priority vectors

Any given bridge B, the LANs connected through its Forwarding Designated Ports, the further bridges connected to those LANs through their Root Ports, the LANs connected to their Forwarding Designated Ports, and so on, recursively, constitute a subtree S_B. Any LAN L that is part of S_B will be connected to B through a Forwarding Designated Port P_{CL} on a bridge C also in S_B. L cannot be directly connected to any port P_B on bridge B unless B and C are one and the same, since the message priority vector for P_B is better than that of any port of any other bridge in $\mathrm{S_B}$, and prior to Forwarding P_{CL} will have advertised its spanning port priority vector for long enough for it to receive any better message priority vector (within the design probabilities of protocol failure due to repeated BPDU loss) or will have engaged in an explicit confirmed exchange (see below) with all other Bridge Ports attached to that LAN.

NOTE—The analysis for the distance vector based RSTP and MSTP differs from that for the link state based ISIS-SPB (see 13.17). In the latter C's priority vector can become better than B's while C remains in S_B, without B being aware of the improvement first.

13.16.2 ~~13.15.2~~ Root Port transition to Forwarding

It follows from the above that B's Root Port can be transitioned to Forwarding immediately whether it is attached to a LAN in S_B or in the rest of the network, provided that all prior recent Root Ports on B (that might be similarly arbitrarily attached) have been transitioned to Discarding and the Root Port was not a Backup Port recently (B and C the same as above).

13.16.3 ~~13.15.3~~ Designated Port transition to Forwarding

On any given bridge A, the Designated Port P_{AM} connected to a LAN M can be transitioned to Forwarding provided that the message priority advertised by the Designated Port P_{CL} on any LAN L in any subtree S_{M1}, S_{M2}, etc. connected to M is worse than that advertised by P_{AM}; that any bridge D attached to L has agreed that P_{CL} is the Designated Port; and that only the Root Port and Designated Ports on D are Forwarding. A sufficient condition for P_{AM} to transition to Forwarding is that M is a point-to-point link attached to the Root Port P_{BM} of a bridge B, that the port priority of P_{BM} is the same as or worse than that of P_{AM}, and any port P_{BN} on B is Discarding or similarly attached to a bridge C. P_{BM} signals this condition to P_{AM} by setting the Agreement flag in a Configuration Message carrying P_{BM}'s designated priority and Port Role.

NOTE 1—RSTP and MSTP use adminPointToPointMAC and operPointToPointMAC (6.6.3) to allow the point-to-point status of LANs to be managed and used by the Port Role Transition state machines for Designated Ports. A newly selected Root Port can be transitioned to Forwarding rapidly, even if attached to a shared media LAN.

Figure 13-10 illustrates the generation of an Agreement at a bridge's Root Port from an Agreement received or a Port State of Discarding at each of its Designated Ports, and a Port State of Discarding at each of its Alternate and Backup Ports. A bridge receiving a Proposal transitions any Designated Port not already synchronized to Discarding so it can send the Agreement, and that port solicits an Agreement by sending a Proposal in its turn.

NOTE 2—Agreements can be generated without prior receipt of a Proposal as soon as the necessary conditions are met. Subsequent receipt of a Proposal serves to elicit a further Agreement. If all other ports have already been synchronized (allSynced in Figure 13-10) and the Proposal's priority vector does not convey worse information, synchronization is maintained and there is no need to transition Designated Ports to Discarding once more, or to transmit further Proposals.

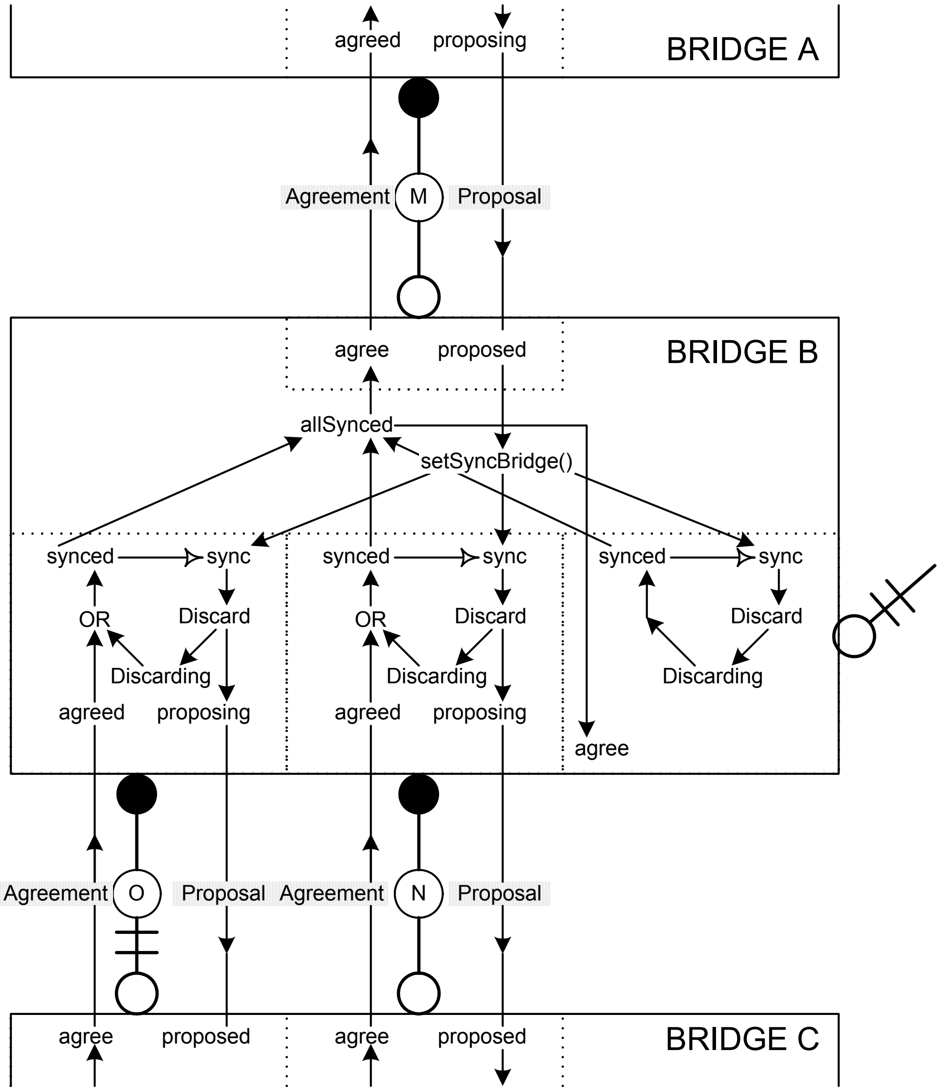


Figure 13-10—Agreements and Proposals

13.16.4 ~~13.15.4~~ Master Port transition to Forwarding

While the connectivity of the CIST from the CIST Regional Root through an MST Region to the rest of the CST comprises a subtree rooted in the CIST Regional Root, the connectivity of the MSTI from the Master Port includes both a subtree below the CIST Regional Root and a subtree rooted in the MSTI Regional Root and connected to the CIST Regional Root by an MSTI Root Port. In the example network of Figure 13-6, this latter subtree continues CST connectivity, from the Master Port on Bridge 86 through to LAN N, for frames allocated to the MSTI within Region 2 (see Figure 13-11). In general either MSTI subtree could be providing CST connectivity through a prior Master Port: the connectivity of both subtrees has to agree with the new CIST Regional Root, before a new Master Port transitions to Forwarding.

NOTE 1—The physical layout shown in the two halves of Figure 13-11 differs in order to reflect the different priorities and logical topologies for the two spanning tree instances. The layout convention is that Designated Ports are shown as horizontal lines, Root Ports as vertical lines, and Alternate Ports as diagonal lines.

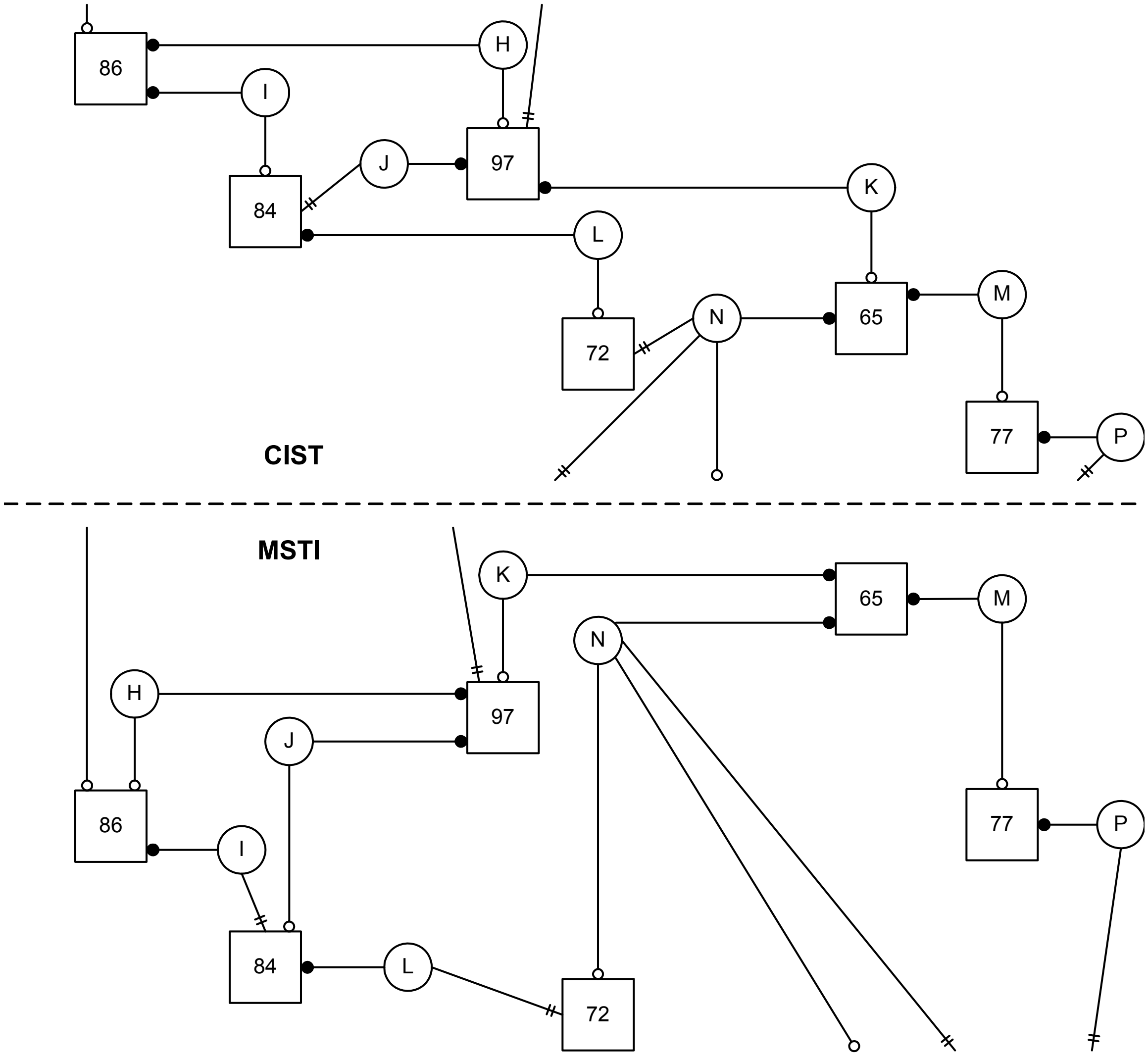


Figure 13-11—CIST and MSTI Active Topologies in Region 2 of Figure 13-6

Figure 13-12 illustrates the extension of the Agreement mechanism to signal from Designated Ports to Root Ports as well as vice versa. To ensure that an MSTI does not connect alternate Master Ports, an Agreement is only recognized at an MSTI Port when the associated CIST Regional Root information matches that selected by the reception Port. Proposals, eliciting Agreements, necessarily flow from Designated Ports to Root Ports with the propagation of spanning tree information, so a new CIST Regional Root cannot transmit a Proposal directly on its MSTI Root Ports. However, updating a CIST Designated Port's port priority vector with a new Regional Root Identifier forces the port to discard frames for all MSTIs, thus initiating the Proposal from the first bridge nearer the MSTI Regional Root that learns of the new Regional Root.

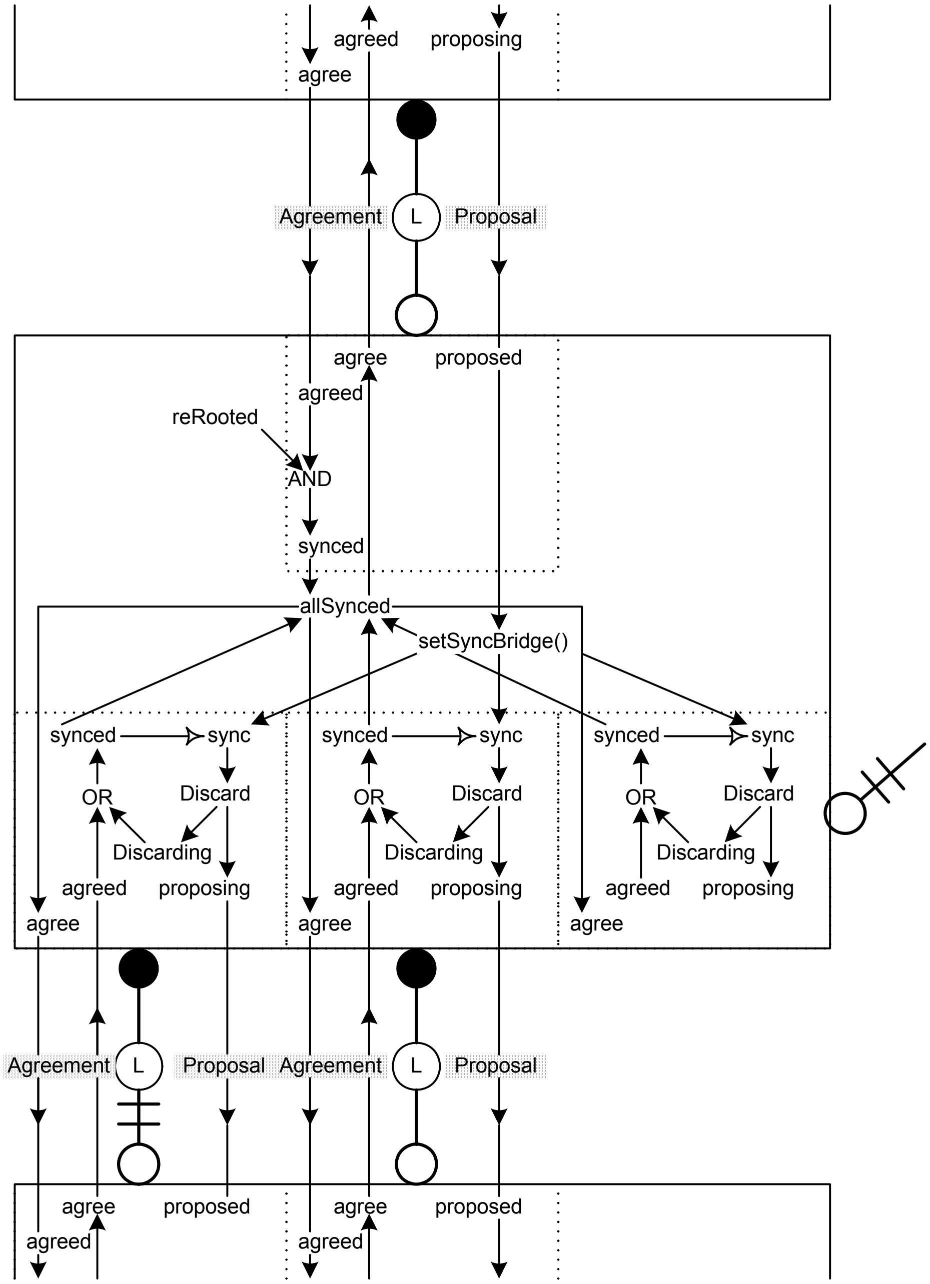


Figure 13-12—Enhanced Agreements

When an Agreement A_{MR} is sent by a Root Port P_{MR} on a Regional Root M, it attests that the CIST Root Identifier and External Root Path Cost components of the message priority advertised on all LANs connected to the CIST by P_{MR} through M are the same as or worse than those accompanying A_{MR}. The connectivity provided by each MSTI can be independent of that provided by the CIST within the MST Region and can therefore connect P_{MR} and one or more CIST Root Ports external to but attached at the boundary of the region even as CIST connectivity within the region is interrupted in order to satisfy the conditions for generating $\mathrm{A_{MR}}$. The Agreement cannot therefore be generated unless all MSTI subtrees as well as the CIST subtree internal to the region are in Agreement. To ensure that an MSTI does not connect to a CIST subtree external to the region that does not meet the constraints on the CST priority vector components, an Agreement received at an MSTI Designated Port from a Bridge Port not internal to the region is only recognized if the CIST Root Identifier and External Root Path Cost of the CIST root priority vector selected by the transmitting Bridge Port are equal to or worse than those selected by the receiver. Updating of a CIST Designated Port's port priority vector with a worse CIST Root Identifier and External Root Path Cost forces the port to discard frames for all MSTIs, thus initiating a Proposal that will elicit agreement.

NOTE 2—MSTI Designated Ports are prompted to discard frames, as required above, as follows. The CIST Port Information state machine sets sync for all MSTIs on a transition into the UPDATE state if updating the port priority with the designated priority changes the Regional Root Identifier or replaces the CIST Root Identifier or External Path Cost with a worse tuple. The MSTI's Port Role Transition machine acts on the sync, instructing the port to discard frames, and setting synced and cancelling sync when the port is discarding or an agreement is received.

NOTE 3—A "cut" in an MSTI can be transferred to the CST, either at a Designated Port attached to the same LAN as an STP bridge or at the Root Port of a bridge in an adjacent region. If the CST priority components have already been synced, as is likely if the original cut was caused by changes in physical topology within the region, the cut will terminate there. Otherwise the transferred cut precedes a cut in the CIST, and the synced port may terminate the latter. In that way, cuts in the CST will proceed through an MST Region by the quickest tree that will carry them.

NOTE 4—In the important topology where the CIST Root Bridge is within an MST Region, cuts are not transferred from the region's IST to any MSTI. CIST cuts propagating in the region will not disrupt MSTI connectivity.

13.17 Changing Port States with SPB

While the link state routing protocols are less likely to create temporary loops than distance vector based protocols, loops are still possible if (for example) two (or more) links fail about the same time. Any potential loop is a serious problem for bridged networks, as a frame that is multicast or whose destination has not been learned and is travelling around a loop is copied to each of the connected subtrees on every circuit, consuming bandwidth throughout the network. Each SPT Bridge Port uses SPT BPDUs and/or SPB Hello PDUs (28.2) to ensure that additions to each active topology do not occur unless there is sufficient agreement between neighboring bridges (with some ports temporarily transitioned to Discarding if necessary) to ensure that a temporary loop will not be created.

Like RSTP and MSTP, agreements (see 13.16, Figure 13-12) are used to ensure that ports that transition to forwarding do not create loops. However, IS-IS distributes information in parallel with computation, rather than computing a new active topology hop-by-hop as priority vector information is passed from the Root Bridge to members of a potential subtree. Moreover the distribution path is not restricted to the potential subtree. Arbitrary distribution and parallel computation means that different bridges can complete topology calculations at quite different times, with two major consequences as follows.

First, each bridge is not necessarily aware of priority vector changes passed to any subtree connected through one of its Designated Ports before any other bridge in that subtree. A forwarding Designated Port on a given bridge can only provide connectivity to another bridge through the latter's Root Port; and new connectivity is only allowed if the Designated Port has received an agreement showing that the Root Port's *designated priority* is worse than its own. When RSTP (or MSTP) is used priority vectors propagate down the tree: so each bridge knows that its forwarding parents (connected through its Root Port) either have a better *designated priority* or have discarded any agreement previously received. None of those parents can therefore provide an agreement allowing connectivity through their Root Port to one of the given bridge's Designated Ports (potentially creating a loop), unless one of the Designated Ports between the parent and the given bridge becomes Discarding—allowing an agreement to be sent but interrupting the potential loop. When the CIST or an SPT is supported by ISIS-SPB, a Bridge Port needs to receive an explicit message from its neighbors discarding any outstanding agreement before it can start forwarding as a Designated Port or (if it is the Root Port) before the bridge's Designated Ports can use an improved designated priority to transition to forwarding. A given port cannot discard outstanding agreements until all the other ports of the bridge have either received suitable agreements or stopped forwarding, or until the given port itself is Discarding.

The second major consequence of arbitrary distribution and parallel computation is that received Agreements that appear to contradict the results of the last local link state computation can become useful after a future computation completes. Agreements received after the last computation can be retained until they prove useful (following a further link state update) or are superseded by the receipt of a further agreement or a changed Port Role. The bridge transmitting the agreement will not create any connectivity that is inconsistent with the agreement until it has been explicitly discarded. For example, two Designated Ports attached to the same LAN cannot both transition to forwarding, each using an old Agreement with a Root Port role from the other. To minimize the number of messages needed to create new connectivity, unsatisfactory received Agreements are voluntarily discarded when a link state topology calculation completes. In addition to the Agreement Digest, the Agreement Protocol (13.16, 13.17) uses a two-bit Agreement Number and a Discarded Agreement Number (in SPT BPDUs and SPB Hello PDUs) to allow for message delay, loss, duplication, and misordering. Once an Agreement has been sent it is considered *outstanding* until the Agreement Protocol declares a new topology match.

Agreements for the IST use CIST information present in MST and SPT BPDUs. This minimizes active topology disruption while allowing parallel adoption of a new topology since neighboring bridges can communicate IST priority vectors, propagate agreements, and signal the need for temporary cuts to speed transitions to forwarding: even while those neighbors' views of the entire active topology differ as a consequence of differences in their knowledge of the latest physical topology. A single BPDU or SPB Hello PDU cannot convey the same per tree state for all possible SPTs, so the input to their link state calculation is summarized in a Agreement Digest (28.4). If a received digest matches the bridge's own digest, the bridge can compute the implied received Agreements. Using a digest means neighboring bridges cannot synchronize their local views of any SPT's active topology until they have synchronized their views of the physical topology of the entire SPT Region, but reducing the number of messages is worthwhile.

NOTE 1—Management VLANs with modest bandwidth needs should be allocated to the IST in SPT Regions.

The per port per tree state machine variable agreed (see Figure 13-12) is recomputed after ISIS-SPB completes a link state calculation, including the calculation of each port's designated priority vector, and after each BPDU is received: agreed is TRUE if and only if the port is a Root or Alternate Port with no outstanding agreements worse than its designated priority vector, or a Designated Port with no outstanding agreements and a received agreement that is worse than its designated priority vector and any outstanding agreement for the bridge's Root Port. As in Figure 13-12 each port is synced if it is either agreed or Discarding for the tree, and agree is TRUE for a given port when all other ports are synced. Unused outstanding agreements are discarded, by their recipient, only when agree is TRUE, since any continued connectivity has to be supported by continued agreement if loops are to be prevented.

To avoid delays that might otherwise occur waiting for Agreements to percolate up the tree, sync is set for all Designated Ports to force those that are not agreed to Discarding, so that the Root Port can transmit an Agreement to its neighbor. However it is an implementation issue as to how long sync requires to take effect: the likelihood of receiving an Agreement for the CIST from the Designated Port's neighbor before the port can be transitioned to Discarding and then to Forwarding again can be taken into account. Each BPDU's CIST Proposal flag should be set when a Designated Port is Discarding, thus providing a prompt for the return of an Agreement just as described above (13.16) for RSTP and MSTP.

The Agreement Digest conveys fresh Agreements for all SPTs (all identified by the same Agreement Number used for the CIST) and discards those received for prior topologies (identified by the Discarded Agreement Number). Since different trees will have different Root Ports (in the same bridge), if all bridges waited for agreed to be set for all other ports for all other trees before transmitting a fresh digest the protocol would deadlock. If the Root Port is not agreed but all other ports are synced, then agree is also set for any port that is Discarding, thus avoiding the potential deadlock. When a link state computation completes, each bridge transitions any Designated Port that is not agreed for an SPT to Discarding as rapidly as possible so that it can send the fresh digest to its neighbors, and does not attempt to optimize Port State transitions by waiting to receive a matching digest.

NOTE 2—In all comparisons between SPT priority vectors, the specified comparisons take place within the CST context, and agreed is only set if the CST Root and External Path Cost match exactly. This allow the Port Role Transition state machine for the Master Port at the SPT Region boundary, where the IST and each SPT are connected into the CST, to function just as for MSTP.

The considerations and state machine conditions detailed in this clause (13.17) can be summarized (for an SPT Bridge with all ports providing connectivity within an SPT Region) as follows:

A Root Port satisfies the conditions for Forwarding if and only if:

a) The Agreement Protocol has declared a topology match; and
b) Since (and including) that topology match, no outstanding Agreement Digest sent through the port is for a topology where:
    1) This bridge is or was farther from the SPT's Root Bridge than in the current topology; or
    2) The port was a Designated Port or Backup Port.

A Designated Port satisfies the conditions for Forwarding if and only if:

c) The Agreement Protocol has declared a topology match; and
d) Since (and including) that topology match, no outstanding Agreement Digest sent through the port is for a topology where:
    1) Any other Bridge attached to the same LAN is or was as close or closer to the SPT's Root Bridge than this bridge is in the current topology; or
    2) The port was a Root Port or Alternate Port.

Note that a Root Port can only become Forwarding following receipt of a fresh Agreement Digest if all Designated Ports either satisfy the conditions for Forwarding or are made Discarding. A Designated Port can only become Forwarding if the Root Port also satisfies the conditions for Forwarding, or is made Discarding.

An Agreement Digest is sent on all ports as soon as possible after ISIS-SPB completes a link state calculation, but is only sent if (for every SPT):

e) The port sending the Agreement Digest is Discarding; or
f) The port is a Root Port or a Designated Port, that satisfies the necessary conditions for Forwarding (item (a) to item (d) above), and every other port of the bridge is either Discarding or is a Root or Designated Port that also satisfies those conditions.

13.17.1 Agreement Digest

The Agreement Digest is calculated as specified in 28.4.

The purpose of the Agreement Digest is to prevent temporary loops and its use as an input to the Port Role Transition and Port State Transition state machines is specified in 13.29, 13.37, and 13.38. Successful operation of ISIS-SPB relies on the fact that SPT Bridges attached to the same LAN will, in the absence of continual physical topology changes in the network or continual loss of ISIS-SPB frames, advertise the same Agreement Digest in transmitted BPDUs and SPB Hello PDUs. Conditions that would result in permanent disagreement, and consequent failure to provide shortest path bridging, are captured in the MST Configuration Identifier (13.8). Differences in the MST Configuration Identifier result in both MST and SPT Bridge Ports attached to the LAN being identified as bounding a region, so connectivity can be provided using the CST.

13.18 ~~13.16~~ Managing spanning tree topologies

The active topology of the CIST, and the topologies that can result after the failure or addition of network components, may be managed by assigning values to some or all of the following:

- The Bridge Priority component of the CIST Bridge Identifier for one or more bridges.
- The External Port Path Cost (also referred to as the Port Path Cost for RSTP) for some Bridge Ports.
- Components of the MST Configuration Identifier for bridges with the same Configuration Digest.
- The CIST Internal Port Path Cost Port (for MSTP and ISIS-SPB) for some Bridge Ports.
- The Port Priority component of the Port Identifier for some Bridge Ports.

Within an MST Region, the active topology of each MSTI may be managed by assigning values to some or all of the following:

- The Bridge Priority component of the MSTI Regional Root Identifier.
- The Internal Port Path Cost for the MSTI for some Bridge Ports.
- The Port Priority component of the MSTI's Port Identifier for some Bridge Ports.

In general topology management objectives can be met by modifying only a few parameter values in a few bridges in the network. Table 13-3 specifies default values and ranges for Bridge Priorities and Port Priorities. If these parameters can be updated by management, the bridge shall have the capability to use the full range of values with the granularity specified.

Table 13-3—Bridge and Port Priority values

Parameter	Recommended or default value	Range
Bridge Priority	32 768	0–61 440 in steps of 4096
Port Priority	128	0–240 in steps of 16

NOTE 1—The stated ranges and granularities for Bridge Priority and Port Priority differ from those in IEEE Std 802.1D, 1998 Edition, and earlier revisions of that standard. Expressing these values in steps of 4096 and 16 allows consistent management of old and new implementations of this standard; the steps chosen ensure that bits that have been re-assigned are not modified, but priority values can be directly compared.

Table 13-4 recommends defaults and ranges for Port Path Cost and Internal Port Path Cost values, chosen according to the speed of the attached LAN, to minimize the administrative effort required to provide reasonable active topologies. If these values can be set by management, the bridge shall be able to use the full range of values in the parameter ranges specified, with a granularity of 1.

When two or more links are aggregated (see IEEE 802.1AX), Port Path Cost and Internal Port Path Cost values can be modified to reflect the actual throughput. However, as the primary purpose of Path Cost is to select active topologies, it can be inappropriate to track throughput too closely, as the resultant active topology could fluctuate or differ from that intended by the network administrator. For example, if the network administrator had chosen aggregated links for resilience (rather than for increased data rate), it would be inappropriate to change topology as a result of one of the links in an aggregation failing. Similarly, with links that can autonegotiate their data rate, reflecting such changes of data rate in changes to Path Cost is not necessarily appropriate. As a default behavior, dynamic changes of data rate should not automatically cause changes in Port Path Cost.

Table 13-4—Port Path Cost values

Link Speed	Recommended value	Recommended range	Range
<=100 Kb/s	200 000 000	20 000 000–200 000 000	1–200 000 000
1 Mb/s	20 000 000	2 000 000–200 000 000	1–200 000 000
10 Mb/s	2 000 000	200 000–20 000 000	1–200 000 000
100 Mb/s	200 000	20 000–2 000 000	1–200 000 000
1 Gb/s	20 000	2 000–200 000	1–200 000 000
10 Gb/s	2 000	200–20 000	1–200 000 000
100 Gb/s	200	20–2 000	1–200 000 000
1 Tb/s	20	2–200	1–200 000 000
10 Tb/s	2	1–20	1–200 000 000

NOTE 2—BPDUs are capable of carrying 32 bits of Root Path Cost information, though IEEE Std 802.1D, 1998 Edition, and its earlier revisions limited the range of the Port Path Cost parameter to a 16-bit unsigned integer value. Table 13-4 uses the full 32-bit range to extend the range of supported link speeds. Additional recommended values can be calculated as 20 000 000 000/(Link Speed in Kb/s). Limiting the range of the Path Cost parameter to 1–200 000 000 ensures that the accumulated Path Cost cannot exceed 32 bits over a concatenation of 20 hops. Where bridges using the IEEE Std 802.1D, 1998 Edition, recommendations and others using Table 13-4 are mixed in the same Bridged Local Area Network, explicit configuration is likely to be necessary to obtain reasonable CST topologies.

13.19 ~~13.17~~ Updating learned station location information

In normal stable operation, learned station location information held in the Filtering Database need only change as a consequence of the physical relocation of stations. It is therefore desirable to employ a long ageing time for Dynamic Filtering Entries (8.8.3), especially as many end stations transmit frames following power-up causing the information to be relearned.

However, when the active topology reconfigures, stations can appear to move from the point of view of any given bridge even if that bridge's Port States have not changed. If a Bridge Port is no longer part of an active topology, stations are no longer reachable through that port, and its Dynamic Filtering Entries are removed from that bridge's Filtering Database. Conversely, stations formerly reachable through other ports might be reachable through a newly active port. Dynamic Filtering Entries for the other ports are removed, and RSTP and MSTP transmit Topology Change Notification Messages both through the newly active Port and through the other active Ports on that Bridge. TCNs signal additional connectivity, not just changes in connectivity, as relearning a station's location is only possible if it can be reached, and if that is possible when a port is removed from the active topology another port will be added. A TCN is sent when a Bridge Port joins the active topology, and not before, so that bridges can relearn removed station location information and minimize unnecessary flooding of frames. A bridge that receives a TCN on an active port removes Dynamic Filtering Entries for their other active ports and propagates the TCN through those ports.

NOTE 1—STP allowed for the presence of LAN repeaters that could partition a shared media LAN, thus causing stations to appear to move when the partition was repaired later—with the only Bridge Port changing Port Role or Port State transitioning to Discarding at that time. This scenario does not occur with current technology, and its future likelihood does not justify the use of TCNs to signal connectivity loss. Bridge Ports that participate in the MAC status propagation protocol should be capable of originating TCNs when that protocol signals additional connectivity.

The Topology Change state machine (13.39) avoids removing learned information when ports temporarily revert to Discarding to suppress loops. It treats a port as joining the active topology when it becomes forwarding, and no longer active when it becomes an Alternate, Backup, or Disabled Port and stops forwarding and learning. TCNs are not generated following Edge Port (operEdge, 13.27.44) Port State changes, as these do not affect connectivity or station location information in the rest of the network, nor are Dynamic Filtering Entries for Edge Ports removed when TCNs are received.

Dynamic Filtering Entries for MAC addresses previously learned on a Root Port may be modified to move those addresses to an Alternate Port that becomes the new Root Port and a TCN sent only through the new Root Port (and not through other active ports), reducing the need to flood frames. This optimization is possible because a retiring Root Port that becomes Discarding temporarily partitions the active topology into two subtrees, one including all bridges and LANs hitherto reachable through the retiring Root Port, and the other including all the others. If the new Root Port simply provides a new path to the first of these subtrees, its stations will not appear to move from the point of view of bridges in the other subtree. Alternatively if a the tree reconfiguration is more complex one or more newly Designated Port will become active and will transmit the necessary TCNs.

NOTE 2—The rules described require removal of potentially invalid learned information for a minimum set of ports on each bridge. A bridge implementation can flush information from more ports than strictly necessary, removing (for example) all Dynamic Filtering Entries rather than just those for the specified ports. This does not result in incorrect operation, but will result in more flooding of frames with unknown destination addresses.

Changes in the active topology of any given MSTI or SPT do not change Dynamic Filtering Entries for the CIST or any other MSTI or SPT, unless the underlying changes in the physical topology that gave rise to the reconfiguration also cause those trees to reconfigure. Changes to the CST, i.e., the connectivity provided between regions, can cause end station location changes for all trees. Changes to an IST can cause CST end station location changes but do not affect MSTIs in that region unless those trees also reconfigure.

On receipt of a CIST TCN Message from a Bridge Port not internal to the region, or on a change in Port Role for a Bridge Port at the region boundary, TCN Messages are transmitted through each of the other ports of the receiving bridge for each MSTI and the Dynamic Filtering Entries for those ports are removed.

NOTE 3—The port receiving the CIST TCN Message can be a Master Port, a Designated Port attached to the same LAN as an STP bridge, or a Designated Port attached to the same LAN as the Root Ports of bridges in other regions.

TCN Messages for the CIST are always encoded in the same way, irrespective of whether they are perceived to have originated from topology changes internal to the region or outside it. This allows RST~~P~~ Bridges whose Root Ports attach to a LAN within an MST Region to receive these TCN Messages correctly.

Since each SPT is rooted at the source of the frames assigned to that SPT, or at the point where those frames enter the SPT Region, entries are only learned for the Root Port of an SPT. Any apparent changes in station location due to changes in an SPT's active topology only occur when a bridge changes the Root Port for that SPT, and are accommodated by removing the Dynamic Filtering Entries for the prior Root Port and FID associated with the SPT Set. If the same new Root Port is chosen for all SPTs, in the SPT set, with that prior Root Port, the entries can be moved to the new Root Port rather than simply being deleted.

NOTE 4—Dynamic Filtering Entries created by ISIS-SPB as a result of that protocol's direct knowledge of the location of some stations are not removed when a TCN is received, but can be changed by ISIS-SPB as result of its calculations.

NOTE 5—Topology changes for the IST are always propagated by TCNs received and transmitted in BPDUs, and are not injected as a result of ISIS-SPB calculations as the latter complete prior to new connectivity being established.

13.20 ~~13.18~~ Managing reconfiguration

The priority component of the Bridge Identifier can be managed for the CIST, and independently for each MSTI, to allow preferential selection of the first choice Root Bridge and of alternate Roots that will be used if the better choices have failed or lack connectivity. The Port Path Cost of each Bridge Port can also be managed to allow preferential selection of the path from each bridge to the Root, with the priority component of each bridge's Bridge Identifier providing a manageable tie-breaker between equal cost paths.

In the event of LAN or bridge failure, fastest reconfiguration (and thus highest service availability) can usually be provided by a bridge that can substitute a prior Alternate Port for a Root Port that now lacks or provides inferior connectivity to the Root. Figure 13-2 illustrates a simple topology where most failures can be handled in this way. Configuration to place a backup Root Bridge adjacent to the primary (bridges 222 and 111 in the figure ~~neighbor~~, respectively) enables Alternate Port to Root Port failover in the other bridges. If the original Root Port and its replacement Alternate Port have the same Root Port Path Cost, bridges further from the Root will not see a topology change. Figure 13-5 illustrates a ring topology, where simple failover copes with few failures. However locating a backup Root Bridge adjacent to the preferred Root remains effective in reducing the effect of any change.

RSTP and MSTP are distance vector protocols: following the failure of a Root Bridge (or selection of a costlier path to the Root) spanning tree priority vectors conveying the prior Root can circulate in the network until that information ages out (i.e., until Message Age exceeds Max Age, or the remainingHops parameter is decremented to zero). To accommodate a wide range of networks, the default setting of these parameters permit 20 hops. However, if the preferred and alternate Roots are located at the center of the network, many networks can be configured with minimal values (6 hops, Table 13-5) so each message can only be received by any node at most twice. In larger networks, for example ring backbones with other bridges redundantly attached to adjacent bridges in the ring, the unwanted effects of recirculating information can be prevented by configuring the restrictedRole parameter (13.27.64). This parameter prevents information from being propagated through ports that should never provide connectivity toward the Root. In Figure 13-3, for example, restrictedRole could be configured for any or all of the Designated Ports.

Other potentially disruptive effects of reconfiguration can be prevented or mitigated by setting the restrictedTcn parameter (13.27.65) for a port.

The configuration of a subtree of a spanning tree defined by the connectivity provided by a given bridge *B* (say) can be made independent of the rest of the tree by configuring *B*'s Root Port as a Layer 2 Gateway Port (L2GP). An L2GP Port behaves as if it is in continual receipt of a fixed, manageable, spanning tree priority vector. Provided that this fixed priority vector is worse than the actual priority vector transmitted by the Designated Port for the LAN attached to *B*'s Root Port, there is no potential for a loop. Each port in the subtree will transmit a worse priority vector. If a received message priority vector is worse than the fixed value the disputed flag is set, causing the L2GP port to transition to Discarding. Thus stability of the spanning tree priority vector information in the subtree is maintained, at the possible expense of the connectivity between the subtree and the rest of the tree. Alternatively, if another bridge within *B*'s subtree becomes the source of better priority information, causing *B*'s L2GP port to assume a Designated Port Role, a dispute will exist between the newly propagated spanning tree information and the fixed information for the L2GP port, and will also cause that port to become Discarding.

The L2GP capability is optional, and is primarily intended to be used for service access protection (*25.9*) when a single customer's Bridged Local Area Network is connected to a Provider Backbone Bridged Network (PBBN) or a Provider Bridged Network (PBN). Two or more Bridge Ports on one or more of the bridges within the customer's network, are connected to a single service instance supported by the provider's network, and are configured as L2GP ports. The L2GP ports' fixed priority vectors are each defined by a different, configured pseudoRootId (13.27.51). Only the L2GP port with the best pseudoRootId can provide connectivity to the provider's network. Spanning tree information from that port will be disseminated throughout the customer's network and will be disputed by the other L2GP port, causing the latter to become Discarding. If all the customer network's ports that are bridged to other networks (such as PBNs or PBBNs) are configured as L2GP ports, those ports neither have to transmit or receive BPDUs to prevent loops—though connectivity through providers' networks to other networks will not be protected. Enabling BPDU reception for each L2GP port prevents loops that might otherwise be caused by attaching another of the customer's network's ports to the provider network.

13.21 ~~13.19~~ Partial and disputed connectivity

It is possible for the connectivity between Bridge Ports attached to the same LAN to fail, in system or media access method dependent ways, so that BPDUs and data frames are transmitted in one direction only. As a result it is possible for more than one port attached to the same LAN to believe itself to be the Designated Port. To ensure that the active topology remains loop-free, a Designated Port will recognise that a dispute is in progress and stop learning from or forwarding frames, if it receives a BPDU with a worse message priority and the Learning flag set from another port that claims to be Designated.

If two (Designated) ports attached to the same shared media LAN cannot communicate with each other at all, but can each communicate with a third (Root) port, a potential loop exists if one of the Designated Ports has a priority vector that is worse than that of the Root Port. To ensure that such loop does not occur, a Root Port that receives an inferior message from a Designated Port detects a dispute if the Learning flag is set, and transitions to Discarding.

13.22 ~~13.20~~ In-service upgrades

It can be desirable to upgrade the control plane software of a bridging system, without interrupting existing data connectivity, while the Spanning Tree Protocol Entity is not operating. This can be done, without the risk of creating data loops, providing that the other bridges in the network are suitably configured. However the failure of a network component (bridge or LAN) while the upgrade is in progress can result in a loss of connectivity, as the ways in which the network can reconfigure are restricted. This clause (13.22) describes the necessary configuration of the other bridges in the network; the behavior of the upgrading system is assumed to be system dependent and is not specified in detail, except as follows:

a) Frames can be received from and transmitted to ports that were forwarding prior to beginning the upgrade, and are not forwarded to or from any other port.
b) Frames received on ports that were learning prior to beginning the upgrade can be submitted to the Learning Process, while frames received on other ports are not.
c) BPDUs are not transmitted, and received BPDUs are discarded, while the upgrade is in progress.
d) If there is to be no interruption in connectivity when the upgrade is complete, the parameters of BPDUs received prior to beginning the upgrade are retained, or recovered using ISIS-SPB.

NOTE 1—If received BPDU information is not retained, the spanning tree protocol variables and state machines are reinitialized by asserting BEGIN, and there will be a brief interruption in connectivity.

The need for a control plane software upgrade can result from the need to change any component of that software, and rarely from upgrades to the Spanning Tree Protocol Entity itself. The in-service upgrade procedures described here depend on the operation of specifically identified features of the RSTP, ~~and~~ MSTP, and ISIS-SPB specified in this standard (Clause 13), and do not ensure loop-free operation if the necessary criteria are not met by all bridges in the network. In particular safe upgrades are not supported in networks including bridges operating STP as specified in the IEEE Std 802.1D, 1998 Edition, and earlier revisions. It is essential for the network administrator to base-line the network, i.e., verify its configuration and the configuration of its components, prior to attempting an in-service upgrade. It is recommended that a single system be upgraded at a time, and its subsequent correct operation verified prior to making further changes.

If all the LANs that connect the bridges in the network provide point-to-point connectivity, connecting at most two bridges, and each Bridge Port thus connecting to another bridge meets the following conditions:

e) operPointToPointMAC (13.28.16) is TRUE; and
f) operEdge (13.27.44) is FALSE;

NOTE 2—Apart from the requirement for a point-to-point physical topology, these conditions can be ensured by setting adminPointToPointMAC (6.6.3) TRUE, and both AdminEdge and AutoEdge FALSE.

then a Designated Port will not transition from Discarding to Learning or Forwarding without receiving an Agreement from its immediate neighbor (see 13.37). This prevents the introduction of data loops while one or more bridges are being upgraded, even if other bridges or LANs fail or are added to the network. To prevent existing connectivity being disrupted (provided no other network additions or failures occur) the following are also necessary:

g) Prior to the upgrade, each of the upgrading bridge's immediately neighboring ports that is a Root Port (for a given tree) has to be configured as an Layer 2 Gateway Port (13.20) (for that tree), with the same information that it was previously receiving from the upgrading bridge.
h) After the upgrade, each of the ports thus temporarily configured as an Layer 2 Gateway Port needs to have its normal configuration restored.

NOTE 3—If the upgrading bridge is capable of sending periodic 'canned BPDUs' containing the same information as immediately prior to the upgrade, there is no requirement for neighbor L2GP configuration, or for the point-to-point and operEdge conditions. Attempts by neighboring bridges to operate outside the parameters dictated by the 'canned BPDUs' will result in disputes, preventing new connectivity.

On the SPB ports within an SPT Region, the functions of the Layer 2 Gateway Port described in items g) and h) above are replaced by those of 'Restart Signaling for IS-IS' , IETF RFC 5306, October 2008 [Annex M], which allows the forwarding state across an adjacency to be maintained whilst an IS-IS element is restarted. When the neighbor (running) shortest path bridge receives a restart request, it performs the procedures of IETF RFC 5306, and for as long as it maintains the state of the IS-IS adjacency as "UP" it also maintains its received Agreement Digest at the last value received from the restarting bridge before it requested restart. If the adjacency is successfully reacquired by the procedures of IETF RFC 5306, as signaled by the restarting bridge clearing the Restart Request bit in an SPB Hello PDU, its running neighbor shall include its current Agreement Digest value in its SPB Hello response [and update it as required thereafter according to the Agreement Protocol (13.17)].

Once the restarting bridge has achieved LSP Database Synchronization according to the procedures of IETF RFC 5306 [Annex M], it shall compute its local value of the Agreement Digest prior to making any changes to its FDB. On ports for which a Digest match is achieved with a neighbor, full installation of unicast and multicast state may be performed immediately. On ports that do not have a Digest match (caused for example by a propagating topology\change), the restarting bridge shall assume when computing allowed forwarding entries that Digest Agreement has never been achieved on the affected adjacency (where "allowed" is determined by the value of the Agreement Digest Convention Identifier; see 28.4.3). In either event, a Digest Agreement shall be sent to the neighbor as soon as possible, as specified by the Agreement Protocol.

SPB ports on the boundary of an SPT Region use the procedures of a) to h) unaltered.

13.23 ~~13.21~~ Fragile bridges

Some, nonconformant, bridging systems are known to be "fragile," i.e., they can suffer from unpredictable interruptions to Spanning Tree Protocol Entity operation and will forward data frames while no longer sending or receiving BPDUs, on some or all ports. If the spanning tree protocol implementations of the other bridges in the network conform to prior revisions of this standard and IEEE Std 802.1D, these interruptions can result data loops even in networks of point-to-point LANs. This revision of this standard allows a Bridge Port attached to point-to-point LAN to ensure that its neighbor remains capable of receiving and transmitting BPDUs, even if that neighbor's RSTP or MSTP implementation conforms to a prior revision of this standard, and to block connectivity (by transitioning to Discarding) otherwise. The CIST Proposal flag is set in all BPDUs transmitted by a Designated Port, and used to solicit an Agreement from the neighbor (which might otherwise not transmit). This capability is controlled by the AutoIsolate (13.27.19) variable, is disabled by default to allow for in-service upgrades, and is only effective if the neighboring bridge is capable of RSTP or MSTP operation.

13.24 ~~13.22~~ Spanning tree protocol state machines

Each Spanning Tree Protocol Entity's operation of the protocols specified in this clause (Clause 13) is specified by the following state machines:

a) Port Role Selection (PRS, 13.36)

for the CIST and for each MSTI, with the following state machines for each Bridge Port:

b) Port Timers (PTI, 13.30)
c) Port Protocol Migration (PPM, 13.32)
d) Port Receive (PRX, 13.31)
e) Port Transmit (PRT, 13.34)
f) Bridge Detection (BDM, 13.33)

and the following optional state machine for each Bridge Port:

g) Layer 2 Gateway Port Receive (L2GPRX, 13.40)

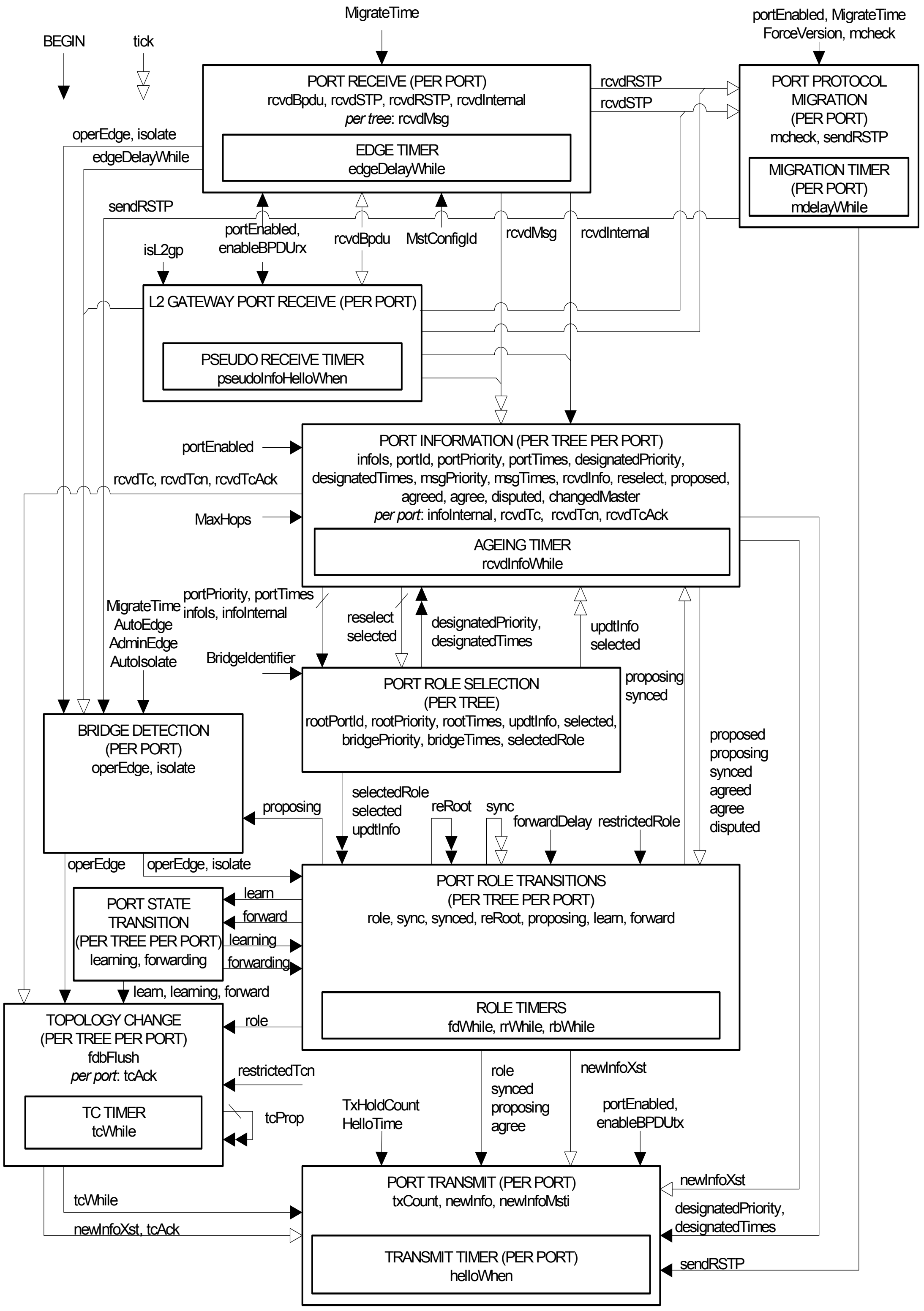
and the following state machines for each Bridge Port for the CIST and for each MSTI:

h) Port Information (PIM, 13.35)
i) Topology Change (TCM, 13.39)

and the following state machines for each Bridge Port for the CIST, for each MSTI, and for each SPT:

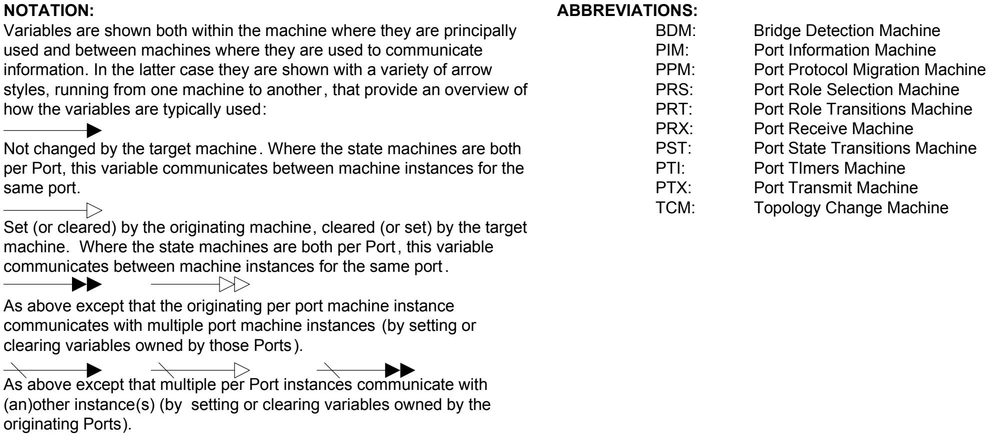
j) Port Role Transitions (PRT, 13.37)
k) Port State Transition (PST, 13.38)

Each state machine and its associated variable and procedural definitions are specified in detail in 13.25 through 13.40. The state machine notation is specified in Annex E. Figure 13-13 is not itself a state machine but provides an overview of the state machines, their state variables, and communication between machines. Figure 13-14 describes its notation.



NOTE: For convenience all timers are collected together into one state machine .

Figure 13-13—Spanning tree protocol state machines—overview and relationships



NOTATION:

Variables are shown both within the machine where they are principally used and between machines where they are used to communicate information. In the latter case they are shown with a variety of arrow styles, running from one machine to another, that provide an overview of how the variables are typically used:

Not changed by the target machine. Where the state machines are both per Port, this variable communicates between machine instances for the same port.

Set (or cleared) by the originating machine, cleared (or set) by the target machine. Where the state machines are both per Port, this variable communicates between machine instances for the same port.

As above except that the originating per port machine instance communicates with multiple port machine instances (by setting or clearing variables owned by those Ports).

As above except that multiple per Port instances communicate with (an)other instance(s) (by setting or clearing variables owned by the originating Ports).

ABBREVIATIONS:

BDM:	Bridge Detection Machine
PIM:	Port Information Machine
PPM:	Port Protocol Migration Machine
PRS:	Port Role Selection Machine
PRT:	Port Role Transitions Machine
PRX:	Port Receive Machine
PST:	Port State Transitions Machine
PTI:	Port TImers Machine
PTX:	Port Transmit Machine
TCM:	Topology Change Machine

Figure 13-14—MSTP overview notation

13.25 ~~13.23~~ State machine timers

The timer variables declared in this subclause are part of the specification. The accompanying descriptions are provided to aid in the comprehension of the protocol only, and are not part of the specification. Each timer variable represents an integral number of seconds before timer expiry.

One instance of the following shall be implemented per port:

a) edgeDelayWhile (13.25.1)
b) helloWhen (13.25.3)
c) mdelayWhile (13.25.4)

One instance of the following shall be implemented per port when L2GP functionality is provided:

d) pseudoInfoHelloWhen (13.25.10)

One instance per port of the following shall be implemented for the CIST and one per port for each MSTI:

e) fdWhile (13.25.2)
f) rrWhile (13.25.7)
g) rbWhile (13.25.5)
h) tcWhile (13.25.9)
i) rcvdInfoWhile (13.25.6)
j) tcDetected (13.25.8)

Table 13-5 specifies values and ranges for the initial values of timers and for transmission rate limiting performance parameters. Default values are specified to avoid the need to set values prior to operation in most cases, and are widely applicable to networks using the spanning tree protocols specified in this standard. The table recommends Bridge Hello Time, Bridge Max Age, and Bridge Forward Delay values that maximise interoperability in networks that include bridges using STP as specified in IEEE Std 802.1D, 1998 Edition. Ranges are specified to ensure that the protocols operate correctly.

NOTE—Changes to Bridge Forward Delay do not affect reconfiguration times, unless the network includes bridges that do not conform to this revision of this standard. Changes to Bridge Max Age can have an effect, as it is possible for old information to persist in loops in the physical topology for a number of “hops” equal to the value of Max Age in seconds, and thus exhaust the Transmit Hold Count in small loops.

Table 13-5—Timer and related parameter values

Parameter	Default	Permitted range	Interoperability recommendations
Migrate Time	3.0	—[a]	—[a]
(Bridge) Hello Time	2.0	—[a]	—[a]
Bridge Max Age	20.0	6.0–40.0	20.0
Bridge Forward Delay	15.0	4.0–30.0	15.0
Transmit Hold Count	6	1–10	6
Max Hops	20	6–40	—

All times are in seconds. —[a] Not applicable, value is fixed.

Bridge Max Age, Bridge Forward Delay, and Transmit Hold Count may be set by management, if this capability is provided the bridge shall have the capability to use the full range of values in the parameter ranges specified in the "Permitted range" column of Table 13-5, with a timer resolution of *r* seconds, where 0 < *r* <= 1. To support interoperability with previous revisions of this standard and IEEE Std 802.1D, a bridge shall enforce the following relationships:

$$2 \times (Bridge\_Forward\_Delay - 1.0\ seconds) >= Bridge\_Max\_Age$$

$$Bridge\_Max\_Age >= 2 \times (Bridge\_Hello\_Time + 1.0\ seconds)$$

13.25.1 ~~13.23.1~~ edgeDelayWhile

The Edge Delay timer. The time remaining, in the absence of a received BPDU, before this port is identified as an operEdgePort.

13.25.2 ~~13.23.2~~ fdWhile

The Forward Delay timer. Used to delay Port State transitions until other Bridges have received spanning tree information.

13.25.3 ~~13.23.3~~ helloWhen

The Hello timer. Used to ensure that at least one BPDU is transmitted by a Designated Port in each HelloTime period.

13.25.4 ~~13.23.4~~ mdelayWhile

The Migration Delay timer. Used by the Port Protocol Migration state machine to allow time for another RST~~P~~ Bridge on the same LAN to synchronize its migration state with this port before the receipt of a BPDU can cause this port to change the BPDU types it transmits. Initialized to MigrateTime (13.26.6).

13.25.5 ~~13.23.5~~ rbWhile

The Recent Backup timer. Maintained at its initial value, twice HelloTime, while the port is a Backup Port.

13.25.6 ~~13.23.6~~ rcvdInfoWhile

The Received Info timer. The time remaining before information, i.e., portPriority (13.27.47) and portTimes (13.27.48), received in a Configuration Message is aged out if a further message is not received.

13.25.7 ~~13.23.7~~ rrWhile

The Recent Root timer.

13.25.8 ~~13.23.8~~ tcDetected

The Topology Change timer for MRP application usage. ‘New’ messages are sent while this timer is running (see 10.2).

13.25.9 ~~13.23.9~~ tcWhile

The Topology Change timer. TCN Messages are sent while this timer is running.

13.25.10 ~~13.23.10~~ pseudoInfoHelloWhen

The Pseudo Info Hello When timer. Used by the Layer 2 Gateway Port Receive (L2GPRX, 13.40) state machine to prompt the simulated reception at HelloTime intervals of a BPDU (see 13.29.11).

13.26 ~~13.24~~ Per bridge variables

The variables declared in this clause (13.26) are part of the specification. The accompanying descriptions are provided to aid in the comprehension of the protocol only, and are not part of the specification.

There is one instance per bridge component of the following variable(s):

a) ForceProtocolVersion (13.26.5)
b) TxHoldCount (13.26.12)
c) MigrateTime (13.26.6)

One instance of the following shall be implemented per bridge component if MSTP or ISIS-SPB is implemented:

d) MstConfigId (13.26.7)

The above parameters ((a) through (d)) are not modified by the operation of the spanning tree protocols, but are treated as constants by the state machines. If ForceProtocolVersion or MSTConfigId are modified by management, BEGIN shall be asserted for all state machines.

There is one instance per bridge of each of the following for the CIST and one for each MSTI.

e) BridgeIdentifier (13.26.2)
f) BridgePriority (13.26.3)
g) BridgeTimes (13.26.4)
h) rootPortId (13.26.9)
i) rootPriority (13.26.10)
j) rootTimes (13.26.11)

BridgeIdentifier, BridgePriority, and BridgeTimes are not modified by the operation of the spanning tree protocols but are treated as constants by the state machines. If they are modified by management, spanning tree priority vectors and Port Role assignments for all trees shall be recomputed, as specified by the operation of the Port Role Selection state machine (13.36) by clearing selected (13.27.67) and setting reselect (13.27.62) for all Bridge Ports for the relevant MSTI and for all trees if the CIST parameter is changed.

If the ISIS-SPB is implemented there is one instance per bridge component, of the following variable(s), with that single instance supporting all SPTs:

k) agreementDigest (13.26.1)
l) AuxMstConfigId (13.26.8)

13.26.1 agreementDigest

The Agreement Digest calculated by ISIS-SPB (28.4), and updated with other calculated parameters (see 13.36).

13.26.2 ~~13.24.1~~ BridgeIdentifier

The unique Bridge Identifier assigned to this bridge for this tree (CIST or MSTI).

The 12-bit system ID extension component of a Bridge Identifier (14.4) shall be set to zero for the CIST and to the value of the MSTID for an MSTI, thus allocating distinct Bridge Identifiers to the CIST and each MSTI—all based on the use of a single Bridge Address component value for the MST Bridge as a whole.

NOTE—This convention is used to convey the MSTID for each MSTI Configuration Message in an MST BPDU.

The four most significant bits of the Bridge Identifier (the settable Priority component) for the CIST and for each MSTI can be modified independently of the setting of those bits for all other trees, as a part of allowing full and independent configuration control to be exerted over each Spanning Tree instance.

13.26.3 ~~13.24.2~~ BridgePriority

For the CIST, the value of the CIST bridge priority vector, as defined in 13.10. The CIST Root Identifier, CIST Regional Root Identifier, and Designated Bridge Identifier components are all equal to the value of the CIST Bridge Identifier. The remaining components (External Root Path Cost, Internal Root Path Cost, and Designated Port Identifier) are set to zero.

For a given MSTI, the value of the MSTI bridge priority vector, as defined in 13.11. The MSTI Regional Root Identifier and Designated Bridge Identifier components are equal to the value of the MSTI Bridge Identifier (13.26.2). The remaining components (MSTI Internal Root Path Cost, Designated Port Identifier) are set to zero.

BridgePriority is used by updtRolesTree() in determining the value of the rootPriority variable (see 13.26.10).

13.26.4 ~~13.24.3~~ BridgeTimes

For the CIST, BridgeTimes comprises:

a) The current values of Bridge Forward Delay and Bridge Max Age (13.25, Table 13-5).
b) A Message Age value of zero.
c) The current value of Max Hops (13.25, Table 13-5). This parameter value is determined only by management.

For a given MSTI, BridgeTimes comprises:

d) The current value of MaxHops (Max Hops in Table 13-5), the initial value of remainingHops for MSTI information generated at the boundary of an MSTI region (see 13.26.11).

BridgeTimes is used by updtRolesTree() in determining the value of the rootTimes variable (13.26.11).

13.26.5 ~~13.24.4~~ ForceProtocolVersion

The Force Protocol Version parameter for the Bridge (13.7.2).

13.26.6 ~~13.24.5~~ MigrateTime

The value of the Migrate Time parameter as specified in Table 13-5. This value shall not be changed.

13.26.7 ~~13.24.6~~ MstConfigId

The current value of the bridge's MST Configuration Identifier (13.8). Changes in this parameter cause BEGIN to be asserted for the state machines for the bridge, for all trees, and for each port.

13.26.8 AuxMstConfigId

The previous value of the bridge's MST Configuration Identifier (13.8). This value is saved in nonvolatile memory when certain non-critical configuration is made if the bridge supports the ISIS-SPB non critical changes (27.4.1).

13.26.9 ~~13.24.7~~ rootPortId

For the CIST, the Port Identifier of the Root Port, and a component of the CIST root priority vector (13.10).

For a given MSTI, the Port Identifier of the Root Port, and a component of the MSTI root priority vector (13.11).

13.26.10 ~~13.24.8~~ rootPriority

For the CIST: the Root Identifier, External Root Path Cost, Regional Root Identifier, Internal Root Path Cost, Designated Bridge Identifier, and Designated Port Identifier components of the bridge's CIST root priority vector (13.10).

For a given MSTI: the MSTI Regional Root Identifier, Internal Root Path Cost, Designated Bridge Identifier, and Designated Port Identifier components of the bridge's MSTI root priority vector (13.11).

13.26.11 ~~13.24.9~~ rootTimes

For the CIST, the Bridge's timer parameter values (Message Age, Max Age, Forward Delay, and remainingHops). The values of these timers are derived (see 13.29.34) from the values stored in the CIST's portTimes parameter (13.27.48) for the Root Port or from BridgeTimes (13.26.4).

For a given MSTI, the value of remainingHops derived (13.29.34) from the value stored in the MSTI's portTimes parameter (13.27.48) for the Root Port or from BridgeTimes (13.26.4).

13.26.12 ~~13.24.10~~ TxHoldCount

The value of Transmit Hold Count (Table 13-5) for the bridge. If this is modified, the value of txCount (13.27) for all ports shall be set to zero.

13.27 ~~13.25~~ Per port variables

The variables declared in this clause (13.27) are part of the specification. The accompanying descriptions are provided to aid in the comprehension of the protocol only, and are not part of the specification.

There is one instance per port of each of the following variables:

a) AdminEdge (13.27.1)
b) ageingTime (13.27.2)
c) AutoEdge (13.27.18)
d) AutoIsolate (13.27.19)
e) enableBPDUrx (13.27.23)
f) enableBPDUtx (13.27.24)
g) ExternalPortPathCost (13.27.25)
h) ~~g)~~ isL2gp (13.27.26)
i) ~~h)~~ isolate (13.27.27)
j) ~~i)~~ mcheck (13.27.38)
k) ~~j)~~ newInfo (13.27.42)
l) ~~k)~~ operEdge (13.27.44)
m) ~~l)~~ portEnabled (13.27.45)
n) ~~m)~~ rcvdBpdu (13.27.52)
o) ~~n)~~ rcvdRSTP (13.27.56)
p) ~~o)~~ rcvdSTP (13.27.57)
q) ~~p)~~ rcvdTcAck (13.27.59)
r) ~~q)~~ rcvdTcn (13.27.60)
s) ~~r)~~ restrictedRole (13.27.64)
t) ~~s)~~ restrictedTcn (13.27.65)
u) ~~t)~~ sendRSTP (13.27.69)
v) ~~u)~~ tcAck (13.27.72)
w) ~~v)~~ tick (13.27.74)
x) ~~w)~~ txCount (13.27.75)

If MSTP or the ISIS-SPB is implemented, there is one instance per port, applicable to the CIST and to all MSTIs and SPTs, of the following variable(s):

y) ~~x)~~ rcvdInternal (13.27.54)
z) restrictedDomainRole (13.27.63)

If MSTP or the ISIS-SPB is implemented, there is one instance per port of each of the following variables for the CIST:

~~y) ExternalPortPathCost (13.27.25)~~
aa) ~~z)~~ infoInternal (13.27.31)
ab) ~~aa)~~ master (13.27.36)
ac) ~~ab)~~ mastered (13.27.37)

A single per port instance of the following variable(s) applies to all MSTIs:

ad) ~~ae)~~ newInfoMsti (13.27.43)

If the Layer 2 Gateway Port Receive state machine (13.20, 13.40) is implemented for a port, there is one instance for the CIST, and one instance for each MSTI, for that port of the following variable:

ae) ~~ad)~~ pseudoRootId (13.27.51)

If the ISIS-SPB is implemented, there is one instance per port of the following variable(s):

af) agreedMisorder (13.27.10)
ag) agreedN (13.27.11)

ah) agreedND (13.27.12)
ai) agreeN (13.27.16)
aj) agreeND (13.27.17)

If the ISIS-SPB is implemented, there is one instance per port of the following variable(s), with that single instance supporting all SPTs:

ak) agreedDigest (13.27.6)
al) agreedDigestValid (13.27.7)
am) agreeDigest (13.27.8)
an) agreeDigestValid (13.27.9)
ao) agreedTopology (13.27.14)

There is one instance per port of each of the following variables for the CIST, one per port for each MSTI and one per port for each SPT:

ap) ~~ae)~~ agree (13.27.3)
aq) ~~af)~~ agreed (13.27.4)
ar) ~~ag)~~ designatedPriority (13.27.20)
as) ~~ah)~~ designatedTimes (13.27.21)
at) ~~ai)~~ disputed (13.27.22)
au) ~~aj)~~ fdbFlush (13.27.28)
av) ~~ak)~~ forward (13.27.29)
aw) ~~al)~~ forwarding (13.27.30)
ax) ~~am)~~ infoIs (13.27.32)
ay) ~~an)~~ InternalPortPathCost (13.27.33)
az) ~~ao)~~ learn (13.27.34)
ba) ~~ap)~~ learning (13.27.35)
bb) ~~aq)~~ msgPriority (13.27.39)
bc) ~~ar)~~ msgTimes (13.27.40)
bd) ~~as)~~ portId (13.27.46)
be) ~~at)~~ portPriority (13.27.47)
bf) ~~au)~~ portTimes (13.27.48)
bg) ~~av)~~ proposed (13.27.49)
bh) ~~aw)~~ proposing (13.27.50)
bi) ~~ax)~~ rcvdInfo (13.27.53)
bj) ~~ay)~~ rcvdMsg (13.27.55)
bk) ~~az)~~ rcvdTc (13.27.58)
bl) ~~ba)~~ reRoot (13.27.61)
bm) ~~bb)~~ reselect (13.27.62)
bn) ~~bc)~~ role (13.27.66)
bo) ~~bd)~~ selected (13.27.67)
bp) ~~be)~~ selectedRole (13.27.68)
bq) ~~bf)~~ sync (13.27.70)
br) ~~bg)~~ synced (13.27.71)
bs) ~~bh)~~ tcProp (13.27.73)
bt) ~~bi)~~ updtInfo (13.27.76)

If the ISIS-SPB is implemented, there is one instance per port of the following variable(s) for the CIST and one per port for each SPT:

bu) agreedPriority (13.27.13)
bv) agreementOutstanding (13.27.15)

If the ISIS-SPB is implemented, there is one instance per port of the following variables for each SPT:

bw) agreedAbove (13.27.5)
bx) neighbourPriority (13.27.41)

13.27.1 ~~13.25.1~~ AdminEdge

A Boolean. Set by management if the port is to be identified as operEdge immediately on initialization, without a delay to detect other bridges attached to the LAN. The recommended default value is FALSE.

13.27.2 ~~13.25.2~~ ageingTime

Filtering database entries for this port are aged out after ageingTime has elapsed since they were first created or refreshed by the Learning Process. The value of this parameter is normally Ageing Time (8.8.3, Table 8-6), and is changed to FwdDelay (13.28.10) for a period of FwdDelay after fdbFlush (13.27.28) is set by the topology change state machine if stpVersion (13.28.23) is TRUE.

13.27.3 ~~13.25.3~~ agree

A Boolean. See 13.16 and Figure 13-12.

13.27.4 ~~13.25.4~~ agreed

A Boolean. Set, for the CIST or an MSTI, if an Agreement has been received indicating that the Port States and transmitted priority vectors for the other bridge attached to this LAN are (and, in the absence of further communication with this bridge and within the design probabilities of protocol failure due to repeated BPDU loss, will remain) compatible with a loop-free active topology determined by this port's priority vectors (13.16, 13.24).

13.27.5 agreedAbove

A Boolean controlled by the updtAgreement() procedure to indicate agreement on the neighbor(s) superior priority vector.

13.27.6 agreedDigest

The Agreement Digest conveyed in the last BPDU received, if the port is internal to an SPT Region.

13.27.7 agreedDigestValid

A Boolean. TRUE if the Agreement Valid flag was set in the last BPDU or SPB Hello PDU that conveyed an Agreement Digest.

13.27.8 agreeDigest

The Agreement Digest currently encoded in SPT BPDUs transmitted from the port.

13.27.9 agreeDigestValid

A Boolean. TRUE if the Agreement Valid flag is to be set in transmitted SPT BPDUs and SPB Hello PDUs.

13.27.10 agreedMisorder

A Boolean. TRUE if the Agreement Protocol (13.16, 13.17) has received an out of order message as indicated by the received Agreement Number (13.27.16), reset when an in order message is received.

13.27.11 agreedN

The Agreement Number associated with the last BPDU received, if the port is within an SPT Region. It takes values 0, 1, 2, or 3, and all arithmetic using this variable is modulo 4, e.g., 1 + 1 = 2, 3 + 1 = 0, 0 – 1 = 3.

13.27.12 agreedND

The Discarded Agreement Number (see 13.17) to be included in SPT BPDUs transmitted from the port. The variable values 0, 1, 2, or 3, are encoded in a two-bit field and all arithmetic using this variable is modulo 4.

13.27.13 agreedPriority

The priority vector associated with the worst priority received by a CIST or SPT Designated Port from an SPT Region in SPT BPDUs that are sent by Root Ports, zero if there is no such agreement.

13.27.14 agreedTopology

A Boolean. TRUE if the Agreement Protocol has declared a topology match for agreeDigest (13.27.8), and agreeDigest has the same value as the last calculated agreementDigest (13.26.1). Reset to FALSE whenever agreementDigest is recalculated by ISIS-SPB, and set TRUE by updtDigest().

13.27.15 agreementOutstanding

The priority vector for the worst priority associated with outstanding Agreements made by a CIST or SPT Root Port, Alternate Port, or Backup Port within an SPT Region. in SPT BPDUs. When an SPT BPDU, from the same SPT Region, is received with a Discarded Agreement Number that is the immediate precursor of the current agreeN (13.27.16), the CIST's agreementOutstanding is set to the port's designatedPriority. On reception of a BPDU that is either not an SPT BPDU or is from a different SPT Region, the agreeND (13.27.17) is set to the value of the agreeN (13.27.16), so there is no outstanding Agreement.

13.27.16 agreeN

The Agreement Number (see 13.17) transmitted in SPT BPDUs. The variable values 0, 1, 2, or 3, are encoded in a two-bit field and arithmetic using this variable is modulo 4.

13.27.17 agreeND

The last Discarded Agreement Number (see 13.17) received in an SPT BPDU. It takes values 0, 1, 2, or 3, and arithmetic using this variable is modulo 4. If agreeND matches agreeN, there is no outstanding agreement.

13.27.18 ~~13.25.5~~ AutoEdge

A Boolean. Set by management if the bridge detection state machine (BDM, 13.33) is to detect other bridges attached to the LAN, and set operEdge automatically. The recommended default is TRUE.

13.27.19 ~~13.25.6~~ AutoIsolate

A Boolean. Set by management if isolate (13.27.27) is to be set, causing a Designated Port to transition to Discarding if both AdminEdge (13.27.1) and AutoEdge (13.27.18) are FALSE and the other bridge presumed to be attached to the same point-to-point LAN does not transmit periodic BPDUs—either as a Designated Port or in response to BPDUs with the Proposal flag set (see 13.23). The recommended default of this parameter is FALSE. AdminEdge and AutoEdge are both reset only on ports that are known to connect to other bridges.

13.27.20 ~~13.25.7~~ designatedPriority

For the CIST and a given port, the CIST Root Identifier, External Root Path Cost, Regional Root Identifier, Internal Root Path Cost, Designated Bridge Identifier, and Designated Port Identifier components of the port's CIST designated priority vector, as defined in 13.10.

For a given MSTI and port, the Regional Root Identifier, Internal Root Path Cost, Designated Bridge Identifier, and Designated Port Identifier components of the designated priority vector, as defined in 13.11.

13.27.21 ~~13.25.8~~ designatedTimes

For the CIST and a given port, the set of timer parameter values (Message Age, Max Age, Forward Delay, and remainingHops) that are used to update Port Times when updtInfo is set. These timer parameter values are used in BPDUs transmitted from the port. The value of designatedTimes is copied from the CIST rootTimes Parameter (13.26.11) by the operation of the updtRolesTree() procedure.

For a given MSTI and port, the value of remainingHops used to update this MSTI's portTimes parameter when updtInfo is set. This timer parameter value is used in BPDUs transmitted from the port. The updtRolesTree() procedure (13.29.34) copies designatedTimes from the MSTI's rootTimes (13.26.11).

13.27.22 ~~13.25.9~~ disputed

A Boolean. See 13.21, 13.20, Figure 13-20, 13.29.16, Figure 13-24, and Figure 13-25.

13.27.23 ~~13.25.10~~ enableBPDUrx

A Boolean. This per port management parameter is set by default, and should not be clear unless the port is configured as a Layer Two Gateway Port (i.e., isL2gp is set). When clear it can allow loops to be created, or can result in no connectivity. When cleared, BPDUs received on the port are discarded and not processed.

13.27.24 ~~13.25.11~~ enableBPDUtx

A Boolean. This per port management parameter is set by default, and should not be clear unless the port is configured as a Layer Two Gateway Port (i.e., isL2gp is set). When clear it can allow loops to be created, or can result in no connectivity. When cleared, no BPDUs are transmitted by this port.

13.27.25 ~~13.25.12~~ ExternalPortPathCost

The port's contribution, when it is the CIST Master Port (for MSTP and ISIS-SPB) or the CST Root Port (for RSTP and MSTP), to the External Root Path Cost (13.9) for the bridge.

13.27.26 ~~13.25.19~~ isL2gp

A Boolean. Set by management to identify a port configured to be a Layer Two Gateway Port. This parameter is set to FALSE by default. When set, enableBPDUtx should be cleared.

13.27.27 ~~13.25.20~~ isolate

A Boolean. Set by the bridge detection state machine (BDM, 13.33) when the Spanning Tree Protocol Entity of a neighboring bridge has apparently failed (see 13.23, 13.27.19).

13.27.28 ~~13.25.13~~ fdbFlush

A Boolean. Set by the topology change state machine to instruct the filtering database to remove entries for this port, immediately if rstpVersion (13.28.21) is TRUE, or by rapid ageing (13.27.2) if stpVersion (13.28.23) is TRUE. Reset by the filtering database once the entries are removed if rstpVersion is TRUE, and immediately if stpVersion is TRUE. Setting the fdbFlush variable does not result in removal of filtering database entries in the case that the port is an Edge Port (i.e., operEdge is TRUE). The filtering database removes entries only for those VIDs that have a fixed registration (see 10.7.2) on any port of the bridge that is not an Edge Port.

NOTE—If MVRP or MIRP is in use, the topology change notification and flushing mechanisms defined in MRP (Clause 10) and MVRP (11.2.5) are responsible for filtering entries in the Filtering Database for VIDs that are dynamically registered using MVRP or MIRP (i.e., for which there is no fixed registration in the bridge on non-Edge Ports).

13.27.29 ~~13.25.14~~ forward

A Boolean. Set or cleared by the Port Role Transitions state machine (13.37) to instruct the Port State Transitions state machine (13.38) to enable or disable forwarding.

13.27.30 ~~13.25.15~~ forwarding

A Boolean. Set or cleared by the Port State Transitions state machine (13.38) to indicate that forwarding has been enabled or disabled.

13.27.31 ~~13.25.16~~ infoInternal

If infoIs is Received, indicating that the port has received current information from the Designated Bridge for the attached LAN, infoInternal is set if that Designated Bridge is in the same MST Region as the receiving bridge and reset otherwise.

13.27.32 ~~13.25.17~~ infoIs

A variable that takes the values Mine, Aged, Received, or Disabled, to indicate the origin/state of the port’s Spanning Tree information (portInfo) held for the port, as follows:

a) If infoIs is Received, the port has received current (not aged out) information from the Designated Bridge for the attached LAN (a point-to-point bridge link being a special case of a LAN).
b) If infoIs is Mine, information for the port has been derived from the Root Port for the Bridge (with the addition of root port cost information). This includes the possibility that the Root Port is “Port 0,” i.e., the bridge is the Root Bridge for the Bridged Local Area Network.
c) If infoIs is Aged, information from the Root Bridge has been aged out. Just as for reselect (13.27.62), the state machine does not formally allow the Aged state to persist. However, if there is a delay in recomputing the new root port, correct processing of a received BPDU is specified.
d) Finally if the port is disabled, infoIs is Disabled.

13.27.33 ~~13.25.18~~ InternalPortPathCost

The port’s contribution, when it is an IST or SPT Root Port (for MSTP and ISIS-SPB) to the Internal Root Path Cost (13.9) for the bridge.

13.27.34 ~~13.25.21~~ learn

A Boolean. Set or cleared by the Port Role Transitions state machine (13.37) to instruct the Port State Transitions state machine (13.38) to enable or disable learning.

13.27.35 ~~13.25.22~~ learning

A Boolean. Set or cleared by the Port State Transitions state machine (13.38) to indicate that learning has been enabled or disabled.

13.27.36 ~~13.25.23~~ master

A Boolean. Used to determine the value of the Master flag for a given MSTI and port in transmitted MST BPDUs.

Set TRUE if the Port Role for the MSTI and Port is Root Port or Designated Port, and the bridge has selected one of its ports as the Master Port for this MSTI or the mastered flag is set for this MSTI for any other Bridge Port with a Root Port or Designated Port Role. Set FALSE otherwise.

13.27.37 ~~13.25.24~~ mastered

A Boolean. Used to record the value of the Master flag for a given MSTI and port in MST BPDUs received from the attached LAN.

NOTE—master and mastered signal the connection of the MSTI to the CST via the Master Port throughout the MSTI. These variables and their supporting procedures do not affect the connectivity provided by this standard but permit future enhancements to MSTP providing increased flexibility in the choice of Master Port without abandoning plug-and-play network migration. They are, therefore, omitted from the overviews of protocol operation, including Figure 13-13.

13.27.38 ~~13.25.25~~ mcheck

A Boolean. May be set by management to force the Port Protocol Migration state machine to transmit RST (or MST or SPT) BPDUs for a MigrateTime (13.26.6, Table 13-5) period, to test whether all STP Bridges on the attached LAN have been removed and the port can continue to transmit RST~~P~~ BPDUs. Setting mcheck has no effect if stpVersion (13.28.23) is TRUE.

13.27.39 ~~13.25.26~~ msgPriority

For the CIST and a given port, the CIST Root Identifier, External Root Path Cost, Regional Root Identifier, Internal Root Path Cost, Designated Bridge Identifier, and Designated Port Identifier components of the CIST message priority vector conveyed in a received BPDU, as defined in 13.10.

For a given MSTI and port, the Regional Root Identifier, Internal Root Path Cost, Designated Bridge Identifier, and Designated Port Identifier components of the MSTI message priority vector, as defined in 13.11 and conveyed in a received BPDU for this MSTI.

13.27.40 ~~13.25.27~~ msgTimes

For the CIST and a given port, the timer parameter values (Message Age, Max Age, Forward Delay, Hello Time, and remainingHops) conveyed in a received BPDU. If the BPDU is an STP or RST BPDU without MSTP parameters, remainingHops is set to the value of the MaxHops component of BridgeTimes (13.26.4).

For a given MSTI and port, the value of remainingHops received in the same BPDU as the message priority components of this MSTI's msgPriority parameter.

13.27.41 neighbourPriority

For a port attached to a point-to-point LAN within the same SPT Region, the designated priority of the other bridge attached to that LAN as calculated by ISIS-SPB for a given SPT. Receipt of an Agreement Digest matching the bridge component's agreementDigest implies receipt of an Agreement, with agreedPriority == neighbourPriority, for each SPT for which the port's selectedRole is DesignatedPort.

If the port is attached to a shared media LAN that is within the same SPT Region, i.e., all bridges attached to that LAN are within the Region, then neighbourPriority is the best (13.9) of all the neighbor's calculated designated priority vectors, and an SPT BPDU with a matching digest has to be received from all the neighbors before agreedPriority is updated.

13.27.42 ~~13.25.28~~ newInfo

A Boolean. Set TRUE if a BPDU conveying changed CIST information is to be transmitted. It is set FALSE by the Port Transmit state machine.

13.27.43 ~~13.25.29~~ newInfoMsti

A Boolean. Set TRUE if a BPDU conveying changed MSTI information is to be transmitted. It is set FALSE by the Port Transmit state machine.

13.27.44 ~~13.25.30~~ operEdge

A Boolean. The value of the operEdgePort parameter [item l) in 12.18.2.1.3], as determined by the operation of the Bridge Detection state machine (13.33).

13.27.45 ~~13.25.31~~ portEnabled

A Boolean. Set if the Bridge's MAC Relay Entity and Spanning Tree Protocol Entity can use the MAC Service provided by the Bridge Port's MAC entity to transmit and receive frames to and from the attached LAN, i.e., portEnabled is TRUE if and only if:

a) MAC_Operational (6.6.2) is TRUE; and
b) Administrative Bridge Port State (8.4, 13.12) for the port is Enabled.

13.27.46 ~~13.25.32~~ portId

The Port Identifier for this port. This variable forms a component of the port priority and designated priority vectors (13.10, 13.11).

The four most significant bits of the Port Identifier (the settable Priority component) for the CIST and for each MSTI can be modified independently of the setting of those bits for all other trees, as a part of allowing full and independent configuration control to be exerted over each Spanning Tree instance.

13.27.47 ~~13.25.33~~ portPriority

For the CIST and a given port, the CIST Root Identifier, External Root Path Cost, Regional Root Identifier, Internal Root Path Cost, Designated Bridge Identifier, and Designated Port Identifier components of the port's port priority vector (13.10).

For a given MSTI and port, the Regional Root Identifier, Internal Root Path Cost, Designated Bridge Identifier, and Designated Port Identifier components of the port's MSTI port priority vector (13.11).

13.27.48 ~~13.25.34~~ portTimes

For the CIST and a given port, the port's timer parameter values (Message Age, Max Age, Forward Delay, Hello Time, and remainingHops). The Hello Time timer parameter value is used in transmitted BPDUs.

For a given MSTI and port, the value of remainingHops for this MSTI in transmitted BPDUs.

13.27.49 ~~13.25.35~~ proposed

A Boolean. See 13.16, Figure 13-10, Figure 13-12, 13.29.14, 13.29.19, 13.37, and Figure 13-24.

13.27.50 ~~13.25.36~~ proposing

A Boolean. See 13.16, Figure 13-10, Figure 13-12, 13.29.14, 13.29.15, 13.33, and Figure 13-25.

13.27.51 ~~13.25.37~~ pseudoRootId

A Bridge Identifier configured by management for L2GP operation. By default, it is set to the BridgeIdentifier (13.26.2).

13.27.52 ~~13.25.38~~ rcvdBPDU

A Boolean. Set by system dependent processes, this variable notifies the Port Receive state machine (13.31) when a valid (Clause 14) Configuration, TCN, RST, ~~or~~ MST, or SPT BPDU (14.3) is received on the port. Reset by the Port Receive state machine.

13.27.53 ~~13.25.39~~ rcvdInfo

Set to the result of the rcvInfo() procedure (13.29.12).

13.27.54 ~~13.25.40~~ rcvdInternal

A Boolean. Set TRUE by the Receive Machine if the BPDU received was transmitted by a bridge in the same MST Region as the receiving bridge.

13.27.55 ~~13.25.41~~ rcvdMsg

A Boolean. See 13.29.13 and 13.31.

13.27.56 ~~13.25.42~~ rcvdRSTP

A Boolean. See 13.31, 13.29.31, and 13.32.

13.27.57 ~~13.25.43~~ rcvdSTP

A Boolean. See 13.31, 13.29.31, and 13.32.

13.27.58 ~~13.25.44~~ rcvdTc

A Boolean. See 13.29.13, 13.29.24, and 13.39.

13.27.59 ~~13.25.45~~ rcvdTcAck

A Boolean. See 13.29.24 and 13.39.

13.27.60 ~~13.25.46~~ rcvdTcn

A Boolean. See 13.29.13, 13.29.24, and 13.39.

13.27.61 ~~13.25.47~~ reRoot

A Boolean. Set by a newly selected Root Port to force prior Root Ports to Discarding, before it becomes forwarding. See Figure 13-24, 13.29.21, Figure 13-23, Figure 13-25, and Figure 13-26.

13.27.62 ~~13.25.48~~ reselect

A Boolean. Set to prompt recomputation of the CIST or an MSTI. See 13.35 and 13.36.

13.27.63 restrictedDomainRole

A Boolean. Set by management. If TRUE, causes a port that is a Boundary Port of an SPT Region not to be selected as Root Port for the CIST, any MSTI, or any SPT, even it has the best spanning tree priority vector. See 13.20, 27.7, and restrictedRole (13.27.64).

13.27.64 ~~13.25.49~~ restrictedRole

A Boolean. Set by management. If TRUE causes the port not to be selected as Root Port for the CIST or any MSTI, even it has the best spanning tree priority vector. Such a port will be selected as an Alternate Port after the Root Port has been selected. This parameter should be FALSE by default. If set, it can cause lack of spanning tree connectivity. It is set by a network administrator to prevent bridges external to a core region of the network influencing the spanning tree active topology, possibly because those bridges are not under the full control of the administrator.

13.27.65 ~~13.25.50~~ restrictedTcn

A Boolean. Set by management. If TRUE causes the port not to propagate received topology change notifications and topology changes to other ports. This parameter should be FALSE by default. If set it can cause temporary loss of connectivity after changes in a spanning trees active topology as a result of persistent incorrectly learned station location information. It is set by a network administrator to prevent bridges external to a core region of the network, causing address flushing in that region, possibly because those bridges are not under the full control of the administrator or MAC_Operational for the attached LANs transitions frequently.

13.27.66 ~~13.25.51~~ role

The current Port Role. DisabledPort, RootPort, DesignatedPort, AlternatePort, BackupPort, or MasterPort.

NOTE—The MasterPort role applies to each MSTI when the CIST Port Role is RootPort and connects to another MST Region. An MSTI Master Port is part of the stable active topology for frames assigned to that MSTI, just as the CIST Root Port forwards frames for the IST. The Port State for each MSTI may differ as required to suppress temporary loops.

13.27.67 ~~13.25.52~~ selected

A Boolean. See 13.36 and 13.29.22.

13.27.68 ~~13.25.53~~ selectedRole

A newly computed role for the port.

13.27.69 ~~13.25.54~~ sendRSTP

A Boolean. See 13.32 and 13.34.

13.27.70 ~~13.25.55~~ sync

A Boolean. Set to force the Port State to be compatible with the loop-free active topology determined by the priority vectors held by this bridge (13.16, 13.24) for this tree (CIST or MSTI), transitioning the Port State to Discarding, and soliciting an Agreement if possible, if the port is not already synchronized (13.27.71).

13.27.71 ~~13.25.56~~ synced

A Boolean. TRUE only if the Port State is compatible with the loop-free active topology determined by the priority vectors held by this bridge for this tree (13.16, 13.24).

13.27.72 ~~13.25.57~~ tcAck

A Boolean. Set to transmit a Configuration Message with a topology change acknowledge flag set.

13.27.73 ~~13.25.58~~ tcProp

A Boolean. Set by the Topology Change state machine of any other port, to indicate that a topology change should be propagated through this port.

13.27.74 ~~13.25.59~~ tick

A Boolean. See the Port Timers state machine (13.30).

13.27.75 ~~13.25.60~~ txCount

A counter. Incremented by the Port Transmission (13.34) state machine on every BPDU transmission, and decremented used by the Port Timers state machine (13.30) once a second. Transmissions are delayed if txCount reaches TxHoldCount (13.26.12).

13.27.76 ~~13.25.61~~ updtInfo

A boolean. Set by the Port Role Selection state machine (13.36, 13.29.34) to tell the Port Information state machine that it should copy designatedPriority to portPriority and designatedTimes to portTimes.

13.28 ~~13.26~~ State machine conditions and parameters

The following variable evaluations are defined for notational convenience in the state machines:

a) allSptAgree (13.28.1)
b) ~~a)~~ allSynced (13.28.2)
c) ~~b)~~ allTransmitReady (13.28.3)
d) BestAgreementPriority (13.28.4)
e) ~~c)~~ cist (13.28.5)
f) ~~d)~~ cistRootPort (13.28.6)
g) ~~e)~~ cistDesignatedPort (13.28.7)
h) ~~f)~~ EdgeDelay (13.28.8)
i) ~~g)~~ forwardDelay (13.28.9)
j) ~~h)~~ FwdDelay (13.28.10)
k) ~~i)~~ HelloTime (13.28.11)
l) ~~j)~~ MaxAge (13.28.12)
m) ~~k)~~ msti (13.28.13)
n) ~~l)~~ mstiDesignatedOrTCpropagatingRootPort (13.28.14)
o) ~~m)~~ mstiMasterPort (13.28.15)
p) ~~n)~~ operPointToPointMAC (13.28.16)
q) ~~o)~~ rcvdAnyMsg (13.28.17)
r) ~~p)~~ rcvdCistMsg (13.28.18)
s) ~~q)~~ rcvdMstiMsg (13.28.19)
t) ~~r)~~ reRooted (13.28.20)
u) ~~s)~~ rstpVersion (13.28.21)
v) spt (13.28.22)
w) ~~t)~~ stpVersion (13.28.23)
x) ~~u)~~ updtCistInfo (13.28.24)
y) ~~v)~~ updtMstiInfo (13.28.25)

13.28.1 allSptAgree

TRUE, if and only if, agree is TRUE for the given port for all SPTs.

13.28.2 ~~13.26.1~~ allSynced

The condition allSynced is TRUE for a given port, for a given tree, if and only if

a) For all ports for the given tree, selected is TRUE, the port’s role is the same as its selectedRole, and updtInfo is FALSE; and
b) The role of the given port is
   1) Root Port or Alternate Port and synced is TRUE for all ports for the given tree other than the Root Port; or
   2) Designated Port and synced is TRUE for all ports for the given tree other than the given port; or
   3) Designated Port, and the tree is an SPT or the IST, and the Root Port of the tree and the given port are both within the bridge’s SPT Region, and both learning and forwarding are FALSE for the given port; or
   4) ~~3)~~ Master Port and synced is TRUE for all ports for the given tree other than the given port.

13.28.3 ~~13.26.2~~ allTransmitReady

TRUE, if and only if, for the given port for all trees

a) selected is TRUE; and
b) updtInfo is FALSE.

13.28.4 BestAgreementPriority

The best of all possible priority vectors, i.e., numerically the lowest.

13.28.5 ~~13.26.3~~ cist

TRUE only for CIST state machines; i.e., FALSE for MSTI or SPT state machine instances.

13.28.6 ~~13.26.4~~ cistRootPort

TRUE if the CIST role for the given port is RootPort.

13.28.7 ~~13.26.5~~ cistDesignatedPort

TRUE if the CIST role for the given port is DesignatedPort.

13.28.8 ~~13.26.6~~ EdgeDelay

Returns the value of MigrateTime if operPointToPointMAC is TRUE, and the value of MaxAge otherwise.

13.28.9 ~~13.26.7~~ forwardDelay

Returns the value of HelloTime if sendRSTP is TRUE, and the value of FwdDelay otherwise.

13.28.10 ~~13.26.8~~ FwdDelay

The Forward Delay component of the CIST's designatedTimes parameter (13.27.21).

13.28.11 ~~13.26.9~~ HelloTime

The Hello Time component of the CIST's portTimes parameter (13.27.48) with the recommended default value given in Table 13-5.

13.28.12 ~~13.26.11~~ MaxAge

The Max Age component of the CIST's designatedTimes parameter (13.27.21).

13.28.13 ~~13.26.10~~ msti

TRUE only for MSTI state machines; i.e., FALSE for CIST or SPT state machine instances.

13.28.14 ~~13.26.12~~ mstiDesignatedOrTCpropagatingRootPort

TRUE if the role for any MSTI for the given port is either:

a) DesignatedPort; or
b) RootPort, and the instance for the given MSTI and port of the tcWhile timer is not zero.

13.28.15 ~~13.26.13~~ mstiMasterPort

TRUE if the role for any MSTI for the given port is MasterPort.

13.28.16 ~~13.26.14~~ operPointToPoint

TRUE if operPointToPointMAC (6.6.3) is TRUE for the Bridge Port.

13.28.17 ~~13.26.15~~ rcvdAnyMsg

TRUE for a given port if rcvdMsg is TRUE for the CIST or any MSTI for that port.

13.28.18 ~~13.26.16~~ rcvdCistMsg

TRUE for a given port if and only if rcvdMsg is TRUE for the CIST for that port.

13.28.19 ~~13.26.17~~ rcvdMstiMsg

TRUE for a given port and MSTI if and only if rcvdMsg is FALSE for the CIST for that port and rcvdMsg is TRUE for the MSTI for that port.

13.28.20 ~~13.26.18~~ reRooted

TRUE if the rrWhile timer is clear (zero) for all Ports for the given tree other than the given Port.

13.28.21 ~~13.26.19~~ rstpVersion

TRUE if ForceProtocolVersion (13.7.2) is greater than or equal to 2.

13.28.22 spt

TRUE only for SPT state machines, in an SPT Bridge; i.e., FALSE for CIST and MSTI state machine instances.

13.28.23 ~~13.26.20~~ stpVersion

TRUE if Force Protocol Version (13.7.2) is less than 2.

13.28.24 ~~13.26.21~~ updtCistInfo

TRUE for a given port if and only if updtInfo is TRUE for the CIST for that port.

13.28.25 ~~13.26.22~~ updtMstiInfo

TRUE for a given port and MSTI if and only if updtInfo is TRUE for the MSTI for that port or updtInfo is TRUE for the CIST for that port.

NOTE—The dependency of rcvdMstiMsg and updtMstiInfo on CIST variables for the port reflects the fact that MSTIs exist in a context of CST parameters. The state machines ensure that the CIST parameters from received BPDUs are processed and updated prior to processing MSTI information.

13.29 ~~13.27~~ State machine procedures

The following procedures perform the functions specified for both the state machines for all trees, or specifically for the CIST, ~~or~~ a given MSTI, or a given SPT:

a) betterorsameInfo(newInfoIs) (13.29.1)
b) clearAllRcvdMsgs() (13.29.2)

c) clearReselectTree() (13.29.3)
d) disableForwarding() (13.29.4)
e) disableLearning() (13.29.5)
f) enableForwarding() (13.29.6)
g) enableLearning() (13.29.7)
h) fromSameRegion() (13.29.8)
i) newTcDetected() (13.29.9)
j) newTcWhile() (13.29.10)
k) pseudoRcvMsgs() (13.29.11)
l) rcvInfo() (13.29.12)
m) rcvMsgs() (13.29.13)
n) recordAgreement() (13.29.15)
o) recordDispute() (13.29.16)
p) recordMastered() (13.29.17)
q) recordPriority() (13.29.18)
r) recordProposal() (13.29.19)
s) recordTimes() (13.29.20)
t) setReRootTree() (13.29.21)
u) setSelectedTree() (13.29.22)
v) setSyncTree() (13.29.23)
w) setTcFlags() (13.29.24)
x) setTcPropTree() (13.29.25)
y) syncMaster() (13.29.26)
z) txConfig() (13.29.27)
aa) txRstp() (13.29.28)
ab) txTcn() (13.29.29)
ac) updtBPDUVersion() (13.29.31)
ad) updtRcvdInfoWhile() (13.29.33)
ae) updtRolesTree() (13.29.34)
af) updtRolesDisabledTree() (13.29.35)

The following procedures perform the functions specified for all SPTs, or for a given SPT or the CIST when ISIS-SPB is implemented:

ag) rcvAgreements() (13.29.14)
ah) updtAgreement() (13.29.30)
ai) updtDigest() (13.29.32)

All references to named variables in the specification of procedures are to instances of the variables corresponding to the instance of the state machine using the function, i.e., to the CIST or the given MSTI or the given SPT as appropriate. References to forwarding and learning apply to frames assigned to the specified tree.

13.29.1 ~~13.27.1~~ betterorsameInfo(newInfoIs)

Returns TRUE if, for a given port and tree (CIST or MSTI), either

a) The procedure’s parameter newInfoIs is Received, and infoIs is Received and the msgPriority vector is better than or the same as (13.10) the portPriority vector; or,
b) The procedure’s parameter newInfoIs is Mine, and infoIs is Mine and the designatedPriority vector is better than or the same as (13.10) the portPriority vector.

Returns False otherwise.

NOTE—This procedure is not invoked (in the case of a MSTI) if the received BPDU carrying the MSTI information was received from another MST Region. In that event, the Port Receive Machine (using rcvMsgs()) does not set rcvdMsg for any MSTI, and the Port Information Machine's SUPERIOR_DESIGNATED state is not entered.

13.29.2 ~~13.27.2~~ clearAllRcvdMsgs()

Clears rcvdMsg for the CIST and all MSTIs, for this port.

13.29.3 ~~13.27.3~~ clearReselectTree()

Clears reselect for the tree (the CIST or a given MSTI) for all ports of the bridge.

13.29.4 ~~13.27.4~~ disableForwarding()

An implementation dependent procedure that causes the Forwarding Process (8.6) to stop forwarding frames through the port. The procedure does not complete until forwarding has stopped.

13.29.5 ~~13.27.5~~ disableLearning()

An implementation dependent procedure that causes the Learning Process (8.7) to stop learning from the source address of frames received on the port. The procedure does not complete until learning has stopped.

13.29.6 ~~13.27.6~~ enableForwarding()

An implementation dependent procedure that causes the Forwarding Process (8.6) to start forwarding frames through the port. The procedure does not complete until forwarding has been enabled.

13.29.7 ~~13.27.7~~ enableLearning()

An implementation dependent procedure that causes the Learning Process (8.7) to start learning from frames received on the port. The procedure does not complete until learning has been enabled.

13.29.8 ~~13.27.8~~ fromSameRegion()

Returns TRUE if rcvdRSTP is TRUE, and the received BPDU conveys a MST Configuration Identifier that matches that held for the bridge. Returns FALSE otherwise. If SPB protocols are implemented, fromSameRegion() returns TRUE if the bridge detects that either its MCID (13.26.7) or its Auxiliary MCID (13.26.8) match either the MST Configuration Identifier or Auxiliary MST Configuration Identifier carried by the received SPT BPDU.

13.29.9 ~~13.27.9~~ newTcDetected()

If the value of tcDetected is zero and sendRSTP is TRUE, this procedure sets the value of tcDetected to HelloTime plus one second. The value of HelloTime is taken from the CIST's portTimes parameter (13.27.48) for this port.

If the value of tcDetected is zero and sendRSTP is FALSE, this procedure sets the value of tcDetected to the sum of the Max Age and Forward Delay components of rootTimes.

Otherwise the procedure takes no action.

13.29.10 ~~13.27.10~~ newTcWhile()

If the value of tcWhile is zero and sendRSTP is TRUE, this procedure sets the value of tcWhile to HelloTime plus one second and sets either newInfo TRUE for the CIST or newInfoMsti TRUE for a given MSTI. The value of HelloTime is taken from the CIST's portTimes parameter (13.27.48) for this port.

If the value of tcWhile is zero and sendRSTP is FALSE, this procedure sets the value of tcWhile to the sum of the Max Age and Forward Delay components of rootTimes and does not change the value of either newInfo or newInfoMsti.

Otherwise the procedure takes no action.

13.29.11 ~~13.27.11~~ pseudoRcvMsgs()

Using local parameters, this procedure simulates the processing that would be applied by rcvInfo() and rcvMsgs() to a BPDU received on the port, from the same region and with the following parameters:

a) Message Age, Max Age, Hello Time and Forward Delay are derived from BridgeTimes (13.26.4);
b) The CIST information carries the message priority vector (13.10) with a value of {pseudoRootId, 0, pseudoRootId, 0, 0, 0};
c) A CIST Port Role of Designated Port, with the Learning and Forwarding flags set;
d) The Version 1 Length is 0 and Version 3 Length calculated appropriately;
e) For each MSTI configured on the bridge, the corresponding MSTI Configuration Message carries:
    1) A message priority vector with a value of {pseudoRootId, 0, 0, 0};
    2) A Port Role of Designated Port, with the Learning and Forwarding flags set;
    3) MSTI Remaining Hops set to the value of the MaxHops component of BridgeTimes (13.26.4).

NOTE—If two L2GP ports are configured with the same CIST pseudoRootId then the IST may partition within the MST Region, but either of the L2GP ports can be selected to provide connectivity from the Region/customer network to a provider's network on an MSTI by MSTI basis.

13.29.12 ~~13.27.12~~ rcvInfo()

Returns SuperiorDesignatedInfo if, for a given port and tree (CIST or MSTI):

a) The received CIST or MSTI message conveys a Designated Port Role, and
    1) The message priority (msgPriority—13.27.39) is superior (13.10 or 13.11) to the port's port priority vector, or
    2) The message priority is the same as the port's port priority vector, and any of the received timer parameter values (msgTimes—13.27.40) differ from those already held for the port (portTimes—13.27.48).

Otherwise, returns RepeatedDesignatedInfo if, for a given port and tree (CIST or MSTI):

b) The received CIST or MSTI message conveys a Designated Port Role, and
    1) A message priority vector and timer parameters that are the same as the port's port priority vector and timer values; and
    2) infoIs is Received.

Otherwise, returns InferiorDesignatedInfo if, for a given port and tree (CIST or MSTI):

c) The received CIST or MSTI message conveys a Designated Port Role.

Otherwise, returns InferiorRootAlternateInfo if, for a given port and tree (CIST or MSTI):

d) The received CIST or MSTI message conveys a Root Port, Alternate Port, or Backup Port Role and a CIST or MSTI message priority that is the same as or worse than the CIST or MSTI port priority vector.

Otherwise, returns OtherInfo.

NOTE—A Configuration BPDU implicitly conveys a Designated Port Role.

13.29.13 ~~13.27.13~~ rcvMsgs()

This procedure is invoked by the Port Receive state machine (13.31) to decode a received BPDU. Sets rcvdTcn and rcvdTc for each and every MSTI if a TCN BPDU has been received, and extracts the message priority and timer values from the received BPDU storing them in the msgPriority and msgTimes variables.

If ISIS-SPB is implemented, ForceProtocolVersion is 4 (or greater), the BPDU is an SPT BPDU, and has been received on a Bridge Port that is internal to the SPT Region (i.e., is not a Boundary Port, see 13.12), then the rcvAgreements() procedure processes the CIST and SPT information conveyed by the BPDU.

Otherwise (i.e., if rcvAgreements() is not used) this ~~The~~ procedure sets rcvdMsg for the CIST and makes the received CST or CIST message available to the CIST Port Information state machines.

If and only if rcvdInternal is set, this procedure sets rcvdMsg for each and every MSTI for which a MSTI message is conveyed in the BPDU, and makes available each MSTI message and the common parts of the CIST message priority (the CIST Root Identifier, External Root Path Cost, and Regional Root Identifier) to the Port Information state machine for that MSTI.

13.29.14 rcvAgreements()

This procedure is used to process agreements received in SPT BPDUs and in SPB Hello PDUs. The same variables are used in both cases. In the case of SPT BPDUs, this procedure is invoked by the rcvMsgs() procedure that decodes received BPDUs.

If the received Agreement Number is equal to the value of agreedN plus three, agreedMisorder is set TRUE.

The variables agreedN and agreeND are set to the values of the received Agreement Number and Discard Agreement Number respectively. The variables agreedDigest and agreedDigestValid are updated with the values of the received Agreement Digest and Agreement Digest flag respectively.

The CIST proposed flag is set to the value of the Proposing flag for the CIST in the received BPDU.

The updtDigest() procedure is invoked. The updtAgreement() procedure is invoked for the port, for the CIST and for each SPT.

13.29.15 ~~13.27.14~~ recordAgreement()

For the CIST and a given port, if rstpVersion is TRUE, operPointToPointMAC (6.6.3) is TRUE, and the received CIST Message has the Agreement flag set, then the CIST agreed flag is set and the CIST proposing flag is cleared. Otherwise the CIST agreed flag is cleared. Additionally, if the CIST message was received from a bridge in a different MST Region, i.e., the rcvdInternal flag is clear, the agreed and proposing flags for this port for all MSTIs are set or cleared to the same value as the CIST agreed and proposing flags. If the CIST message was received from a bridge in the same MST Region, the MSTI agreed and proposing flags are not changed.

For a given MSTI and port, if operPointToPointMAC (6.6.3) is TRUE, and

a) The message priority vector of the CIST Message accompanying the received MSTI Message (i.e., received in the same BPDU) has the same CIST Root Identifier, CIST External Root Path Cost, and Regional Root Identifier as the CIST port priority vector, and
b) The received MSTI Message has the Agreement flag set,

the MSTI agreed flag is set and the MSTI proposing flag is cleared. Otherwise the MSTI agreed flag is cleared.

NOTE—MSTI Messages received from bridges external to the MST Region are discarded and not processed by recordAgreeement() or recordProposal().

13.29.16 ~~13.27.15~~ recordDispute()

For the CIST and a given port, if the CIST message has the learning flag set:

a) The disputed variable is set; and
b) The agreed variable is cleared.

Additionally, if the CIST message was received from a bridge in a different MST region (i.e., if the rcvdInternal flag is clear), then for all the MSTIs:

c) The disputed variable is set; and
d) The agreed variable is cleared.

For a given MSTI and port, if the received MSTI message has the learning flag set:

e) The disputed variable is set; and
f) The agreed variable is cleared.

13.29.17 ~~13.27.16~~ recordMastered()

For the CIST and a given port, if the CIST message was received from a bridge in a different MST Region, i.e., the rcvdInternal flag is clear, the mastered variable for this port is cleared for all MSTIs.

For a given MSTI and port, if the MSTI message was received on a point-to-point link and the MSTI Message has the Master flag set, set the mastered variable for this MSTI. Otherwise reset the mastered variable.

13.29.18 ~~13.27.17~~ recordPriority()

Sets the components of the portPriority variable to the values of the corresponding msgPriority components.

13.29.19 ~~13.27.18~~ recordProposal()

For the CIST and a given port, if the received CIST Message conveys a Designated Port Role, and has the Proposal flag set, the CIST proposed flag is set. Otherwise the CIST proposed flag is not changed. Additionally, if the CIST Message was received from a bridge in a different MST Region, i.e., the rcvdInternal flag is clear, the proposed flags for this port for all MSTIs are set or cleared to the same value as the CIST proposed flag. If the CIST message was received from a bridge in the same MST Region, the MSTI proposed flags are not changed.

For a given MSTI and port, if the received MSTI Message conveys a Designated Port Role, and has the Proposal flag set, the MSTI proposed flag is set. Otherwise the MSTI proposed flag is not changed.

13.29.20 ~~13.27.19~~ recordTimes()

For the CIST and a given port, sets portTimes' Message Age, Max Age, Forward Delay, and remainingHops to the received values held in msgTimes and portTimes' Hello Time to the default specified in Table 13-5.

For a given MSTI and port, sets portTime's remainingHops to the received value held in msgTimes.

13.29.21 ~~13.27.20~~setReRootTree()

Sets reRoot TRUE for this tree (the CIST or a given MSTI) for all ports of the bridge.

13.29.22 ~~13.27.21~~ setSelectedTree()

Sets selected TRUE for this tree (the CIST or a given MSTI) for all ports of the bridge if reselect is FALSE for all ports in this tree. If reselect is TRUE for any port in this tree, this procedure takes no action.

13.29.23 ~~13.27.22~~ setSyncTree()

Sets sync TRUE for this tree (the CIST or a given MSTI) for all ports of the bridge.

13.29.24 ~~13.27.23~~ setTcFlags()

For the CIST and a given port:

a) If the Topology Change Acknowledgment flag is set for the CIST in the received BPDU, sets rcvdTcAck TRUE.
b) If rcvdInternal is clear and the Topology Change flag is set for the CIST in the received BPDU, sets rcvdTc TRUE for the CIST and for each and every MSTI.
c) If rcvdInternal is set, sets rcvdTc for the CIST if the Topology Change flag is set for the CIST in the received BPDU.

For a given MSTI and port, sets rcvdTc for this MSTI if the Topology Change flag is set in the corresponding MSTI message.

13.29.25 ~~13.27.24~~ setTcPropTree()

If and only if restrictedTcn is FALSE for the port that invoked the procedure, sets tcProp TRUE for the given tree (the CIST or a given MSTI) for all other ports.

13.29.26 ~~13.27.25~~ syncMaster()

For all MSTIs, for each port that has infoInternal set:

a) Clears the agree, agreed, and synced variables; and
b) Sets the sync variable.

13.29.27 ~~13.27.26~~ txConfig()

Transmits a Configuration BPDU. The first four components of the message priority vector (13.27.39) conveyed in the BPDU are set to the value of the CIST Root Identifier, External Root Path Cost, Bridge Identifier, and Port Identifier components of the CIST's designatedPriority parameter (13.27.20) for this port. The topology change flag is set if (tcWhile != 0) for the port. The topology change acknowledgment flag is set to the value of tcAck for the port. The remaining flags are set to zero. The value of the Message Age, Max Age, and Fwd Delay parameters conveyed in the BPDU are set to the values held in the CIST's designatedTimes parameter (13.27.21) for the port. The value of the Hello Time parameter conveyed in the BPDU is set to the value held in the CIST's portTimes parameter (13.27.48) for the port.

13.29.28 ~~13.27.27~~ txRstp()

Transmits a RST BPDU, ~~or~~ MST BPDU, or SPT BPDU as determined by the value of ForceProtocolVersion (13.7.2), and encoded as specified by Clause 14. All per port variables referenced in this clause (13.29.28) are those for the transmitting port.

The first six components of the CIST message priority vector (13.27.39) conveyed in the BPDU are set to the value of the CIST's designatedPriority parameter (13.27.20). The Port Role in the BPDU (14.2.9) is set to the current value of the CIST's role (13.27.66). The Agreement and Proposal flags in the BPDU are set to the values of agree (13.27.3) and proposing (13.27.50), respectively. The CIST topology change flag is set if (tcWhile != 0) for the port. The topology change acknowledge flag in the BPDU is never used and is set to zero. The Learning and Forwarding flags in the BPDU are set to the values of learning (13.27.35) and forwarding (13.27.30) for the CIST, respectively. The value of the Message Age, Max Age, and Fwd Delay parameters conveyed in the BPDU are set to the values held in the CIST's designatedTimes parameter (13.27.21). The value of the Hello Time parameter conveyed in the BPDU is set to the value held in the CIST's portTimes parameter (13.27.48).

If the value of the Force Protocol Version parameter is less than 3, no further parameters are encoded in the BPDU and the protocol version parameter is set to 2 (denoting a RST BPDU). Otherwise, MST BPDU parameters are encoded:

a) The version 3 length.
b) The MST Configuration Identifier parameter of the BPDU is the value of MstConfigId (13.26.7).
c) The CIST Internal Root Path Cost (13.27.20).
d) The CIST Bridge Identifier (CIST Designated Bridge Identifier—13.27.20).
e) The CIST Remaining Hops (13.27.21).
f) The parameters of each MSTI message, encoded in MSTID order.

NOTE—No more than 64 MSTIs may be supported. The parameter sets for all of these can be encoded in a standard-sized Ethernet frame. The number of MSTIs supported can be zero: an SPT Bridge, for example, is not obliged to have MSTIs configured in order to support shortest path bridging.

If the value of the Force Protocol Version parameter is less than 3, no further parameters are encoded in the BPDU and the protocol version parameter is set to 3 (denoting a MST BPDU). Otherwise, SPT BPDU parameters are encoded, and the protocol version parameter is set to 4 (denoting an SPT BPDU):

g) agreeN (13.27.16) is encoded in the Agreement Number field.
h) agreedND (13.27.17) is encoded in the Discarded Agreement Number field.
i) agreeDigest (13.27.8) is encoded in the Agreement Digest field.

13.29.29 ~~13.27.28~~ txTcn()

Transmits a TCN BPDU.

13.29.30 updtAgreement()

If ISIS-SPB is implemented, this procedure is invoked for a given port, for the CIST or a given SPT, by the updtRolesTree() procedure used by the Port Role Selection machine (13.36), and by the rcvAgreements()

procedure. The Port Role Selection machine itself could have been invoked following receipt of a BPDU on a port that is at the boundary of the SPT Region, possibly changing the context for the link state calculation, or following a link state update computed by ISIS-SPB following receipt of an IS-IS PDU.

If the procedure has been invoked for the CIST:

a) If the port's selectedRole is DesignatedPort, agreed is TRUE if and only if:
    1) agreedTopology is TRUE; and
    2) designatedPriority is the same or better (13.9) than agreedPriority.

b) If the port's selectedRole is not DesignatedPort, agreed is TRUE if and only if:
    1) agreementOutstanding is the same or better than designatedPriority.

If the procedure has been invoked for an SPT, and the port's selectedRole is DesignatedPort:

c) agreedAbove is reset to FALSE;

d) if agreedTopology is TRUE:
    1) agreementOutstanding (13.27.15) is set to BestAgreementPriority (13.28.4); and
    2) agreedPriority (13.27.13) is set to the sum of neighbourPriority (13.27.41) and InternalPortPathCost.

e) if agreedTopology is FALSE and agreedPriority is less than the sum of neighbourPriority and InternalPortPathCost:
    1) agreedPriority (13.27.13) is set to the sum of neighbourPriority (13.27.41) and InternalPortPathCost.

f) agreed (13.27.4) is TRUE if and only if:
    1) designatedPriority is better than agreedPriority.

If the procedure has been invoked for an SPT and the port's selectedRole is not DesignatedPort:

g) agreedPriority is set to BestAgreementPriority;

h) if agreedTopology is TRUE:
    1) agreedAbove is set TRUE, and
    2) agreementOutstanding (13.27.15) is set is set to the sum of neighbourPriority (13.27.41) and InternalPortPathCost.

i) if agreedTopology is FALSE and agreementOutstanding is better than the sum of neighbourPriority and InternalPortPathCost:
    1) agreementOutstanding is set to the sum of neighbourPriority (13.27.41) and InternalPortPathCost.

j) agreed is TRUE if and only if:
    1) agreedAbove is TRUE, and
    2) agreementOutstanding is the same as or better than designatedPriority.

Independently of whether the procedure was invoked for the CIST or an SPT:

k) if agreed is FALSE, then the sync variable is set TRUE.

13.29.31 ~~13.27.29~~ updtBPDUVersion()

Sets rcvdSTP TRUE if the BPDU received is a version 0 or version 1 TCN or a Config BPDU. Sets rcvdRSTP TRUE if the received BPDU is a RST BPDU or a MST BPDU.

13.29.32 updtDigest()

Updates agreeDigest, agreeN, and agreeND, following calculation of a new topology or topologies by ISIS-SPB, and checks for a topology match with the updated values of those variables, as follows.

If agreeDigest is not equal to agreementDigest and:

a) agreeN is equal to agreedND, or
b) agreeN plus one is equal to agreedND;

then:

c) agreeDigest is set equal to agreementDigest, and
d) agreeDigestValid is reset to FALSE, and
e) agreeN is set equal to agreeN plus one.

Additionally, if agreeDigest is now equal to agreementDigest and:

f) agreeDigest equals agreedDigest, and
g) agreedDigestValid is TRUE,

then:

h) if agreeND is not equal to agreedN plus one, then:
    1) agreeND is set equal to agreedN plus one, and
    2) newInfoMsti is set.

and

i) if agreedND is equal to agreeN and agreedMisorder is FALSE, or
j) if agreedND is set equal to agreeN plus one, then:
    1) agreedTopology is set TRUE, and
    2) agreedMisorder is set FALSE;

otherwise, i.e., if (f) and (g) above are not both TRUE:

k) if agreeND is not equal to agreedN, then:
    1) agreeND is set equal to agreedN, and
    2) newInfoMsti is set.

The agreeDigest, agreeN, and agreeND, variables determine the values of the Agreement Digest, Agreement Number (AN), and Discarded Agreement Number (DAN), transmitted in SPT BPDU and SPB Hello PDUs. This procedure is used by updtRolesTree() before using updtAgreement() for each SPT, and also by rcvAgreements(), since the latter can rotate the AN sequence window number and thus allow agreeDigest to be updated with agreementDigest.

Wherever newInfoMsti is set in this procedure, transmission of an SPB Hello PDU to convey the updated digest and sequence number information is also requested. SPT BPDUs and SPB Hello PDUs perform additional functions and are subject to different rate limiters, but both convey the Agreement Digest and related information.

13.29.33 ~~13.27.30~~ updtRcvdInfoWhile()

Updates rcvdInfoWhile (13.25). The value assigned to rcvdInfoWhile is three times the Hello Time, if either:

a) Message Age, incremented by 1 second and rounded to the nearest whole second, does not exceed Max Age and the information was received from a bridge external to the MST Region (rcvdInternal FALSE);

or

b) remainingHops, decremented by one, is greater than zero and the information was received from a bridge internal to the MST Region (rcvdInternal TRUE);

and is zero otherwise.

The values of Message Age, Max Age, remainingHops, and Hello Time used in these calculations are taken from the CIST's portTimes parameter (13.27.48) and are not changed by this procedure.

13.29.34 ~~13.27.31~~ updtRolesTree()

This procedure calculates the following priority vectors (13.9, 13.10 for the CIST, 13.11 for a MSTI) and timer values, for the CIST or a given MSTI:

a) The *root path priority vector* for each Bridge Port that is not Disabled and has a *port priority vector* (portPriority plus portId—see 13.27.47 and 13.27.46) that has been recorded from a received message and not aged out (infoIs == Received).

b) The *root path priority vector* propagated and calculated by ISIS-SPB (if ISIS-SPB is implemented and ForceProtocolVersion is 4).

c) ~~b)~~ The Bridge's *root priority vector* (rootPortId, rootPriority—13.26.9, 13.26.10), chosen as the best of the set of priority vectors comprising the bridge's own *bridge priority vector* (BridgePriority—13.26.3) plus all calculated root path priority vectors whose:
   1) DesignatedBridgeID Bridge Address component is not equal to that component of the bridge's own bridge priority vector (13.10) and,
   2) Port's restrictedRole parameter is FALSE and,
   3) Port's restrictedDomainRole parameter is FALSE or the port is not a Boundary Port of the SPT Region (13.12).

NOTE—If ISIS-SPB is being used but did not provided the selected *root priority vector* for the CIST, that priority vector will be associated with the Master Port of the SPT Region (a Boundary Port of, and not internal to, the Region) and will be used by ISIS-SPB to propagate the CST component of the priority vector throughout the Region.

d) ~~c)~~ The bridge's *root times*, (rootTimes—13.26.11), set equal to:
   1) BridgeTimes (13.26.4), if the chosen root priority vector is the bridge priority vector, or was calculated by ISIS-SPB; otherwise,
   2) portTimes (13.27.48) for the port associated with the selected root priority vector, with the Message Age component incremented by 1 second and rounded to the nearest whole second if the information was received from a bridge external to the MST Region (rcvdInternal FALSE), and with remainingHops decremented by one if the information was received from a bridge internal to the MST Region (rcvdInternal TRUE).

e) ~~d)~~ The *designated priority vector* (designatedPriority—13.27.20) for each port that is not internal an SPT Region; and

f) ~~e)~~ The *designated times* (designatedTimes—13.27.21) for each port set equal to the value of *root times*.

If the root priority vector for the CIST is recalculated, and has a different Regional Root Identifier than that previously selected, and has or had a nonzero CIST External Root Path Cost, the syncMaster() procedure (13.29.26) is invoked.

NOTE—Changes in Regional Root Identifier will not cause loops if the Regional Root is within an MST Region, as is the case if and only if the MST Region is the Root of the CST. This important optimization allows the MSTIs to be fully independent of each other in the case where they compose the core of a network.

The CIST, ~~or~~ MSTI, or SPT Port Role for each port is assigned, and its port priority vector and timer information are updated as specified in the remainder of this clause (13.41.2).

If the port is Disabled (infoIs == Disabled), selectedRole is set to DisabledPort.

Otherwise, if this procedure was invoked for an MSTI or an SPT, for a port that is not Disabled, and that has CIST port priority information that was received from a bridge external to its bridge's Region (infoIs == Received and infoInternal == FALSE), then:

g) ~~f)~~ If the selected CIST Port Role (calculated for the CIST prior to invoking this procedure for an MSTI or SPT) is RootPort, selectedRole is set to MasterPort.
h) ~~g)~~ If selected CIST Port Role is AlternatePort, selectedRole is set to AlternatePort.
i) ~~h)~~ Additionally, updtInfo is set if the port priority vector differs from the designated priority vector or the port's associated timer parameter differs from the one for the Root Port.

Otherwise, for the CIST for a port that is not Disabled and not internal to an SPT Region, or for an MSTI for a port of that is not Disabled and whose CIST port priority information was not received from a bridge external to the Region (infoIs != Received or infoInternal == TRUE), the CIST or MSTI port role is assigned, and the port priority vector and timer information updated as follows:

j) ~~i)~~ If the port priority vector information was aged (infoIs = Aged), updtInfo is set and selectedRole is set to DesignatedPort;
k) ~~j)~~ If the port priority vector was derived from another port on the bridge or from the bridge itself as the Root Bridge (infoIs = Mine), selectedRole is set to DesignatedPort. Additionally, updtInfo is set if the port priority vector differs from the designated priority vector or the port's associated timer parameter(s) differ(s) from the Root Port's associated timer parameters;
l) ~~k)~~ If the port priority vector was received in a Configuration Message and is not aged (infoIs == Received), and the root priority vector is now derived from it, selectedRole is set to RootPort, and updtInfo is reset;
m) ~~l)~~ If the port priority vector was received in a Configuration Message and is not aged (infoIs == Received), the root priority vector is not now derived from it, the designated priority vector is not better than the port priority vector, and the designated bridge and designated port components of the port priority vector do not reflect another port on this bridge, selectedRole is set to AlternatePort, and updtInfo is reset;
n) ~~m)~~ If the port priority vector was received in a Configuration Message and is not aged (infoIs == Received), the root priority vector is not now derived from it, the designated priority vector is not better than the port priority vector, and the designated bridge and designated port components of the port priority vector reflect another port on this bridge, selectedRole is set to BackupPort, and updtInfo is reset;
o) ~~n)~~ If the port priority vector was received in a Configuration Message and is not aged (infoIs == Received), the root priority vector is not now derived from it, the designated priority vector is better than the port priority vector, selectedRole is set to DesignatedPort, and updtInfo is set.

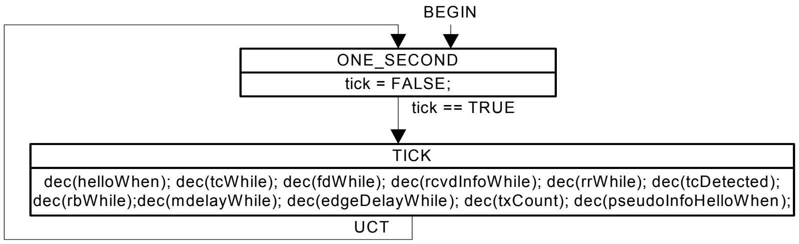
Otherwise, for the CIST or an SPT, for a port that is not Disabled and is internal to an SPT Region, ISIS-SPB determines the selectedRole (and other parameters, see 13.36), and uses updtAgreement() (13.29.30).

13.29.35 ~~13.27.32~~ uptRolesDisabledTree()

This procedure sets selectedRole to DisabledPort for all ports of the bridge for a given tree (CIST, ~~or~~ MSTI, or SPT).

13.30 ~~13.28~~ The Port Timers state machine

The Port Timers state machine shall implement the function specified by the state diagram in Figure 13-15 and the attendant definitions in 13.25 through 13.29.


BEGIN
ONE_SECOND
tick = FALSE;
tick == TRUE
TICK
dec(helloWhen); dec(tcWhile); dec(fdWhile); dec(rcvdInfoWhile); dec(rrWhile); dec(tcDetected);
dec(rbWhile);dec(mdelayWhile); dec(edgeDelayWhile); dec(txCount); dec(pseudoInfoHelloWhen);
UCT

The following abbreviation is used in this state diagram:
dec(x)
{ if (x !=0) x= x-1;
}

If there is more than one instance of x, e.g. per port, all instances are separately decremented.

Figure 13-15—Port Timers state machine

The state machine uses tick (13.27), a signal set by an implementation-specific system clock function at one second intervals, to decrement the timer variables for the CIST and all MSTIs for the port. The state machine that uses a given timer variable is responsible for setting its initial value.

13.31 ~~13.29~~ Port Receive state machine

The Port Receive state machine shall implement the function specified by the state diagram in Figure 13-16 and the attendant definitions in 13.25 through 13.29.

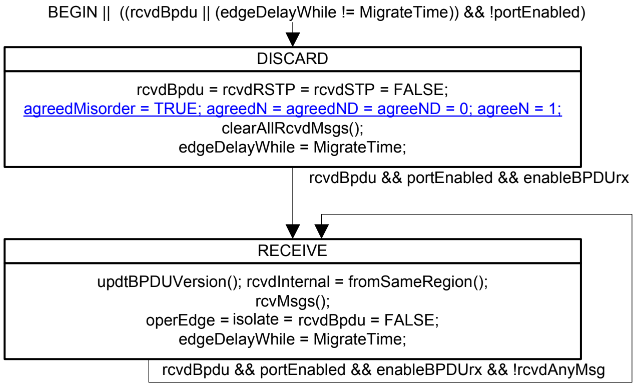

BEGIN || ((rcvdBpdu || (edgeDelayWhile != MigrateTime)) && !portEnabled)
DISCARD
rcvdBpdu = rcvdRSTP = rcvdSTP = FALSE;
agreedMisorder = TRUE; agreedN = agreedND = agreeND = 0; agreeN = 1;
clearAllRcvdMsgs();
edgeDelayWhile = MigrateTime;
rcvdBpdu && portEnabled && enableBPDUrx
RECEIVE
updtBPDUVersion(); rcvdInternal = fromSameRegion();
rcvMsgs();
operEdge = isolate = rcvdBpdu = FALSE;
edgeDelayWhile = MigrateTime;
rcvdBpdu && portEnabled && enableBPDUrx && !rcvdAnyMsg

Figure 13-16—Port Receive state machine

This state machine receives and decodes validated BPDUs. The rcvdMsg flag is set for the CIST and for each MSTI supported by the receiving bridge if the received BPDU is from the same MST or SPT Region. The next BPDU is not processed until all rcvdMsg flags have been cleared by the per tree state machines.

NOTE—This standard does not specify or constrain the means used by the rcvMsgs() procedure to identify or extract per tree information from a BPDU for processing by rcvInfo() in the Port Information Machine RECEIVE state.

13.32 ~~13.30~~ Port Protocol Migration state machine

The Port Protocol Migration state machine shall implement the function specified by the state diagram in Figure 13-17 and the attendant definitions in 13.25 through 13.29.

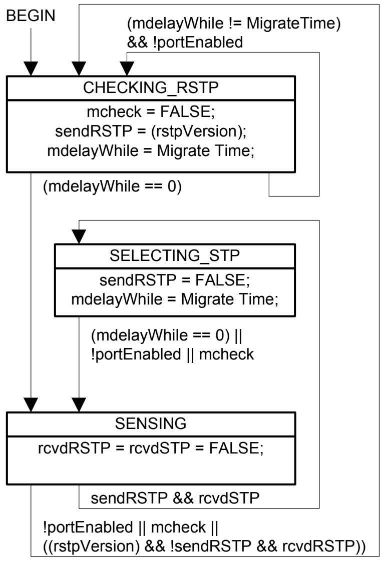

BEGIN
(mdelayWhile != MigrateTime)
&& !portEnabled
CHECKING_RSTP
mcheck = FALSE;
sendRSTP = (rstpVersion);
mdelayWhile = Migrate Time;
(mdelayWhile == 0)
SELECTING_STP
sendRSTP = FALSE;
mdelayWhile = Migrate Time;
(mdelayWhile == 0) ||
!portEnabled || mcheck
SENSING
rcvdRSTP = rcvdSTP = FALSE;
sendRSTP && rcvdSTP
!portEnabled || mcheck ||
((rstpVersion) && !sendRSTP && rcvdRSTP))

Figure 13-17—Port Protocol Migration state machine

13.33 ~~13.31~~ Bridge Detection state machine

The Bridge Detection state machine shall implement the function specified by the state diagram in Figure 13-18 and the attendant definitions in 13.25 through 13.29.

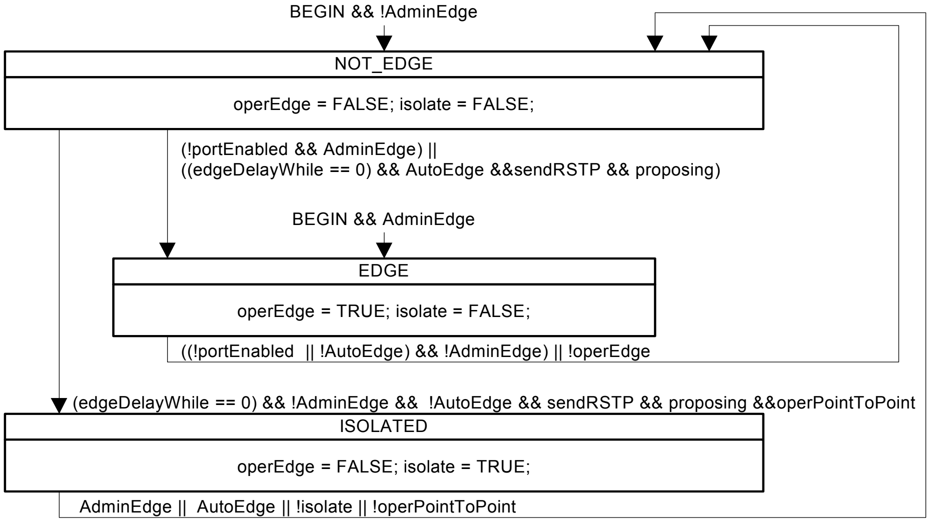
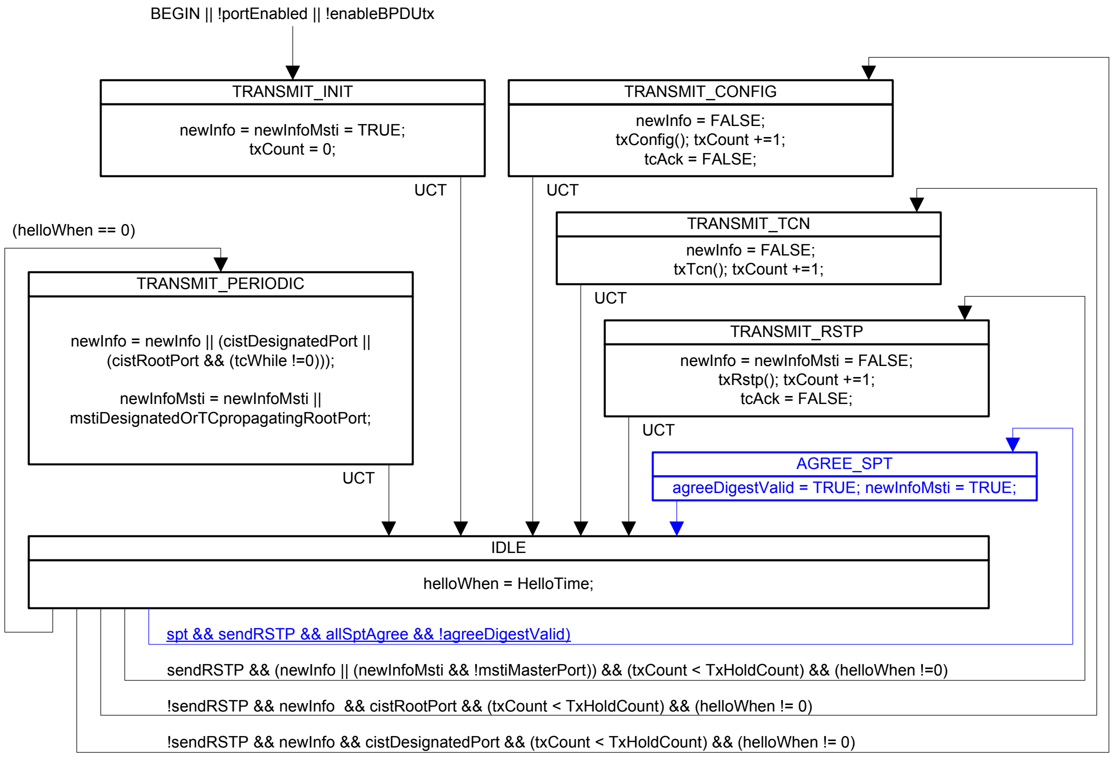

BEGIN && !AdminEdge
NOT_EDGE
operEdge = FALSE; isolate = FALSE;
(!portEnabled && AdminEdge) ||
((edgeDelayWhile == 0) && AutoEdge &&sendRSTP && proposing)
BEGIN && AdminEdge
EDGE
operEdge = TRUE; isolate = FALSE;
((!portEnabled || !AutoEdge) && !AdminEdge) || !operEdge
(edgeDelayWhile == 0) && !AdminEdge && !AutoEdge && sendRSTP && proposing &&operPointToPoint
ISOLATED
operEdge = FALSE; isolate = TRUE;
AdminEdge || AutoEdge || !isolate || !operPointToPoint

Figure 13-18—Bridge Detection state machine

13.34 ~~13.32~~ Port Transmit state machine

The Port Transmit state machine shall implement the function specified by the state diagram in Figure 13-19 and the attendant definitions in 13.25 through 13.29.


BEGIN || !portEnabled || !enableBPDUtx
TRANSMIT_INIT
newInfo = newInfoMsti = TRUE;
txCount = 0;
UCT
TRANSMIT_CONFIG
newInfo = FALSE;
txConfig(); txCount +=1;
tcAck = FALSE;
UCT
TRANSMIT_TCN
newInfo = FALSE;
txTcn(); txCount +=1;
UCT
(helloWhen == 0)
TRANSMIT_PERIODIC
newInfo = newInfo || (cistDesignatedPort ||
(cistRootPort && (tcWhile !=0)));
newInfoMsti = newInfoMsti ||
mstiDesignatedOrTCpropagatingRootPort;
UCT
TRANSMIT_RSTP
newInfo = newInfoMsti = FALSE;
txRstp(); txCount +=1;
tcAck = FALSE;
UCT
AGREE_SPT
agreeDigestValid = TRUE; newInfoMsti = TRUE;
IDLE
helloWhen = HelloTime;
spt && sendRSTP && allSptAgree && !agreeDigestValid)
sendRSTP && (newInfo || (newInfoMsti && !mstiMasterPort)) && (txCount < TxHoldCount) && (helloWhen !=0)
!sendRSTP && newInfo && cistRootPort && (txCount < TxHoldCount) && (helloWhen != 0)
!sendRSTP && newInfo && cistDesignatedPort && (txCount < TxHoldCount) && (helloWhen != 0)

All transtions, except UCT, are qualified by "&& allTransmitReady".

Figure 13-19—Port Transmit state machine

This state machine is responsible for transmitting BPDUs. It also determines when the Agreement Digest in SPT BPDUs can be updated, and prompts transmission when that has been done.

NOTE 1—Any single received BPDU that changes the CIST Root Identifier, CIST External Root Path Cost, or CIST Regional Root associated with MSTIs should be processed entirely, or not at all, before encoding BPDUs for transmission.This recommendation minimizes the number of BPDUs to be transmitted following receipt of a BPDU with new information. It is not required for correctness and has not therefore been incorporated into the state machines.

NOTE 2—If a CIST state machine sets newInfo, this machine will ensure that a BPDU is transmitted conveying the new CIST information. If MST BPDUs can be transmitted through the port, this BPDU will also convey new MSTI information for all MSTIs. If a MSTI state machine sets newInfoMsti, and MST BPDUs can be transmitted through the port, this machine will ensure that a BPDU is transmitted conveying information for the CIST and all MSTIs. Separate newInfo and newInfoMsti variables are provided to avoid requiring useless transmission of a BPDU through a port that can only transmit STP BPDUs (as required by the Force Protocol Version parameter or Port Protocol Migration machine) following a change in MSTI information without any change to the CIST.

13.35 ~~13.33~~ Port Information state machine

The Port Information state machine for each tree shall implement the function specified by the state diagram in Figure 13-20 and the attendant definitions in 13.25 through 13.29.

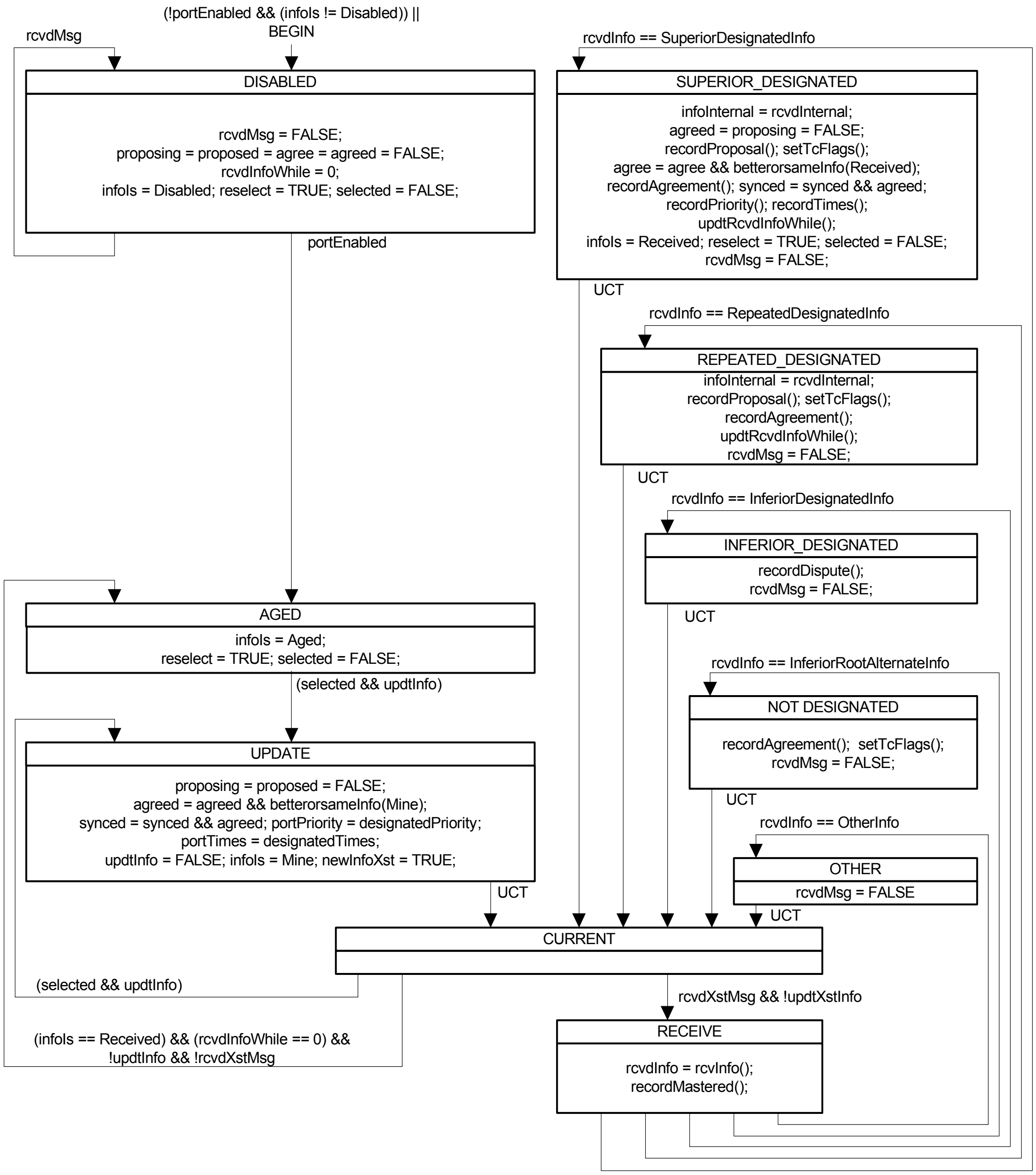

(!portEnabled && (infoIs != Disabled)) ||
BEGIN
rcvdMsg
DISABLED
rcvdMsg = FALSE;
proposing = proposed = agree = agreed = FALSE;
rcvdInfoWhile = 0;
infoIs = Disabled; reselect = TRUE; selected = FALSE;
portEnabled
rcvdInfo == SuperiorDesignatedInfo
SUPERIOR_DESIGNATED
infoInternal = rcvdInternal;
agreed = proposing = FALSE;
recordProposal(); setTcFlags();
agree = agree && betterorsameInfo(Received);
recordAgreement(); synced = synced && agreed;
recordPriority(); recordTimes();
updtRcvdInfoWhile();
infoIs = Received; reselect = TRUE; selected = FALSE;
rcvdMsg = FALSE;
UCT
rcvdInfo == RepeatedDesignatedInfo
REPEATED_DESIGNATED
infoInternal = rcvdInternal;
recordProposal(); setTcFlags();
recordAgreement();
updtRcvdInfoWhile();
rcvdMsg = FALSE;
UCT
rcvdInfo == InferiorDesignatedInfo
INFERIOR_DESIGNATED
recordDispute();
rcvdMsg = FALSE;
UCT
rcvdInfo == InferiorRootAlternateInfo
NOT DESIGNATED
recordAgreement(); setTcFlags();
rcvdMsg = FALSE;
UCT
rcvdInfo == OtherInfo
OTHER
rcvdMsg = FALSE
UCT
AGED
infoIs = Aged;
reselect = TRUE; selected = FALSE;
(selected && updtInfo)
UPDATE
proposing = proposed = FALSE;
agreed = agreed && betterorsameInfo(Mine);
synced = synced && agreed; portPriority = designatedPriority;
portTimes = designatedTimes;
updtInfo = FALSE; infoIs = Mine; newInfoXst = TRUE;
UCT
CURRENT
(selected && updtInfo)
(infoIs == Received) && (rcvdInfoWhile == 0) &&
!updtInfo && !rcvdXstMsg
rcvdXstMsg && !updtXstInfo
RECEIVE
rcvdInfo = rcvInfo();
recordMastered();

Figure 13-20—Port Information state machine

This state machine is responsible for recording the spanning tree information currently in use by the CIST or a given MSTI for a given port, ageing that information out if it was derived from an incoming BPDU, and recording the origin of the information in the infoIs variable. The selected variable is cleared and reselect set to signal to the Port Role Selection machine that port roles need to be recomputed. The infoIs and portPriority variables from all ports are used in that computation and, together with portTimes, determine new values of designatedPriority and designatedTimes. The selected variable is set by the Port Role Selection machine once the computation is complete.

13.36 ~~13.34~~ Port Role Selection state machine

The Port Role Selection state machine shall implement the function specified by the state diagram in Figure 13-21 and the attendant definitions in 13.25 through 13.29.

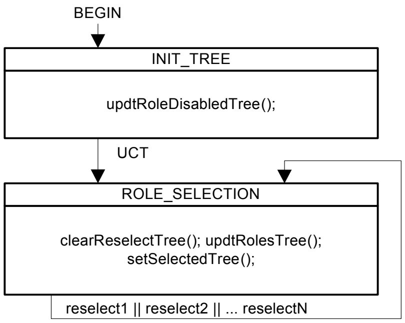

BEGIN
INIT_TREE
updtRoleDisabledTree();
UCT
ROLE_SELECTION
clearReselectTree(); updtRolesTree();
setSelectedTree();
reselect1 || reselect2 || ... reselectN

Figure 13-21—Port Role Selection state machine

When ISIS-SPB finishes a link state calculation, it clears the selected variable for each port for each SPT before updating agreementDigest (13.26.1) for the bridge component, the *root priority vector (*rootPortId, rootPriority—13.26.9, 13.26.10) for each SPT and the selectedRole (13.12, 13.27.68), *designated priority vector (*designatedPriority—13.27.20), and neighbourPriority for each port for each SPT. ISIS-SPB then sets reselect for each SPT.

13.37 ~~13.35~~ Port Role Transitions state machine

The Port Role Transitions state machine shall implement the function specified by the state diagram in the following figures ~~neighbors~~:

a) Part 1: Figure 13-22 for both the initialization of this state machine and the states associated with the DisabledPort role; and
b) Part 2: Figure 13-23 for the states associated with the MasterPort role; and
c) Part 3: Figure 13-24 for the states associated with the RootPort role; and
d) Part 4: Figure 13-25 for the states associated with the DesignatedPort role; and
e) Part 5: Figure 13-26 for the states associated with the AlternatePort and BackupPort roles;

and the attendant definitions in 13.25 through 13.29.

As Figure 13-22, Figure 13-23, Figure 13-24, Figure 13-25, and Figure 13-26 are component parts of the same state machine, the global transitions associated with these diagrams are possible exit transitions from the states shown in any of the diagrams.

Figure 13-22 and Figure 13-26 show the Port Roles for ports that do not form part of the active topology of the given tree.

Figure 13-23, Figure 13-24, and Figure 13-25 show the Port Roles that form part of the active topology.

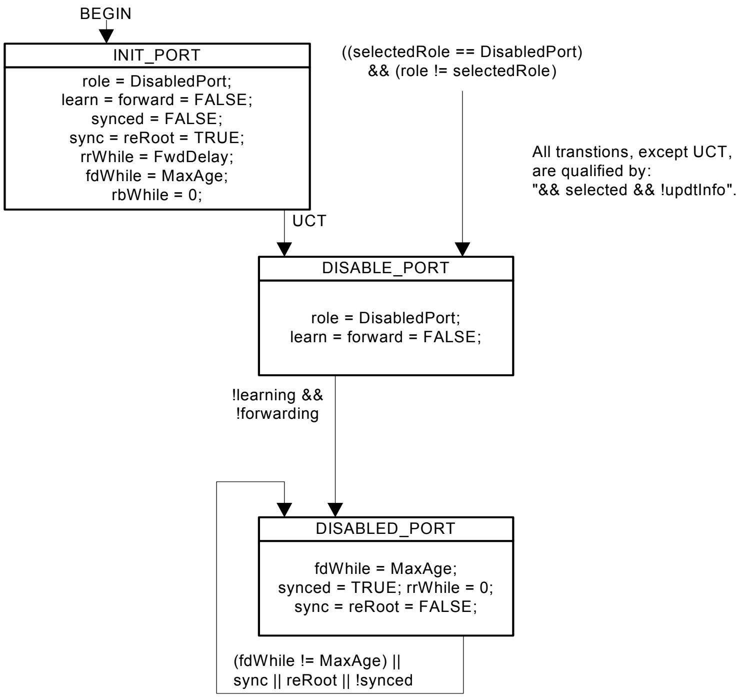


Figure 13-22—Disabled Port role transitions

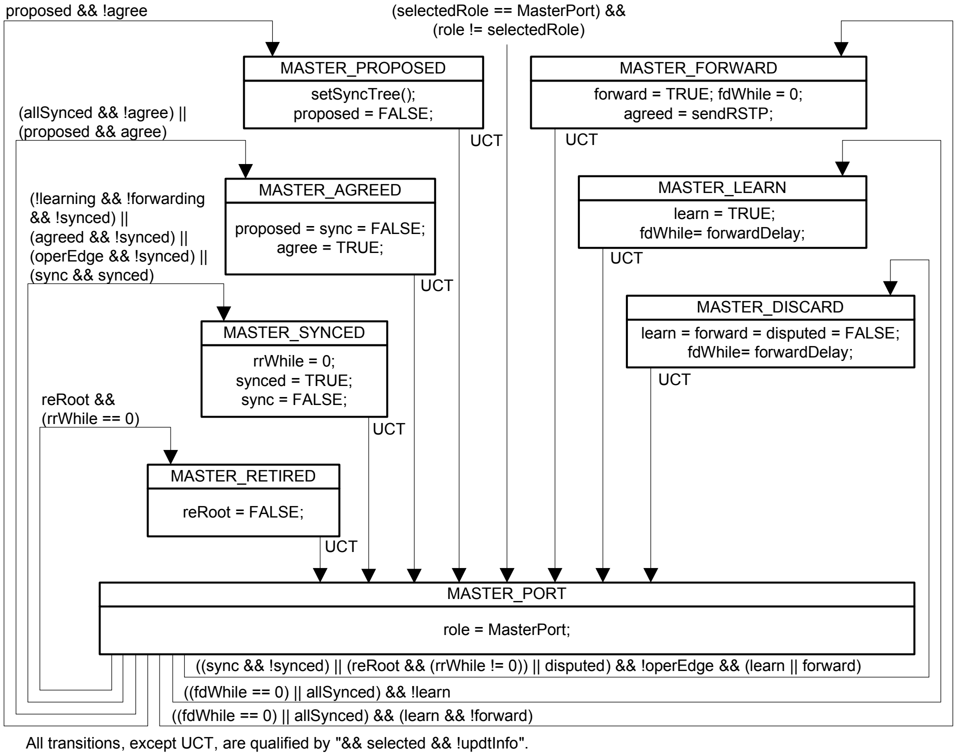
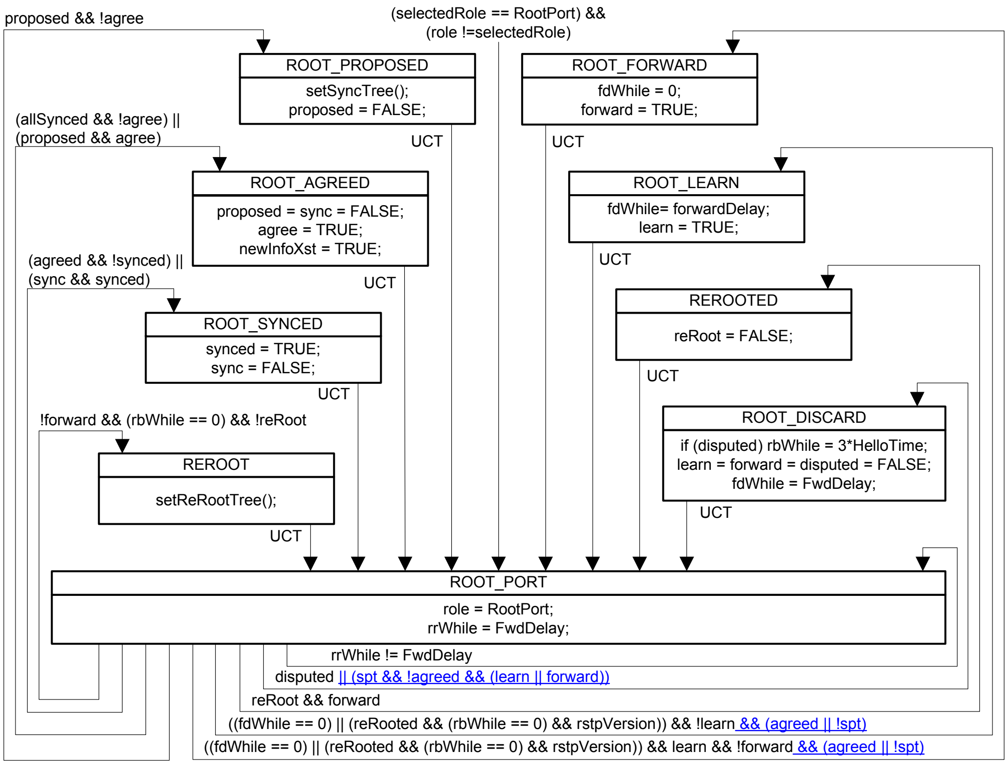
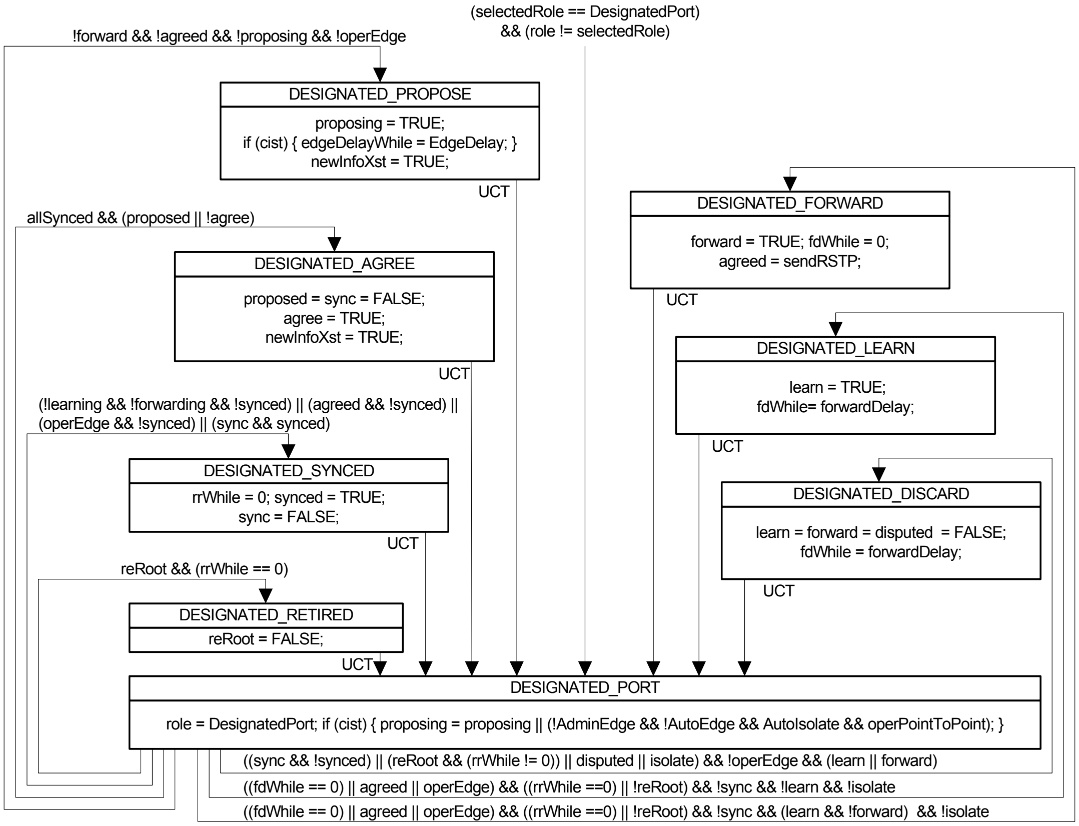


Figure 13-23—Port Role Transitions state machine—MasterPort



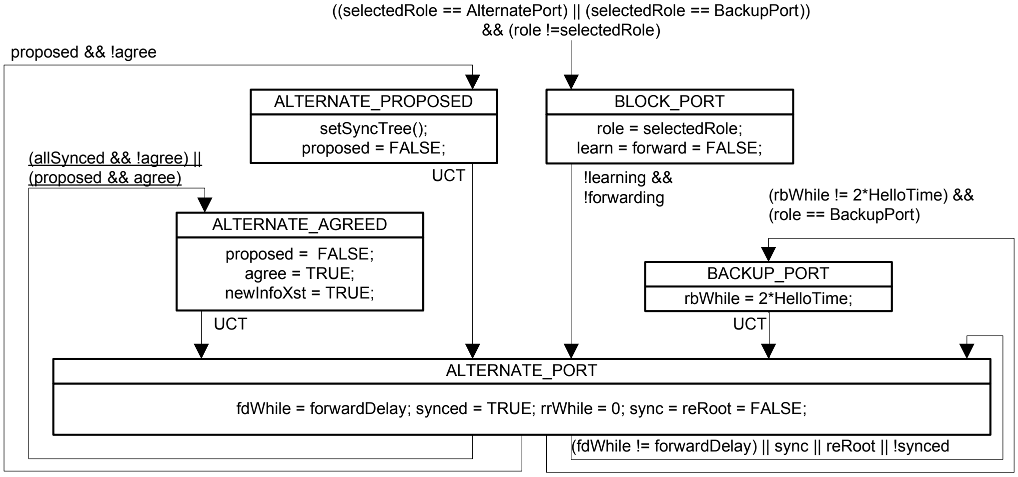
All transitions, except UCT, are qualified by "&& selected && !updtInfo".

Figure 13-24—Port Role Transitions state machine—RootPort


(selectedRole == DesignatedPort)
&& (role != selectedRole)
!forward && !agreed && !proposing && !operEdge
DESIGNATED_PROPOSE
proposing = TRUE;
if (cist) { edgeDelayWhile = EdgeDelay; }
newInfoXst = TRUE;
UCT
DESIGNATED_FORWARD
forward = TRUE; fdWhile = 0;
agreed = sendRSTP;
UCT
allSynced && (proposed || !agree)
DESIGNATED_AGREE
proposed = sync = FALSE;
agree = TRUE;
newInfoXst = TRUE;
UCT
DESIGNATED_LEARN
learn = TRUE;
fdWhile= forwardDelay;
UCT
(!learning && !forwarding && !synced) || (agreed && !synced) ||
(operEdge && !synced) || (sync && synced)
DESIGNATED_SYNCED
rrWhile = 0; synced = TRUE;
sync = FALSE;
UCT
DESIGNATED_DISCARD
learn = forward = disputed = FALSE;
fdWhile = forwardDelay;
UCT
reRoot && (rrWhile == 0)
DESIGNATED_RETIRED
reRoot = FALSE;
UCT
DESIGNATED_PORT
role = DesignatedPort; if (cist) { proposing = proposing || (!AdminEdge && !AutoEdge && AutoIsolate && operPointToPoint); }
((sync && !synced) || (reRoot && (rrWhile != 0)) || disputed || isolate) && !operEdge && (learn || forward)
((fdWhile == 0) || agreed || operEdge) && ((rrWhile ==0) || !reRoot) && !sync && !learn && !isolate
((fdWhile == 0) || agreed || operEdge) && ((rrWhile ==0) || !reRoot) && !sync && (learn && !forward) && !isolate

All transitions, except UCT, are qualified by "&& selected && !updtInfo".

Figure 13-25—Port Role Transitions state machine—DesignatedPort



All transitions, except UCT, are qualified by "&& selected && !updtInfo".

Figure 13-26—Port Role Transitions state machine—AlternatePort and BackupPort

13.38 ~~13.36~~ Port State Transition state machine

The Port State Transition state machine shall implement the function specified by the state diagram in Figure 13-27 and the attendant definitions in 13.25 through 13.29.

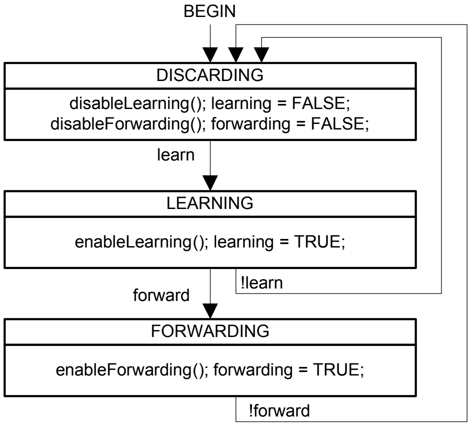

BEGIN
DISCARDING
disableLearning(); learning = FALSE;
disableForwarding(); forwarding = FALSE;
learn
LEARNING
enableLearning(); learning = TRUE;
!learn
forward
FORWARDING
enableForwarding(); forwarding = TRUE;
!forward

Figure 13-27—Port State Transition state machine

NOTE—A small system-dependent delay may occur on each of the transitions shown in the referenced state machine.

This state machine operates independently of the type of tree (CIST, ~~or~~ MSTI, or SPT) and whether or not backbone bridging is being supported. However the way in which the bridge supports the learn and forward variables and the disableForwarding(), disableLearning(), enableForwarding(), and enableLearning() procedures are supported does vary (see 8.4, 8.6, 8.6.1). The forwarding and learning variables provide implementation independent reporting of the current state.

13.38.1 ~~13.36.1~~ Port State transitions for the CIST and MSTIs

The CIST and each MSTI are always supported by an explicit Port State, enforced by bridge's implementation of the Forwarding Process, and the procedures prompt that implementation to take the necessary action to forward and/or learn from received frames (as requested).

13.38.2 Port State transitions for SPTs

The enableLearning() and disableLearning() procedures return without taking an action: for SPBV mode learning does not occur until forwarding is enabled (8.6.1); for SPBM mode the Learning Process (8.7) neither creates or deletes Dynamic Filtering Entries.

When a VLAN is supported by an SPT Bridge using SPBV mode, a Dynamic VLAN Registration Entry is created for each SPVID and Enable Ingress Filtering (8.6.2) enabled on each port. Connectivity is provided, and loops prevented, by adding and removing ports from the Dynamic VLAN Registration Entry (27.13).

The enableForwarding() procedure adds the port to the Dynamic VLAN Registration Entry for each SPVID that is associated with the SPT (provided the VLAN topology extends through the port, see 27.13). Similarly disableForwarding() removes the port from the registration entry.

When frames with a given VID are supported by SPBM mode, MAC address based ingress filtering is used to discard a received frame if there is no corresponding Dynamic Filtering Entry (8.8, 8.6.1, 27.14) for that VID and the frame's source MAC address with the receiving port in the Port Map.

ISIS-SPB identifies the port that provides the shortest path to a given bridge: this is the Root Port for the SPT rooted at that bridge. The individual MAC addresses (there can be more than one) known to ISIS-SPB as identifying sources for which that given bridge first transmits frames within the SPT Region (entering the region from an attached bridge or station, or from a protocol entity within the bridge) or as identifying destinations for which that bridge is the last recipient within the region (delivering frames to an attached bridge or station, or to a protocol entity within the bridge) are associated with that Root Port. If loop mitigation (6.5.4.2) is used for unicast frames, a Dynamic Filtering Entry is created for those addresses and VID, permitting forwarding through that port and no others, without reference to the state machines specified in this clause (Clause 13), and the setting or clearing of the forward variable for any port. The enableForwarding() procedure creates or modifies these Dynamic Filtering Entries, so that forwarding to and from a given port is permitted, if and only if loop prevention (6.5.4.1) is used for both unicast and multicast frames and the port's role is RootPort, and has no effect otherwise. Similarly if loop prevention is being used, disableForwarding() modifies or removes any existing Dynamic Filtering Entry so that forwarding to or from the port is not permitted.

NOTE—The Port Role Transition machine allows an SPT to transition a Root Port to Discarding in support of enableForwarding() and disableForwarding() as specified in this clause (13.38.2). The need to specify that transition could have been avoided by specifying that the Dynamic Filtering Entry would only permit forwarding through a given port if forward was TRUE for the port for every other SPT for which it was a Designated Port. While equivalent, such a specification approach would have been obscure and many implementors would have failed to spot the way to avoid checking variables for all SPTs when processing a change for a single tree.

An SPT Bridge using SPBM mode uses source specific multicast addresses, so that the destination address alone can be used to identify the SPT (Clause 27) on egress, and a Static Filtering Entry (8.8.1) is created for that multicast address and the B-VIDs supported by the SPT. If enableForwarding() is invoked for a Designated Port, the Static Filtering Entry's control element for that port is updated to specify that frames should be Forwarded through the port. The control element is modified to specify Filtered if enableForwarding() is invoked and the port's role is not Designated Port, or if disabledForwarding() is used.

13.39 ~~13.37~~ Topology Change state machine

The Topology Change state machine for each tree shall implement the function specified by the state diagram in Figure 13-28 and the attendant definitions in 13.25 through 13.29.

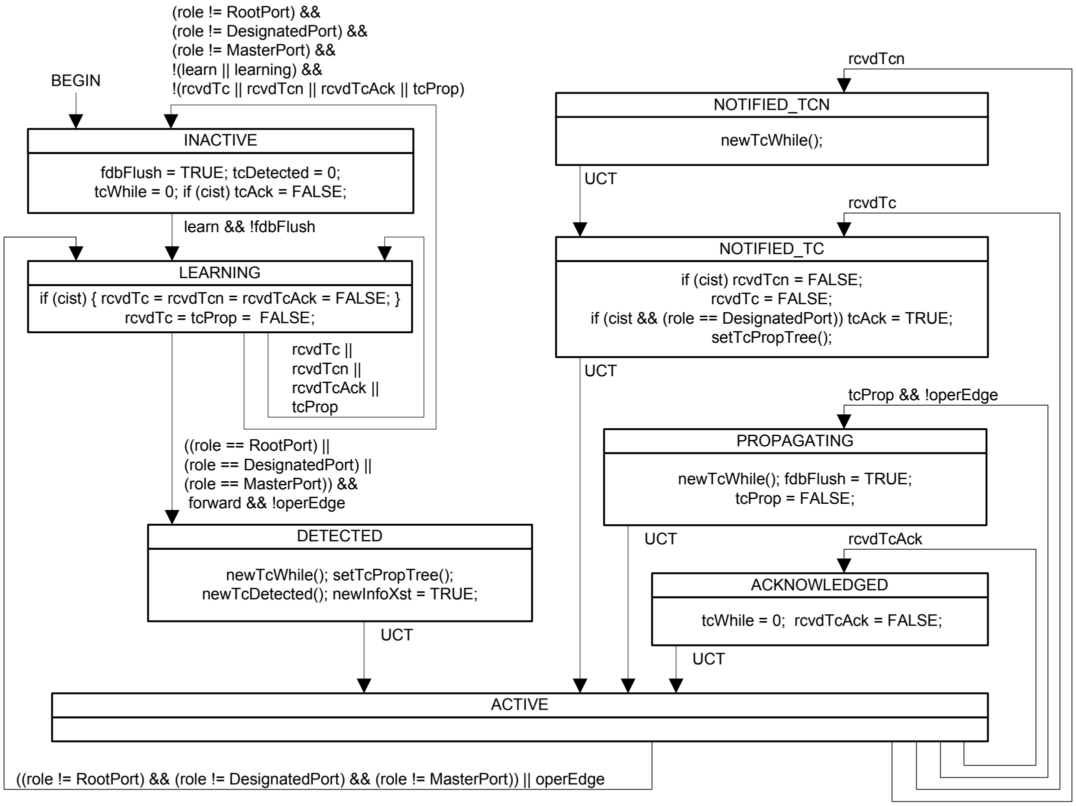

BEGIN
(role != RootPort) &&
(role != DesignatedPort) &&
(role != MasterPort) &&
!(learn || learning) &&
!(rcvdTc || rcvdTcn || rcvdTcAck || tcProp)
INACTIVE
fdbFlush = TRUE; tcDetected = 0;
tcWhile = 0; if (cist) tcAck = FALSE;
learn && !fdbFlush
LEARNING
if (cist) { rcvdTc = rcvdTcn = rcvdTcAck = FALSE; }
rcvdTc = tcProp = FALSE;
rcvdTc ||
rcvdTcn ||
rcvdTcAck ||
tcProp
((role == RootPort) ||
(role == DesignatedPort) ||
(role == MasterPort)) &&
forward && !operEdge
DETECTED
newTcWhile(); setTcPropTree();
newTcDetected(); newInfoXst = TRUE;
UCT
rcvdTcn
NOTIFIED_TCN
newTcWhile();
UCT
rcvdTc
NOTIFIED_TC
if (cist) rcvdTcn = FALSE;
rcvdTc = FALSE;
if (cist && (role == DesignatedPort)) tcAck = TRUE;
setTcPropTree();
UCT
tcProp && !operEdge
PROPAGATING
newTcWhile(); fdbFlush = TRUE;
tcProp = FALSE;
UCT
rcvdTcAck
ACKNOWLEDGED
tcWhile = 0; rcvdTcAck = FALSE;
UCT
ACTIVE
((role != RootPort) && (role != DesignatedPort) && (role != MasterPort)) || operEdge

Figure 13-28—Topology Change state machine

NOTE—MRP (Clause 10) uses the tcDetected variable maintained by this state machine.

13.40 ~~13.38~~ Layer 2 Gateway Port Receive state machine

If implemented, the Layer 2 Gateway Port state machine for each port shall implement the function specified by the state diagram in Figure 13-29 and the attendant definitions in 13.25 through 13.29.

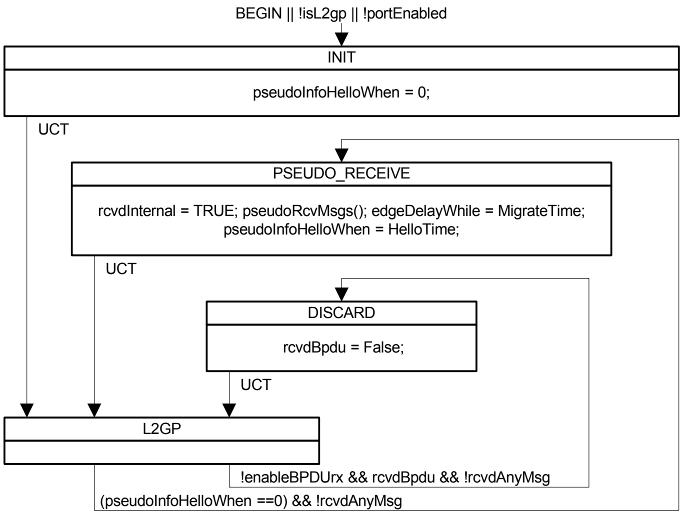

BEGIN || !isL2gp || !portEnabled
INIT
pseudoInfoHelloWhen = 0;
UCT
PSEUDO_RECEIVE
rcvdInternal = TRUE; pseudoRcvMsgs(); edgeDelayWhile = MigrateTime;
pseudoInfoHelloWhen = HelloTime;
UCT
DISCARD
rcvdBpdu = False;
UCT
L2GP
!enableBPDUrx && rcvdBpdu && !rcvdAnyMsg
(pseudoInfoHelloWhen ==0) && !rcvdAnyMsg

Figure 13-29—L2 Gateway Port Receive state machine

NOTE—The functionality provided by this state machine is discussed in 13.20, 25.9.2, 26.4.3, and 13.29.11.

13.41 ~~13.39~~ Customer Edge Port Spanning Tree operation

This subclause specifies the operation of the Spanning Tree Protocol Entity within a C-VLAN component that supports a Customer Edge Port (Figure 15-4) of a Provider Edge Bridge. The Customer Edge Port and each Provider Edge Port are treated as separate Bridge Ports by the spanning tree protocol.

If the C-VLAN component connects to the S-VLAN component with a single Provider Edge Port, and the associated service instance supports no more than two customer interfaces, then all frames (including Spanning Tree BPDUs) addressed to the Bridge Group Address may be relayed between the two ports of the C-VLAN component without modification. Otherwise, the Spanning Tree Protocol Entity shall execute RSTP, as modified by the provisions of this subclause (13.41).

The RSTP enhancements specified do not reduce Provider Bridged Network connectivity between Customer Edge Ports to a single spanning tree of service instances but ensure that connectivity for frames assigned to any given C-VLAN is loop-free. In this respect, the C-VLAN component's spanning tree protocol operation is equivalent to, but simpler to manage than, the operation of MSTP.

13.41.1 ~~13.39.1~~ Provider Edge Port operPointToPointMAC and operEdge

The value of the adminPointToPointMAC parameter for a Provider Edge Port is always Auto, and no management control over its setting is provided. The value of the operPointToPointMAC parameter, used by the RSTP state machines, shall be true if the service instance corresponding to the Provider Edge Port connects at most two customer interfaces, and false otherwise.

The value of the AdminEdge, AutoEdge, and operEdge parameters for a Provider Edge Port are always false, true, and false, respectively. No management control over their setting is provided.

13.41.2 ~~13.39.2~~ updtRolesTree()

The spanning tree priority vectors timer values are calculated, and the port role for each port, its port priority vector, and timer information updated as specified in 13.29.34, with one exception. If selectedRole was to be set to AlternatePort, the port is an Provider Edge Port, and the root priority vector was derived from another Provider Edge Port, then the selectedRole shall be set to Root Port.

NOTE—The effect of this enhancement is to allow the C-VLAN component to have multiple Root Ports (just as if separate per S-VLAN trees were being provided), if they are all Provider Edge Ports. As the C-VLAN component assigns each frame to a single C-VLAN and maps any given C-VLAN to and from at most one Provider Edge Port, no loop is created.

13.41.3 ~~13.39.3~~ setReRootTree(), setSyncTree(), setTcPropTree()

The setReRootTree() and setSyncTree() procedures specified in 13.29.21 and 13.29.23 set the reRoot and sync variables for all ports of the bridge, and the setTcPropTree() as specified in 13.29.25 sets the tcProp variable for all ports other than the port that invoked the procedure. If the port invoking the procedure is a Customer Edge Port, then this behavior is unchanged; if it is a Provider Edge Port, then the behavior of each procedure shall be as follows.

The setReRootTree() procedure sets reRoot for the port invoking the procedure and for the Customer Edge Port.

The setSyncTree() procedure sets sync for the port invoking the procedure and for the Customer Edge Port.

The setTcPropTree() procedure sets tcProp for the Customer Edge Port.

13.41.4 ~~13.39.4~~ allSynced, reRooted

RSTP specifies a single value of the allSynced and reRooted state machine conditions for all Bridge Ports. This specification requires an independent value of each of these conditions for each port of the C-VLAN component. If that port is the Customer Edge Port, then allSynced shall be true if and only if synced is true for all Provider Edge Ports, and reRooted shall be true if and only if rrWhile is zero for all Provider Edge Ports. If the port for which the condition is being evaluated is a Provider Edge Port, then allSynced shall take the value of synced for the Customer Edge Port, and reRooted shall be true if and only if rrWhile is zero for the Customer Edge Port.

13.41.5 ~~13.39.5~~ Configuration parameters

All configuration parameters for RSTP should be set to their recommended defaults, with the exception of the following, which are chosen to minimize the chance of interfering with the customer's configuration (e.g., by the C-VLAN component becoming the root of the customer spanning tree), as follows:

a) The Bridge Priority (13.18, Table 13-3, 13.26.2) should be set to 61 440. This sets the priority part of the Bridge Identifier (the most significant 4 bits) to hex F.
b) The following 12 bits (the Bridge Identifier system ID extension) should be set to hex FFF.
c) The Port Priority (13.18, Table 13-3, 13.27.47) should be set to 32. This sets the priority part of the Port Identifier (the most significant 4 bits) to hex 2, a higher priority than the default (128, or hex 8).
d) The Port Path Cost values for Provider Edge Ports should be set to 128.

All BPDUs generated by the Spanning Tree Protocol Entity within a C-VLAN component use the MAC address of the Customer Edge Port as a source address and as the bridge address portion of the Bridge Identifier. For each internal Provider Edge Port, the protocol uses the S-VID associated with the corresponding internal Customer Network Port on the S-VLAN component as a port number. For the Customer Edge Port, the value 0xFFF is used as the port number.

13.42 ~~13.40~~ Virtual Instance Port Spanning Tree operation

This subclause specifies the operation of the Spanning Tree Protocol Entity within an I-component in a Backbone Edge Bridge. The Customer Network Ports (CNP) and Virtual Instance Ports (VIP) are treated as separate Bridge Ports by the spanning tree protocol.

If the I-component has a single CNP and a single VIP supported by a point-to-point backbone service instance, then all frames (including Spanning Tree BPDUs) addressed to the Provider Bridge Group address may be relayed between the two ports of the I-component without modification. Otherwise, the Spanning Tree Protocol Entity shall execute RSTP, as modified by the provisions of this subclause.

The RSTP enhancements specified ensure that connectivity for frames assigned to any given S-VLAN is loop-free.

The parameters and functions of the RSTP protocol used on the VIPs get the same values and functionality as defined for Provider Edge Ports of a C-VLAN component as defined in 13.41. The Bridge Identifier Priority and system ID extension get the values specified in 13.41.5. These changes in the RSTP protocol ensure that no VIP is blocked due to the operation of the RSTP protocol and the I-component will never be elected as root.

NOTE—The effect of not blocking any VIP in the I-component (never set the port role alternate to a VIP) will not cause a loop since the I-component maps any given S-VID to at most one VIP.

Change Clause 14 as follows:

14. Encoding of Bridge Protocol Data Units (BPDUs)~~Use of BPDUs by MSTP~~

~~This clause specifies the BPDU formats, encoding, and decoding used to exchange protocol parameters with other Bridges operating MSTP, RSTP, or STP, by a Bridge Protocol Entity operating MSTP (Clause 13).~~

This clause specifies formats, and the encoding, decoding, and validation of BPDUs exchanged by protocol entities operating RSTP, MSTP, and other protocols enhancing or designed to interoperate with STP, RSTP, or MSTP (Clause 13).

NOTE—The text of this clause was updated by the IEEE Std 802.1aq amendment to avoid using references and specified changes to IEEE Std 802.1D-2004 for a number of normative provisions, in order to make the standard both easier to use and to develop, but without making technical changes in the use of BPDUs by RSTP or MSTP.

Interoperability with STP, the Spanning Tree Protocol specified in IEEE Std 802.1D, 1998 Edition, and prior revisions, is provided as specified by the state machines and procedures in Clause 13 of this standard. Parameter type encodings for STP BPDUs are a subset of those used for RSTP, MSTP, and SPT BPDUs, and the fields encoded in STP Configuration BPDUs are a subset of those used in RST BPDUs.

The format of MST BPDUs is compatible with that specified for RST BPDUs, with the addition of fields to convey information for the IST and each MSTI, and is shown in Figure 14-1.

The format of SPT BPDUs comprises the fields specified for MST BPDUs, with the addition of fields that communicate the Agreement Number, Discarded Agreement Number, and Agreement Digest (13.17).

The Protocol Version Identifier encoded in all BPDUs serves to distinguish RST BPDUs, MST BPDUs, and SPT BPDUs. A BPDU Type is also encoded in all BPDUs, and distinguishes the Configuration and TCN BPDUs used by STP and by the spanning tree protocols specified in this standard when interoperating with STP implementations.

Figure 14-1 shows the overall format of RST, MST, SPT, and STP Configuration BPDUs. Figure 14-2 shows the format of STP TCN BPDUs.

14.1 BPDU Structure

14.1.1 Transmission and representation of octets

All BPDUs shall contain an integral number of octets. The octets in a BPDU are numbered starting from 1 and increasing in the order they are put into a Data Link Service Data Unit (DLSDU).When bit positions in an octet or a sequence of octets encode a number, the number is encoded as an unsigned binary numeral with bit positions in lower octet numbers having more significance. Within an octet the bits are numbered from 8 to 1, where 1 is the low-order bit. Where sequences of bits are represented, high order bits are shown to the left of lower order bits in the same octet, and bits in lower octet numbers are shown to the left of bits in higher octet numbers.

	Octet
Protocol Identifier	1–2
Protocol Version Identifier	3
BPDU Type	4
CIST Flags	5
CIST Root Identifier	6–13
CIST External Path Cost	14–17
CIST Regional Root Identifier	18–25
CIST Port Identifier	26–27
Message Age	28–29
Max Age	30–31
Hello Time	32–33
Forward Delay	34–35
Version 1 Length = 0 (RST, MST, SPT BPDUs only)	36
Version 3 Length (MST, SPT BPDUs only)	37–38
MST Configuration Identifier	39–89
CIST Internal Root Path Cost	90–93
CIST Bridge Identifier	94–101
CIST Remaining Hops	102
MSTI Configuration Messages (may be absent)	103–*39* + *Version 3 Length*
Version 4 Length (SPT BPDUs only)	(40 + *Version 3 Length)* – (41 + *Version 3 Length)*
Auxiliary MCID (SPT BPDUs only)	(42 + *Version 3 Length)* – (92 + *Version 3 Length*)
SPT Agreement Number, Discarded Agreement Number, Agreement Digest (SPT BPDUs only)	(93 + *Version 3 Length)* – (41 + *Version 3 Length* + *Version 4 Length)*

NOTE—BPDUs are encoded in LLC Type 1 frames following the DSAP, LSAP, and UI fields, so if that frame is received from directly an IEEE 802.3 MAC with the MAC Addresses aligned on an even octet boundaries then the BPDU octet pairs 6 and 7, 14 and 15, 18 and 19, etc. will also be aligned on those boundaries.

Figure 14-1—RST, MST, SPT, and STP Configuration BPDU format

	Octet
Protocol Identifier	1–2
Protocol Version Identifier	3
BPDU Type	4

Figure 14-2—STP TCN BPDU format

14.1.2 Common BPDU fields ~~Components~~

A Protocol Identifier is encoded in the initial octets of all BPDUs. The single Protocol Identifier value of 0000 0000 0000 0000 identifies the Spanning Tree family of protocols (the Spanning Tree Algorithm and Protocol, the Rapid Spanning Tree Algorithm and Protocol, and the Multiple Spanning Tree Protocol) and BPDUs whose formats are compatible with:

a) STP BPDUs, as specified in this clause (Clause 14) and IEEE Std 802.1D, 1998 Edition
b) RST BPDUs, as specified in this clause and Clause 9 of IEEE Std 802.1D-2004
c) The BPDUs specified in IEEE Std 802.1G, now withdrawn
d) MST BPDUs, as specified in this clause
e) SPT BPDUs, as specified in this clause

NOTE—ISIS-SPB (Clause 27, Clause 28) sends and receives its own PDUs to support the distributed computation of symmetric SPTs.

A Protocol Version Identifier and BPDU Type are also encoded in all BPDUs.

14.2 Encoding of parameter types

~~The following parameter types are encoded as specified in 9.2 of IEEE Std 802.1D:~~

~~a) Protocol Identifiers~~
~~b) Protocol Version Identifiers~~
~~c) BPDU Types~~
~~d) Flags~~
~~e) Port Identifiers~~
~~f) Timer values~~
~~g) Length values~~

~~Additional considerations follow for encoding:~~

~~h) Port Roles~~
~~i) Bridge Identifiers~~
~~j) Port Identifiers~~
~~k) External Root Path Costs~~
~~l) Internal Root Path Costs~~

~~This standard specifies new or extended parameter types and encodings for~~

~~m) Hop Counts~~

14.2.1 Encoding of ~~Port Role values~~Protocol Identifiers

~~Port Role values shall be encoded in two consecutive flag bits, taken to represent an unsigned integer, as follows:~~

~~a) A value of 0 indicates Master Port;~~
~~b) A value of 1 indicates Alternate or Backup;~~
~~c) A value of 2 indicates Root;~~
~~d) A value of 3 indicates Designated.~~

A Protocol Identifier shall be encoded in two octets.

14.2.2 ~~Allocation and encoding of Bridge Identifiers~~Encoding of Protocol Version Identifiers

~~The 12-bit system ID extension component of a Bridge Identifier (9.2.5 of IEEE Std 802.1D) is used toallocate distinct Bridge Identifiers to each Spanning Tree instance supported by the operation of MSTP, based on the use of a single Bridge Address component value for the MST Bridge as a whole. The system ID extension value zero shall be allocated to the Bridge Identifier used by MSTP in support of the CIST; the system ID extension value allocated to the Bridge Identifier used by a given MSTI shall be equal to the MSTID.~~

~~NOTE 1—This convention is used to convey the MSTID for each MSTI parameter set in an MST BPDU.~~

~~The four most significant bits of the Bridge Identifier for a given Spanning Tree instance (the settable Priority component) can be modified independently of the other Bridge Identifiers supported by the Bridge, allowing full configuration control to be exerted over each Spanning Tree instance with regard to bridge priority.~~

~~NOTE 2—Only these four bits of the transmitting Bridge's Bridge Identifier are encoded in BPDUs for each MSTI. The remainder of the Bridge Identifier is derived from the CIST Bridge Identifier and the MSTID using the system ID extension convention described previously.~~

A Protocol Version Identifier shall be encoded in one octet. If two Protocol Version Identifiers are interpreted as unsigned binary numbers, the greater identifies the more recently defined Protocol Version.

14.2.3 ~~Allocation and encoding of Port Identifiers~~Encoding of BPDU types

~~The four most significant bits of the Port Identifier for a given Spanning Tree instance (the settable Priority component) can be modified independently for each Spanning Tree instance supported by the Bridge. NOTE—Only these four bits of the transmitting Bridge's Port Identifier are encoded in a BPDU for each MSTI. The remainder of the Port Identifier is derived from the CIST Port Identifier.~~

The type of the BPDU shall be encoded as a single octet. The bit pattern contained in the octet merely serves to distinguish the type; no ordering relationship between BPDUs of different types is implied.

14.2.4 Encoding of ~~External Root Path Cost~~flags

~~The External Root Path Cost shall be encoded as specified by 9.2.6 of IEEE Std 802.1D for Root Path Cost in four octets, taken to represent a number of arbitrary cost units. Subclause 17.4 of IEEE Std 802.1D be placed on this parameter without requiring a management installation practice for Bridges in a network.~~

A flag shall be encoded as a bit in a single octet. A flag is set if the bit takes the value 1. A number of flags may be encoded in a single octet. Bits in the octet that do not correspond to flags defined for the BPDU's type are reset, i.e., shall take the value 0. No additional flags will be defined for a BPDU of given protocol version and type.

14.2.5 Encoding of ~~Internal Root Path Cost~~Bridge Identifiers

~~The Internal Root Path Cost shall be encoded in four octets, taken to represent a number of arbitrary cost units that may differ from those used for External Path Cost. Table 13-4 contains recommendations for the use of these units. These recommendations allow higher LAN speeds to be represented in support of both current and future technologies, while still allowing common values to be assigned without a management installation practice.~~

~~NOTE—This revision from the original IEEE Std 802.1D recommendations for STP Path Cost causes no operational difficulties because there was no installed base of Bridges using the Internal Root Path Cost parameter prior to approval of this standard.~~

A Bridge Identifier is a 64-bit unsigned integer. If two Bridge Identifiers are numerically compared, the lesser number denotes the Bridge of the better priority.

NOTE 1—Use of the terms “higher” and “lower” to describe both the relative numerical values and the relative priority of Spanning Tree information can cause confusion, as lesser numbers convey better priorities. In this clause and in Clause 13, relative numeric values are described as “least,” “lesser,” “equal,” and “greater,” and their comparisons as “less than,” “equal to,” or “greater than,” while relative Spanning Tree priorities are described as “best,” “better,” “the same,” “different,” and “worse” and their comparisons as “better than,” “the same as,” “different from,” and “worse than.” The terms “superior” and “inferior” describe comparisons not simply based on strict ordered comparison of priority components.

The 4 most significant bits of a Bridge Identifier comprise a settable priority component (13.20, 13.26.2), and can be modified for the CIST or for an MSTI independently of those for other MSTIs (or for the CIST), allowing the Bridge Identifier priority to be used to provide full and independent control over the configuration of each of those trees. Each SPT uses the same Bridge Identifier as the CIST, so SPT Bridge Identifiers are not separately encoded.

NOTE 2—To maintain management compatibility with implementations of IEEE Std 802.1D, 1998 Edition, and prior revisions, and of IEEE Std 802.1Q prior to the IEEE Std 802.1s-2002 amendment, the priority component is considered to be a 16-bit value for management purposes, but the values that it can be set to are restricted to only those values where the least significant 12 bits are zero (i.e., only the most significant 4 bits are settable).

The next most significant twelve bits of a Bridge Identifier (when encoded, the four least significant bits of the most significant octet plus the second most significant octet) comprise a locally assigned system ID extension, that provides a distinct Bridge Identifier for each MSTI. The CIST is identified by the system ID extension of zero. Each MSTI has a system ID extension value equal to its MSTID, a convention used to convey the MSTID for each MSTI parameter set in an MST BPDU.

The 48 least significant bits of the Bridge Identifier ensure its uniqueness and are derived from the globally unique Bridge Address (8.13.8) and are encoded in the third through eight most significant octets of the encoded identifier according to the following procedure.

The third most significant octet is derived from the initial octet of the MAC Address; the least significant bit of the octet (Bit 1) is assigned the value of the first bit of the Bridge Address, the next most significant bit is assigned the value of the second bit of the Bridge Address, and so on. The fourth through eighth octets are similarly assigned the values of the second to the sixth octets of the Bridge Address.

14.2.6 Encoding of ~~Hop Counts~~External Root Path Cost and Internal Root Path Cost

~~The number of remaining Hops parameter shall be encoded in a single octet.~~

The External Root Path Cost shall be encoded in four octets as a number of arbitrary cost units. The Internal Root Path Cost is similarly encoded in four octets. Subclause 13.18 recommends Port Path Cost values so that a common interpretation can be placed on path cost values, and reasonable spanning tree configurations obtained without explicit management.

NOTE—IEEE 802.1D refers to the External Root Path Cost as Root Path Cost and does not use Internal Root Path Cost.

14.2.7 Encoding of Port Identifiers

A Port Identifier is a 16-bit unsigned integer. If two Port Identifiers are numerically compared, the lesser number denotes the port of better priority. The 4 most significant bits of a Port Identifier is a settable priority component that permits the relative priority of ports on the same bridge to be managed (13.27.46), and can be modified independently for the CIST and each MSTI. The less significant 12 bits is the Port Number expressed as an unsigned binary number, and is same for the CIST and all MSTIs. The value 0 is not used as a Port Number.

The Port Identifier for the CIST is encoded in two octets, comprising both the CIST's priority component and the Port Number used for all spanning trees. The 4-bit priority component for each MSTI is encoded as a binary number, in the MSTI Configuration Message for that MSTI. Each SPT uses the same Port Identifier as the CIST, so SPT Port Identifiers are not separately encoded.

NOTE—IEEE Std 802.1D, 1998 Edition, and prior revisions specified a priority component of 8 bits and a Port Number of 8 bits. To maintain management compatibility with prior implementations the priority component is still considered to be an 8-bit value, but its values are restricted to those where the least significant 4 bits are zero (and hence ignored in encoding).

14.2.8 Encoding of Timer Values

Timer Values shall be encoded in two octets, taken to represent an unsigned binary number multiplied by a unit of time of 1/256 of a second. This permits times in the range 0 to, but not including, 256 s to be represented.

14.2.9 Encoding of Port Role values

Port Role values shall be encoded in two consecutive flag bits, taken to represent an unsigned integer, as follows:

a) A value of 0 indicates Master Port;
b) A value of 1 indicates Alternate or Backup;
c) A value of 2 indicates Root;
d) A value of 3 indicates Designated.

NOTE—IEEE Std 802.1D-2004 identified the Port Role value of 0 as Unknown, as it not used as a CIST Port Role in transmitted BPDUs. A received BPDU with a CIST Port Role value of 0 is identified as a Configuration BPDU.

14.2.10 Encoding of Length Values

Version 1 Length Values are encoded in one octet, taken to represent an unsigned binary number. No further length values are encoded for Version 2. Length Values for Version 3 and 4 are encoded in two octets.

14.2.11 Encoding of Hop Counts

The number of remaining Hops parameter shall be encoded in a single octet.

~~14.3 BPDU formats and parameters~~

~~14.3.1 STP BPDUs~~

~~The formats of STP BPDU Configuration and TCN BPDUs are as specified in Clause 9 of IEEE Std 802.1D.~~

~~14.3.2 RST BPDUs~~

~~The format of RST BPDUs is as specified in Clause 9 of IEEE Std 802.1D.~~

~~14.3.3 MST BPDUs~~

~~The format of MST BPDUs is compatible with that specified for RST BPDUs (Clause 9 of IEEE Std 802.1D), with the addition of fields to convey information for the IST and each MSTI and is shown in Figure 14-1. Each transmitted MST BPDU shall contain the parameters specified and no others. NOTE—The BPDU specified in this clause is carried in an LLC Type 1 frame following the DSAP, LSAP, and UI fields~~

~~(7.12 on Addressing in IEEE Std 802.1D). The consequence of the inclusion of those three octets in an IEEE 802.3 or Ethernet MAC frame is that, if the MAC Addresses in the frame are aligned on an even octet boundary, then so are the BPDU octet pairs 6 and 7, 14 and 15, 18 and 19, etc.~~

~~14.4 Validation of received BPDUs~~

See new 14.5 for former 14.4 text.

14.3 ~~14.5~~ Transmission of BPDUs

A~~n MST~~ Bridge Protocol Entity shall encode 0000 0000 0000 0000 in Octets 1 and 2 (conveying the Protocol Identifier), the remaining fields shall be encoded to convey an STP Configuration BPDU, an STP TCN BPDU, an RST BPDU, or an MST BDU as required by the Force Protocol Version parameter, the Port Protocol Migration state machine, and other protocol parameters, all as specified in Clause 13.

a) If transmission of an STP Configuration BPDU is required, the Protocol Version Identifier shall be 0, and the BPDU Type shall be 0000 0000.
b) If transmission of an STP TCN BPDU is required, the Protocol Version Identifier shall be 0, and the BPDU Type shall be 1000 0000.
c) If transmission of an RST BPDU is required, the Protocol Version Identifier shall be 2, and the BPDU Type shall be 0000 0010.
d) If transmission of an MST BPDU is required, the Protocol Version Identifier shall be 3, and the BPDU Type shall be 0000 0010.
e) If transmission of an SPT BPDU is required, the Protocol Version Identifier shall be 4, and the BPDU Type shall be 0000 0010.

The remaining parameters for STP Configuration, RST, ~~and~~ MST, and SPT BPDUs shall be encoded as specified below (14.4).

14.4 ~~14.6~~ Encoding and decoding of STP Configuration, RST, ~~and~~ MST, and SPT BPDUs

STP Configuration, RST, ~~and~~ MST, and SPT BPDU protocol parameters are encoded for transmission, and decoded, checked or ignored on receipt as follows:

a) Bit 1 of Octet 5 conveys the CIST Topology Change flag.
b) Bit 2 of Octet 5 conveys the CIST Proposal flag in RST, ~~and~~ MST, and SPT BPDUs. It is unused in STP Configuration BPDUs, and shall be transmitted as 0 and ignored on receipt.
c) Bits 3 and 4 of Octet 5 conveys the CIST Port Role in RST, ~~and~~ MST, and SPT BPDUs. It is unused in STP Configuration BPDUs, and shall be transmitted as 0 and ignored on receipt.
d) Bit 5 of Octet 5 conveys the CIST Learning flag in RST, ~~and~~ MST, and SPT BPDUs. It is unused in STP Configuration BPDUs, and shall be transmitted as 0 and ignored on receipt.
e) Bit 6 of Octet 5 conveys the CIST Forwarding flag in RST, ~~and~~ MST, and SPT BPDUs. It is unused in STP Configuration BPDUs, and shall be transmitted as 0 and ignored on receipt.
f) Bit 7 of Octet 5 conveys the CIST Agreement flag in RST, ~~and~~ MST, and SPT BPDUs. It is unused in STP Configuration BPDUs, and shall be transmitted as 0 and ignored on receipt.
g) Bit 8 of Octet 5 conveys the Topology Change Acknowledge Flag in STP Configuration BPDUs. It is unused in RST, ~~and~~ MST, and SPT BPDUs, and shall be transmitted as 0 and ignored on receipt.
h) Octets 6 through 13 convey the CIST Root Identifier.

NOTE 1—The 12 bit system id extension component of the CIST Root Identifier can be received and subsequently transmitted as an arbitrary value, even in MST BPDUs, since the CIST Root may be an STP Bridge.

i) Octets 14 through 17 convey the CIST External Root Path Cost.

j) Octets 18 through 25 shall take the value of the CIST Regional Root Identifier when transmitted in RST and MST BPDUs, and the value of the CIST Bridge Identifier of the transmitting bridge when transmitted in STP Configuration BPDUs. On receipt of an STP Configuration or RST BPDU both the CIST Regional Root Identifier and the CIST Designated Bridge Identifier shall be decoded from this field. On receipt of an MST BPDU the CIST Regional Root Identifier shall be decoded from this field.

k) Octets 26 and 27 convey the CIST Port Identifier of the transmitting Bridge Port.

l) Octets 28 and 29 convey the Message Age timer value.

m) Octets 30 and 31 convey the Max Age timer value.

n) Octets 32 and 33 convey the Hello Time timer value used by the transmitting Bridge Port.

o) Octets 34 and 35 convey the ~~Max Age~~ Forward Delay timer value.

No further octets shall be encoded in STP Configuration BPDUs. Additional octets in received BPDUs identified by the validation procedure (14.5) as STP Configuration BPDUs shall be ignored. The specification of encoding or decoding of further octets in this subclause refers only to RST, ~~and~~ MST, and SPT BPDUs.

p) Octet 36 conveys the Version 1 Length. This shall be transmitted as 0. It is checked on receipt by the validation procedure (14.5).

No further octets shall be encoded in RST BPDUs. Additional octets in received BPDUs identified by the validation procedure (14.5) as RST BPDUs shall be ignored. The specification of encoding or decoding of further octets in this subclause refers only to MST and SPT BPDUs.

NOTE 2—As Version 2 does not specify any additional fields beyond the end of the Version 0 information, there is no Version 2 Length field specified in Version 2 of the protocol (see Clause 9 of IEEE Std 802.1D-2004), and therefore no need for a Version 2 length field here.

q) Octets 37 and 38 convey the Version 3 Length. Its value is the number of octets taken by the parameters that follow in the BPDU. It is checked on receipt by the validation procedure (14.5).

r) Octets 39 through 89 convey the elements of the MST Configuration Identifier (13.8):
   1) The Configuration Identifier Format Selector is encoded in octet 39 and shall take the value 0000 0000;
   2) The Configuration Name is encoded in octets 40 through 71;
   3) The Revision Level is encoded as a number in octets 72 through 73;
   4) The Configuration Digest is encoded in octets 74 through 89.

s) Octets 90 through 93 convey the CIST Internal Root Path Cost.

t) Octets 94 through 101 convey the CIST Bridge Identifier of the transmitting bridge. The 12-bit system id extension component of the CIST Bridge Identifier shall be transmitted as 0. The behavior on receipt is unspecified if it is non-zero.

NOTE 3—The 4 most significant bits of the Bridge Identifier constitute the manageable priority component for each MSTI and are separately encoded in MSTI Configuration Messages in the BPDU.

~~NOTE 4—The 4 most significant bits constitute the manageable priority component of each MSTI and are separately encoded in MSTI Configuration Messages in the BPDU.~~

u) Octet 102 encodes the value of remaining Hops for the CIST.

v) A sequence of zero or more, up to a maximum of 64, MSTI Configuration Messages follows, each encoded as specified in 14.4.1.

No further octets shall be encoded in MST Configuration BPDUs. Additional octets in received BPDUs identified by the validation procedure (14.5) as MST Configuration BPDUs shall be ignored. The specification of encoding or decoding of further octets in this subclause refers only to SPT BPDUs.

w) Octets 1 and 2 following the Version 3 information convey the Version 4 Length. Its value is the number of octets that follow the Version 4 Length in the BPDU, up to but not including octets added by subsequent revisions of this standard and associated with Versions 5 or greater. It is checked on receipt by the validation procedure (14.5).
x) Octets 1 through 51 following the Version 4 Length encode the Auxiliary MCID (if present) as follows:
   1) The Configuration Identifier Format Selector is encoded in octet 1 and shall take the value 0000 0000;
   2) The Configuration Name is encoded in octets 2 through 33;
   3) The Revision Level is encoded as a number in octets 34 through 35;
   4) The Configuration Digest is encoded in octets 36 through 51.
y) Octets 1 and 2 following the Auxiliary MCID are encoded (if present) as follows:
   1) Bits 1 and 2 of Octet 1 convey the Agreement Number (13.27.11, 13.27.16).
   2) Bits 3 and 4 of Octet 1 convey the Discarded Agreement Number (13.27.12, 13.27.17).
   3) Bit 5 of Octet 1 conveys the Agreement Valid flag (13.27.7, 13.27.9).
   4) Bit 6 of Octet 1 conveys the Restricted Role flag (13.27.63, 27.19).
   5) Bits 7 through 8 of Octet 1 are unused, and shall be transmitted as 0 and ignored on receipt.
   6) Octet 2 is unused, and shall be transmitted as 0 and ignored on receipt.
z) The remaining Octets following the Version 4 Length and covered by that length comprise the Agreement Digest (13.17.1, 28.4).

No further octets shall be encoded in BPDUs. Additional octets in received BPDUs shall be ignored.

14.4.1 ~~14.6.1~~MSTI Configuration Messages

A single instance of the following set of parameters is encoded for each MSTI supported by the transmitting bridge.

a) Bits 1, 2, 3 and 4, 5, 6, 7, and 8, respectively, of Octet 1 convey the Topology Change flag, Proposal flag, Port Role, Learning flag, Forwarding flag, Agreement flag, and Master flag for this MSTI.
b) Octets 2 through 9 convey the Regional Root Identifier (13.27.20) as illustrated in Figure 14-3 ~~Figure 14-2~~. This includes the value of the MSTID for this Configuration Message encoded in bits 4 through 1 of Octet 1, and bits 8 through 1 of Octet 2.

NOTE—The 4 most significant bits of each MSTI’s Regional Root Identifier constitute a manageable priority component.

c) Octets 10 through 13 convey the Internal Root Path Cost.
d) Bits 5 through 8 of Octet 14 convey the value of the Bridge Identifier Priority for this MSTI. Bits 1 through 4 of Octet 14 shall be transmitted as 0, and ignored on receipt.
e) Bits 5 through 8 of Octet 15 convey the value of the Port Identifier Priority for this MSTI. Bits 1 through 4 of Octet 15 shall be transmitted as 0, and ignored on receipt.
f) Octet 16 conveys the value of remainingHops for this MSTI (13.27.21).

	Octet
MSTI Flags	1
MSTI Regional Root Identifier	2–9
MSTI Internal Root Path Cost	10–13
MSTI Bridge Priority	14
MSTI Port Priority	15
MSTI Remaining Hops	16

Figure 14-3—~~Figure 14.2—~~MSTI Configuration Message parameters and format

14.5 ~~14.4~~ Validation of received BPDUs

The receiving protocol entity ~~An MST Bridge Protocol Entity~~ shall examine Octets 1 and 2 (conveying the Protocol Identifier), Octet 3 (conveying the Protocol Version Identifier encoded as a number), Octet 4 (conveying the BPDU Type) and the total length of the received BPDU (including the preceding fields, but none prior to the Protocol Identifier) to determine the further processing required as follows:

a) If the Protocol Identifier is 0000 0000 0000 0000, the BPDU Type is 0000 0000, and the BPDU contains 35 or more octets, it shall be decoded as an STP Configuration BPDU.

b) If the Protocol Identifier is 0000 0000 0000 0000, the BPDU Type is 1000 0000 (where bit 8 is shown at the left of the sequence), and the BPDU contains 4 or more octets, it shall be decoded as an STP TCN BPDU (9.3.2 of IEEE Std 802.1D-2004).

c) If the Protocol Identifier is 0000 0000 0000 0000, the Protocol Version Identifier is 2, and the BPDU Type is 0000 0010 (where bit 8 is shown at the left of the sequence), and the BPDU contains 36 or more octets, it shall be decoded as an RST BPDU.

d) If the Protocol Identifier is 0000 0000 0000 0000, the Protocol Version Identifier is 3 or greater, the BPDU Type is 0000 0010, the bridge is configured as an MST Bridge or an SPT Bridge or according to a future revision of this standard that intends to provide interoperability with prior revisions, and the BPDU:
   1) Contains 35 or more but less than 103 octets; or
   2) Contains a Version 1 Length that is not 0; or
   3) Contains a Version 3 length that does not represent an integral number, from 0 to 64 inclusive, of MSTI Configuration Messages,

   it shall be decoded as an RST BPDU.

e) If the Protocol Identifier is 0000 0000 0000 0000, the Protocol Version Identifier is 3 or greater and the bridge is configured as an MST Bridge or the Protocol Version Identifier is 3 and the bridge is configured as an SPT Bridge or according to a future revision of this standard that intends to provide interoperability with prior revisions, the BPDU Type is 0000 0010, and the BPDU:
   1) Contains 102 or more octets; and
   2) Contains a Version 1 Length of 0; and
   3) Contains a Version 3 length representing an integral number, from 0 to 64 inclusive, of MSTI Configuration Messages,

   it shall be decoded as an MST BPDU.

f) If the Protocol Identifier is 0000 0000 0000 0000, the Protocol Version Identifier is 3 or greater, the BPDU Type is 0000 0010, the bridge is configured as an SPT Bridge or according to a future revision of this standard that intends to provide interoperability with prior revisions, and the BPDU:
   1) Contains 102 or more octets; and
   2) Contains a Version 1 Length of 0; and
   3) Contains a Version 3 length representing an integral number, from 0 to 64 inclusive, of MSTI Configuration Messages; and
   4) Is not a well-formed an SPT BPDU, i.e.,
      i) Contains less than 6 octets following the octets specified by the Version 3 length;
      ii) Has a Version 4 length that is less than 55;
      iii) Does not contain an MST Configuration Identifier Format Selector of 1,

   it shall be decoded as an MST BPDU.

g) If the Protocol Identifier is 0000 0000 0000 0000, the Protocol Version Identifier is 4 or greater, the BPDU Type is 0000 0010, the bridge is configured as an MST Bridge or an SPT Bridge or according to a future revision of this standard that intends to provide interoperability with prior revisions, and the BPDU:
   1) Contains 106 or more octets; and
   2) Contains a Version 1 Length of 0; and
   3) Contains a Version 3 length representing an integral number, from 0 to 64 inclusive, of MSTI Configuration Messages; and
   4) Is a well-formed an SPT BPDU, i.e., contains:

i) At least 6 octets following the octets specified by the Version 3 length
ii) A Version 4 length of 55 or greater
iii) An MST Configuration Identifier Format Selector of 1,

it shall be decoded as an SPT BPDU.

h) ~~f)~~ Otherwise the BPDU shall be discarded and not processed.

NOTE 1—The LLC LSAP that identifies BPDUs is reserved for standard protocols, no other protocols using that LSAP have been standardized though they may be at some future time. At that time, BPDUs with different Protocol Identifiers may be processed according to the rules of those protocols but will still be discarded from the point of view of MSTP.

NOTE 2—These validation rules do not contain a loopback check of the form specified for use with STP Configuration BPDUs in 9.3.4 of IEEE Std 802.1D-2004.

~~NOTE 2—These validation rules are in accord with the approach to backward compatibility of future version enhancements set out in 9.3.4 of IEEE Std 802.1D-2004. Test a) and test b) do not check the Protocol Version Identifier.~~

~~NOTE 3—These validation rules do not contain a loopback check of the form specified in 9.3.4 of IEEE Std 802.1D-2004.~~

14.6 Validation and interoperability

The validation rules above (14.5) follow a consistent general approach that allows future version enhancements to be made while retaining backwards compatibility. In particular, tests (a) and (b) in 14.5 above do not check the Protocol Version Identifier.

In general, for an implementation that supports version A of a given protocol, a received PDU of a given type that carries a protocol version number B is interpreted as follows:

a) Where B is greater than or equal to A, the PDU shall be interpreted as if it carried the supported version number, A. Specifically:
   1) All PDU types, parameters, and flags that are defined in version A shall be interpreted in the manner specified for version A of the protocol for the given BPDU type.
   2) All PDU types, parameters, and flags that are undefined in version A for the given BPDU type shall be ignored.
   3) All octets that appear in the PDU beyond the largest numbered octet defined for version A for the given BPDU type shall be ignored.
b) Where B is less than A, the PDU shall be interpreted as specified for the version number, B, carried in the BPDU. Specifically:
   1) All PDU parameters and flags shall be interpreted in the manner specified for version B of the protocol for the given PDU type.
   2) All PDU parameters and flags that are undefined in version B for the given BPDU type shall be ignored.
   3) All octets that appear in the PDU beyond the largest numbered octet defined for version B for the given BPDU type shall be ignored.

NOTE—In other words, if the protocol version implemented differs from the protocol version number carried in the PDU, then only those PDU types, parameters, and flags that are specified within the lesser numbered protocol version are interpreted by the implementation (in accordance with the lesser numbered protocol version's specification), and no attempt is made to interpret any additional PDU types, parameters, and flags that may be specified within the greater numbered protocol version. In the specific case of STP (version 0) and RSTP (version 2), as there is only a single RST BPDU type defined in version 2, and as the RST BPDU type is undefined in version 0, a version 0 implementation will ignore all RST BPDUs. Version 2 implementations, however, recognize and process both version 0 and version 2 BPDUs. As version 2 makes no changes to the BPDU types defined for version 0 (and always transmits such BPDU types with 0 as the version identifier), version 0 BPDUs are always interpreted by version 2 implementations according to their version 0 definition.

17. Management Information Base (MIB)

17.2 Structure of the MIB

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

Table 17-1—Structure of the MIB modules

Module	Subclause	Defining IEEE standard	Reference	Notes
IEEE8021-SPB-MIB	17.2.4	802.1aq	27, 28	Initial version in IEEE Std 802.1aq

17.2.4 Structure of the IEEE8021-Q-BRIDGE-MIB

Insert the following rows at the end of Table 17-7:

Table 17-7—IEEE8021-QBRIDGE MIB structure and relationship to IETF RFC 4363 and this standard

IEEE MIB table/object	IETF MIB table/object	Reference
ieee8021QBridgeVidXTable		12.10.1.8 12.13.2.1
ieee8021QBridgeBasePortComponentId*		
ieee8021QBridgeBasePort*		
ieee8021QBridgeVidXLocalVid*		
ieee8021QBridgeVidXRelayVid		
ieee8021QBridgeVidXRowStatus		
ieee8021QBridgeEgressVidXTable		12.10.1.9 12.13.2.1
ieee8021BridgeEgressBasePortComponentId*		
ieee8021QBridgeEgressBasePort*		
ieee8021QBridgeEgressVidXRelayVid*		
ieee8021QBridgeEgressVidXLocalVid		
ieee8021QBridgeEgressVidXRowStatus		

Insert the following list item after item e) in 17.2.4:

f) The ieee8021QBridgeX Subtree
This subtree contains objects that control configuration of VID translation for VLAN ports. A single symmetric translation value or an individual ingress and egress translation is allowed depending on the management table supported.

17.2.5 Structure of the IEEE8021-PB MIB

Delete the following rows from Table 17-9:

Table 17-9—IEEE8021-PB MIB structure and relationship to this standard

IEEE MIB table	IEEE MIB Object	Reference
~~ieee8021PbVidTranslationTable~~		~~12.13.2.1~~ ~~12.13.2.2~~
	~~ieee8021BridgeBasePortComponentId*~~	
	~~ieee8021BridgeBasePort*~~	
	~~ieee8021PbVidTranslationLocalVid*~~	
	~~ieee8021PbVidTranslationRelayVid~~	
	~~ieee8021PbVidTranslationRowStatus~~	

17.2.6 Structure of the IEEE8021-MSTP MIB

Change Table 17-10 as follows (note that the entire table is not shown here):

Table 17-10—IEEE8021-MSTP MIB structure and relationship to this standard

IEEE MIB table	IEEE MIB Object	Reference
~~ieee8021MstpFidToMstiTable~~ieee8021MstpFidToMstiV2Table		12.12.2.2
	~~ieee8021MstpFidToMstiComponentId~~ ieee8021MstpFidToMstiV2ComponentId*	
	~~ieee8021MstpFidToMstiFid~~ ieee8021MstpFidToMstiV2Fid*	
	~~ieee8021MstpFidToMstiMstpId~~ ieee8021MstpFidToMstiV2MstpId*	
~~ieee8021MstpVlanTable~~ieee8021MstpVlanV2Table		12.12.3.1
	~~ieee8021MstpVlanComponentId~~ ieee8021MstpVlanV2ComponentId*	
	~~ieee8021MstpVlanId~~ ieee8021MstpVlanV2Id*	
	~~ieee8021MstpVlanMstpId~~ ieee8021MstpVlanV2MstpId*	

Insert the following subclause, 17.2.19 (including Table 17-25), after 17.2.18.3:

17.2.19 Structure of the IEEE8021-SPB-MIB

The IEEE8021-SPB-MIB is an new MIB to control the configuration of ISIS-SPB. SPB builds on the 802.1 MIBs and reuses aspects of existing MIBs.

This SPB MIB provides objects to configure SPB defined in Clause 27 and Clause 28. This is the complete MIB for SPBV and SPBM. Table 17-25 describes the relationship between the SMIv2 objects defined in the MIB module in (17.7.19) and the variables and managed objects defined in Clause 12.

Table 17-25—IEEE8021-SPB MIB structure and relationship to this standard

Clause 17 MIB table/object	Reference
ieee8021SpbSys	12.25.1
ieee8021SpbSysAreaAddress	12.25.1.3.2, 12.25.1.3.3
ieee8021SpbSysId	12.25.1.3.3, Clause 3
ieee8021SpbSysControlAddr	12.25.1.3.3
ieee8021SpbSysName	12.25.1.3.3
ieee8021SpbSysBridgePriority	12.25.1.3.3, 13.26.3
ieee8021SpbmSysSPSourceId	12.25.1.3.3, Clause 3
ieee8021SpbvSysMode	12.25.1.3.3, Clause 3
ieee8021SpbmSysMode	12.25.1.3.3, Clause 3
ieee8021SpbSysDigestConvention	12.25.1.3.3, 28.4.3
ieee8021SpbMtidStaticTable	12.25.2
ieee8021SpbMtidStaticTableEntry	12.25.2.3.3
ieee8021SpbMtidStaticEntryMtid	12.25.1.3.2, 12.25.2.3.3, 28.12
ieee8021SpbMTidStaticEntryMtidOverload	12.25.2.3.3, 27.8.1
ieee8021SpbMtidStaticEntryRowStatus	12.25.2.3.3
ieee8021SpbTopIx	12.25.2.3.3
ieee8021SpbTopIxDynamicTable	12.25.3
ieee8021SpbTopIxDynamicTableEntry	12.25.3
ieee8021SpbTopIxDynamicEntryTopIx	12.25.3.1.2, 28.12
ieee8021SpbTopIxDynamicEntryAgreeDigest	12.25.3.1.3, 28.4
ieee8021SpbTopIxDynamicEntryMCID	12.25.3.1.3, 13.8
ieee8021SpbTopIxDynamicEntryAuxMCID	12.25.3.1.3, 28.9
ieee8021SpbEctStaticTable	12.25.4
ieee8021SpbEctStaticTableEntry	12.25.4
ieee8021SpbEctStaticEntryTopIx	12.25.4.1.2,28.12

Table 17-25—IEEE8021-SPB MIB structure and relationship to this standard *(continued)*

Clause 17 MIB table/object	Reference
ieee8021SpbEctStaticEntryBaseVid	12.25.4.1.2, Clause 3
ieee8021SpbEctStaticEntryEctAlgorithm	12.25.4.1.2, Clause 3
ieee8021SpbvEctStaticEntrySpvid	12.25.4.1.2, Clause 3
ieee8021SpbEctStaticEntryRowStatus	12.25.4.1.2
ieee8021SpbEctDynamicTable	12.25.5
ieee8021SpbEctDynamicTableEntry	12.25.5
ieee8021SpbEctDynamicEntryTopIx	12.25.5.1.2,12.25.5.1.3, 28.12
ieee8021SpbEctDynamicEntryBaseVid	12.25.5.1.2,12.25.5.1.3, Clause 3
ieee8021SpbEctDynamicEntryMode	12.25.5.1.3, 28.12.4
ieee8021SpbEctDynamicEntryLocalUse	12.25.5.1.3, 28.12.4
ieee8021SpbEctDynamicEntryRemoteUse	12.25.5.1.3, 28.12.4
ieee8021SpbEctDynamicEntryIngressCheckDiscards	12.25.5.1.3
ieee8021SpbAdjStaticTable	12.25.6
ieee8021SpbAdjStaticTableEntry	12.25.6
ieee8021SpbAdjStaticEntryTopIx	12.25.6.1.2,12.25.6.1.3,28.12
ieee8021SpbAdjStaticEntryIfIndex	12.25.6.1.2,12.25.6.1.3
ieee8021SpbAdjStaticEntryMetric	12.25.6.1.2,12.25.6.1.3,28.12.7
ieee8021SpbAdjStaticEntryIfAdminState	12.25.6.1.2,12.25.6.1.3
ieee8021SpbAdjStaticEntryRowStatus	12.25.6.1.3
ieee8021SpbAdjDynamicTable	12.25.7
ieee8021SpbAdjDynamicTableEntry	12.25.7
ieee8021SpbAdjDynamicEntryTopIx	12.25.7.1.2, 12.25.7.1.3, 28.12
ieee8021SpbAdjDynamicEntryIfIndex	12.25.7.1.2, 12.25.7.1.3
ieee8021SpbAdjDynamicEntryPeerSysId	12.25.7.1.3, Clause 3
ieee8021SpbAdjDynamicEntryPort	12.25.7.1.3
ieee8021SpbAdjDynamicEntryIfOperState	12.25.7.1.3
ieee8021SpbAdjDynamicEntryPeerSysName	12.25.7.1.3
ieee8021SpbAdjDynamicEntryPeerAgreeDigest	12.25.7.1.3, 28.4
ieee8021SpbAdjDynamicEntryPeerMCID	12.25.7.1.3, 28.12.2, 27.4.1
ieee8021SpbAdjDynamicEntryPeerAuxMCID	12.25.7.1.3, 28.12.2
ieee8021SpbAdjDynamicEntryLocalCircuitID	12.25.7.1.3, 28.11
ieee8021SpbAdjDynamicEntryPeerLocalCircuitID	12.25.7.1.3, 28.11
ieee8021SpbAdjDynamicEntryPortIdentifier	12.25.7.1.3, 28.11, 14.2.7

Table 17-25—IEEE8021-SPB MIB structure and relationship to this standard *(continued)*

Clause 17 MIB table/object	Reference
ieee8021SpbAdjDynamicEntryPeerPortIdentifier	12.25.7.1.3, 28.11, 14.2.7
ieee8021SpbAdjDynamicEntryIsisCircIndex	12.25.7.1.3, 28.11
ieee8021SpbTopNodeTable	12.25.8
ieee8021SpbTopNodeTableEntry	12.25.8
ieee8021SpbTopNodeEntryTopIx	12.25.8.1.2,12.25.8.1.3,28.12
ieee8021SpbTopNodeEntrySysId	12.25.8.1.2,12.25.8.1.3, Clause 3
ieee8021SpbTopNodeEntryBridgePriority	12.25.8.1.3,13.26.3
ieee8021SpbmTopNodeEntrySPsourceID	12.25.8.1.3, Clause 3
ieee8021SpbTopNodeEntrySysName	12.25.8.1.3
ieee8021SpbTopEctTable	12.25.9
ieee8021SpbTopEctTableEntry	12.25.9
ieee8021SpbTopEctEntryTopIx	12.25.9.1.2,12.25.9.1.3
ieee8021SpbTopEctEntrySysId	12.25.9.1.2,12.25.9.1.3, Clause 3
ieee8021SpbTopEctEntryBaseVid	12.25.9.1.2,12.25.9.1.3, Clause 3
ieee8021SpbTopEctEntryEctAlgorithm	12.25.9.1.3, Clause 3
ieee8021SpbTopEctEntryMode	12.25.9.1.3
ieee8021SpbvTopEctSysMode	12.25.9.1.3
ieee8021SpbvTopEctEntrySpvid	12.25.9.1.3
ieee8021SpbTopEctEntryLocalUse	12.25.9.1.3, 28.12.5
ieee8021SpbTopEdgeTable	12.25.10
ieee8021SpbTopEdgeTableEntry	12.25.10
ieee8021SpbTopEdgeEntryTopIx	12.25.10.1.2, 12.25.10.1.3, 28.12
ieee8021SpbTopEdgeEntrySysIdNear	12.25.10.1.2, 12.25.10.1.3, Clause 3
ieee8021SpbTopEdgeEntrySysIdFar	12.25.10.1.2, 12.25.10.1.3, Clause 3
ieee8021SpbTopEdgeEntryMetricNear2Far	12.25.10.1.3, 28.12.7
ieee8021SpbTopEdgeEntryMetricFar2Near	12.25.10.1.3, 28.12.7
ieee8021SpbmTopSrvTable	12.25.11
ieee8021SpbmTopSrvTableEntry	12.25.11
ieee8021SpbmTopSrvEntryTopIx	12.25.11.1.2,12.25.10.1.3, 28.12
ieee8021SpbmTopSrvEntrySysId	12.25.11.1.2,12.25.10.1.3, Clause 3
ieee8021SpbmTopSrvEntryIsid	12.25.11.1.2,12.25.10.1.3,28.12.10

Table 17-25—IEEE8021-SPB MIB structure and relationship to this standard *(continued)*

Clause 17 MIB table/object	Reference
ieee8021SpbmTopSrvEntryBaseVid	12.25.11.1.2,12.25.10.1.3,28.12.10
ieee8021SpbmTopSrvEntryIsidFlags	12.25.11.1.2,12.25.10.1.3,28.12.10
ieee8021SpbmTopSrvEntryMac	12.25.10.1.3,28.12.10
ieee8021SpbvTopSrvTable	12.25.12
ieee8021SpbvTopSrvTableEntry	12.25.12
ieee8021SpbvTopSrvEntryTopIx	12.25.12.1.2,12.25.12.1.3, 28.12
ieee8021SpbvTopSrvEntrySysId	12.25.12.1.2,12.25.12.1.3, Clause 3
ieee8021SpbvTopSrvEntryMMac	12.25.12.1.2,12.25.12.1.3, 28.12.9
ieee8021SpbvTopSrvEntryMMacFlags	12.25.12.1.3,28.12.9
ieee8021SpbvTopSrvEntryBaseVid	12.25.12.1.3, Clause 3

17.3 Relationship to other MIBs

17.3.5 Relationship of the IEEE8021-PB-BRIDGE MIB to other MIB modules

Change 17.3.5 as follows:

To supplement the Q-BRIDGE-MIB, this module contains the following:

1) ~~Support for VID translation.~~ The Q-Bridge MIB directly supports VID translation. The original bidirectional VID translation of PB MIB is supported by the Q-Bridge MIB as well as separate ingress and egress VID translation.
2) Support for Provider Edge Bridges.

Insert the following subclause, 17.3.19, after 17.3.18:

17.3.19 Relationship of the of the IEEE8021-SPB-MIB to other MIB modules

The IEEE8021-SPB-MIB, because it provides mainly control plane functions and reuses VLANs in the Q-BRIDGE-MIB depends on that MIB. It also imports items from the SNMPv2-SMI, SNMPv2-TC, IF-MIB and IEEE8021-TC-MIB, and requires SNMPv2-CONF for conformance.

17.4 Security considerations

Insert the following subclause, 17.4.19, after 17.4.18:

17.4.19 Security considerations of the IEEE8021-SPB MIB

There are a number of management objects defined in this 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 tables and objects and their sensitivity/vulnerability are described next.

ieee8021SpbMtidStaticEntryMtid

The following tables and objects in the SPB-MIB could be manipulated to interfere with the operation of Shortest Path Bridges. This could, for example, be used to misconfigure the network to cause loss of connectivity, or misconnect traffic. The following are vulnerable writable objects from the IEEE8021-SPB-MIB:

ieee8021SpbSys
ieee8021SpbSysAreaAddress
ieee8021SpbSysId
ieee8021SpbSysControlAddr
ieee8021SpbSysName
ieee8021SpbSysBridgePriority
ieee8021SpbMtidStaticTable
ieee8021SpbMtidStaticTableEntry
ieee8021SpbMTidStaticEntryMtidOverload

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 all types of access (including GET and/or NOTIFY) to these objects and possibly to even encrypt the values of these objects when sending them over the network via SNMP. The following read-only tables with there respective objects in this MIB could be used by an attacker to understand the logical topology of the network:

ieee8021SpbEctDynamicTable
ieee8021SpbAdjDynamicTable
ieee8021SpbTopNodeTable
ieee8021SpbTopEctTable
ieee8021SpbTopEdgeTable

17.7 MIB modules

17.7.4 Definitions for the IEEE8021-Q-BRIDGE MIB module

Delete the entire text of 17.7.4, and insert the following text:

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

-- =============================================================
-- MIB for IEEE 802.1Q Devices
-- =============================================================

IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE, Gauge32,
    Counter64, Unsigned32, TimeTicks, Integer32
        FROM SNMPv2-SMI
    RowStatus, StorageType, TruthValue, MacAddress
        FROM SNMPv2-TC
    SnmpAdminString
        FROM SNMP-FRAMEWORK-MIB
    MODULE-COMPLIANCE, OBJECT-GROUP
        FROM SNMPv2-CONF
    ieee8021BridgeBasePortComponentId, ieee8021BridgeBasePort,
    ieee8021BridgeBasePortEntry
        FROM IEEE8021-BRIDGE-MIB
    ieee802dot1mibs, IEEE8021PbbComponentIdentifier,
    IEEE8021BridgePortNumber, IEEE8021BridgePortNumberOrZero,
    IEEE8021VlanIndex, IEEE8021VlanIndexOrWildcard,
    IEEE8021PortAcceptableFrameTypes
        FROM IEEE8021-TC-MIB
    PortList, VlanId
        FROM Q-BRIDGE-MIB
    TimeFilter
        FROM RMON2-MIB;

ieee8021QBridgeMib MODULE-IDENTITY
    LAST-UPDATED "201112120000Z" -- December 12, 2011
    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 VLAN Bridge MIB module for managing Virtual Bridged
        Local Area Networks, as defined by IEEE 802.1Q-2011.

        This MIB module is derived from the IETF Q-BRIDGE-MIB
        from RFC 4363.

        Unless otherwise indicated, the references in this MIB
        module are to IEEE 802.1Q-2010.

        Copyright (C) IEEE.
        This version of this MIB module is part of IEEE802.1Q;
        see the draft itself for full legal notices."
    REVISION     "201112120000Z" -- December 12, 2011
    DESCRIPTION
         "Addition of the VID Translation MIB Subtree for 802.1aq"

    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 version, derived from RFC 4363."
    ::= { ieee802dot1mibs 4 }

ieee8021QBridgeMibObjects OBJECT IDENTIFIER ::= { ieee8021QBridgeMib 1 }

-- ===============================================================
-- subtrees in the Q-BRIDGE MIB
-- ===============================================================

ieee8021QBridgeBase      OBJECT IDENTIFIER ::= { ieee8021QBridgeMibObjects 1 }
ieee8021QBridgeTp        OBJECT IDENTIFIER ::= { ieee8021QBridgeMibObjects 2 }
ieee8021QBridgeStatic    OBJECT IDENTIFIER ::= { ieee8021QBridgeMibObjects 3 }
ieee8021QBridgeVlan      OBJECT IDENTIFIER ::= { ieee8021QBridgeMibObjects 4 }
ieee8021QBridgeProtocol  OBJECT IDENTIFIER ::= { ieee8021QBridgeMibObjects 5 }
ieee8021QBridgeVIDX      OBJECT IDENTIFIER ::= { ieee8021QBridgeMibObjects 6 }
```

```
-- ===============================================================
-- ieee8021QBridgeBase subtree
-- ===============================================================

-- ===============================================================
-- ieee8021QBridgeTable - Table of VLAN bridges
-- ===============================================================

ieee8021QBridgeTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021QBridgeEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table that contains generic information about every
        VLAN bridge."
    REFERENCE   "12.4"
    ::= { ieee8021QBridgeBase 1 }

ieee8021QBridgeEntry OBJECT-TYPE
    SYNTAX      Ieee8021QBridgeEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A list of objects containing information for each VLAN bridge."
    INDEX   { ieee8021QBridgeComponentId }
    ::= { ieee8021QBridgeTable 1 }

Ieee8021QBridgeEntry ::=
    SEQUENCE {
        ieee8021QBridgeComponentId        IEEE8021PbbComponentIdentifier,
        ieee8021QBridgeVlanVersionNumber  INTEGER,
        ieee8021QBridgeMaxVlanId          VlanId,
        ieee8021QBridgeMaxSupportedVlans  Unsigned32,
        ieee8021QBridgeNumVlans           Gauge32,
        ieee8021QBridgeMvrpEnabledStatus  TruthValue
    }

ieee8021QBridgeComponentId 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."
    ::= { ieee8021QBridgeEntry 1 }

ieee8021QBridgeVlanVersionNumber OBJECT-TYPE
    SYNTAX      INTEGER {
                    version1(1),
                    version2(2)
                }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The version number of IEEE 802.1Q that this device
        supports.  Reported as 1 by VLAN Bridges that support
        only SST operation, and reported as 2 by VLAN Bridges
        that support MST operation."
    REFERENCE   "12.10.1.1"
    ::= { ieee8021QBridgeEntry 2 }
```

```
ieee8021QBridgeMaxVlanId OBJECT-TYPE
    SYNTAX      VlanId
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The maximum IEEE 802.1Q VLAN-ID that this device
        supports."
    REFERENCE   "9.6"
    ::= { ieee8021QBridgeEntry 3 }

ieee8021QBridgeMaxSupportedVlans OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "vlans"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The maximum number of IEEE 802.1Q VLANs that this
        device supports."
    REFERENCE   "12.10.1.1"
    ::= { ieee8021QBridgeEntry 4 }

ieee8021QBridgeNumVlans OBJECT-TYPE
    SYNTAX      Gauge32
    UNITS       "vlans"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The current number of IEEE 802.1Q VLANs that are
        configured in this device."
    REFERENCE   "12.7.1.1"
    ::= { ieee8021QBridgeEntry 5 }

ieee8021QBridgeMvrpEnabledStatus OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The administrative status requested by management for
        MVRP.  The value true(1) indicates that MVRP should
        be enabled on this device, on all ports for which it has
        not been specifically disabled.  When false(2), MVRP
        is disabled on all ports, and all MVRP packets will be
        forwarded transparently.  This object affects all MVRP
        Applicant and Registrar state machines.  A transition
        from false(2) to true(1) will cause a reset of all
        MVRP state machines on all ports.

        The value of this object MUST be retained across
        reinitializations of the management system."
    DEFVAL      { true }
    ::= { ieee8021QBridgeEntry 6 }

-- =============================================================
-- ieee8021QBridgeCVlanPortTable - Table of C-VLAN ports
-- =============================================================

ieee8021QBridgeCVlanPortTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021QBridgeCVlanPortEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This table provides the capability to create and delete
        customer VLAN ports.  Entries in this table 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"
    ::= { ieee8021QBridgeBase 2 }

ieee8021QBridgeCVlanPortEntry OBJECT-TYPE
    SYNTAX      Ieee8021QBridgeCVlanPortEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A list of objects containing information for each VLAN bridge."
    INDEX    { ieee8021QBridgeCVlanPortComponentId,
               ieee8021QBridgeCVlanPortNumber }
    ::= { ieee8021QBridgeCVlanPortTable 1 }

Ieee8021QBridgeCVlanPortEntry ::=
    SEQUENCE {
        ieee8021QBridgeCVlanPortComponentId  IEEE8021PbbComponentIdentifier,
        ieee8021QBridgeCVlanPortNumber       IEEE8021BridgePortNumber,
        ieee8021QBridgeCVlanPortRowStatus    RowStatus
    }

ieee8021QBridgeCVlanPortComponentId OBJECT-TYPE
    SYNTAX      IEEE8021PbbComponentIdentifier
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The componentcontaining the customer VLAN port represented
        by this row."
    ::= { ieee8021QBridgeCVlanPortEntry 1 }

ieee8021QBridgeCVlanPortNumber OBJECT-TYPE
    SYNTAX      IEEE8021BridgePortNumber
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The customer VLAN port number represented by this row."
    ::= { ieee8021QBridgeCVlanPortEntry 2 }

ieee8021QBridgeCVlanPortRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This indicates the status of the entry, and is used to create
        and delete entries in this table.  Each entry in this table that
        is valid will have a corresponding entry in the
        ieee8021BridgeBasePortTable whose value for
        ieee8021BridgeBasePortType is customerVlanPort(2).  The
        corresponding value of ieee8021BridgeBasePortIfIndex must
        be set at the time the value of this object transitions
        to valid(1).

        Entries in this table must be persistent across
        reinitializations of the management system."
    ::= { ieee8021QBridgeCVlanPortEntry 3 }

-- ============================================================
-- the ieee8021QBridgeTp subtree
-- ============================================================

-- ============================================================
-- the current Filtering Database Table
-- ============================================================
ieee8021QBridgeFdbTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021QBridgeFdbEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table that contains configuration and control
        information for each Filtering Database currently
        operating on this device.  Entries in this table appear
        automatically when VLANs are assigned FDB IDs in the
        ieee8021QBridgeVlanCurrentTable."
    REFERENCE   "12.7.1"
    ::= { ieee8021QBridgeTp 1 }

ieee8021QBridgeFdbEntry OBJECT-TYPE
    SYNTAX      Ieee8021QBridgeFdbEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Information about a specific Filtering Database."
    INDEX   { ieee8021QBridgeFdbComponentId,
              ieee8021QBridgeFdbId }
    ::= { ieee8021QBridgeFdbTable 1 }

Ieee8021QBridgeFdbEntry ::=
    SEQUENCE {
        ieee8021QBridgeFdbComponentId
            IEEE8021PbbComponentIdentifier,
        ieee8021QBridgeFdbId
            Unsigned32,
        ieee8021QBridgeFdbDynamicCount
            Gauge32,
        ieee8021QBridgeFdbLearnedEntryDiscards
            Counter64,
        ieee8021QBridgeFdbAgingTime
            Integer32
    }

ieee8021QBridgeFdbComponentId 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."
    ::= { ieee8021QBridgeFdbEntry 1 }

ieee8021QBridgeFdbId OBJECT-TYPE
    SYNTAX       Unsigned32 (0..4294967295)
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "The identity of this Filtering Database."
    ::= { ieee8021QBridgeFdbEntry 2 }

ieee8021QBridgeFdbDynamicCount OBJECT-TYPE
    SYNTAX       Gauge32
    UNITS        "database entries"
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The current number of dynamic entries in this
        Filtering Database."
    REFERENCE   "12.7.1.1.3"
    ::= { ieee8021QBridgeFdbEntry 3 }

ieee8021QBridgeFdbLearnedEntryDiscards OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "database entries"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The total number of Filtering Database entries that
        have been or would have been learned, but have been
        discarded due to a lack of storage space in the
        Filtering Database.  If this counter is increasing, it
        indicates that the Filtering Database is regularly
        becoming full (a condition that has unpleasant
        performance effects on the subnetwork).  If this counter
        has a significant value but is not presently increasing,
        it indicates that the problem has been occurring but is
        not persistent.

        Discontinuities in the value of the counter can occur
        at re-initialization of the management system."
    ::= { ieee8021QBridgeFdbEntry 4 }

ieee8021QBridgeFdbAgingTime OBJECT-TYPE
    SYNTAX      Integer32 (10..1000000)
    UNITS       "seconds"
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The timeout period in seconds for aging out
        dynamically-learned forwarding information.
        802.1D-1998 recommends a default of 300 seconds.

        The value of this object MUST be retained across
        reinitializations of the management system."
    REFERENCE   "12.7.1.2"
    ::= { ieee8021QBridgeFdbEntry 5 }

-- ===============================================================
-- Multiple Filtering Databases for 802.1Q Transparent Devices
-- This table is an alternative to the ieee8021BridgeTpFdbTable,
-- previously defined for 802.1D devices that only support a
-- single Filtering Database.
-- ===============================================================

ieee8021QBridgeTpFdbTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021QBridgeTpFdbEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table that contains information about unicast entries
        for which the device has forwarding and/or filtering
        information.  This information is used by the
        transparent bridging function in determining how to
        propagate a received frame."
    REFERENCE   "12.7.1"
    ::= { ieee8021QBridgeTp 2 }

ieee8021QBridgeTpFdbEntry OBJECT-TYPE
    SYNTAX      Ieee8021QBridgeTpFdbEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
```

```
        "Information about a specific unicast MAC address for
        which the device has some forwarding and/or filtering
        information."
    INDEX   { ieee8021QBridgeFdbComponentId,
              ieee8021QBridgeFdbId,
              ieee8021QBridgeTpFdbAddress }
    ::= { ieee8021QBridgeTpFdbTable 1 }

Ieee8021QBridgeTpFdbEntry ::=
    SEQUENCE {
        ieee8021QBridgeTpFdbAddress
            MacAddress,
        ieee8021QBridgeTpFdbPort
            IEEE8021BridgePortNumberOrZero,
        ieee8021QBridgeTpFdbStatus
            INTEGER
    }

ieee8021QBridgeTpFdbAddress OBJECT-TYPE
    SYNTAX      MacAddress
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A unicast MAC address for which the device has
        forwarding and/or filtering information."
    ::= { ieee8021QBridgeTpFdbEntry 1 }

ieee8021QBridgeTpFdbPort OBJECT-TYPE
    SYNTAX      IEEE8021BridgePortNumberOrZero
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Either the value '0', or the port number of the port on
        which a frame having a source address equal to the value
        of the corresponding instance of ieee8021QBridgeTpFdbAddress has
        been seen.  A value of '0' indicates that the port
        number has not been learned but that the device does
        have some forwarding/filtering information about this
        address (e.g., in the ieee8021QBridgeStaticUnicastTable).
        Implementors are encouraged to assign the port value to
        this object whenever it is learned, even for addresses
        for which the corresponding value of ieee8021QBridgeTpFdbStatus is
        not learned(3)."
    ::= { ieee8021QBridgeTpFdbEntry 2 }

ieee8021QBridgeTpFdbStatus OBJECT-TYPE
    SYNTAX      INTEGER {
                    other(1),
                    invalid(2),
                    learned(3),
                    self(4),
                    mgmt(5)
                }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The status of this entry.  The meanings of the values
        are:
            other(1) - none of the following.  This may include
                the case where some other MIB object (not the
                corresponding instance of ieee8021QBridgeTpFdbPort, nor an
                entry in the ieee8021QBridgeStaticUnicastTable) is being
                used to determine if and how frames addressed to
                the value of the corresponding instance of
                ieee8021QBridgeTpFdbAddress are being forwarded.
            invalid(2) - this entry is no longer valid (e.g., it
                was learned but has since aged out), but has not
                yet been flushed from the table.
            learned(3) - the value of the corresponding instance
                of ieee8021QBridgeTpFdbPort was learned and is being used.
            self(4) - the value of the corresponding instance of
                ieee8021QBridgeTpFdbAddress represents one of the device's
                addresses.  The corresponding instance of
                ieee8021QBridgeTpFdbPort indicates which of the device's
                ports has this address.
            mgmt(5) - the value of the corresponding instance of
                ieee8021QBridgeTpFdbAddress is also the value of an
                existing instance of ieee8021QBridgeStaticUnicastAddress."
    ::= { ieee8021QBridgeTpFdbEntry 3 }

-- ============================================================
-- Dynamic Group Registration Table
-- ============================================================

ieee8021QBridgeTpGroupTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021QBridgeTpGroupEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table containing filtering information for VLANs
        configured into the bridge by (local or network)
        management, or learned dynamically, specifying the set of
        ports to which frames received on a VLAN for this FDB
        and containing a specific Group destination address are
        allowed to be forwarded."
    REFERENCE   "12.7.4"
    ::= { ieee8021QBridgeTp 3 }

ieee8021QBridgeTpGroupEntry OBJECT-TYPE
    SYNTAX      Ieee8021QBridgeTpGroupEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Filtering information configured into the bridge by
        management, or learned dynamically, specifying the set of
        ports to which frames received on a VLAN and containing
        a specific Group destination address are allowed to be
        forwarded.  The subset of these ports learned dynamically
        is also provided."
    INDEX   { ieee8021QBridgeVlanCurrentComponentId,
              ieee8021QBridgeVlanIndex,
              ieee8021QBridgeTpGroupAddress }
    ::= { ieee8021QBridgeTpGroupTable 1 }

Ieee8021QBridgeTpGroupEntry ::=
    SEQUENCE {
        ieee8021QBridgeTpGroupAddress
            MacAddress,
        ieee8021QBridgeTpGroupEgressPorts
            PortList,
        ieee8021QBridgeTpGroupLearnt
            PortList
    }

ieee8021QBridgeTpGroupAddress OBJECT-TYPE
    SYNTAX      MacAddress
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The destination Group MAC address in a frame to which
        this entry's filtering information applies."
    ::= { ieee8021QBridgeTpGroupEntry 1 }

ieee8021QBridgeTpGroupEgressPorts OBJECT-TYPE
    SYNTAX      PortList
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The complete set of ports, in this VLAN, to which
        frames destined for this Group MAC address are currently
        being explicitly forwarded.  This does not include ports
        for which this address is only implicitly forwarded, in
        the ieee8021QBridgeForwardAllPorts list."
    ::= { ieee8021QBridgeTpGroupEntry 2 }

ieee8021QBridgeTpGroupLearnt OBJECT-TYPE
    SYNTAX      PortList
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The subset of ports in ieee8021QBridgeTpGroupEgressPorts that
        were learned by MMRP or some other dynamic mechanism, in
        this Filtering database."
    ::= { ieee8021QBridgeTpGroupEntry 3 }

-- =============================================================
-- Service Requirements subtree
-- =============================================================

ieee8021QBridgeForwardAllTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021QBridgeForwardAllEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table containing forwarding information for each
        VLAN, specifying the set of ports to which forwarding of
        all multicasts applies, configured statically by
        management or dynamically by MMRP.  An entry appears in
        this table for all VLANs that are currently
        instantiated."
    REFERENCE   "12.7.2, 12.7.7"
    ::= { ieee8021QBridgeTp 4 }

ieee8021QBridgeForwardAllEntry OBJECT-TYPE
    SYNTAX      Ieee8021QBridgeForwardAllEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Forwarding information for a VLAN, specifying the set
        of ports to which all multicasts should be forwarded,
        configured statically by management or dynamically by
        MMRP."
    INDEX   { ieee8021QBridgeVlanCurrentComponentId,
              ieee8021QBridgeForwardAllVlanIndex }
    ::= { ieee8021QBridgeForwardAllTable 1 }

Ieee8021QBridgeForwardAllEntry ::=
    SEQUENCE {
        ieee8021QBridgeForwardAllVlanIndex
            IEEE8021VlanIndexOrWildcard,
        ieee8021QBridgeForwardAllPorts
            PortList,
        ieee8021QBridgeForwardAllStaticPorts
            PortList,
        ieee8021QBridgeForwardAllForbiddenPorts
            PortList
    }

ieee8021QBridgeForwardAllVlanIndex OBJECT-TYPE
    SYNTAX      IEEE8021VlanIndexOrWildcard
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The VLAN-ID or other identifier referring to this VLAN."
    ::= { ieee8021QBridgeForwardAllEntry 1 }

ieee8021QBridgeForwardAllPorts OBJECT-TYPE
    SYNTAX      PortList
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The complete set of ports in this VLAN to which all
        multicast group-addressed frames are to be forwarded.
        This includes ports for which this need has been
        determined dynamically by MMRP, or configured statically
        by management."
    ::= { ieee8021QBridgeForwardAllEntry 2 }

ieee8021QBridgeForwardAllStaticPorts OBJECT-TYPE
    SYNTAX      PortList
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The set of ports configured by management in this VLAN
        to which all multicast group-addressed frames are to be
        forwarded.  Ports entered in this list will also appear
        in the complete set shown by ieee8021QBridgeForwardAllPorts.  This
        value will be restored after the device is reset.  This
        only applies to ports that are members of the VLAN,
        defined by ieee8021QBridgeVlanCurrentEgressPorts.  A port may not
        be added in this set if it is already a member of the
        set of ports in ieee8021QBridgeForwardAllForbiddenPorts.  The
        default value is a string of ones of appropriate length,
        to indicate the standard behaviour of using basic
        filtering services, i.e., forward all multicasts to all
        ports.

        The value of this object MUST be retained across
        reinitializations of the management system."
    ::= { ieee8021QBridgeForwardAllEntry 3 }

ieee8021QBridgeForwardAllForbiddenPorts OBJECT-TYPE
    SYNTAX      PortList
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The set of ports configured by management in this VLAN
        for which the Service Requirement attribute Forward All
        Multicast Groups may not be dynamically registered by
        MMRP.  This value will be restored after the device is
        reset.  A port may not be added in this set if it is
        already a member of the set of ports in
        ieee8021QBridgeForwardAllStaticPorts.  The default value is a
        string of zeros of appropriate length.

        The value of this object MUST be retained across
        reinitializations of the management system."
    ::= { ieee8021QBridgeForwardAllEntry 4 }

ieee8021QBridgeForwardUnregisteredTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021QBridgeForwardUnregisteredEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table containing forwarding information for each
        VLAN, specifying the set of ports to which forwarding of
        multicast group-addressed frames for which no
        more specific forwarding information applies.  This is
        configured statically by management and determined
        dynamically by MMRP.  An entry appears in this table for
        all VLANs that are currently instantiated."
    REFERENCE   "12.7.2, 12.7.7"
    ::= { ieee8021QBridgeTp 5 }

ieee8021QBridgeForwardUnregisteredEntry OBJECT-TYPE
    SYNTAX      Ieee8021QBridgeForwardUnregisteredEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Forwarding information for a VLAN, specifying the set
        of ports to which all multicasts for which there is no
        more specific forwarding information shall be forwarded.
        This is configured statically by management or
        dynamically by MMRP."
    INDEX   { ieee8021QBridgeVlanCurrentComponentId,
              ieee8021QBridgeForwardUnregisteredVlanIndex }
    ::= { ieee8021QBridgeForwardUnregisteredTable 1 }

Ieee8021QBridgeForwardUnregisteredEntry ::=
    SEQUENCE {
        ieee8021QBridgeForwardUnregisteredVlanIndex
            IEEE8021VlanIndexOrWildcard,
        ieee8021QBridgeForwardUnregisteredPorts
            PortList,
        ieee8021QBridgeForwardUnregisteredStaticPorts
            PortList,
        ieee8021QBridgeForwardUnregisteredForbiddenPorts
            PortList
    }

ieee8021QBridgeForwardUnregisteredVlanIndex OBJECT-TYPE
    SYNTAX      IEEE8021VlanIndexOrWildcard
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The VLAN-ID or other identifier referring to this VLAN."
    ::= { ieee8021QBridgeForwardUnregisteredEntry 1 }

ieee8021QBridgeForwardUnregisteredPorts OBJECT-TYPE
    SYNTAX      PortList
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The complete set of ports in this VLAN to which
        multicast group-addressed frames for which there is no
        more specific forwarding information will be forwarded.
        This includes ports for which this need has been
        determined dynamically by MMRP, or configured statically
        by management."
    ::= { ieee8021QBridgeForwardUnregisteredEntry 2 }
```

```
ieee8021QBridgeForwardUnregisteredStaticPorts OBJECT-TYPE
    SYNTAX      PortList
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The set of ports configured by management, in this
        VLAN, to which multicast group-addressed frames for
        which there is no more specific forwarding information
        are to be forwarded.  Ports entered in this list will
        also appear in the complete set shown by
        ieee8021QBridgeForwardUnregisteredPorts.  This value will be
        restored after the device is reset.  A port may not be
        added in this set if it is already a member of the set
        of ports in ieee8021QBridgeForwardUnregisteredForbiddenPorts.  The
        default value is a string of zeros of appropriate
        length, although this has no effect with the default
        value of ieee8021QBridgeForwardAllStaticPorts.

        The value of this object MUST be retained across
        reinitializations of the management system."
    ::= { ieee8021QBridgeForwardUnregisteredEntry 3 }

ieee8021QBridgeForwardUnregisteredForbiddenPorts OBJECT-TYPE
    SYNTAX      PortList
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The set of ports configured by management in this VLAN
        for which the Service Requirement attribute Forward
        Unregistered Multicast Groups may not be dynamically
        registered by MMRP.  This value will be restored after
        the device is reset.  A port may not be added in this
        set if it is already a member of the set of ports in
        ieee8021QBridgeForwardUnregisteredStaticPorts.  The default value
        is a string of zeros of appropriate length.

        The value of this object MUST be retained across
        reinitializations of the management system."
    ::= { ieee8021QBridgeForwardUnregisteredEntry 4 }

-- ===============================================================
-- The Static (Destination-Address Filtering) Database
-- ===============================================================

ieee8021QBridgeStaticUnicastTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021QBridgeStaticUnicastEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table containing filtering information for Unicast
        MAC addresses for each Filtering Database, configured
        into the device by (local or network) management
        specifying the set of ports to which frames received
        from specific ports and containing specific unicast
        destination addresses are allowed to be forwarded.
        Entries are valid for unicast addresses only.

        Two modes of operation are supported by this table. When
        the receive port index is non-zero, this table is
        supporting an 802.1D filtering database as specified in
        14.7.6.1.  If the receive port is zero, the
        table is operating as specified in 802.1Q
        8.8.1 and 12.7.7.  An agent must at least
        support the 802.1Q mode of operation."
    REFERENCE   "802.1D 7.9.1, 14.7.6.1;
                 802.1Q 12.7.7, 8.8.1"
    ::= { ieee8021QBridgeStatic 1 }

ieee8021QBridgeStaticUnicastEntry OBJECT-TYPE
    SYNTAX      Ieee8021QBridgeStaticUnicastEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Filtering information configured into the device by
        (local or network) management specifying the set of
        ports to which frames received from a specific port and
        containing a specific unicast destination address are
        allowed to be forwarded."
    INDEX   {
        ieee8021QBridgeStaticUnicastComponentId,
        ieee8021QBridgeStaticUnicastVlanIndex,
        ieee8021QBridgeStaticUnicastAddress,
        ieee8021QBridgeStaticUnicastReceivePort
    }
    ::= { ieee8021QBridgeStaticUnicastTable 1 }

Ieee8021QBridgeStaticUnicastEntry ::=
    SEQUENCE {
        ieee8021QBridgeStaticUnicastComponentId
            IEEE8021PbbComponentIdentifier,
        ieee8021QBridgeStaticUnicastVlanIndex
            IEEE8021VlanIndexOrWildcard,
        ieee8021QBridgeStaticUnicastAddress
            MacAddress,
        ieee8021QBridgeStaticUnicastReceivePort
            IEEE8021BridgePortNumberOrZero,
        ieee8021QBridgeStaticUnicastStaticEgressPorts
            PortList,
        ieee8021QBridgeStaticUnicastForbiddenEgressPorts
            PortList,
        ieee8021QBridgeStaticUnicastStorageType
            StorageType,
        ieee8021QBridgeStaticUnicastRowStatus
            RowStatus
    }

ieee8021QBridgeStaticUnicastComponentId 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."
    ::= { ieee8021QBridgeStaticUnicastEntry 1 }

ieee8021QBridgeStaticUnicastVlanIndex OBJECT-TYPE
    SYNTAX       IEEE8021VlanIndexOrWildcard
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "The Vlan to which this entry applies."
    ::= { ieee8021QBridgeStaticUnicastEntry 2 }

ieee8021QBridgeStaticUnicastAddress OBJECT-TYPE
    SYNTAX      MacAddress
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The destination MAC address in a frame to which this
        entry's filtering information applies.  This object must
        take the value of a unicast address."
    ::= { ieee8021QBridgeStaticUnicastEntry 3 }

ieee8021QBridgeStaticUnicastReceivePort OBJECT-TYPE
    SYNTAX      IEEE8021BridgePortNumberOrZero
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Either the value '0' or the port number of the port
        from which a frame must be received in order for this
        entry's filtering information to apply.  A value of zero
        indicates that this entry applies on all ports of the
        device for which there is no other applicable entry. An
        implementation is required to support the '0' value and
        may optionally support non-zero values for this column."
    ::= { ieee8021QBridgeStaticUnicastEntry 4 }

ieee8021QBridgeStaticUnicastStaticEgressPorts OBJECT-TYPE
    SYNTAX      PortList
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The set of ports to which frames received from a
        specific port and destined for a specific unicast address
        must be forwarded, regardless of
        any dynamic information, e.g., from MMRP.  A port may not
        be added in this set if it is already a member of the
        set of ports in ieee8021QBridgeStaticUnicastForbiddenEgressPorts.
        The default value of this object is a string of ones of
        appropriate length."
    DEFVAL      { ''H }
    ::= { ieee8021QBridgeStaticUnicastEntry 5 }


ieee8021QBridgeStaticUnicastForbiddenEgressPorts OBJECT-TYPE
    SYNTAX      PortList
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The set of ports to which frames received from a
        specific port and destined for a specific unicast
        MAC address must not be forwarded, regardless
        of any dynamic information, e.g., from MMRP.  A port may
        not be added in this set if it is already a member of the
        set of ports in ieee8021QBridgeStaticUnicastStaticEgressPorts.
        The default value of this object is a string of zeros of
        appropriate length."
    DEFVAL      { ''H }
    ::= { ieee8021QBridgeStaticUnicastEntry 6 }

ieee8021QBridgeStaticUnicastStorageType OBJECT-TYPE
    SYNTAX      StorageType
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The storage type for this conceptual row.  If this object
        has a value of permanent(4), then no other objects are
        required to be able to be modified."
    DEFVAL { nonVolatile }
    ::= { ieee8021QBridgeStaticUnicastEntry 7 }

ieee8021QBridgeStaticUnicastRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This object indicates the status of this entry, and is used
        to create/delete entries in the table.

        An entry of this table may be set to active without setting
        any other columns of the table.  Also, other columns of this
        table may be set while the value of this object is active(1)."
    ::= { ieee8021QBridgeStaticUnicastEntry 8 }

ieee8021QBridgeStaticMulticastTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021QBridgeStaticMulticastEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table containing filtering information for Multicast
        and Broadcast MAC addresses for each VLAN, configured
        into the device by (local or network) management
        specifying the set of ports to which frames received
        from specific ports and containing specific Multicast
        and Broadcast destination addresses are allowed to be
        forwarded.  A value of zero in this table (as the port
        number from which frames with a specific destination
        address are received) is used to specify all ports for
        which there is no specific entry in this table for that
        particular destination address.  Entries are valid for
        Multicast and Broadcast addresses only."
    REFERENCE   "12.7.7, 8.8.1"
    ::= { ieee8021QBridgeStatic 2 }

ieee8021QBridgeStaticMulticastEntry OBJECT-TYPE
    SYNTAX      Ieee8021QBridgeStaticMulticastEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Filtering information configured into the device by
        (local or network) management specifying the set of
        ports to which frames received from this specific port
        for this VLAN and containing this Multicast or Broadcast
        destination address are allowed to be forwarded."
    INDEX   {
        ieee8021QBridgeVlanCurrentComponentId,
        ieee8021QBridgeVlanIndex,
        ieee8021QBridgeStaticMulticastAddress,
        ieee8021QBridgeStaticMulticastReceivePort
    }
    ::= { ieee8021QBridgeStaticMulticastTable 1 }

Ieee8021QBridgeStaticMulticastEntry ::=
    SEQUENCE {
        ieee8021QBridgeStaticMulticastAddress
            MacAddress,
        ieee8021QBridgeStaticMulticastReceivePort
            IEEE8021BridgePortNumberOrZero,
        ieee8021QBridgeStaticMulticastStaticEgressPorts
            PortList,
        ieee8021QBridgeStaticMulticastForbiddenEgressPorts
            PortList,
        ieee8021QBridgeStaticMulticastStorageType
```

```
            StorageType,
        ieee8021QBridgeStaticMulticastRowStatus
            RowStatus
    }

ieee8021QBridgeStaticMulticastAddress OBJECT-TYPE
    SYNTAX      MacAddress
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The destination MAC address in a frame to which this
        entry's filtering information applies.  This object must
        take the value of a Multicast or Broadcast address."
    ::= { ieee8021QBridgeStaticMulticastEntry 1 }

ieee8021QBridgeStaticMulticastReceivePort OBJECT-TYPE
    SYNTAX      IEEE8021BridgePortNumberOrZero
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Either the value '0' or the port number of the port
        from which a frame must be received in order for this
        entry's filtering information to apply.  A value of zero
        indicates that this entry applies on all ports of the
        device for which there is no other applicable entry. An
        implementation is required to support the '0' value and
        may optionally support non-zero values for this column."
    ::= { ieee8021QBridgeStaticMulticastEntry 2 }

ieee8021QBridgeStaticMulticastStaticEgressPorts OBJECT-TYPE
    SYNTAX      PortList
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The set of ports to which frames received from a
        specific port and destined for a specific Multicast or
        Broadcast MAC address must be forwarded, regardless of
        any dynamic information, e.g., from MMRP.  A port may not
        be added in this set if it is already a member of the
        set of ports in ieee8021QBridgeStaticMulticastForbiddenEgressPorts.
        The default value of this object is a string of ones of
        appropriate length."
    DEFVAL      { ''H }
    ::= { ieee8021QBridgeStaticMulticastEntry 3 }

ieee8021QBridgeStaticMulticastForbiddenEgressPorts OBJECT-TYPE
    SYNTAX      PortList
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The set of ports to which frames received from a
        specific port and destined for a specific Multicast or
        Broadcast MAC address must not be forwarded, regardless
        of any dynamic information, e.g., from MMRP.  A port may
        not be added in this set if it is already a member of the
        set of ports in ieee8021QBridgeStaticMulticastStaticEgressPorts.
        The default value of this object is a string of zeros of
        appropriate length."
    DEFVAL      { ''H }
    ::= { ieee8021QBridgeStaticMulticastEntry 4 }

ieee8021QBridgeStaticMulticastStorageType OBJECT-TYPE
    SYNTAX      StorageType
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The storage type for this conceptual row.  If this object
        has a value of permanent(4), then no other objects are
        required to be able to be modified."
    DEFVAL      { nonVolatile }
    ::= { ieee8021QBridgeStaticMulticastEntry 5 }

ieee8021QBridgeStaticMulticastRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This object indicates the status of this entry, and is used
        to create/delete entries in the table.

        An entry of this table may be set to active without setting
        any other columns of the table.  Also, other columns of this
        table may be set while the value of this object is active(1)."
    ::= { ieee8021QBridgeStaticMulticastEntry 6 }

-- =============================================================
-- The Current VLAN Database
-- =============================================================

ieee8021QBridgeVlanNumDeletes OBJECT-TYPE
    SYNTAX     Counter64
    UNITS      "vlan deletions"
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
        "The number of times a VLAN entry has been deleted from
        the ieee8021QBridgeVlanCurrentTable (for any reason).
        If an entry is deleted, then inserted, and then deleted,
        this counter will be incremented by 2.  Discontinuities
        in this value can only occur at a reboot."
    ::= { ieee8021QBridgeVlan 1 }

ieee8021QBridgeVlanCurrentTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021QBridgeVlanCurrentEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table containing current configuration information
        for each VLAN currently configured into the device by
        (local or network) management, or dynamically created
        as a result of MVRP requests received."
    REFERENCE   "12.10.2"
    ::= { ieee8021QBridgeVlan 2 }

ieee8021QBridgeVlanCurrentEntry OBJECT-TYPE
    SYNTAX      Ieee8021QBridgeVlanCurrentEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Information for a VLAN configured into the device by
        (local or network) management, or dynamically created
        as a result of MVRP requests received."
    INDEX   { ieee8021QBridgeVlanTimeMark,
              ieee8021QBridgeVlanCurrentComponentId,
              ieee8021QBridgeVlanIndex }
    ::= { ieee8021QBridgeVlanCurrentTable 1 }

Ieee8021QBridgeVlanCurrentEntry ::=
    SEQUENCE {
        ieee8021QBridgeVlanTimeMark
            TimeFilter,
        ieee8021QBridgeVlanCurrentComponentId
            IEEE8021PbbComponentIdentifier,
        ieee8021QBridgeVlanIndex
            IEEE8021VlanIndex,
        ieee8021QBridgeVlanFdbId
            Unsigned32,
        ieee8021QBridgeVlanCurrentEgressPorts
            PortList,
        ieee8021QBridgeVlanCurrentUntaggedPorts
            PortList,
        ieee8021QBridgeVlanStatus
            INTEGER,
        ieee8021QBridgeVlanCreationTime
            TimeTicks
    }

ieee8021QBridgeVlanTimeMark OBJECT-TYPE
    SYNTAX      TimeFilter
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A TimeFilter for this entry.  See the TimeFilter
        textual convention to see how this works."
    ::= { ieee8021QBridgeVlanCurrentEntry 1 }

ieee8021QBridgeVlanCurrentComponentId 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."
    ::= { ieee8021QBridgeVlanCurrentEntry 2 }

ieee8021QBridgeVlanIndex OBJECT-TYPE
    SYNTAX      IEEE8021VlanIndex
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The VLAN-ID or other identifier referring to this VLAN."
    ::= { ieee8021QBridgeVlanCurrentEntry 3 }

ieee8021QBridgeVlanFdbId OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The Filtering Database used by this VLAN.  This is one
        of the ieee8021QBridgeFdbId values in the ieee8021QBridgeFdbTable.
        This value is allocated automatically by the device whenever
        the VLAN is created: either dynamically by MVRP, or by
        management, in ieee8021QBridgeVlanStaticTable.  Allocation of this
        value follows the learning constraints defined for this
        VLAN in ieee8021QBridgeLearningConstraintsTable."
    ::= { ieee8021QBridgeVlanCurrentEntry 4 }

ieee8021QBridgeVlanCurrentEgressPorts OBJECT-TYPE
    SYNTAX      PortList
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The set of ports that are transmitting traffic for
        this VLAN as either tagged or untagged frames."
    REFERENCE   "12.10.2.1"
    ::= { ieee8021QBridgeVlanCurrentEntry 5 }

ieee8021QBridgeVlanCurrentUntaggedPorts OBJECT-TYPE
    SYNTAX      PortList
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The set of ports that are transmitting traffic for
        this VLAN as untagged frames."
    REFERENCE   "12.10.2.1"
    ::= { ieee8021QBridgeVlanCurrentEntry 6 }

ieee8021QBridgeVlanStatus OBJECT-TYPE
    SYNTAX      INTEGER {
                    other(1),
                    permanent(2),
                    dynamicMvrp(3)
                }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This object indicates the status of this entry.
            other(1) - this entry is currently in use, but the
                conditions under which it will remain so differ
                from the following values.
            permanent(2) - this entry, corresponding to an entry
                in ieee8021QBridgeVlanStaticTable, is currently in use and
                will remain so after the next reset of the
                device.  The port lists for this entry include
                ports from the equivalent ieee8021QBridgeVlanStaticTable
                entry and ports learned dynamically.
            dynamicMvrp(3) - this entry is currently in use
                and will remain so until removed by MVRP.  There
                is no static entry for this VLAN, and it will be
                removed when the last port leaves the VLAN."
    ::= { ieee8021QBridgeVlanCurrentEntry 7 }

ieee8021QBridgeVlanCreationTime OBJECT-TYPE
    SYNTAX      TimeTicks
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The value of sysUpTime when this VLAN was created."
    ::= { ieee8021QBridgeVlanCurrentEntry 8 }

-- =============================================================
-- The Static VLAN Database
-- =============================================================

ieee8021QBridgeVlanStaticTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021QBridgeVlanStaticEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table containing static configuration information for
        each VLAN configured into the device by (local or
        network) management.  All entries are persistent and will
        be restored after the device is reset."
    REFERENCE   "12.7.5"
    ::= { ieee8021QBridgeVlan 3 }

ieee8021QBridgeVlanStaticEntry OBJECT-TYPE
    SYNTAX      Ieee8021QBridgeVlanStaticEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Static information for a VLAN configured into the
        device by (local or network) management."
    INDEX   { ieee8021QBridgeVlanStaticComponentId,
              ieee8021QBridgeVlanStaticVlanIndex }
    ::= { ieee8021QBridgeVlanStaticTable 1 }

Ieee8021QBridgeVlanStaticEntry ::=
    SEQUENCE {
        ieee8021QBridgeVlanStaticComponentId
            IEEE8021PbbComponentIdentifier,
        ieee8021QBridgeVlanStaticVlanIndex
            IEEE8021VlanIndex,
        ieee8021QBridgeVlanStaticName
            SnmpAdminString,
        ieee8021QBridgeVlanStaticEgressPorts
            PortList,
        ieee8021QBridgeVlanForbiddenEgressPorts
            PortList,
        ieee8021QBridgeVlanStaticUntaggedPorts
            PortList,
        ieee8021QBridgeVlanStaticRowStatus
            RowStatus
    }

ieee8021QBridgeVlanStaticComponentId 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."
    ::= { ieee8021QBridgeVlanStaticEntry 1 }

ieee8021QBridgeVlanStaticVlanIndex OBJECT-TYPE
    SYNTAX      IEEE8021VlanIndex
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The VLAN-ID or other identifier referring to this VLAN."
    ::= { ieee8021QBridgeVlanStaticEntry 2 }

ieee8021QBridgeVlanStaticName OBJECT-TYPE
    SYNTAX      SnmpAdminString (SIZE (0..32))
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "An administratively assigned string, which may be used
        to identify the VLAN."
    REFERENCE   "12.10.2.1"
    ::= { ieee8021QBridgeVlanStaticEntry 3 }

ieee8021QBridgeVlanStaticEgressPorts OBJECT-TYPE
    SYNTAX      PortList
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The set of ports that are permanently assigned to the
        egress list for this VLAN by management.  Changes to a
        bit in this object affect the per-port, per-VLAN
        Registrar control for Registration Fixed for the
        relevant MVRP state machine on each port.  A port may
        not be added in this set if it is already a member of
        the set of ports in ieee8021QBridgeVlanForbiddenEgressPorts.  The
        default value of this object is a string of zeros of
        appropriate length, indicating not fixed."
    REFERENCE   "12.7.7.3, 11.2.3.2.3"
    ::= { ieee8021QBridgeVlanStaticEntry 4 }

ieee8021QBridgeVlanForbiddenEgressPorts OBJECT-TYPE
    SYNTAX      PortList
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The set of ports that are prohibited by management
        from being included in the egress list for this VLAN.
        Changes to this object that cause a port to be included
        or excluded affect the per-port, per-VLAN Registrar
        control for Registration Forbidden for the relevant MVRP
        state machine on each port.  A port may not be added in
        this set if it is already a member of the set of ports
        in ieee8021QBridgeVlanStaticEgressPorts.  The default value of
        this object is a string of zeros of appropriate length,
        excluding all ports from the forbidden set."
    REFERENCE   "12.7.7.3, 11.2.3.2.3"
    ::= { ieee8021QBridgeVlanStaticEntry 5 }

ieee8021QBridgeVlanStaticUntaggedPorts OBJECT-TYPE
    SYNTAX      PortList
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The set of ports that should transmit egress packets
        for this VLAN as untagged.  The default value of this
        object for the default VLAN (ieee8021QBridgeVlanIndex = 1) is a string
        of appropriate length including all ports.  There is no
        specified default for other VLANs.  If a device agent cannot
        support the set of ports being set, then it will reject the
        set operation with an error.  For example, a
        manager might attempt to set more than one VLAN to be untagged
        on egress where the device does not support this IEEE 802.1Q
        option."
    REFERENCE   "12.10.2.1"
    ::= { ieee8021QBridgeVlanStaticEntry 6 }

ieee8021QBridgeVlanStaticRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This object indicates the status of this entry, and is used
        to create/delete entries.  Any object in an entry of this table
        may be modified while the value of the corresponding instance
        of this object is active(1)."
    ::= { ieee8021QBridgeVlanStaticEntry 7 }

ieee8021QBridgeNextFreeLocalVlanTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021QBridgeNextFreeLocalVlanEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table that contains information about the next free VLAN
        value for a statically configured VLAN bridge."
    ::= { ieee8021QBridgeVlan 4 }

ieee8021QBridgeNextFreeLocalVlanEntry OBJECT-TYPE
    SYNTAX      Ieee8021QBridgeNextFreeLocalVlanEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The next free VLAN value for a statically configured VLAN bridge"
    INDEX   { ieee8021QBridgeNextFreeLocalVlanComponentId }
    ::= { ieee8021QBridgeNextFreeLocalVlanTable 1 }

Ieee8021QBridgeNextFreeLocalVlanEntry ::=
    SEQUENCE {
        ieee8021QBridgeNextFreeLocalVlanComponentId
            IEEE8021PbbComponentIdentifier,
        ieee8021QBridgeNextFreeLocalVlanIndex
            Unsigned32
    }

ieee8021QBridgeNextFreeLocalVlanComponentId 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."
    ::= { ieee8021QBridgeNextFreeLocalVlanEntry 1 }

ieee8021QBridgeNextFreeLocalVlanIndex OBJECT-TYPE
    SYNTAX      Unsigned32 (0|4096..4294967295)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The next available value for ieee8021QBridgeVlanIndex of a local
        VLAN entry in ieee8021QBridgeVlanStaticTable.  This will report
        values >=4096 if a new Local VLAN may be created or else
        the value 0 if this is not possible.

        A row creation operation in this table for an entry with a local
        VlanIndex value may fail if the current value of this object
        is not used as the index.  Even if the value read is used,
        there is no guarantee that it will still be the valid index
        when the create operation is attempted; another manager may
        have already got in during the intervening time interval.
        In this case, ieee8021QBridgeNextFreeLocalVlanIndex should be re-read
        and the creation re-tried with the new value.

        This value will automatically change when the current value is
        used to create a new row."
    ::= { ieee8021QBridgeNextFreeLocalVlanEntry 2 }

-- ===============================================================
-- The VLAN Port Configuration Table
-- ===============================================================

ieee8021QBridgePortVlanTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021QBridgePortVlanEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table containing per-port control and status
        information for VLAN configuration in the device."
    REFERENCE   "12.10.1"
    ::= { ieee8021QBridgeVlan 5 }

ieee8021QBridgePortVlanEntry OBJECT-TYPE
    SYNTAX      Ieee8021QBridgePortVlanEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Information controlling VLAN configuration for a port
        on the device.  This is indexed by ieee8021BridgeBasePort."
    AUGMENTS { ieee8021BridgeBasePortEntry }
    ::= { ieee8021QBridgePortVlanTable 1 }

Ieee8021QBridgePortVlanEntry ::=
    SEQUENCE {
        ieee8021QBridgePvid
            IEEE8021VlanIndex,
        ieee8021QBridgePortAcceptableFrameTypes
            IEEE8021PortAcceptableFrameTypes,
        ieee8021QBridgePortIngressFiltering
            TruthValue,
        ieee8021QBridgePortMvrpEnabledStatus
            TruthValue,
        ieee8021QBridgePortMvrpFailedRegistrations
            Counter64,
        ieee8021QBridgePortMvrpLastPduOrigin
            MacAddress,
        ieee8021QBridgePortRestrictedVlanRegistration
            TruthValue
    }

ieee8021QBridgePvid OBJECT-TYPE
    SYNTAX      IEEE8021VlanIndex
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The PVID, the VLAN-ID assigned to untagged frames or
        Priority-Tagged frames received on this port.

        The value of this object MUST be retained across
        reinitializations of the management system."
    REFERENCE   "12.10.1.1"
    DEFVAL      { 1 }
    ::= { ieee8021QBridgePortVlanEntry 1 }

ieee8021QBridgePortAcceptableFrameTypes OBJECT-TYPE
    SYNTAX      IEEE8021PortAcceptableFrameTypes
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "When this is admitTagged(3), the device will
        discard untagged frames or Priority-Tagged frames
        received on this port.  When admitAll(1), untagged
        frames or Priority-Tagged frames received on this port
        will be accepted and assigned to a VID based on the
        PVID and VID Set for this port.

        This control does not affect VLAN-independent Bridge
        Protocol Data Unit (BPDU) frames, such as MVRP and
        Spanning Tree Protocol (STP).  It does affect VLAN-
        dependent BPDU frames, such as MMRP.
        The value of this object MUST be retained across
        reinitializations of the management system."
    REFERENCE   "12.10.1.3"
    DEFVAL      { admitAll }
    ::= { ieee8021QBridgePortVlanEntry 2 }

ieee8021QBridgePortIngressFiltering OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "When this is true(1), the device will discard incoming
        frames for VLANs that do not include this Port in its
        Member set.  When false(2), the port will accept all
        incoming frames.

        This control does not affect VLAN-independent BPDU
        frames, such as MVRP and STP.  It does affect VLAN-
        dependent BPDU frames, such as MMRP.

        The value of this object MUST be retained across
        reinitializations of the management system."
    REFERENCE   "12.10.1.4"
    DEFVAL      { false }
    ::= { ieee8021QBridgePortVlanEntry 3 }

ieee8021QBridgePortMvrpEnabledStatus OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The state of MVRP operation on this port.  The value
        true(1) indicates that MVRP is enabled on this port,
        as long as ieee8021QBridgeMvrpEnabledStatus is also enabled
        for this device.  When false(2) but
        ieee8021QBridgeMvrpEnabledStatus is still
        enabled for the device, MVRP is disabled on this port:
        any MVRP packets received will be silently discarded, and
        no MVRP registrations will be propagated from other
        ports.  This object affects all MVRP Applicant and
        Registrar state machines on this port.  A transition
        from false(2) to true(1) will cause a reset of all
        MVRP state machines on this port.

        The value of this object MUST be retained across
        reinitializations of the management system."
    DEFVAL      { true }
    ::= { ieee8021QBridgePortVlanEntry 4 }

ieee8021QBridgePortMvrpFailedRegistrations OBJECT-TYPE
    SYNTAX      Counter64
    UNITS      "failed MVRP registrations"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The total number of failed MVRP registrations, for any
        reason, on this port.

        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)."
    ::= { ieee8021QBridgePortVlanEntry 5 }

ieee8021QBridgePortMvrpLastPduOrigin OBJECT-TYPE
    SYNTAX      MacAddress
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The Source MAC Address of the last MVRP message
        received on this port."
    ::= { ieee8021QBridgePortVlanEntry 6 }

ieee8021QBridgePortRestrictedVlanRegistration OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The state of Restricted VLAN Registration on this port.
         If the value of this control is true(1), then creation
         of a new dynamic VLAN entry is permitted only if there
         is a Static VLAN Registration Entry for the VLAN concerned,
         in which the Registrar Administrative Control value for
         this port is Normal Registration.

        The value of this object MUST be retained across
        reinitializations of the management system."
    REFERENCE   "11.2.3.2.3, 12.10.1.7."
    DEFVAL      { false }
    ::= { ieee8021QBridgePortVlanEntry 7 }

-- ===============================================================
-- Per port VLAN Statistics Table
-- ===============================================================

ieee8021QBridgePortVlanStatisticsTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021QBridgePortVlanStatisticsEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table containing per-port, per-VLAN statistics for
        traffic received."
    ::= { ieee8021QBridgeVlan 6 }

ieee8021QBridgePortVlanStatisticsEntry OBJECT-TYPE
    SYNTAX      Ieee8021QBridgePortVlanStatisticsEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Traffic statistics for a VLAN on an interface."
    INDEX   { ieee8021BridgeBasePortComponentId,
              ieee8021BridgeBasePort,
              ieee8021QBridgeVlanIndex }
    ::= { ieee8021QBridgePortVlanStatisticsTable 1 }

Ieee8021QBridgePortVlanStatisticsEntry ::=
    SEQUENCE {
        ieee8021QBridgeTpVlanPortInFrames
            Counter64,
        ieee8021QBridgeTpVlanPortOutFrames
            Counter64,
        ieee8021QBridgeTpVlanPortInDiscards
            Counter64
    }

ieee8021QBridgeTpVlanPortInFrames OBJECT-TYPE
```

```
    SYNTAX      Counter64
    UNITS      "frames"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of valid frames received by this port from
        its segment that were classified as belonging to this
        VLAN.  Note that a frame received on this port is
        counted by this object if and only if it is for a
        protocol being processed by the local forwarding process
        for this VLAN.  This object includes received bridge
        management frames classified as belonging to this VLAN
        (e.g., MMRP, but not MVRP or STP.

        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)"
    ::= { ieee8021QBridgePortVlanStatisticsEntry 1 }

ieee8021QBridgeTpVlanPortOutFrames OBJECT-TYPE
    SYNTAX      Counter64
    UNITS      "frames"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of valid frames transmitted by this port to
        its segment from the local forwarding process for this
        VLAN.  This includes bridge management frames originated
        by this device that are classified as belonging to this
        VLAN (e.g., MMRP, but not MVRP or STP).

        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)"
    ::= { ieee8021QBridgePortVlanStatisticsEntry 2 }

ieee8021QBridgeTpVlanPortInDiscards OBJECT-TYPE
    SYNTAX      Counter64
    UNITS      "frames"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of valid frames received by this port from
        its segment that were classified as belonging to this
        VLAN and that were discarded due to VLAN-related reasons.
        Specifically, the IEEE 802.1Q counters for Discard
        Inbound and Discard on Ingress Filtering.

        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"
    ::= { ieee8021QBridgePortVlanStatisticsEntry 3 }

-- =============================================================
-- The VLAN Learning Constraints Table
-- =============================================================

ieee8021QBridgeLearningConstraintsTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021QBridgeLearningConstraintsEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table containing learning constraints for sets of
        Shared and Independent VLANs.  Entries in this table are
        persistent and are preserved across reboots."
    REFERENCE   "12.10.3.1"
    ::= { ieee8021QBridgeVlan 8 }

ieee8021QBridgeLearningConstraintsEntry OBJECT-TYPE
    SYNTAX      Ieee8021QBridgeLearningConstraintsEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A learning constraint defined for a VLAN."
    INDEX   { ieee8021QBridgeLearningConstraintsComponentId,
              ieee8021QBridgeLearningConstraintsVlan,
              ieee8021QBridgeLearningConstraintsSet }
    ::= { ieee8021QBridgeLearningConstraintsTable 1 }

Ieee8021QBridgeLearningConstraintsEntry ::=
    SEQUENCE {
        ieee8021QBridgeLearningConstraintsComponentId
            IEEE8021PbbComponentIdentifier,
        ieee8021QBridgeLearningConstraintsVlan
            IEEE8021VlanIndex,
        ieee8021QBridgeLearningConstraintsSet
            Integer32,
        ieee8021QBridgeLearningConstraintsType
            INTEGER,
        ieee8021QBridgeLearningConstraintsStatus
            RowStatus
    }

ieee8021QBridgeLearningConstraintsComponentId 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."
    ::= { ieee8021QBridgeLearningConstraintsEntry 1 }

ieee8021QBridgeLearningConstraintsVlan OBJECT-TYPE
    SYNTAX      IEEE8021VlanIndex
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The index of the row in ieee8021QBridgeVlanCurrentTable for the
        VLAN constrained by this entry."
    ::= { ieee8021QBridgeLearningConstraintsEntry 2 }

ieee8021QBridgeLearningConstraintsSet OBJECT-TYPE
    SYNTAX      Integer32 (0..65535)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The identity of the constraint set to which
        ieee8021QBridgeLearningConstraintsVlan belongs.  These values may
        be chosen by the management station."
    ::= { ieee8021QBridgeLearningConstraintsEntry 3 }

ieee8021QBridgeLearningConstraintsType OBJECT-TYPE
    SYNTAX      INTEGER {
                    independent(1),
                    shared(2)
                }
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The type of constraint this entry defines.
            independent(1) - the VLAN, ieee8021QBridgeLearningConstraintsVlan,
                uses a filtering database independent from all
                other VLANs in the same set, defined by
                ieee8021QBridgeLearningConstraintsSet.
            shared(2) - the VLAN, ieee8021QBridgeLearningConstraintsVlan,
                shares the same filtering database as all other VLANs
                in the same set, defined by
                ieee8021QBridgeLearningConstraintsSet."
    ::= { ieee8021QBridgeLearningConstraintsEntry 4 }

ieee8021QBridgeLearningConstraintsStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The status of this entry.  Any object in an entry of this table
        may be modified while the value of the corresponding instance
        of this object is active(1)."
    ::= { ieee8021QBridgeLearningConstraintsEntry 5 }

ieee8021QBridgeLearningConstraintDefaultsTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021QBridgeLearningConstraintDefaultsEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table containing learning constraints for sets of
        Shared and Independent VLANs."
    REFERENCE   "12.10.3.1"
    ::= { ieee8021QBridgeVlan 9 }

ieee8021QBridgeLearningConstraintDefaultsEntry OBJECT-TYPE
    SYNTAX      Ieee8021QBridgeLearningConstraintDefaultsEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A learning constraint defined for a VLAN."
    INDEX   { ieee8021QBridgeLearningConstraintDefaultsComponentId }
    ::= { ieee8021QBridgeLearningConstraintDefaultsTable 1 }

Ieee8021QBridgeLearningConstraintDefaultsEntry ::=
    SEQUENCE {
        ieee8021QBridgeLearningConstraintDefaultsComponentId
            IEEE8021PbbComponentIdentifier,
        ieee8021QBridgeLearningConstraintDefaultsSet
            Integer32,
        ieee8021QBridgeLearningConstraintDefaultsType
            INTEGER
    }

ieee8021QBridgeLearningConstraintDefaultsComponentId 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."
    ::= { ieee8021QBridgeLearningConstraintDefaultsEntry 1 }

ieee8021QBridgeLearningConstraintDefaultsSet OBJECT-TYPE
    SYNTAX      Integer32 (0..65535)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The identity of the constraint set to which a VLAN
        belongs, if there is not an explicit entry for that VLAN
        in ieee8021QBridgeLearningConstraintsTable.

        The value of this object MUST be retained across
        reinitializations of the management system."
    ::= { ieee8021QBridgeLearningConstraintDefaultsEntry 2 }

ieee8021QBridgeLearningConstraintDefaultsType OBJECT-TYPE
    SYNTAX      INTEGER {
                    independent(1),
                    shared(2)
                }
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The type of constraint set to which a VLAN belongs, if
        there is not an explicit entry for that VLAN in
        ieee8021QBridgeLearningConstraintsTable.  The types are as defined
        for ieee8021QBridgeLearningConstraintsType.

        The value of this object MUST be retained across
        reinitializations of the management system."
    ::= { ieee8021QBridgeLearningConstraintDefaultsEntry 3 }

-- ===========================================================
-- ieee8021QBridgeProtocol subtree
-- ===========================================================

ieee8021QBridgeProtocolGroupTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021QBridgeProtocolGroupEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table that contains mappings from Protocol
         Templates to Protocol Group Identifiers used for
         Port-and-Protocol-based VLAN Classification.

         Entries in this table must be persistent over power
         up restart/reboot."
    REFERENCE   "12.10.1"
    ::= { ieee8021QBridgeProtocol 1 }

ieee8021QBridgeProtocolGroupEntry OBJECT-TYPE
    SYNTAX      Ieee8021QBridgeProtocolGroupEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A mapping from a Protocol Template to a Protocol
         Group Identifier."
    REFERENCE   "12.10.1.1.3 d)"
    INDEX       { ieee8021QBridgeProtocolGroupComponentId,
                  ieee8021QBridgeProtocolTemplateFrameType,
                  ieee8021QBridgeProtocolTemplateProtocolValue }
    ::= { ieee8021QBridgeProtocolGroupTable 1 }

Ieee8021QBridgeProtocolGroupEntry ::=
    SEQUENCE {
        ieee8021QBridgeProtocolGroupComponentId
            IEEE8021PbbComponentIdentifier,
        ieee8021QBridgeProtocolTemplateFrameType
            INTEGER,
        ieee8021QBridgeProtocolTemplateProtocolValue
            OCTET STRING,
        ieee8021QBridgeProtocolGroupId
            Integer32,
        ieee8021QBridgeProtocolGroupRowStatus
            RowStatus
    }

ieee8021QBridgeProtocolGroupComponentId 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."
    ::= { ieee8021QBridgeProtocolGroupEntry 1 }

ieee8021QBridgeProtocolTemplateFrameType OBJECT-TYPE
    SYNTAX      INTEGER {
                  ethernet  (1),
                  rfc1042   (2),
                  snap8021H (3),
                  snapOther (4),
                  llcOther  (5)
                }
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The data-link encapsulation format or the
         'detagged_frame_type' in a Protocol Template."
    REFERENCE   "12.10.1.8"
    ::= { ieee8021QBridgeProtocolGroupEntry 2 }

ieee8021QBridgeProtocolTemplateProtocolValue OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE (2 | 5))
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The identification of the protocol above the data-link
         layer in a Protocol Template.  Depending on the
         frame type, the octet string will have one of the
         following values:

         For 'ethernet', 'rfc1042' and 'snap8021H',
             this is the 16-bit (2-octet) IEEE 802.3 Type Field.
         For 'snapOther',
             this is the 40-bit (5-octet) PID.
         For 'llcOther',
             this is the 2-octet IEEE 802.2 Link Service Access
             Point (LSAP) pair: first octet for Destination Service
```

```
              Access Point (DSAP) and second octet for Source Service
              Access Point (SSAP)."
    REFERENCE   "12.10.1.8"
    ::= { ieee8021QBridgeProtocolGroupEntry 3 }

ieee8021QBridgeProtocolGroupId OBJECT-TYPE
    SYNTAX      Integer32 (0..2147483647)
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Represents a group of protocols that are associated
         together when assigning a VID to a frame."
    REFERENCE   "12.10.1.8"
    ::= { ieee8021QBridgeProtocolGroupEntry 4 }

ieee8021QBridgeProtocolGroupRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This object indicates the status of this entry."
    ::= { ieee8021QBridgeProtocolGroupEntry 5 }

ieee8021QBridgeProtocolPortTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021QBridgeProtocolPortEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table that contains VID sets used for
         Port-and-Protocol-based VLAN Classification."
    REFERENCE   "12.10.1"
    ::= { ieee8021QBridgeProtocol 2 }

ieee8021QBridgeProtocolPortEntry OBJECT-TYPE
    SYNTAX      Ieee8021QBridgeProtocolPortEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A VID set for a port."
    REFERENCE   "12.10.1.1.3 c)"
    INDEX       { ieee8021BridgeBasePortComponentId,
                  ieee8021BridgeBasePort,
                  ieee8021QBridgeProtocolPortGroupId }
    ::= { ieee8021QBridgeProtocolPortTable 1 }

Ieee8021QBridgeProtocolPortEntry ::=
    SEQUENCE {
        ieee8021QBridgeProtocolPortGroupId
            Integer32,
        ieee8021QBridgeProtocolPortGroupVid
            VlanId,
        ieee8021QBridgeProtocolPortRowStatus
            RowStatus
    }

ieee8021QBridgeProtocolPortGroupId OBJECT-TYPE
    SYNTAX      Integer32 (1..2147483647)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Designates a group of protocols in the Protocol
         Group Database."
    REFERENCE   "12.10.1.2"
    ::= { ieee8021QBridgeProtocolPortEntry 1 }
```

```
ieee8021QBridgeProtocolPortGroupVid OBJECT-TYPE
    SYNTAX      VlanId
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The VID associated with a group of protocols for
         each port."
    REFERENCE   "12.10.1.2"
    ::= { ieee8021QBridgeProtocolPortEntry 2 }

ieee8021QBridgeProtocolPortRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This object indicates the status of this entry."
    ::= { ieee8021QBridgeProtocolPortEntry 3 }

-- ============================================================
-- ieee8021QBridgeVIDX subtree
--
-- ============================================================

ieee8021QBridgeVIDXTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021QBridgeVIDXEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This table is used to configure the VID Translation
        Table defined in 12.10.1.8 and 6.9.  The Bridge VID
        Translation Table is used to implement a mapping between a
        local VID, and a relay VID, used by the filtering and
        forwarding process.  Each row in this table is indexed by
        component, port, and local VID value and a value to be used
        for the specified VID as specified in (6.9).  Entries in
        this table must be persistent over power up restart/reboot."
 REFERENCE   "12.10.1.8 "
    ::= { ieee8021QBridgeVIDX 1 }

ieee8021QBridgeVIDXEntry OBJECT-TYPE
    SYNTAX      Ieee8021QBridgeVIDXEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An entry for the S-VID translation table which includes
        both the Local and Relay S-VIDs."

    INDEX { ieee8021BridgeBasePortComponentId,
            ieee8021BridgeBasePort,
            ieee8021QBridgeVIDXLocalVid }
    ::= { ieee8021QBridgeVIDXTable 1 }

Ieee8021QBridgeVIDXEntry ::=
      SEQUENCE {
          ieee8021QBridgeVIDXLocalVid VlanId,
          ieee8021QBridgeVIDXRelayVid VlanId,
          ieee8021QBridgeVIDXRowStatus RowStatus
}

ieee8021QBridgeVIDXLocalVid OBJECT-TYPE
    SYNTAX      VlanId
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The Local VID after translation received at the ISS or EISS."
    REFERENCE   "12.10.1.8.1, 12.10.1.8.2 "
    ::= { ieee8021QBridgeVIDXEntry 1 }

ieee8021QBridgeVIDXRelayVid OBJECT-TYPE
    SYNTAX      VlanId
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The Relay VID received before translation received at ISS or EISS."
    REFERENCE   "12.10.1.8.1, 12.10.1.8.2 "
    ::= { ieee8021QBridgeVIDXEntry 2 }

ieee8021QBridgeVIDXRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This indicates the status of an entry in this table,
        and is used to create/delete entries. It is an
        implementation specific decision as to whether
        any column in this table may be set while the
        corresponding instance of this object is valid(1)."
    REFERENCE   "12.10.1.8.1, 12.10.1.8.2 "
    ::= { ieee8021QBridgeVIDXEntry 3 }

-- ==========================================================
-- ieee8021QBridgeEgressVidXTable:
-- ==========================================================

ieee8021QBridgeEgressVidXTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021QBridgeEgressVidXEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This table is used to configure the VID Translation
        Table defined in 12.10.1.9 and 6.9.  The Bridge VID
        Egress Translation Table is used to implement a mapping between a
        relay VID, and a local VID, used by the filtering and
        forwarding process.  Each row in this table is indexed by
        component, port, and relay VID value and a value to be used
        for the specified local VID as specified in (6.9).  Entries in
        this table must be persistent over power up restart/reboot."
    REFERENCE   "12.10.1.9, 6.9"
    ::= { ieee8021QBridgeVIDX 2 }

ieee8021QBridgeEgressVidXEntry OBJECT-TYPE
    SYNTAX      Ieee8021QBridgeEgressVidXEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An entry for the Egress VID translation table which includes
        both the relay and local IDs between which the PNP or CNP
        translates."

    INDEX { ieee8021BridgeBaseEgressPortComponentId,
            ieee8021BridgeEgressBasePort,
            ieee8021QBridgeEgressVidXRelayVid }
    ::= { ieee8021QBridgeEgressVidXTable 1 }

Ieee8021QBridgeEgressVidXEntry ::=
      SEQUENCE {
          ieee8021QBridgeEgressVidXRelayVid VlanId,
          ieee8021QBridgeEgressVidXLocalVid VlanId,
          ieee8021QBridgeEgressVidXRowStatus RowStatus
}

ieee8021QBridgeEgressVidXRelayVid OBJECT-TYPE
    SYNTAX      VlanId
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The Relay VID after translation transmitted to the
         ISS or EISS."
    REFERENCE   "12.10.1.9.1, 12.10.1.9.2 "
     ::= { ieee8021QBridgeEgressVidXEntry 1 }

ieee8021QBridgeEgressVidXLocalVid OBJECT-TYPE
    SYNTAX      VlanId
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The Local VID before translation transmitted to the
         ISS or EISS."
    REFERENCE   "12.10.1.9.1, 12.10.1.9.2 "
    ::= { ieee8021QBridgeEgressVidXEntry 2 }


ieee8021QBridgeEgressVidXRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
       "This indicates the status of an entry in this table,
        and is used to create/delete entries. It is an
        implementation specific decision as to whether
        any column in this table may be set while the
        corresponding instance of this object is valid(1)."
    REFERENCE   "12.10.1.9.1, 12.10.1.9.2 "
   ::= { ieee8021QBridgeEgressVidXEntry 3 }

-- ===============================================================
-- IEEE 802.1Q MIB - Conformance Information
-- ===============================================================

ieee8021QBridgeConformance
    OBJECT IDENTIFIER ::= { ieee8021QBridgeMib 2 }

ieee8021QBridgeGroups
    OBJECT IDENTIFIER ::= { ieee8021QBridgeConformance 1 }

ieee8021QBridgeCompliances
    OBJECT IDENTIFIER ::= { ieee8021QBridgeConformance 2 }

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

ieee8021QBridgeBaseGroup OBJECT-GROUP
    OBJECTS {
        ieee8021QBridgeVlanVersionNumber,
        ieee8021QBridgeMaxVlanId,
        ieee8021QBridgeMaxSupportedVlans,
        ieee8021QBridgeNumVlans,
        ieee8021QBridgeMvrpEnabledStatus
    }
    STATUS      current
    DESCRIPTION
        "A collection of objects providing device-level control
        and status information for the Virtual LAN bridge
        services."
    ::= { ieee8021QBridgeGroups 1 }

ieee8021QBridgeFdbUnicastGroup OBJECT-GROUP
    OBJECTS {
        ieee8021QBridgeFdbDynamicCount,
        ieee8021QBridgeFdbLearnedEntryDiscards,
        ieee8021QBridgeFdbAgingTime,
        ieee8021QBridgeTpFdbPort,
        ieee8021QBridgeTpFdbStatus
    }
    STATUS      current
    DESCRIPTION
        "A collection of objects providing information about all
        unicast addresses, learned dynamically or statically
        configured by management, in each Filtering Database."
    ::= { ieee8021QBridgeGroups 2 }

ieee8021QBridgeFdbMulticastGroup OBJECT-GROUP
    OBJECTS {
        ieee8021QBridgeTpGroupEgressPorts,
        ieee8021QBridgeTpGroupLearnt
    }
    STATUS      current
    DESCRIPTION
        "A collection of objects providing information about all
        multicast addresses, learned dynamically or statically
        configured by management, in each Filtering Database."
    ::= { ieee8021QBridgeGroups 3 }

ieee8021QBridgeServiceRequirementsGroup OBJECT-GROUP
    OBJECTS {
        ieee8021QBridgeForwardAllPorts,
        ieee8021QBridgeForwardAllStaticPorts,
        ieee8021QBridgeForwardAllForbiddenPorts,
        ieee8021QBridgeForwardUnregisteredPorts,
        ieee8021QBridgeForwardUnregisteredStaticPorts,
        ieee8021QBridgeForwardUnregisteredForbiddenPorts
    }
    STATUS      current
    DESCRIPTION
        "A collection of objects providing information about
        service requirements, learned dynamically or statically
        configured by management, in each Filtering Database."
    ::= { ieee8021QBridgeGroups 4 }

ieee8021QBridgeFdbStaticGroup OBJECT-GROUP
    OBJECTS {
        ieee8021QBridgeStaticUnicastStaticEgressPorts,
        ieee8021QBridgeStaticUnicastForbiddenEgressPorts,
        ieee8021QBridgeStaticUnicastStorageType,
        ieee8021QBridgeStaticUnicastRowStatus,
        ieee8021QBridgeStaticMulticastStaticEgressPorts,
        ieee8021QBridgeStaticMulticastForbiddenEgressPorts,
        ieee8021QBridgeStaticMulticastStorageType,
        ieee8021QBridgeStaticMulticastRowStatus
    }
    STATUS      current
    DESCRIPTION
        "A collection of objects providing information about
        unicast and multicast addresses statically configured by
        management, in each Filtering Database or VLAN."
    ::= { ieee8021QBridgeGroups 5 }

ieee8021QBridgeVlanGroup OBJECT-GROUP
    OBJECTS {
        ieee8021QBridgeVlanNumDeletes,
        ieee8021QBridgeVlanFdbId,
        ieee8021QBridgeVlanCurrentEgressPorts,
        ieee8021QBridgeVlanCurrentUntaggedPorts,
        ieee8021QBridgeVlanStatus,
        ieee8021QBridgeVlanCreationTime
    }
    STATUS      current
    DESCRIPTION
        "A collection of objects providing information about
        all VLANs currently configured on this device."
    ::= { ieee8021QBridgeGroups 6 }

ieee8021QBridgeVlanStaticGroup OBJECT-GROUP
    OBJECTS {
        ieee8021QBridgeVlanStaticName,
        ieee8021QBridgeVlanStaticEgressPorts,
        ieee8021QBridgeVlanForbiddenEgressPorts,
        ieee8021QBridgeVlanStaticUntaggedPorts,
        ieee8021QBridgeVlanStaticRowStatus,
        ieee8021QBridgeNextFreeLocalVlanIndex
    }
    STATUS      current
    DESCRIPTION
        "A collection of objects providing information about
        VLANs statically configured by management."
    ::= { ieee8021QBridgeGroups 7 }

ieee8021QBridgeVlanStatisticsGroup OBJECT-GROUP
    OBJECTS {
        ieee8021QBridgeTpVlanPortInFrames,
        ieee8021QBridgeTpVlanPortOutFrames,
        ieee8021QBridgeTpVlanPortInDiscards
    }
    STATUS      current
    DESCRIPTION
        "A collection of objects providing per-port packet
        statistics for all VLANs currently configured on this
        device."
    ::= { ieee8021QBridgeGroups 8 }

ieee8021QBridgeLearningConstraintsGroup OBJECT-GROUP
    OBJECTS {
        ieee8021QBridgeLearningConstraintsType,
        ieee8021QBridgeLearningConstraintsStatus
    }
    STATUS      current
    DESCRIPTION
        "A collection of objects defining the Filtering Database
        constraints all VLANs have with each other."
    ::= { ieee8021QBridgeGroups 9 }

ieee8021QBridgeLearningConstraintDefaultGroup OBJECT-GROUP
    OBJECTS {
        ieee8021QBridgeLearningConstraintDefaultsSet,
        ieee8021QBridgeLearningConstraintDefaultsType
    }
    STATUS      current
    DESCRIPTION
```

```
        "A collection of objects defining the default Filtering
        Database constraints for VLANs that have no specific
        constraints defined."
    ::= { ieee8021QBridgeGroups 10 }

ieee8021QBridgeClassificationDeviceGroup OBJECT-GROUP
    OBJECTS {
        ieee8021QBridgeProtocolGroupId,
        ieee8021QBridgeProtocolGroupRowStatus
    }
    STATUS      current
    DESCRIPTION
        "VLAN classification information for the bridge."
    ::= { ieee8021QBridgeGroups 11 }

ieee8021QBridgeClassificationPortGroup OBJECT-GROUP
    OBJECTS {
        ieee8021QBridgeProtocolPortGroupVid,
        ieee8021QBridgeProtocolPortRowStatus
    }
    STATUS      current
    DESCRIPTION
        "VLAN classification information for individual ports."
    ::= { ieee8021QBridgeGroups 12 }

ieee8021QBridgePortGroup2 OBJECT-GROUP
    OBJECTS {
        ieee8021QBridgePvid,
        ieee8021QBridgePortAcceptableFrameTypes,
        ieee8021QBridgePortIngressFiltering,
        ieee8021QBridgePortMvrpEnabledStatus,
        ieee8021QBridgePortMvrpFailedRegistrations,
        ieee8021QBridgePortMvrpLastPduOrigin,
        ieee8021QBridgePortRestrictedVlanRegistration
    }
    STATUS      current
    DESCRIPTION
        "A collection of objects providing port-level VLAN
        control and status information for all ports."
    ::= { ieee8021QBridgeGroups 13 }

ieee8021QBridgeCVlanPortGroup OBJECT-GROUP
    OBJECTS {
        ieee8021QBridgeCVlanPortRowStatus
    }
    STATUS      current
    DESCRIPTION
        "Objects used to create/delete customer VLAN ports."
    ::= { ieee8021QBridgeGroups 14 }

ieee8021QBridgeVIDXGroup OBJECT-GROUP
    OBJECTS {
        ieee8021QBridgeVIDXRelayVid,
        ieee8021QBridgeVIDXRowStatus
    }
    STATUS      current
    DESCRIPTION
        "Ingress or Ingress/Egress VID translation for
        individual ports."
    ::= { ieee8021QBridgeGroups 15 }

ieee8021QBridgeEgressVIDXGroup OBJECT-GROUP
    OBJECTS {
        ieee8021QBridgeEgressVidXLocalVid,
        ieee8021QBridgeEgressVidXRowStatus
    }
    STATUS      current
    DESCRIPTION
        "Egress VID translation for individual ports."
    ::= { ieee8021QBridgeGroups 16 }

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

ieee8021QBridgeCompliance MODULE-COMPLIANCE
    STATUS  current
    DESCRIPTION
        "The compliance statement for device support of Virtual
        LAN Bridge services."

    MODULE
        MANDATORY-GROUPS {
            ieee8021QBridgeBaseGroup,
            ieee8021QBridgeVlanGroup,
            ieee8021QBridgeVlanStaticGroup,
            ieee8021QBridgePortGroup2
        }

        GROUP       ieee8021QBridgeFdbUnicastGroup
        DESCRIPTION
            "This group is mandatory for bridges that implement
            802.1Q transparent bridging."

        GROUP       ieee8021QBridgeFdbMulticastGroup
        DESCRIPTION
            "This group is mandatory for bridges that implement
            802.1Q transparent bridging."
        GROUP       ieee8021QBridgeServiceRequirementsGroup
        DESCRIPTION
            "This group is mandatory for bridges that implement
            extended filtering services.  All objects must be
            read-write if extended-filtering services are
            enabled."

        GROUP       ieee8021QBridgeFdbStaticGroup
        DESCRIPTION
            "This group is optional."

        GROUP       ieee8021QBridgeVlanStatisticsGroup
        DESCRIPTION
            "This group is optional as there may be significant
            implementation cost associated with its support."

        GROUP       ieee8021QBridgeLearningConstraintsGroup
        DESCRIPTION
            "This group is mandatory for devices implementing
             both Independent VLAN Learning (IVL) and Shared
             VLAN Learning (SVL) modes of operation of the
             filtering database, as defined by IEEE 802.1Q."

        GROUP       ieee8021QBridgeLearningConstraintDefaultGroup
        DESCRIPTION
            "This group is mandatory for devices implementing
             both Independent VLAN Learning (IVL) and Shared
             VLAN Learning (SVL) modes of operation of the
             filtering database, as defined by IEEE 802.1Q."
        GROUP       ieee8021QBridgeClassificationDeviceGroup
        DESCRIPTION
            "This group is mandatory ONLY for devices implementing
             VLAN Classification as specified in IEEE 802.1v."

        GROUP       ieee8021QBridgeClassificationPortGroup
        DESCRIPTION
            "This group is mandatory ONLY for devices implementing
             VLAN Classification as specified in IEEE 802.1v."

        GROUP       ieee8021QBridgeCVlanPortGroup
        DESCRIPTION
            "This group is mandatory ONLY for devices supporting
             creation/deletion of customer VLAN ports."

        GROUP       ieee8021QBridgeVIDXGroup
        DESCRIPTION
            "This group is mandatory ONLY for devices supporting
             VID translation of customer and/or provider VLAN ports."

        GROUP       ieee8021QBridgeEgressVIDXGroup
        DESCRIPTION
            "This group is mandatory ONLY for devices supporting
             separate Ingress and Egress VID translation of
             of customer and provider VLAN ports."

        OBJECT      ieee8021QBridgePortAcceptableFrameTypes
        MIN-ACCESS  read-only
        DESCRIPTION
            "Write access is not required as this is an optional
            capability in IEEE 802.1Q."

        OBJECT      ieee8021QBridgePortIngressFiltering
        MIN-ACCESS  read-only
        DESCRIPTION
            "Write access is not required as this is an optional
            capability in IEEE 802.1Q."

        OBJECT      ieee8021QBridgeLearningConstraintDefaultsSet
        MIN-ACCESS  read-only
        DESCRIPTION
            "Write access is not required as this is an optional
            capability in IEEE 802.1Q."

        OBJECT      ieee8021QBridgeLearningConstraintDefaultsType
        MIN-ACCESS  read-only
        DESCRIPTION
            "Write access is not required as this is an optional
            capability in IEEE 802.1Q."

        OBJECT      ieee8021QBridgeProtocolGroupId
        MIN-ACCESS  read-only
        DESCRIPTION
            "Write access is not required as this is an optional
            capability in IEEE 802.1v."

        OBJECT      ieee8021QBridgeProtocolGroupRowStatus
        MIN-ACCESS  read-only
        DESCRIPTION
            "Write access is not required as this is an optional
            capability in IEEE 802.1v."

        ::= { ieee8021QBridgeCompliances 1 }

END
```

17.7.5 Definitions for the IEEE8021-PB MIB module

Delete the entire text of 17.7.5, and insert the following text:

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

IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE
        FROM SNMPv2-SMI
    TruthValue, RowStatus
        FROM SNMPv2-TC
    ieee802dot1mibs, IEEE8021PbbComponentIdentifierOrZero,
    IEEE8021PbbComponentIdentifier, IEEE8021BridgePortNumber,
    IEEE8021PortAcceptableFrameTypes, IEEE8021PriorityValue,
    IEEE8021BridgePortNumberOrZero, IEEE8021BridgePortType
        FROM IEEE8021-TC-MIB
    ieee8021BridgeBasePortComponentId, ieee8021BridgeBasePort
        FROM IEEE8021-BRIDGE-MIB
    VlanId, VlanIdOrNone
        FROM Q-BRIDGE-MIB
    MODULE-COMPLIANCE, OBJECT-GROUP
        FROM SNMPv2-CONF;

ieee8021PbMib MODULE-IDENTITY
    LAST-UPDATED  "201202100000Z" -- February 10, 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
        "Provider Bridge MIB module.

         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     "201202100000Z" -- February 10, 2012"
    DESCRIPTION
         "Deprecated ieee8021PbVidTranslationTable
          moved the new object to the Q Bridge as part
          of VID translation for 802.1aq."
    REVISION     "201104060000Z" -- April 6, 2011
    DESCRIPTION
         "Additions to support Remote Customer Service Interfaces."
    REVISION     "201102270000Z" -- February 27, 2011
    DESCRIPTION
         "Change to ieee8021PbEdgePortAcceptableFrameTypes
          permissible values, addition of
          IEEE8021BridgePortNumberOrZero to IMPORTS,
          as part of 2011 revision of IEEE Std 802.1Q."
    REVISION     "201008260000Z" -- August 26, 2010
    DESCRIPTION
         "Minor edits to contact information etc. as part of
          revision of Std 802.1Q."

    REVISION     "200810150000Z" -- October 15, 2008
    DESCRIPTION
        "Initial version."
    ::= { ieee802dot1mibs 5 }

ieee8021PbNotifications  OBJECT IDENTIFIER ::= { ieee8021PbMib 0 }
ieee8021PbObjects        OBJECT IDENTIFIER ::= { ieee8021PbMib 1 }
ieee8021PbConformance    OBJECT IDENTIFIER ::= { ieee8021PbMib 2 }

-- =============================================================
-- ieee8021PbVidTranslationTable:
-- Deprecated and moved VID translation to Q Bridge MIB.
-- =============================================================

ieee8021PbVidTranslationTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021PbVidTranslationEntry
    MAX-ACCESS  not-accessible
    STATUS      deprecated
    DESCRIPTION
        "This table is used to configure the VID Translation Table
        defined in 12.13.2 a) of 802.1Q-2006.  The VID
        Translation Table is used to implement a bi-directional
        mapping between a local S-VID, used in data and protocol
        frames transmitted and received through a CNP or PNP,
        and a relay S-VID, used by the filtering and forwarding
        process.  Each row in this table is indexed by component,
        port, and local S-VID value and indicates the relay S-VID
        value to be used for the specified S-VID.  If no entry for
        a component, port, and local-svid is present in this table
        is present then the relay S-VID used for a frame received
        on that port with the local S-VID value will be the S-VID
        that has the same numeric value as the local S-VID of the
        received frame.

        Entries in this table must be persistent over power up
        restart/reboot."
    REFERENCE   "12.13.2 a), 12.13.2.1, 12.13.2.2"
    ::= { ieee8021PbObjects 1 }

ieee8021PbVidTranslationEntry OBJECT-TYPE
    SYNTAX      Ieee8021PbVidTranslationEntry
    MAX-ACCESS  not-accessible
    STATUS      deprecated
    DESCRIPTION
        "An entry for the S-VID translation table which includes
        both the Local and Relay S-VIDs between which the PNP or CNP
        translates.

        Note that the component ID of entries in this table must refer
        to the S-VLAN Component of a Provider Bridge."
    INDEX { ieee8021BridgeBasePortComponentId,
            ieee8021BridgeBasePort,
            ieee8021PbVidTranslationLocalVid }
    ::= { ieee8021PbVidTranslationTable 1 }

Ieee8021PbVidTranslationEntry ::= SEQUENCE {
      ieee8021PbVidTranslationLocalVid
          VlanId,
      ieee8021PbVidTranslationRelayVid
          VlanId,
      ieee8021PbVidTranslationRowStatus
          RowStatus
}

ieee8021PbVidTranslationLocalVid OBJECT-TYPE
    SYNTAX      VlanId
    MAX-ACCESS  not-accessible
    STATUS      deprecated
    DESCRIPTION
        "The S-VID on received(transmitted) at the ISS of a CNP or PNP."
    ::= { ieee8021PbVidTranslationEntry 1 }

ieee8021PbVidTranslationRelayVid OBJECT-TYPE
    SYNTAX      VlanId
    MAX-ACCESS  read-create
    STATUS      deprecated
    DESCRIPTION
        "The translated S-VID delivered(received) over the EISS from a
        CNP or PNP.  The default value of this object on creation will
        be the value of the corresponding instance of
        ieee8021PbVidTranslationLocalVid."
    ::= { ieee8021PbVidTranslationEntry 2 }

ieee8021PbVidTranslationRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      deprecated
    DESCRIPTION
        "This indicates the status of an entry in this table, and is
        used to create/delete entries.

        It is an implementation specific decision as to whether any
        column in this table may be set while the corresponding
        instance of this object is valid(1)."
    ::= { ieee8021PbVidTranslationEntry 3 }

-- ===========================================================
-- ieee8021PbCVidRegistrationTable:
-- ===========================================================

ieee8021PbCVidRegistrationTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021PbCVidRegistrationEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table used in a CEP to create the mapping between a C-VID
        and a service represented by an S-VID.

        Note that the component ID of entries in this table must refer
        to the S-VLAN component of a Provider Edge Bridge and the Port
        Number must refer to the port number of the Customer Edge Port
        associated with that Provider Edge Bridge.

        Entries in this table must be persistent over power up
        restart/reboot."
    REFERENCE   "12.13.2.1, 12.13.2.2"
    ::= { ieee8021PbObjects 2 }

ieee8021PbCVidRegistrationEntry OBJECT-TYPE
    SYNTAX      Ieee8021PbCVidRegistrationEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An element of the C-VID registration table accessed by PB
        C-VLAN component, Customer Edge Port bridge port number, and
        C-VID.  Each element contains the mapping between a C-VID and
        the S-VID which carries the service and booleans for handling
        untagged frames at the PEP and CEP."
    INDEX { ieee8021BridgeBasePortComponentId,
            ieee8021BridgeBasePort,
            ieee8021PbCVidRegistrationCVid }
    ::= { ieee8021PbCVidRegistrationTable 1 }

Ieee8021PbCVidRegistrationEntry ::= SEQUENCE {
      ieee8021PbCVidRegistrationCVid
          VlanId,
      ieee8021PbCVidRegistrationSVid
          VlanId,
      ieee8021PbCVidRegistrationUntaggedPep
          TruthValue,
      ieee8021PbCVidRegistrationUntaggedCep
          TruthValue,
      ieee8021PbCVidRegistrationRowStatus
          RowStatus
}

ieee8021PbCVidRegistrationCVid OBJECT-TYPE
    SYNTAX      VlanId
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "C-VID of this C-VID registration entry."
    ::= { ieee8021PbCVidRegistrationEntry 1 }

ieee8021PbCVidRegistrationSVid OBJECT-TYPE
    SYNTAX      VlanId
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "S-VID for this C-VID registration entry."
    ::= { ieee8021PbCVidRegistrationEntry 2 }

ieee8021PbCVidRegistrationUntaggedPep OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "A flag indicating if this C-VID should be carried untagged
        at the PEP.  A value of true(1) means untagged."
    DEFVAL { true }
    ::= { ieee8021PbCVidRegistrationEntry 3 }

ieee8021PbCVidRegistrationUntaggedCep OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "A flag indicating if this C-VID should be carried untagged
        at the CEP.  A value of true(1) means untagged."
    DEFVAL { true }
    ::= { ieee8021PbCVidRegistrationEntry 4 }

ieee8021PbCVidRegistrationRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This indicates the status of an entry in this table, and is
        used to create/delete entries.

        The value of ieee8021PbCVidRegistrationSVid must be set before
        an entry in this table can be made valid.

        It is an implementation specific decision as to whether any
        column in this table may be set while the corresponding
        instance of this object is valid(1)."
    ::= { ieee8021PbCVidRegistrationEntry 5 }

-- ==========================================================
-- ieee8021PbEdgePortTable:
-- ==========================================================

ieee8021PbEdgePortTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021PbEdgePortEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A Provider Edge Port (PEP) table which indicate the subset of
        parameters needed for each PEP."
    REFERENCE   "12.13.2.3, 12.13.2.4"
    ::= { ieee8021PbObjects 3 }

ieee8021PbEdgePortEntry OBJECT-TYPE
    SYNTAX      Ieee8021PbEdgePortEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An entry in the PEP table indexed by ComponentID and S-VID and
        containing parameters used to configure ingress filtering on
        the PEP, thus affecting traffic transiting from the provider
        network to the customer edge port.  The columns allow the
        default C-VID value and default User Priority to be specified
        and PEP's ingress filtering operation to be controlled.

        Note that the component ID of entries in this table must refer
        to an S-VLAN component of a provider edge bridge and the Bridge
        Port number must refer to the port number of a Customer Edge
        Port associated with that Provider Edge Bridge."
    INDEX { ieee8021BridgeBasePortComponentId,
            ieee8021BridgeBasePort,
            ieee8021PbEdgePortSVid }
    ::= { ieee8021PbEdgePortTable 1 }

Ieee8021PbEdgePortEntry ::= SEQUENCE {
      ieee8021PbEdgePortSVid
          VlanId,
      ieee8021PbEdgePortPVID
          VlanId,
      ieee8021PbEdgePortDefaultUserPriority
          IEEE8021PriorityValue,
      ieee8021PbEdgePortAcceptableFrameTypes
          IEEE8021PortAcceptableFrameTypes,
      ieee8021PbEdgePortEnableIngressFiltering
          TruthValue
}

ieee8021PbEdgePortSVid OBJECT-TYPE
    SYNTAX      VlanId
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The 12 bit S-VID associated with the PEP."
    ::= { ieee8021PbEdgePortEntry 1 }

ieee8021PbEdgePortPVID OBJECT-TYPE
    SYNTAX      VlanId
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "A 12-bit C-VID to be used for untagged frames received at
        the Provider Edge Port."
    ::= { ieee8021PbEdgePortEntry 2 }

ieee8021PbEdgePortDefaultUserPriority OBJECT-TYPE
    SYNTAX      IEEE8021PriorityValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "An integer range 0-7 to be used for untagged frames received
        at the Provider Edge Port."
    ::= { ieee8021PbEdgePortEntry 3 }

ieee8021PbEdgePortAcceptableFrameTypes OBJECT-TYPE
    SYNTAX      IEEE8021PortAcceptableFrameTypes
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "the Acceptable Frame Types for frames received at the PEP.
        The permissible values for the parameter are:
            1) Admit all frames;
            2) Admit only Untagged and Priority-Tagged frames;
            3) Admit only VLAN-Tagged frames."
    DEFVAL { admitAll }
    ::= { ieee8021PbEdgePortEntry 4 }

ieee8021PbEdgePortEnableIngressFiltering OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "Filtering parameter for frames received at the PEP.
        The permissible values for the parameter are:
            true(1)  Enabled;
            false(2) Disabled."
    DEFVAL { true }
    ::= { ieee8021PbEdgePortEntry 5 }

-- ============================================================
-- ieee8021PbServicePriorityRegenerationTable:
-- ============================================================

ieee8021PbServicePriorityRegenerationTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021PbServicePriorityRegenerationEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The priority regeneration table for this PEP."
    REFERENCE   "12.13.2.5, 12.13.2.6"
    ::= { ieee8021PbObjects 4 }

ieee8021PbServicePriorityRegenerationEntry OBJECT-TYPE
    SYNTAX      Ieee8021PbServicePriorityRegenerationEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An element of the PEP priority regeneration table indexed
        by Component ID, bridge port number, S-VID, and received
        priority.  Each element contains the regenerated priority.

        Note that the component ID of entries in this table must refer
        to the S-VLAN component of a Provider Edge Bridge and the Port
        Number must refer to the port number of the Customer Edge Port
        associated with that S-VLAN component."
    INDEX { ieee8021BridgeBasePortComponentId,
            ieee8021BridgeBasePort,
            ieee8021PbServicePriorityRegenerationSVid,
            ieee8021PbServicePriorityRegenerationReceivedPriority }
    ::= { ieee8021PbServicePriorityRegenerationTable 1 }

Ieee8021PbServicePriorityRegenerationEntry ::= SEQUENCE {
      ieee8021PbServicePriorityRegenerationSVid
          VlanId,
      ieee8021PbServicePriorityRegenerationReceivedPriority
          IEEE8021PriorityValue,
      ieee8021PbServicePriorityRegenerationRegeneratedPriority
          IEEE8021PriorityValue
}

ieee8021PbServicePriorityRegenerationSVid OBJECT-TYPE
    SYNTAX      VlanId
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "S-VID for this regeneration table entry."
    ::= { ieee8021PbServicePriorityRegenerationEntry 1 }

ieee8021PbServicePriorityRegenerationReceivedPriority OBJECT-TYPE
    SYNTAX      IEEE8021PriorityValue
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Received priority for this regeneration table entry."
    ::= { ieee8021PbServicePriorityRegenerationEntry 2 }

ieee8021PbServicePriorityRegenerationRegeneratedPriority OBJECT-TYPE
    SYNTAX      IEEE8021PriorityValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The regenerated priority contained in this regeneration table
        entry."
    ::= { ieee8021PbServicePriorityRegenerationEntry 3 }

-- =============================================================
-- ieee8021PbCnpTable
-- =============================================================

ieee8021PbCnpTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021PbCnpEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This table is used for dynamic creation and deletion of
        Customer Network Ports on S-VLAN components or I-components.
        Creation of an entry in this table will implicitly also
        create a corresponding entry in the ieee8021BridgeBasePortTable.

        Entries in this table must be persistent across reinitializations
        of the management system."
    REFERENCE   "12.3.3"
    ::= { ieee8021PbObjects 5 }

ieee8021PbCnpEntry OBJECT-TYPE
    SYNTAX      Ieee8021PbCnpEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Represents a dynamically created Customer Network Port."
    INDEX { ieee8021BridgeBasePortComponentId,
            ieee8021BridgeBasePort }
    ::= { ieee8021PbCnpTable 1 }

Ieee8021PbCnpEntry ::= SEQUENCE {
    ieee8021PbCnpCComponentId
        IEEE8021PbbComponentIdentifierOrZero,
    ieee8021PbCnpSVid
        VlanIdOrNone,
    ieee8021PbCnpRowStatus
        RowStatus
}

ieee8021PbCnpCComponentId OBJECT-TYPE
    SYNTAX      IEEE8021PbbComponentIdentifierOrZero
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The component ID of the C-Vlan component if this is an
        internal customer network port.  The value must be 0 for
        an external customer network port.

        This value must be consistent with the value of the
        corresponding instance of ieee8021PbCnpSVid.
        Both must be non-zero, or both must be zero."
    ::= { ieee8021PbCnpEntry 1 }

ieee8021PbCnpSVid OBJECT-TYPE
    SYNTAX      VlanIdOrNone
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The S-VID for service for an internal customer network port.
        For an external port, this value must be 0.

        This value must be consistent with the value of the
        corresponding instance of ieee8021PbCnpCComponentId.
        Both must be non-zero, or both must be zero."
    ::= { ieee8021PbCnpEntry 2 }

ieee8021PbCnpRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This object is used for creation/deletion of entries in
        this table.

        All columns in this table must have valid values before
        this object can be set to active(1).

        While the value of this object is active(1), the values
        of other columns in the same entry may not be modified."
    ::= { ieee8021PbCnpEntry 3 }
```

```
-- ===========================================================
-- ieee8021PbPnpTable
-- ===========================================================

ieee8021PbPnpTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021PbPnpEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This table is used for dynamic creation and deletion of
        Provider Network Ports on S-VLAN components or B-components.
        Creation of an entry in this table will implicitly also
        create a corresponding entry in the ieee8021BridgeBasePortTable.

        Entries in this table must be persistent across reinitializations
        of the management system."
    REFERENCE   "12.13.2"
    ::= { ieee8021PbObjects 6 }

ieee8021PbPnpEntry OBJECT-TYPE
    SYNTAX      Ieee8021PbPnpEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Represents a dynamically created Provider Network Port."
    INDEX { ieee8021BridgeBasePortComponentId,
            ieee8021BridgeBasePort }
    ::= { ieee8021PbPnpTable 1 }

Ieee8021PbPnpEntry ::= SEQUENCE {
    ieee8021PbPnpRowStatus
        RowStatus
}

ieee8021PbPnpRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This object is used for creation/deletion of entries in
        this table."
    ::= { ieee8021PbPnpEntry 1 }

-- ===========================================================
-- ieee8021PbCepTable
-- ===========================================================

ieee8021PbCepTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021PbCepEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "This table is used to create Customer Edge Ports, CEPs, on a
        provider edge bridge.  It is indexed by the ComponentId of the
        PEB's S-VLAN component and by the port number for the CEP.  Note that
        the CEP's port number belongs to the set of port numbers
        associated with the PEB's S-VLAN component.

        Entries in this table must be persistent across reinitializations
        of the management system.  However, note that some column values,
        as noted below, may change across system reinitializations."
    ::= { ieee8021PbObjects 7 }
```

```
ieee8021PbCepEntry OBJECT-TYPE
    SYNTAX      Ieee8021PbCepEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The entry of the ieee8021PbCepTable.  Note that the component
         index must refer to the S-VLAN component of a PEB, and that the port
         number for the CEP is allocated from the port number space of
         that S-VLAN component."
    INDEX { ieee8021BridgeBasePortComponentId,
            ieee8021BridgeBasePort }
    ::= { ieee8021PbCepTable 1 }

Ieee8021PbCepEntry ::=
    SEQUENCE {
        ieee8021PbCepCComponentId  IEEE8021PbbComponentIdentifierOrZero,
        ieee8021PbCepCepPortNumber IEEE8021BridgePortNumberOrZero,
        ieee8021PbCepRowStatus     RowStatus
    }

ieee8021PbCepCComponentId OBJECT-TYPE
    SYNTAX      IEEE8021PbbComponentIdentifierOrZero
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This column is an implmentation specific column that may be
         used to map the C component associated with this CEP to other
         tables within the system, such as the Entity MIB.  This
         column may not be created or modified by management station
         action.  A value of 0 is always legal, and non-zero values
         will be interpreted in an implementation specific manner.
         The value of this column may or may not persist across system
         restarts."
    ::= { ieee8021PbCepEntry 1 }

ieee8021PbCepCepPortNumber  OBJECT-TYPE
    SYNTAX      IEEE8021BridgePortNumberOrZero
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This column is an implementation specific column that may be
         used to map the CEP to other tables within the system, for
         example the Entity MIB.  This column may not be created or
         modified by management station action.  A value of 0 is
         always legal, and non-zero values will be interpreted in an
         implementation specific manner.  The value of this column
         may or may not persist across system restarts."
    ::= { ieee8021PbCepEntry 2 }

ieee8021PbCepRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This indicates the status of the entry, and is used to create
        and delete entries in this table."
    ::= { ieee8021PbCepEntry 3 }

-- ===========================================================
-- ieee8021PbRcapTable
-- ===========================================================

ieee8021PbRcapTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021PbRcapEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "This table is used to create Remote Customer Access Ports, on a
        provider edge bridge.  It is indexed by the ComponentId of the
        PEB's S-VLAN component and by the port number for the RCAP.  Note that
        the index port number belongs to the set of port numbers
        associated with the PEB's primary S-VLAN component.

        Entries in this table must be persistent across reinitializations
        of the management agent.  However, note that some column values,
        as noted below, may change across system reinitializations."
    ::= { ieee8021PbObjects 8 }

ieee8021PbRcapEntry OBJECT-TYPE
    SYNTAX      Ieee8021PbRcapEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The entry of the ieee8021PbRcapTable.  Note that the component
         index must refer to the primary S-VLAN component of a PEB, and that
         the port number index for the RCAP is allocated from the port
         number space of that S-VLAN component."
    INDEX { ieee8021BridgeBasePortComponentId,
            ieee8021BridgeBasePort }
    ::= { ieee8021PbRcapTable 1 }

Ieee8021PbRcapEntry ::=
    SEQUENCE {
        ieee8021PbRcapSComponentId   IEEE8021PbbComponentIdentifierOrZero,
        ieee8021PbRcapRcapPortNumber IEEE8021BridgePortNumberOrZero,
        ieee8021PbRcapRowStatus      RowStatus
    }

ieee8021PbRcapSComponentId OBJECT-TYPE
    SYNTAX      IEEE8021PbbComponentIdentifierOrZero
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This column is an implmentation specific column that may be
         used to map the Port-mapping S-VLAN component associated with
         this RCAP to other tables within the system, such as the
         Entity MIB.  This column may not be created or modified
         by management station action.  A value of 0 is always legal,
         and non-zero values will be interpreted in an implementation
         specific manner. The value of this column may or may not
         persist across system restarts."
    ::= { ieee8021PbRcapEntry 1 }

ieee8021PbRcapRcapPortNumber  OBJECT-TYPE
    SYNTAX      IEEE8021BridgePortNumberOrZero
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This column is an implementation specific column that may be
         used to map the RCAP to other tables within the system, for
         example the Entity MIB.  This column may not be created or
         modified by management station action.  A value of 0 is
         always legal, and non-zero values will be interpreted in an
         implementation specific manner.  The value of this column
         may or may not persist across system restarts."
    ::= { ieee8021PbRcapEntry 2 }

ieee8021PbRcapRowStatus OBJECT-TYPE
```

```
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This indicates the status of the entry, and is used to create
        and delete entries in this table."
    ::= { ieee8021PbRcapEntry 3 }

-- ============================================================
-- ieee8021PbInternalInterfaceTable:
-- ============================================================

ieee8021PbInternalInterfaceTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021PbIiEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table used in a Port-mapping S-VLAN component associated with
        a Remote Customer Access Port to manage the mapping between
        external S-VIDs and internal interfaces/S-VIDs.

        Note that the component ID of entries in this table must refer
        to the primary S-VLAN component of a Provider Edge Bridge and
        the Port Number must refer to the port number of a Remote
        Customer Access Port associated with that S-VLAN component.

        Entries in this table must be persistent over power up
        restart/reboot."
    REFERENCE   "12.13.4.1, 12.13.4.2"
    ::= { ieee8021PbObjects 9 }

ieee8021PbIiEntry OBJECT-TYPE
    SYNTAX      Ieee8021PbIiEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An element of the internal interface table accessed by PB
        S-VLAN component ID, Remnote Customer Access Port bridge port
        number, and external S-VID.  Each element contains the mapping
        between an external S-VID and the internal port it selects and,
        except in the case of a C-tagged service interface the
        internal S-VID which carries the service ."
    INDEX { ieee8021BridgeBasePortComponentId,
            ieee8021BridgeBasePort,
            ieee8021PbIiExternalSVid }
    ::= { ieee8021PbInternalInterfaceTable 1 }

Ieee8021PbIiEntry ::= SEQUENCE {
      ieee8021PbIiExternalSVid       VlanId,
      ieee8021PbIiInternalPortNumber IEEE8021BridgePortNumberOrZero,
      ieee8021PbIiInternalPortType   IEEE8021BridgePortType,
      ieee8021PbIiInternalSVid       VlanIdOrNone,
      ieee8021PbIiRowStatus          RowStatus
}

ieee8021PbIiExternalSVid OBJECT-TYPE
    SYNTAX      VlanId
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "External S-VID for this internal interface table entry."
    ::= { ieee8021PbIiEntry 1 }

ieee8021PbIiInternalPortNumber OBJECT-TYPE
```

```
    SYNTAX      IEEE8021BridgePortNumberOrZero
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The port number for the internal port on the primary
        S-VLAN component.  This port number is used in FDB entries
        that reference an RCSI.

        The port number of the Remote Customer Access Port can
        be used to identify a PNP on the primary S-VLAN component
        connected to a PNP on the Port-mapping S-VLAN component."
    ::= { ieee8021PbIiEntry 2 }

ieee8021PbIiInternalPortType OBJECT-TYPE
    SYNTAX      IEEE8021BridgePortType
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The port type of the internal port on the primary
        S-VLAN component. This indicates the type of RCSI as follows:

             providerNetworkPort(3) - Indicates a PNP (not an RCSI)
             customerNetworkPort(4) - Indicates a Port-based RCSI
             customerEdgePort(5) - Indicates a C-tagged RCSI

        Other port type values are not valid for this field."
    ::= { ieee8021PbIiEntry 3 }

ieee8021PbIiInternalSVid OBJECT-TYPE
    SYNTAX      VlanIdOrNone
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Internal S-VID for this external S-VID entry."
    ::= { ieee8021PbIiEntry 4 }

ieee8021PbIiRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This indicates the status of an entry in this table, and is
        used to create/delete entries.

        The value of ieee8021PbIiExternalSVid must be set before
        an entry in this table can be made valid.deprecated

        It is an implementation specific decision as to whether any
        column in this table may be set while the corresponding
        instance of this object is valid(1)."
    ::= { ieee8021PbIiEntry 5 }

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

ieee8021PbGroups
    OBJECT IDENTIFIER ::= { ieee8021PbConformance 1 }
ieee8021PbCompliances
    OBJECT IDENTIFIER ::= { ieee8021PbConformance 2 }

-- =============================================================
-- Units of conformance
-- =============================================================
ieee8021PbVidTranslationGroup OBJECT-GROUP
    OBJECTS {
        ieee8021PbVidTranslationRelayVid,
        ieee8021PbVidTranslationRowStatus
    }
    STATUS      deprecated
    DESCRIPTION
        "The collection of objects used to represent a PB
        C-VID/S-VID translation."
    ::= { ieee8021PbGroups 1 }

ieee8021PbCVidRegistrationGroup OBJECT-GROUP
    OBJECTS {
        ieee8021PbCVidRegistrationSVid,
        ieee8021PbCVidRegistrationUntaggedPep,
        ieee8021PbCVidRegistrationUntaggedCep,
        ieee8021PbCVidRegistrationRowStatus
    }
    STATUS      current
    DESCRIPTION
        "The collection of objects used to represent a CEP translation."
    ::= { ieee8021PbGroups 2 }

ieee8021PbEdgePortGroup OBJECT-GROUP
    OBJECTS {
        ieee8021PbEdgePortPVID,
        ieee8021PbEdgePortDefaultUserPriority,
        ieee8021PbEdgePortAcceptableFrameTypes,
        ieee8021PbEdgePortEnableIngressFiltering
    }
    STATUS      current
    DESCRIPTION
        "The collection of objects user to represent a PEP."
    ::= { ieee8021PbGroups 3 }

ieee8021PbServicePriorityRegenerationGroup OBJECT-GROUP
    OBJECTS {
        ieee8021PbServicePriorityRegenerationRegeneratedPriority
    }
    STATUS      current
    DESCRIPTION
        "A regenerated priority value for a PEP."
    ::= { ieee8021PbGroups 4 }

ieee8021PbDynamicCnpGroup OBJECT-GROUP
    OBJECTS {
        ieee8021PbCnpCComponentId,
        ieee8021PbCnpSVid,
        ieee8021PbCnpRowStatus
    }
    STATUS      current
    DESCRIPTION
        "A set of objects used for dynamic creation and deletion
        of customer network ports."
    ::= { ieee8021PbGroups 5 }

ieee8021PbDynamicPnpGroup OBJECT-GROUP
    OBJECTS {
        ieee8021PbPnpRowStatus
    }
    STATUS      current
    DESCRIPTION
        "A set of objects used for dynamic creation and deletion
        of provider network ports."
    ::= { ieee8021PbGroups 6 }

ieee8021PbDynamicCepGroup OBJECT-GROUP
    OBJECTS {
        ieee8021PbCepCComponentId,
        ieee8021PbCepCepPortNumber,
        ieee8021PbCepRowStatus
    }
    STATUS      current
    DESCRIPTION
        "A set of objects used for dynamic creation and deletion
        of customer edge ports."
    ::= { ieee8021PbGroups 7 }

ieee8021PbDynamicRcapGroup OBJECT-GROUP
    OBJECTS {
        ieee8021PbRcapSComponentId,
        ieee8021PbRcapRcapPortNumber,
        ieee8021PbCepRowStatus
    }
    STATUS      current
    DESCRIPTION
        "A set of objects used for dynamic creation and deletion
        of remote customer access ports."
    ::= { ieee8021PbGroups 8 }

ieee8021PbInternalInterfaceGroup OBJECT-GROUP
    OBJECTS {
        ieee8021PbIiInternalPortNumber,
        ieee8021PbIiInternalPortType,
        ieee8021PbIiInternalSVid,
        ieee8021PbIiRowStatus
    }
    STATUS      current
    DESCRIPTION
        "A set of objects used for dynamic creation and deletion
        of internal interfaces on a Port-mapping S-VLAN component."
    ::= { ieee8021PbGroups 9 }


-- ================================================================
-- Compliance statements
-- ================================================================

ieee8021PbCompliance MODULE-COMPLIANCE
    STATUS      current
    DESCRIPTION
        "The compliance statement for device support of Provider
         Bridge services."

    MODULE
        MANDATORY-GROUPS {
            ieee8021PbVidTranslationGroup,
            ieee8021PbCVidRegistrationGroup,
            ieee8021PbEdgePortGroup,
            ieee8021PbServicePriorityRegenerationGroup
        }

        GROUP       ieee8021PbDynamicCnpGroup
        DESCRIPTION
            "This group is optional and supports dynamic creation
            and deletion of customer network ports."
        GROUP       ieee8021PbDynamicPnpGroup
        DESCRIPTION
            "This group is optional and supports dynamic creation
            and deletion of provider network ports."

        GROUP       ieee8021PbDynamicCepGroup
        DESCRIPTION
            "This group is optional and supports dynamic creation
            and deletion of customer edge ports."

        GROUP       ieee8021PbDynamicRcapGroup
        DESCRIPTION
            "This group is optional and supports dynamic creation
            and deletion of remote customer access ports."

        GROUP       ieee8021PbInternalInterfaceGroup
        DESCRIPTION
            "This group is optional and supports dynamic creation
            and deletion of internal interfaces on Port-mapping
            S-VLAN components."

    ::= { ieee8021PbCompliances 1 }

END
```

17.7.6 Definitions for the IEEE8021-MSTP MIB module

Delete the entire text of 17.7.6, and insert the following text:

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

IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE, Integer32, Counter64,
    Unsigned32, TimeTicks
        FROM SNMPv2-SMI
    TruthValue, RowStatus
        FROM SNMPv2-TC
    ieee802dot1mibs, IEEE8021PbbComponentIdentifier,
    IEEE8021BridgePortNumber, IEEE8021VlanIndex,
    IEEE8021MstIdentifier
        FROM IEEE8021-TC-MIB
    BridgeId
        FROM BRIDGE-MIB
    SnmpAdminString
        FROM SNMP-FRAMEWORK-MIB
    MODULE-COMPLIANCE, OBJECT-GROUP
        FROM SNMPv2-CONF;

ieee8021MstpMib MODULE-IDENTITY
    LAST-UPDATED "201112120000Z" -- December 12, 2011
    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 modules for managing devices that support
         IEEE 802.1Q multiple spanning tree groups.

         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     "201112120000Z" -- December 12, 2011
    DESCRIPTION
         "Deprecated ieee8021MstpFidToMstiTable for an identical
          ieee8021MstpFidToMstiV2Table to add 4095 to the range
          of ieee8021MstpFidToMstiV2Fid  and to add 0 and 4095 to
          the range of ieee8021MstpFidToMstiV2MstId for 802.1aq.
          Deprecated ieee8021MstpVlanTable for an identical
          ieee8021MstpVlanV2Table to add 0 & 4095 to the range
          of ieee8021MstpVlanV2MstId for 802.1aq"

    REVISION     "201103230000Z" -- March 23, 2011
    DESCRIPTION
         "Minor edits to contact information, correction to range of
          ieee8021MstpCistMaxHops and addition of fragile bridge
          as part of 2011 revision of IEEE Std 802.1Q."

    REVISION     "200810150000Z" -- October 15, 2008
    DESCRIPTION
        "Initial version."
    ::= { ieee802dot1mibs 6 }

ieee8021MstpNotifications  OBJECT IDENTIFIER ::= { ieee8021MstpMib 0 }
ieee8021MstpObjects        OBJECT IDENTIFIER ::= { ieee8021MstpMib 1 }
ieee8021MstpConformance    OBJECT IDENTIFIER ::= { ieee8021MstpMib 2 }

-- =============================================================
-- MSTP CIST Table
-- =============================================================

ieee8021MstpCistTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021MstpCistEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The Common and Internal Spanning Tree (CIST) Table. Each row in
         the table represents information regarding a Bridge's Bridge
         Protocol Entity for the CIST.

         Note that entries will exist in this table only for bridge
         components for which the corresponding instance of
         ieee8021SpanningTreeVersion (from the IEEE8021-SPANNING-TREE-MIB)
         has a value of mstp(2).

         This table contains objects corresponding to the following items
         from 12.8.1.1 and 12.8.1.3 of IEEE 802.1Q-2005, and the
         802.1ah amendement.  Some of those items are provided in the
         IEEE8021-SPANNING-TREE-MIB as noted below.

         From 12.8.1.1:
             Items a), c), o), p), and q) are defined in this table
             The remaining items are covered in the
             IEEE8021-SPANNING-TREE-MIB:
                 b) ieee8021SpanningTreeTimeSinceTopologyChange
                 c) ieee8021SpanningTreeTopChanges
                 e) ieee8021SpanningTreeDesignatedRoot
                 f) ieee8021SpanningTreeRootCost
                 g) ieee8021SpanningTreeRootPort
                 h) ieee8021SpanningTreeMaxAge
                 i) ieee8021SpanningTreeForwardDelay
                 j) ieee8021SpanningTreeBridgeMaxAge
                 k) ieee8021SpanningTreeBridgeHelloTime
                 l) ieee8021SpanningTreeBridgeForwardDelay
                 m) ieee8021SpanningTreeHoldTime
                 n) ieee8021SpanningTreeVersion
         From 12.8.1.3:
             Item g) is defined in this table
             The remaining items are covered in the
             IEEE8021-SPANNING-TREE-MIB:
                 a) ieee8021SpanningTreeBridgeMaxAge
                 b) ieee8021SpanningTreeBridgeHelloTime
                 c) ieee8021SpanningTreeBridgeForwardDelay
                 d) ieee8021SpanningTreePriority
                 e) ieee8021SpanningTreeVersion
                 f) ieee8021RstpStpExtTxHoldCount"
    REFERENCE   "12.8.1.1, 12.8.1.3"
    ::= { ieee8021MstpObjects 1 }

ieee8021MstpCistEntry OBJECT-TYPE
    SYNTAX      Ieee8021MstpCistEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A CIST Table entry."
    INDEX { ieee8021MstpCistComponentId }
    ::= { ieee8021MstpCistTable 1 }

Ieee8021MstpCistEntry ::= SEQUENCE {
    ieee8021MstpCistComponentId             IEEE8021PbbComponentIdentifier,
    ieee8021MstpCistBridgeIdentifier        BridgeId,
    ieee8021MstpCistTopologyChange          TruthValue,
    ieee8021MstpCistRegionalRootIdentifier  BridgeId,
    ieee8021MstpCistPathCost                Unsigned32,
    ieee8021MstpCistMaxHops                 Integer32
}

ieee8021MstpCistComponentId 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."
    ::= { ieee8021MstpCistEntry 1 }

ieee8021MstpCistBridgeIdentifier OBJECT-TYPE
    SYNTAX      BridgeId
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The Bridge Identifier for the CIST."
    REFERENCE   "9.2.5 of IEEE Std 802.1D-2004"
    ::= { ieee8021MstpCistEntry 2 }
```

```
ieee8021MstpCistTopologyChange OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "In an STP Bridge, the value of the Topology Change parameter
        (14.8.1.1.3, item d of IEEE Std 802.1D, 2004 Edition), or in
        an RST or MST Bridge, asserted if the tcWhile timer for any
        Port for the CIST is non-zero."
    REFERENCE   "14.8.1.1.3:d of IEEE 802.1D-2004"
    ::= { ieee8021MstpCistEntry 3 }

ieee8021MstpCistRegionalRootIdentifier OBJECT-TYPE
    SYNTAX      BridgeId
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "In an MST Bridge, the CIST Regional Root Identifier parameter,
         i.e. the Bridge Identifier of the current CIST Regional Root."
    REFERENCE   "13.16.4"
    ::= { ieee8021MstpCistEntry 4 }

ieee8021MstpCistPathCost OBJECT-TYPE
    SYNTAX      Unsigned32 (0..2147483647)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "In an MST Bridge, the CIST Path Cost parameter, i.e. the CIST
         path cost from the transmitting Bridge to the CIST Regional Root.
         The sum (about 20 possible out of the given range) of multiple
         port path costs. Also, if the 'transmitting Bridge' is
         the 'CIST Regional Root', then this value could be zero."
    ::= { ieee8021MstpCistEntry 5 }

ieee8021MstpCistMaxHops OBJECT-TYPE
    SYNTAX      Integer32 (6..40)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "In an MST Bridge, the MaxHops parameter.

        The value of this object MUST be retained across
        reinitializations of the management system."
    REFERENCE   "13.22.1"
    ::= { ieee8021MstpCistEntry 6 }

-- ================================================================
-- ieee8021MstpTable:
-- ================================================================

ieee8021MstpTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021MstpEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "In an MST Bridge, the MSTP Table. Each row in the Table
         represents information regarding a Bridge's Bridge Protocol
         Entity for the specified Spanning Tree instance.

         Entries in this table MUST be retained across
         reinitializations of the management system.

         Note that entries can be created in this table only for bridge
         components for which the corresponding instance of
         ieee8021SpanningTreeVersion (from the IEEE8021-SPANNING-TREE-MIB)
         has a value of mstp(2)."
    REFERENCE   "12.8.1.2, 12.8.1.4, 12.12.3.2, 12.12.1"
    ::= { ieee8021MstpObjects 2 }

ieee8021MstpEntry OBJECT-TYPE
    SYNTAX      Ieee8021MstpEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A MST Table entry."
    INDEX { ieee8021MstpComponentId, ieee8021MstpId }
    ::= { ieee8021MstpTable 1 }

Ieee8021MstpEntry ::= SEQUENCE {
      ieee8021MstpComponentId              IEEE8021PbbComponentIdentifier,
      ieee8021MstpId                       IEEE8021MstIdentifier,
      ieee8021MstpBridgeId                 BridgeId,
      ieee8021MstpTimeSinceTopologyChange  TimeTicks,
      ieee8021MstpTopologyChanges          Counter64,
      ieee8021MstpTopologyChange           TruthValue,
      ieee8021MstpDesignatedRoot           BridgeId,
      ieee8021MstpRootPathCost             Integer32,
      ieee8021MstpRootPort                 IEEE8021BridgePortNumber,
      ieee8021MstpBridgePriority           Integer32,
      ieee8021MstpVids0                    OCTET STRING,
      ieee8021MstpVids1                    OCTET STRING,
      ieee8021MstpVids2                    OCTET STRING,
      ieee8021MstpVids3                    OCTET STRING,
      ieee8021MstpRowStatus                RowStatus
}

ieee8021MstpComponentId 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."
    ::= { ieee8021MstpEntry 1 }

ieee8021MstpId OBJECT-TYPE
    SYNTAX      IEEE8021MstIdentifier
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "In an MST Bridge, this parameter is the MSTID, i.e. the
         identifier of a Spanning Tree (or MST) Instance."
    ::= { ieee8021MstpEntry 2 }

ieee8021MstpBridgeId OBJECT-TYPE
    SYNTAX      BridgeId
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "In an MST Bridge, the Bridge Identifier for the MSTI."
    REFERENCE   "13.23.2"
    ::= { ieee8021MstpEntry 3 }

ieee8021MstpTimeSinceTopologyChange OBJECT-TYPE
    SYNTAX      TimeTicks
    UNITS       "centi-seconds"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "In an MST Bridge, count in seconds of the time elapsed since
         tcWhile (13.21) was last non-zero for any Port for the MSTI."
    REFERENCE   "13.21"
    ::= { ieee8021MstpEntry 4 }

ieee8021MstpTopologyChanges OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "topology changes"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "In an MST Bridge, count of the times tcWhile (13.21) has been
         non-zero for any Port for the MSTI since the Bridge was powered
         on or initialized."
    REFERENCE   "13.21"
    ::= { ieee8021MstpEntry 5 }

ieee8021MstpTopologyChange OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "In an MST Bridge, the Topology Change parameter value: true(1)
         if tcWhile is non-zero for any Port for the MSTI."
    REFERENCE   "13.21"
    ::= { ieee8021MstpEntry 6 }

ieee8021MstpDesignatedRoot OBJECT-TYPE
    SYNTAX      BridgeId
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "In an MST Bridge, the Designated Root parameter value, i.e. the
         Bridge Identifier of the Root Bridge for the MSTI."
    REFERENCE   "13.24.2"
    ::= { ieee8021MstpEntry 7 }

ieee8021MstpRootPathCost OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "In an MST Bridge, the Root Path Cost parameter value, i.e. the
         path cost from the transmitting Bridge to the Root Bridge for
         the MSTI."
    REFERENCE   "13.24.2"
    ::= { ieee8021MstpEntry 8 }

ieee8021MstpRootPort OBJECT-TYPE
    SYNTAX      IEEE8021BridgePortNumber
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "In an MST Bridge, the Root Port parameter value, i.e. the Root
         Port for the MSTI."
    REFERENCE   "13.23.5"
    ::= { ieee8021MstpEntry 9 }

ieee8021MstpBridgePriority OBJECT-TYPE
    SYNTAX      Integer32 (0..61440)
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "In an MST Bridge, the Bridge Priority parameter value for the
         MSTI, i.e. the four most significant bits of the Bridge Identifier
         for the MSTI."
    REFERENCE   "13.23.2"
    ::= { ieee8021MstpEntry 10 }

ieee8021MstpVids0 OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE(128))
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This object contains the first 1024 bits of the 4096 bit vector
         indicating which VIDs are assigned to this MSTID.  The high order
         bit of the first octet corresponds to the first bit of the vector,
         while the low order bit of the last octet corresponds to the last
         bit of this portion of the vector.  A bit that is on (equal to 1)
         indicates that the corresponding VID is assigned to this MSTID."
    ::= { ieee8021MstpEntry 11 }

ieee8021MstpVids1 OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE(128))
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This object contains the second 1024 bits of the 4096 bit vector
         indicating which VIDs are assigned to this MSTID.  The high order
         bit of the first octet corresponds to the first bit of this
         portion of the vector, while the low order bit of the last octet
         corresponds to the last bit of this portion of the vector.  A bit
         that is on (equal to 1) indicates that the corresponding VID is
         assigned to this MSTID."
    ::= { ieee8021MstpEntry 12 }

ieee8021MstpVids2 OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE(128))
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This object contains the third 1024 bits of the 4096 bit vector
         indicating which VIDs are assigned to this MSTID.  The high order
         bit of the first octet corresponds to the first bit of this
         portion of the vector, while the low order bit of the last octet
         corresponds to the last bit of this portion of the vector.  A bit
         that is on (equal to 1) indicates that the corresponding VID is
         assigned to this MSTID."
    ::= { ieee8021MstpEntry 13 }

ieee8021MstpVids3 OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE(128))
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This object contains the fourth 1024 bits of the 4096 bit vector
         indicating which VIDs are assigned to this MSTID.  The high order
         bit of the first octet corresponds to the first bit of this
         portion of the vector, while the low order bit of the last octet
         corresponds to the last bit of this portion of the vector.  A bit
         that is on (equal to 1) indicates that the corresponding VID is
         assigned to this MSTID."
    ::= { ieee8021MstpEntry 14 }

ieee8021MstpRowStatus OBJECT-TYPE
```

```
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The status of the row.

         Read SNMPv2-TC (RFC2579) for an
         explanation of the possible values this object can take.

         The writable columns in a row can not be changed if the row
         is active. All columns must have a valid value before a row
         can be activated."
    ::= { ieee8021MstpEntry 15 }

-- =============================================================
-- ieee8021MstpCistPortTable:
-- =============================================================

ieee8021MstpCistPortTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021MstpCistPortEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The CIST Port Table. Each row in the Table represents information
         regarding a specific Port within the Bridge's Bridge Protocol
         Entity, for the CIST.

         The values of all writable objects in this table MUST be
         retained across reinitializations of the management system.

         Note that entries will exist in this table only for bridge
         components for which the corresponding instance of
         ieee8021SpanningTreeVersion (from the IEEE8021-SPANNING-TREE-MIB)
         has a value of mstp(2).

         This table contains objects corresponding to the following items
         from 12.8.2.1, 12.8.2.3, and 12.8.1.5 of IEEE 802.1Q-2005, and the
         802.1ah amendement.  Some of those items are provided in the
         IEEE8021-SPANNING-TREE-MIB as noted below.

         From 12.8.2.1:
             Items a), d), e), and i) through w) are defined in this table
             The remaining items are covered in the
             IEEE8021-SPANNING-TREE-MIB:
                 b) ieee8021SpanningTreePortState
                 c) ieee8021SpanningTreePortPriority
                 d) ieee8021SpanningTreePortPathCost32,
                 f) ieee8021SpanningTreePortDesignatedCost
                 g) ieee8021SpanningTreePortDesignatedBridge
                 h) ieee8021SpanningTreePortDesignatedPort
         From 12.8.2.3:
             Items a), b), and d) through h) are defined in this table
                 (item a is the index)
             The remaining items are covered in the
             IEEE8021-SPANNING-TREE-MIB:
                 b) ieee8021SpanningTreePortPathCost,
                 c) ieee8021SpanningTreePortPriority
         From 12.8.2.5:
             All items are defined in this table
         From 802.1ah 12.8.2.1:
             Items u), v), w), and x) are defined in this table
         From 802.1ah 12.8.2.3:
             Items i), j), k), and l) are defined in this table"
    REFERENCE   "12.8.2.1, 12.8.2.3, 12.8.2.5"
    ::= { ieee8021MstpObjects 3 }

ieee8021MstpCistPortEntry OBJECT-TYPE
    SYNTAX      Ieee8021MstpCistPortEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A CIST Port Table entry."
    INDEX { ieee8021MstpCistPortComponentId, ieee8021MstpCistPortNum }
    ::= { ieee8021MstpCistPortTable 1 }

Ieee8021MstpCistPortEntry ::= SEQUENCE {
    ieee8021MstpCistPortComponentId          IEEE8021PbbComponentIdentifier,
    ieee8021MstpCistPortNum                  IEEE8021BridgePortNumber,
    ieee8021MstpCistPortUptime               TimeTicks,
    ieee8021MstpCistPortAdminPathCost        Integer32,
    ieee8021MstpCistPortDesignatedRoot       BridgeId,
    ieee8021MstpCistPortTopologyChangeAck    TruthValue,
    ieee8021MstpCistPortHelloTime            Integer32,
    ieee8021MstpCistPortAdminEdgePort        TruthValue,
    ieee8021MstpCistPortOperEdgePort         TruthValue,
    ieee8021MstpCistPortMacEnabled           TruthValue,
    ieee8021MstpCistPortMacOperational       TruthValue,
    ieee8021MstpCistPortRestrictedRole       TruthValue,
    ieee8021MstpCistPortRestrictedTcn        TruthValue,
    ieee8021MstpCistPortRole                 INTEGER,
    ieee8021MstpCistPortDisputed             TruthValue,
    ieee8021MstpCistPortCistRegionalRootId   BridgeId,
    ieee8021MstpCistPortCistPathCost         Unsigned32,
    ieee8021MstpCistPortProtocolMigration    TruthValue,
    ieee8021MstpCistPortEnableBPDURx         TruthValue,
    ieee8021MstpCistPortEnableBPDUTx         TruthValue,
    ieee8021MstpCistPortPseudoRootId         BridgeId,
    ieee8021MstpCistPortIsL2Gp               TruthValue
}

ieee8021MstpCistPortComponentId 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."
    ::= { ieee8021MstpCistPortEntry 1 }

ieee8021MstpCistPortNum OBJECT-TYPE
    SYNTAX      IEEE8021BridgePortNumber
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The Port's Port Number parameter value for the CIST, i.e. the
         number of the Bridge Port for the CIST."
    ::= { ieee8021MstpCistPortEntry 2 }

ieee8021MstpCistPortUptime OBJECT-TYPE
    SYNTAX      TimeTicks
    UNITS       "centi-seconds"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The Port's Uptime parameter value for the CIST, i.e. the count
         in seconds of the time elapsed since the Port was last reset or
         initialized (BEGIN, 13.23)."
    ::= { ieee8021MstpCistPortEntry 3 }

ieee8021MstpCistPortAdminPathCost OBJECT-TYPE
    SYNTAX      Integer32 (0..200000000)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The administratively assigned value for the contribution
         of this port to the path cost of paths toward the spanning
         tree root.

         Writing a value of '0' assigns the automatically calculated
         default Path Cost value to the port.  If the default Path
         Cost is being used, this object returns '0' when read.

         This complements the object ieee8021MstpCistPortPathCost,
         which returns the operational value of the path cost.

         The value of this object MUST be retained across
         reinitializations of the management system."
    REFERENCE   "13.22:p, 17.13.11 of IEEE Std 802.1D"
    ::= { ieee8021MstpCistPortEntry 4 }

ieee8021MstpCistPortDesignatedRoot OBJECT-TYPE
    SYNTAX      BridgeId
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The CIST Regional Root Identifier component of the Port's port
         priority vector, as defined in 13.10, for the CIST."
    REFERENCE   "13.24.12"
    ::= { ieee8021MstpCistPortEntry 5 }

ieee8021MstpCistPortTopologyChangeAck OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The Port's Topology Change Acknowledge parameter value.
         True(1) if a Configuration Message with a topology change
         acknowledge flag set is to be transmitted. "
    REFERENCE   "17.19.41 of IEEE Std 802.1D"
    ::= { ieee8021MstpCistPortEntry 6 }

ieee8021MstpCistPortHelloTime OBJECT-TYPE
    SYNTAX      Integer32 (100..1000)
    UNITS       "centi-seconds"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The Port's Hello Time timer parameter value, for the CIST.
         In centi-seconds"
    REFERENCE   "13.24.13, 17.19.22 of IEEE Std 802.1D"
    ::= { ieee8021MstpCistPortEntry 7 }

ieee8021MstpCistPortAdminEdgePort OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "In a Bridge that supports the identification of edge ports, the
         Port's Admin Edge Port parameter value, for the CIST."
    REFERENCE   "17.13.1 of IEEE Std 802.1D"
    DEFVAL      { true }
    ::= { ieee8021MstpCistPortEntry 8 }

ieee8021MstpCistPortOperEdgePort OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "In a Bridge that supports the identification of edge ports, the
         Port's operational Edge Port parameter value, for the CIST.
         True(1) if it is an Oper Edge Port."
    REFERENCE   "17.19.17 of IEEE Std 802.1D"
    ::= { ieee8021MstpCistPortEntry 9 }

ieee8021MstpCistPortMacEnabled OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "In a Bridge that supports the MAC Enabled parameter, the current
         state of the MAC Enabled parameter.
         True(1) indicates that adminstratively the MAC is set as if it
         was connected to a point-to-point LAN."
    REFERENCE   "12.8.2.1.3 m)"
    ::= { ieee8021MstpCistPortEntry 10 }

ieee8021MstpCistPortMacOperational OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "In a Bridge that supports the MAC Operational parameter, the
         current state of the MAC Operational parameter.
         True(1) indicates the MAC is operational."
    REFERENCE   "12.8.2.1.3 n)"
    ::= { ieee8021MstpCistPortEntry 11 }

ieee8021MstpCistPortRestrictedRole OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The current state of the restrictedRole parameter for the Port.
         True(1) causes the Port not to be selected as Root Port for the
         CIST or any MSTI. "
    REFERENCE   "13.25.14"
    ::= { ieee8021MstpCistPortEntry 12 }

ieee8021MstpCistPortRestrictedTcn OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The current state of the restrictedTcn parameter for the Port.
         True(1) causes the Port not to propagate topology changes to
         other Ports."
    REFERENCE   "13.25.15"
    ::= { ieee8021MstpCistPortEntry 13 }

ieee8021MstpCistPortRole OBJECT-TYPE
    SYNTAX      INTEGER {
                    root(1),
                    alternate(2),
                    designated(3),
                    backup(4)
                }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The current Port Role for the Port (i.e., Root, Alternate,
         Designated, or Backup), for the CIST."
    REFERENCE   "12.2.8.1.3 s)"
    ::= { ieee8021MstpCistPortEntry 14 }

ieee8021MstpCistPortDisputed OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The current value of the disputed variable for the CIST for
         the Port.  A value of true(1) indicates that the disputed
         variable is set.  A value of false(2) indicates that the
         agreed variable is cleared."
    REFERENCE   "13.24:u, and 17.19.6 of IEEE Std 802.1D"
    ::= { ieee8021MstpCistPortEntry 15 }

ieee8021MstpCistPortCistRegionalRootId OBJECT-TYPE
    SYNTAX      BridgeId
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "In an MST Bridge, the CIST Regional Root Identifier, i.e. the
         Bridge Identifier of the current CIST Regional Root, for the CIST."
    REFERENCE   "13.9:c, 13.10, 13.24.12"
    ::= { ieee8021MstpCistPortEntry 16 }

ieee8021MstpCistPortCistPathCost OBJECT-TYPE
    SYNTAX      Unsigned32 (0..2147483647)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "In an MST Bridge, the CIST Internal Root Path Cost, i.e. the
         CIST path cost from the transmitting Bridge to the CIST Regional
         Root, for the CIST.

         The sum (about 20 possible out of the given range) of multiple
         port path costs. Also, if the 'the transmitting Bridge' is
         'the CIST Regional Root', then this value could be zero."
    REFERENCE   "13.9:d, 13.10, 13.24.12"
    ::= { ieee8021MstpCistPortEntry 17 }

ieee8021MstpCistPortProtocolMigration OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "In an MST Bridge, the current value of the mcheck variable for
         the Port.  A value of true(1) forces the state machine to
         perform functions as per 17.19.13."
    REFERENCE   "17.19.13 of IEEE Std 802.1D"
    ::= { ieee8021MstpCistPortEntry 18 }

ieee8021MstpCistPortEnableBPDURx OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "In an MST Bridge, the enableBPDUrx parameter value.  A value
        of false(2) indicates that BPDUs are ignored."
    REFERENCE   "13.25.18"
    DEFVAL { false }
    ::= { ieee8021MstpCistPortEntry 19 }

ieee8021MstpCistPortEnableBPDUTx OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "In an MST Bridge, the enableBPDUtx parameter value.  A value
        of false(2) indicates that BPDUs are not transmitted."
    REFERENCE   "13.25.19"
    DEFVAL { false }
    ::= { ieee8021MstpCistPortEntry 20 }

ieee8021MstpCistPortPseudoRootId OBJECT-TYPE
    SYNTAX      BridgeId
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "In an MST Bridge, the pseudoRootId parameter value."
    REFERENCE   "13.25.20"
    ::= { ieee8021MstpCistPortEntry 21 }

ieee8021MstpCistPortIsL2Gp OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "In an MST Bridge, the isL2gp parameter value.  A value of
        true(1) indicates this is an L2GP port."
    REFERENCE   "13.25.21"
    DEFVAL { false }
    ::= { ieee8021MstpCistPortEntry 22 }

-- ============================================================
-- ieee8021MstpPortTable:
-- ============================================================

ieee8021MstpPortTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021MstpPortEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The MSTP Port Table. Each row in the Table represents information
         regarding a specific Port within the Bridge's Bridge Protocol
         Entity, for a given MSTI.

         The values of all writable objects in this table MUST be
         retained across reinitializations of the management system.

         Note that entries will exist in this table only for bridge
         components for which the corresponding instance of
         ieee8021SpanningTreeVersion (from the IEEE8021-SPANNING-TREE-MIB)
         has a value of mstp(2)."
    REFERENCE   "12.8.2.2, 12.8.2.4"
    ::= { ieee8021MstpObjects 4 }

ieee8021MstpPortEntry OBJECT-TYPE
    SYNTAX      Ieee8021MstpPortEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
```

```
        "A MSTP Port Table entry."
    INDEX { ieee8021MstpPortComponentId,
            ieee8021MstpPortMstId,
            ieee8021MstpPortNum }
    ::= { ieee8021MstpPortTable 1 }

Ieee8021MstpPortEntry ::= SEQUENCE {
      ieee8021MstpPortComponentId       IEEE8021PbbComponentIdentifier,
      ieee8021MstpPortMstId             IEEE8021MstIdentifier,
      ieee8021MstpPortNum               IEEE8021BridgePortNumber,
      ieee8021MstpPortUptime            TimeTicks,
      ieee8021MstpPortState             INTEGER,
      ieee8021MstpPortPriority          Integer32,
      ieee8021MstpPortPathCost          Integer32,
      ieee8021MstpPortDesignatedRoot    BridgeId,
      ieee8021MstpPortDesignatedCost    Integer32,
      ieee8021MstpPortDesignatedBridge  BridgeId,
      ieee8021MstpPortDesignatedPort    IEEE8021BridgePortNumber,
      ieee8021MstpPortRole              INTEGER,
      ieee8021MstpPortDisputed          TruthValue
}

ieee8021MstpPortComponentId 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."
    ::= { ieee8021MstpPortEntry 1 }

ieee8021MstpPortMstId OBJECT-TYPE
    SYNTAX      IEEE8021MstIdentifier
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "In an MST Bridge, this parameter is the MSTID, i.e. the
         identifier of a Spanning Tree (or MST) Instance."
    ::= { ieee8021MstpPortEntry 2 }

ieee8021MstpPortNum OBJECT-TYPE
    SYNTAX      IEEE8021BridgePortNumber
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "In an MST Bridge, the Port's Port Number parameter value for
         the MSTI, i.e. the number of the Bridge Port for the MSTI."
    ::= { ieee8021MstpPortEntry 3 }

ieee8021MstpPortUptime OBJECT-TYPE
    SYNTAX      TimeTicks
    UNITS       "centi-seconds"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "In an MST Bridge, the Port's Uptime parameter value for the
         MSTI, i.e. the count in seconds of the time elapsed since the
         Port was last reset or initialized (BEGIN, 13.23)."
    ::= { ieee8021MstpPortEntry 4 }

ieee8021MstpPortState OBJECT-TYPE
    SYNTAX      INTEGER {
```

```
                    disabled(1),
                    listening(2),
                    learning(3),
                    forwarding(4),
                    blocking(5)
                }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "In an MST Bridge, the current state of the Port (i.e., Disabled,
         Listening, Learning, Forwarding, or Blocking), for the MSTI."
    REFERENCE   "13.35, and 17.10 of IEEE Std 802.1D"
    ::= { ieee8021MstpPortEntry 5 }

ieee8021MstpPortPriority OBJECT-TYPE
    SYNTAX      Integer32 (0..240)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "In an MST Bridge, the Port's Port Priority parameter value for
         the MSTI, i.e. the priority field for the Port Identifier for the
         Port for the MSTI."
    REFERENCE   "13.24.12"
    ::= { ieee8021MstpPortEntry 6 }

ieee8021MstpPortPathCost OBJECT-TYPE
    SYNTAX      Integer32 (1..200000000)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "In an MST Bridge, the Port's Port Path Cost parameter value for
         the MSTI."
    REFERENCE   "13.37.1"
    ::= { ieee8021MstpPortEntry 7 }

ieee8021MstpPortDesignatedRoot OBJECT-TYPE
    SYNTAX      BridgeId
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "In an MST Bridge, the Regional Root Identifier component of the
         Port's MSTI port priority vector, as defined in 13.11, for the MSTI."
    REFERENCE   "13.24.12"
    ::= { ieee8021MstpPortEntry 8 }

ieee8021MstpPortDesignatedCost OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "In an MST Bridge, the Internal Root Path Cost component of the
         Port's MSTI port priority vector, as defined in 13.11, for the MSTI."
    REFERENCE   "13.24.12"
    ::= { ieee8021MstpPortEntry 9 }

ieee8021MstpPortDesignatedBridge OBJECT-TYPE
    SYNTAX      BridgeId
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "In an MST Bridge, the Designated Bridge Identifier component of
         the Port's MSTI port priority vector, as defined in 13.11, for
         the MSTI."
    REFERENCE   "13.24.12"
    ::= { ieee8021MstpPortEntry 10 }

ieee8021MstpPortDesignatedPort OBJECT-TYPE
    SYNTAX      IEEE8021BridgePortNumber
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "In an MST Bridge, the Designated Port Identifier component of the
         Port's MSTI port priority vector, as defined in 13.11, for the MSTI."
    REFERENCE   "13.24.12"
    ::= { ieee8021MstpPortEntry 11 }

ieee8021MstpPortRole OBJECT-TYPE
    SYNTAX      INTEGER {
                    root(1),
                    alternate(2),
                    designated(3),
                    backup(4)
                }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "In an MST Bridge, the current Port Role for the Port (i.e., Root,
         Alternate, Designated, or Backup), for the MSTI."
    ::= { ieee8021MstpPortEntry 12 }

ieee8021MstpPortDisputed OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "In an MST Bridge, the current value of the disputed variable for
         the MSTI for the Port."
    REFERENCE   "13.24:u, and 17.19.6 of IEEE Std 802.1D"
    ::= { ieee8021MstpPortEntry 13 }

-- =============================================================
-- ieee8021MstpFidToMstiTable  deprecated
-- see ieee8021MstpFidToMstiV2Table below
-- =============================================================

ieee8021MstpFidToMstiTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021MstpFidToMstiEntry
    MAX-ACCESS  not-accessible
    STATUS      deprecated
    DESCRIPTION
        "In an MST Bridge, the fixed-length FID to MSTID Allocation Table
         entry. Each entry in the Table corresponds to a FID, and the value
         of the entry specifies the MSTID of the spanning tree to which the
         set of VLANs supported by that FID are assigned. A value of zero
         in an entry specifies that the set of VLANs supported by that FID
         are assigned to the CST.

         The values of all writable objects in this table MUST be
         retained across reinitializations of the management system.

         Note that entries will exist in this table only for bridge
         components for which the corresponding instance of
         ieee8021SpanningTreeVersion (from the IEEE8021-SPANNING-TREE-MIB)
         has a value of mstp(2)."
    REFERENCE   "12.12.2"
    ::= { ieee8021MstpObjects 5 }

ieee8021MstpFidToMstiEntry OBJECT-TYPE
```

```
    SYNTAX      Ieee8021MstpFidToMstiEntry
    MAX-ACCESS  not-accessible
    STATUS      deprecated
    DESCRIPTION
        "In an MST Bridge, a FID to MSTID Allocation Table entry."
    INDEX { ieee8021MstpFidToMstiComponentId, ieee8021MstpFidToMstiFid }
    ::= { ieee8021MstpFidToMstiTable 1 }

Ieee8021MstpFidToMstiEntry ::= SEQUENCE {
      ieee8021MstpFidToMstiComponentId  IEEE8021PbbComponentIdentifier,
      ieee8021MstpFidToMstiFid          Unsigned32,
      ieee8021MstpFidToMstiMstId        IEEE8021MstIdentifier
}

ieee8021MstpFidToMstiComponentId OBJECT-TYPE
    SYNTAX      IEEE8021PbbComponentIdentifier
    MAX-ACCESS  not-accessible
    STATUS      deprecated
    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."
    ::= { ieee8021MstpFidToMstiEntry 1 }

ieee8021MstpFidToMstiFid OBJECT-TYPE
    SYNTAX      Unsigned32 (1..4094)
    MAX-ACCESS  not-accessible
    STATUS      deprecated
    DESCRIPTION
        "In an MST Bridge, the FID of the entry in the FID to MSTID
         Allocation Table."
    ::= { ieee8021MstpFidToMstiEntry 2 }

ieee8021MstpFidToMstiMstId OBJECT-TYPE
    SYNTAX      IEEE8021MstIdentifier
    MAX-ACCESS  read-write
    STATUS      deprecated
    DESCRIPTION
        "In an MST Bridge, the MSTID to which the FID (of the entry in
         the FID to MSTID Allocation Table) is to be allocated."
    ::= { ieee8021MstpFidToMstiEntry 3 }

-- ================================================================
-- ieee8021MstpFidToMstiV2Table
-- ================================================================

ieee8021MstpFidToMstiV2Table OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021MstpFidToMstiV2Entry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "In an MST Bridge, the fixed-length FID to MSTID Allocation Table
         entry. Each entry in the Table corresponds to a FID, and the value
         of the entry specifies the MSTID of the spanning tree to which the
         set of VLANs supported by that FID are assigned. A value of zero
         in an entry specifies that the set of VLANs supported by that FID
         are assigned to the CST.

         The values of all writable objects in this table MUST be
         retained across reinitializations of the management system.

         Note that entries will exist in this table only for bridge
         components for which the corresponding instance of
         ieee8021SpanningTreeVersion (from the IEEE8021-SPANNING-TREE-MIB)
         has a value of mstp(2)."
    REFERENCE   "12.12.2"
    ::= { ieee8021MstpObjects 9 }

ieee8021MstpFidToMstiV2Entry OBJECT-TYPE
    SYNTAX      Ieee8021MstpFidToMstiV2Entry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "In an MST Bridge, a FID to MSTID Allocation Table entry."
    INDEX { ieee8021MstpFidToMstiV2ComponentId, ieee8021MstpFidToMstV2Fid }
    ::= { ieee8021MstpFidToMstiV2Table 1 }

Ieee8021MstpFidToMstiV2Entry ::= SEQUENCE {
      ieee8021MstpFidToMstiV2ComponentId  IEEE8021PbbComponentIdentifier,
      ieee8021MstpFidToMstiV2Fid          Unsigned32,
      ieee8021MstpFidToMstiV2MstId        Unsigned32
}

ieee8021MstpFidToMstiV2ComponentId 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."
    ::= { ieee8021MstpFidToMstiV2Entry 1 }

ieee8021MstpFidToMstiV2Fid OBJECT-TYPE
    SYNTAX      Unsigned32 (1..4095)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "In an MST Bridge, the FID of the entry in the FID to MSTID
         Allocation Table."
    ::= { ieee8021MstpFidToMstiV2Entry 2 }

ieee8021MstpFidToMstiV2MstId OBJECT-TYPE
    SYNTAX      Unsigned32 (0..4095)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "In an MST Bridge, the MSTID to which the FID (of the entry in
         the FID to MSTID Allocation Table) is to be allocated.
         In an SPT Bridge, the value 4095 is used to indicate unused
         (non-filtering) FIDs."
    ::= { ieee8021MstpFidToMstiV2Entry 3 }

-- ============================================================
-- ieee8021MstpVlanTable  deprecated
-- see ieee8021MstpVlanV2Table below
-- ============================================================

ieee8021MstpVlanTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021MstpVlanEntry
    MAX-ACCESS  not-accessible
    STATUS      deprecated
    DESCRIPTION
        "In an MST Bridge, the fixed-length (4096 elements), read-only,
         MST Configuration Table. Its elements are derived from other
         configuration information held by the Bridge; specifically, the
         current state of the VID to FID Allocation Table (8.8.7.1,
         12.10.3),and the FID to MSTID Allocation Table (8.9.3, 12.12.2).
         Hence, changes made to either of these Tables can in turn affect
         the contents of the MST Configuration Table, and also affect the
         value of the digest element of the MST Configuration Identifier.

         The values of all writable objects in this table MUST be
         retained across reinitializations of the management system.

         Note that entries will exist in this table only for bridge
         components for which the corresponding instance of
         ieee8021SpanningTreeVersion (from the IEEE8021-SPANNING-TREE-MIB)
         has a value of mstp(2)."
    REFERENCE   "12.12.3.1"
    ::= { ieee8021MstpObjects 6 }

ieee8021MstpVlanEntry OBJECT-TYPE
    SYNTAX      Ieee8021MstpVlanEntry
    MAX-ACCESS  not-accessible
    STATUS      deprecated
    DESCRIPTION
        "In an MST Bridge, a MST Configuration Table entry."
    INDEX { ieee8021MstpVlanComponentId, ieee8021MstpVlanId }
    ::= { ieee8021MstpVlanTable 1 }

Ieee8021MstpVlanEntry ::= SEQUENCE {
      ieee8021MstpVlanComponentId  IEEE8021PbbComponentIdentifier,
      ieee8021MstpVlanId           IEEE8021VlanIndex,
      ieee8021MstpVlanMstId        IEEE8021MstIdentifier
}

ieee8021MstpVlanComponentId OBJECT-TYPE
    SYNTAX      IEEE8021PbbComponentIdentifier
    MAX-ACCESS  not-accessible
    STATUS      deprecated
    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."
    ::= { ieee8021MstpVlanEntry 1 }

ieee8021MstpVlanId OBJECT-TYPE
    SYNTAX      IEEE8021VlanIndex
    MAX-ACCESS  not-accessible
    STATUS      deprecated
    DESCRIPTION
        "In an MST Bridge, the VID of the entry in the MST
         Configuration Table."
    ::= { ieee8021MstpVlanEntry 2 }

ieee8021MstpVlanMstId OBJECT-TYPE
    SYNTAX      IEEE8021MstIdentifier
    MAX-ACCESS  read-only
    STATUS      deprecated
    DESCRIPTION
        "In an MST Bridge, the MSTID value corresponding to the VID
         of the entry in the MST Configuration Table."
    ::= { ieee8021MstpVlanEntry 3 }

-- ===========================================================
-- ieee8021MstpVlanV2Table
-- ===========================================================
ieee8021MstpVlanV2Table OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021MstpVlanV2Entry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "In an MST Bridge, the fixed-length (4096 elements), read-only,
         MST Configuration Table. Its elements are derived from other
         configuration information held by the Bridge; specifically, the
         current state of the VID to FID Allocation Table (8.8.7.1,
         12.10.3),and the FID to MSTID Allocation Table (8.9.3, 12.12.2).
         Hence, changes made to either of these Tables can in turn affect
         the contents of the MST Configuration Table, and also affect the
         value of the digest element of the MST Configuration Identifier.

         The values of all writable objects in this table MUST be
         retained across reinitializations of the management system.

         Note that entries will exist in this table only for bridge
         components for which the corresponding instance of
         ieee8021SpanningTreeVersion (from the IEEE8021-SPANNING-TREE-MIB)
         has a value of mstp(2)."
    REFERENCE   "12.12.3.1"
    ::= { ieee8021MstpObjects 10 }

ieee8021MstpVlanV2Entry OBJECT-TYPE
    SYNTAX      Ieee8021MstpVlanV2Entry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "In an MST Bridge, a MST Configuration Table entry."
    INDEX { ieee8021MstpVlanV2ComponentId, ieee8021MstpVlanV2Id }
    ::= { ieee8021MstpVlanV2Table 1 }

Ieee8021MstpVlanV2Entry ::= SEQUENCE {
      ieee8021MstpVlanV2ComponentId  IEEE8021PbbComponentIdentifier,
      ieee8021MstpVlanV2Id           IEEE8021VlanIndex,
      ieee8021MstpVlanV2MstId        Unsigned32
}

ieee8021MstpVlanV2ComponentId 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."
    ::= { ieee8021MstpVlanV2Entry 1 }

ieee8021MstpVlanV2Id OBJECT-TYPE
    SYNTAX      IEEE8021VlanIndex
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "In an MST Bridge, the VID of the entry in the MST
         Configuration Table."
    ::= { ieee8021MstpVlanV2Entry 2 }

ieee8021MstpVlanV2MstId OBJECT-TYPE
    SYNTAX      Unsigned32 (0..4095)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
```

```
        "In an MST Bridge, the MSTID value corresponding to the VID
         of the entry in the MST Configuration Table.
         In an SPT Bridge, a value of 4095 is used to indicate
         SPVIDs."
    ::= { ieee8021MstpVlanV2Entry 3 }

-- ================================================================
-- MST Configuration Identifier Table
-- ================================================================

ieee8021MstpConfigIdTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021MstpConfigIdEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table containing the MST Configuration Identifier for each
         virtual bridge.  In simple situations where there is only
         a single component, there will only be a single entry in
         this table (i.e., only a single MST Configuration Identifier).

         The values of all writable objects in this table MUST be
         retained across reinitializations of the management system.

         Note that entries will exist in this table only for bridge
         components for which the corresponding instance of
         ieee8021SpanningTreeVersion (from the IEEE8021-SPANNING-TREE-MIB)
         has a value of mstp(2)."
    REFERENCE   "12.12.3.3, 12.12.3.4"
    ::= { ieee8021MstpObjects 7 }

ieee8021MstpConfigIdEntry OBJECT-TYPE
    SYNTAX      Ieee8021MstpConfigIdEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
         "An entry contining the MST Configuration Identifier of a bridge."
    INDEX { ieee8021MstpConfigIdComponentId }
    ::= { ieee8021MstpConfigIdTable 1 }

Ieee8021MstpConfigIdEntry ::= SEQUENCE {
    ieee8021MstpConfigIdComponentId      IEEE8021PbbComponentIdentifier,
    ieee8021MstpConfigIdFormatSelector   Integer32,
    ieee8021MstpConfigurationName        SnmpAdminString,
    ieee8021MstpRevisionLevel            Unsigned32,
    ieee8021MstpConfigurationDigest      OCTET STRING
}

ieee8021MstpConfigIdComponentId 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."
    ::= { ieee8021MstpConfigIdEntry 1 }

ieee8021MstpConfigIdFormatSelector OBJECT-TYPE
    SYNTAX      Integer32 (0..0)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "In an MST Bridge, the Configuration Identifier Format Selector
```

```
         in use by the Bridge, in the MST Configuration Identifier. This
         has a value of 0 to indicate the format specified in IEEE Std 802.1Q."
    REFERENCE   "13.7:1"
    ::= { ieee8021MstpConfigIdEntry 2 }

ieee8021MstpConfigurationName OBJECT-TYPE
    SYNTAX      SnmpAdminString (SIZE(32))
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "In an MST Bridge, the Configuration Name in the MST
         Configuration Identifier."
    REFERENCE   "13.7:2"
    ::= { ieee8021MstpConfigIdEntry 3 }

ieee8021MstpRevisionLevel OBJECT-TYPE
    SYNTAX      Unsigned32 (0..65535)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "In an MST Bridge, the Revision Level in the MST
         Configuration Identifier."
    REFERENCE   "13.7:3"
    ::= { ieee8021MstpConfigIdEntry 4 }

ieee8021MstpConfigurationDigest OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE(16))
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "In an MST Bridge, the Configuration Digest in the MST
         Configuration Identifier."
    REFERENCE   "13.7:4"
    ::= { ieee8021MstpConfigIdEntry 5 }


-- =============================================================
-- Ieee8021MstpCistPortExtensionTable:
-- =============================================================

ieee8021MstpCistPortExtensionTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021MstpCistPortExtensionEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
           "The CIST Port Extensions Table. Each row in the Table represents
information
         regarding a specific Port within the Bridge's Bridge Protocol
         Entity, for the CIST."
    REFERENCE   "12.8.2"
    ::= { ieee8021MstpObjects 8 }

ieee8021MstpCistPortExtensionEntry OBJECT-TYPE
    SYNTAX      Ieee8021MstpCistPortExtensionEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A list of additional objects containing information
         maintained by every port about the CIST
         state for that port."
   AUGMENTS { ieee8021MstpCistPortEntry}
    ::= { ieee8021MstpCistPortExtensionTable 1 }

Ieee8021MstpCistPortExtensionEntry ::=
    SEQUENCE {
        ieee8021MstpCistPortAutoEdgePort
            TruthValue,
        ieee8021MstpCistPortAutoIsolatePort
            TruthValue
    }

ieee8021MstpCistPortAutoEdgePort OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The administrative value of the Auto Edge Port parameter.
         A value of true(1) indicates if the bridge detection state
         machine (BDM, 13.31) is to detect other bridges
         attached to the LAN, and set
         ieee8021SpanningTreeRstpPortOperEdgePort automatically.
         The default value is true(1)

        This is optional and provided only by implementations
        that support the automatic identification of edge ports.

         The value of this object MUST be retained across
         reinitializations of the management system."
    REFERENCE   "12.8.2.1.3 )"
    ::= { ieee8021MstpCistPortExtensionEntry 1 }

ieee8021MstpCistPortAutoIsolatePort OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The operational value of the Isolate Port parameter.

        A value of true(1) indicates a Designated Port will
        transition to discarding if both
        ieee8021SpanningTreeRstpPortAdminEdgePort and
        ieee8021SpanningTreeRstpPortAutoEdgePort are FALSE and
        the other bridge presumed to be attached to the same
        point-to-point LAN does not transmit periodic BPDUs.

        This is optional and provided only by implementations
        that support the automatic identification of fragile
         bridges."
    REFERENCE   "12.8.2.1.3"
    ::= { ieee8021MstpCistPortExtensionEntry 2 }


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

ieee8021MstpGroups
    OBJECT IDENTIFIER ::= { ieee8021MstpConformance 1 }
ieee8021MstpCompliances
    OBJECT IDENTIFIER ::= { ieee8021MstpConformance 2 }

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

ieee8021MstpCistGroup OBJECT-GROUP
    OBJECTS {
        ieee8021MstpCistBridgeIdentifier,
        ieee8021MstpCistTopologyChange,
        ieee8021MstpCistRegionalRootIdentifier,
        ieee8021MstpCistPathCost,
        ieee8021MstpCistMaxHops
    }
    STATUS      current
    DESCRIPTION
        "Objects for the CIST group"
    ::= { ieee8021MstpGroups 1 }

ieee8021MstpGroup OBJECT-GROUP
    OBJECTS {
        ieee8021MstpBridgeId,
        ieee8021MstpTimeSinceTopologyChange,
        ieee8021MstpTopologyChanges,
        ieee8021MstpTopologyChange,
        ieee8021MstpDesignatedRoot,
        ieee8021MstpRootPathCost,
        ieee8021MstpRootPort,
        ieee8021MstpBridgePriority,
        ieee8021MstpVids0,
        ieee8021MstpVids1,
        ieee8021MstpVids2,
        ieee8021MstpVids3,
        ieee8021MstpRowStatus
    }
    STATUS      current
    DESCRIPTION
        "Objects for the MST group"
    ::= { ieee8021MstpGroups 2 }

ieee8021MstpCistPortGroup OBJECT-GROUP
    OBJECTS {
        ieee8021MstpCistPortUptime,
        ieee8021MstpCistPortAdminPathCost,
        ieee8021MstpCistPortDesignatedRoot,
        ieee8021MstpCistPortTopologyChangeAck,
        ieee8021MstpCistPortHelloTime,
        ieee8021MstpCistPortAdminEdgePort,
        ieee8021MstpCistPortOperEdgePort,
        ieee8021MstpCistPortMacEnabled,
        ieee8021MstpCistPortMacOperational,
        ieee8021MstpCistPortRestrictedRole,
        ieee8021MstpCistPortRestrictedTcn,
        ieee8021MstpCistPortRole,
        ieee8021MstpCistPortDisputed,
        ieee8021MstpCistPortCistRegionalRootId,
        ieee8021MstpCistPortCistPathCost,
        ieee8021MstpCistPortProtocolMigration,
        ieee8021MstpCistPortEnableBPDURx,
        ieee8021MstpCistPortEnableBPDUTx,
        ieee8021MstpCistPortPseudoRootId,
        ieee8021MstpCistPortIsL2Gp
    }
    STATUS      current
    DESCRIPTION
        "Objects for the CIST Port group"
    ::= { ieee8021MstpGroups 3 }

ieee8021MstpPortGroup OBJECT-GROUP
    OBJECTS {
        ieee8021MstpPortUptime,
        ieee8021MstpPortState,
        ieee8021MstpPortPriority,
        ieee8021MstpPortPathCost,
        ieee8021MstpPortDesignatedRoot,
        ieee8021MstpPortDesignatedCost,
        ieee8021MstpPortDesignatedBridge,
        ieee8021MstpPortDesignatedPort,
        ieee8021MstpPortRole,
        ieee8021MstpPortDisputed
    }
    STATUS      current
    DESCRIPTION
        "Objects for the MST Port group"
    ::= { ieee8021MstpGroups 4 }

ieee8021MstpFidToMstiGroup OBJECT-GROUP
    OBJECTS {
        ieee8021MstpFidToMstiMstId
    }
    STATUS      deprecated
    DESCRIPTION
        "Objects for the MST FID to MSTID Allocation Table group"
    ::= { ieee8021MstpGroups 5 }

ieee8021MstpVlanGroup OBJECT-GROUP
    OBJECTS {
        ieee8021MstpVlanMstId
    }
    STATUS      deprecated
    DESCRIPTION
        "Objects for the MST Configuration Table group"
    ::= { ieee8021MstpGroups 6 }


ieee8021MstpConfigIdGroup OBJECT-GROUP
    OBJECTS {
        ieee8021MstpConfigIdFormatSelector,
        ieee8021MstpConfigurationName,
        ieee8021MstpRevisionLevel,
        ieee8021MstpConfigurationDigest
    }
    STATUS      current
    DESCRIPTION
        "Objects for the MST Configuration Identifier group"
    ::= { ieee8021MstpGroups 7 }


ieee8021MstpCistPortExtensionGroup OBJECT-GROUP
    OBJECTS {
        ieee8021MstpCistPortAutoEdgePort,
        ieee8021MstpCistPortAutoIsolatePort
    }
    STATUS      current
    DESCRIPTION
        "Objects for the CIST Port Extension group
         for fragile bridges"
    ::= { ieee8021MstpGroups 8 }

ieee8021MstpFidToMstiV2Group OBJECT-GROUP
    OBJECTS {
        ieee8021MstpFidToMstiV2MstId
    }
    STATUS      current
    DESCRIPTION
        "Objects for the MST FID to MSTID Allocation Table group
        for SPB"
    ::= { ieee8021MstpGroups 9 }

ieee8021MstpVlanV2Group OBJECT-GROUP
    OBJECTS {
        ieee8021MstpVlanV2MstId
    }
    STATUS      current
    DESCRIPTION
        "Objects for the MST Configuration Table group for SPB"
    ::= { ieee8021MstpGroups 10 }

-- =============================================================
-- Compliance statements
-- =============================================================

ieee8021MstpCompliance MODULE-COMPLIANCE
    STATUS      current
    DESCRIPTION
        "The compliance statement for devices supporting Multiple
        Spanning Tree as defined in 13 of IEEE Std 802.1Q."

    MODULE
        MANDATORY-GROUPS {
            ieee8021MstpCistGroup,
            ieee8021MstpGroup,
            ieee8021MstpCistPortGroup,
            ieee8021MstpPortGroup,
            ieee8021MstpFidToMstiGroup,
            ieee8021MstpVlanGroup,
            ieee8021MstpConfigIdGroup
        }

    GROUP ieee8021MstpCistPortExtensionGroup
    DESCRIPTION
        "Implementation of this group is optional."

    ::= { ieee8021MstpCompliances 1 }

END
```

Insert the following subclause, 17.7.19, after 17.7.18:

17.7.19 SPB MIB module

```
-- ===================================================================
--  IEEE 802.1 Shortest Path Bridging (SPB) MIB                     --
-- ===================================================================

IEEE8021-SPB-MIB DEFINITIONS ::= BEGIN

IMPORTS
       MODULE-IDENTITY, OBJECT-TYPE, Integer32, Unsigned32
              FROM SNMPv2-SMI
       RowStatus, MacAddress, TruthValue, TEXTUAL-CONVENTION
              FROM SNMPv2-TC
       ieee802dot1mibs, IEEE8021PbbIngressEgress,
       IEEE8021BridgePortNumber
              FROM IEEE8021-TC-MIB
       VlanId, VlanIdOrNone, VlanIdOrAny
              FROM Q-BRIDGE-MIB
       InterfaceIndexOrZero
              FROM IF-MIB
       SnmpAdminString
```

```
             FROM SNMP-FRAMEWORK-MIB
       MODULE-COMPLIANCE, OBJECT-GROUP
             FROM SNMPv2-CONF;

ieee8021SpbMib MODULE-IDENTITY
       LAST-UPDATED "201202030000Z"  -- February 3, 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: Don Fedyk
            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  "802.1 SPB MIB"
       REVISION  "201202030000Z"  -- February 3, 2012
       DESCRIPTION  "802.1 Shortest Path Bridging MIB Initial Version"
       ::= { ieee802dot1mibs 26 }
-- ======================================================================
--  TYPE DEFINITIONS                                                    --
-- ======================================================================

IEEE8021SpbAreaAddress ::= TEXTUAL-CONVENTION
       DISPLAY-HINT "1x:"
       STATUS current
       DESCRIPTION
             "This identifier is the 3 Byte IS-IS Area Address.
             Domain Specific part(DSP)."
       REFERENCE "12.25.1.1.2 a), 12.25.1.2.2 a), 12.25.1.3.2 a), 12.25.1.2.2 a)"
       SYNTAX OCTET STRING (SIZE(3))

IEEE8021SpbEctAlgorithm ::= TEXTUAL-CONVENTION
       DISPLAY-HINT "1x-"
       STATUS current
       DESCRIPTION
             "The 4 byte Equal Cost Multiple Tree Algorithm identifier.
             This identifies the tree computation algorithm and tie breakers."
       REFERENCE "12.3 q)"
       SYNTAX OCTET STRING (SIZE(4))

IEEE8021SpbMode ::= TEXTUAL-CONVENTION
       STATUS current
       DESCRIPTION
             "Auto allocation control for this instance
             of SPB. For SPBV it controls SPVIDs and for SPBM it controls
             SPSourceID."
       REFERENCE "27.10"
       SYNTAX INTEGER { auto(1), manual(2) }

IEEE8021SpbEctMode ::= TEXTUAL-CONVENTION
       STATUS current
       DESCRIPTION
             "The mode of the Base VID assigned for this instance of SPB.
             Modes are assigned in the FID to MSTI Allocation table."
       REFERENCE "12.25.5.1.3 c), 12.25.9.1.3 e)"
       SYNTAX INTEGER { disabled(1), spbm(2), spbv(3) }

IEEE8021SpbDigestConvention ::= TEXTUAL-CONVENTION
```

```
       STATUS current
       DESCRIPTION
              "The mode of the current Agreement Digest. This
              determines the level of loop prevention."
       REFERENCE "28.4.3"
       SYNTAX INTEGER { off(1), loopFreeBoth(2), loopFreeMcastOnly(3) }

IEEE8021SpbLinkMetric ::= TEXTUAL-CONVENTION

       DISPLAY-HINT "d"
       STATUS current
       DESCRIPTION
              "The 24 bit cost of an SPB link. A lower metric
              value means better. Value 16777215 equals Infinity."
       REFERENCE "28.2"
       SYNTAX Integer32(1..16777215)

IEEE8021SpbAdjState ::= TEXTUAL-CONVENTION
       STATUS current
       DESCRIPTION
              "The current state of this SPB adjacency or port.
              The values are up, down, and testing."
       REFERENCE "12.25.6.1.3 d), 12.25.6.2.3 d), 12.25.7.1.3 (e"
       SYNTAX INTEGER { up(1), down(2), testing(3) }

IEEE8021SpbmSPsourceId ::= TEXTUAL-CONVENTION
       DISPLAY-HINT "1x:"
       STATUS current
       DESCRIPTION
               "It is the high order 3 bytes for Group Address DA from this
               bridge.
               Note that only the 20 bits not including the top 4 bits are
               the SPSourceID."
       REFERENCE "27.15"
       SYNTAX OCTET STRING (SIZE(3))

IEEE8021SpbDigest ::= TEXTUAL-CONVENTION
       DISPLAY-HINT "1x"
       STATUS current
       DESCRIPTION
              "The Topology Agreement digest hex string."
       REFERENCE "28.4"
       SYNTAX OCTET STRING (SIZE(32))

IEEE8021SpbMCID ::= TEXTUAL-CONVENTION
       DISPLAY-HINT "1x"
       STATUS current
       DESCRIPTION
              "MST Configuration Identifier digest hex string."
       REFERENCE "13.8"
       SYNTAX OCTET STRING (SIZE(51))

IEEE8021SpbBridgePriority ::= TEXTUAL-CONVENTION
       DISPLAY-HINT "1x"
       STATUS current
       DESCRIPTION
              "The Bridge priority is the top 2 bytes of the Bridge Identifier.
              Lower values represent a better priority."
       REFERENCE "13.26.3"
       SYNTAX OCTET STRING (SIZE(2))

IEEE8021SpbMTID ::= TEXTUAL-CONVENTION
       DISPLAY-HINT "d"
       STATUS current
       DESCRIPTION
             "The IS-IS Multi Topology Identifier."
       REFERENCE "3.23, 3.24"
       SYNTAX Unsigned32

IEEE8021SpbServiceIdentifierOrAny ::= TEXTUAL-CONVENTION
       DISPLAY-HINT "d"
       STATUS current
       DESCRIPTION
             "The service instance identifier is used at the Customer Backbone
              port in SPBM to distinguish a service instance.
              The special value of 0xFFFFFF is used for wildcard.
              This range also includes the default I-SID. "
       REFERENCE "3.23, 3.24"
       SYNTAX Unsigned32 (255..16777215)

-- ===================================================================
--  OBJECT DEFINITIONS                                              --
-- ===================================================================

-- ===================================================================
-- ieee8021SpbObjects:
-- ===================================================================
ieee8021SpbObjects OBJECT IDENTIFIER
       ::= { ieee8021SpbMib 1 }

-- ===================================================================
-- ieee8021SpbSys:
-- ===================================================================
ieee8021SpbSys OBJECT IDENTIFIER
       ::= { ieee8021SpbObjects 1 }

ieee8021SpbSysAreaAddress OBJECT-TYPE
       SYNTAX IEEE8021SpbAreaAddress
       MAX-ACCESS read-write
       STATUS current
       DESCRIPTION
             "The three byte IS-IS Area Address to join. Normally
              SPB will use area 00:00:00 however if SPB is being
              used in conjunction with IPV4/V6 it may operate
              using the IS-IS area address already in use.
              This object is persistent."
       REFERENCE "12.25.1.3.2, 12.25.1.3.3"
       ::= { ieee8021SpbSys 1 }

ieee8021SpbSysId OBJECT-TYPE
       SYNTAX MacAddress
       MAX-ACCESS read-write
       STATUS current
       DESCRIPTION
             "SYS ID used for all SPB instances on this bridge.
              A six byte network wide unique identifier. This is
              defaulted to the Bridge Address initially but may
              be overridden.
              This object is persistent."
       REFERENCE "12.25.1.3.3, 3.21"
       ::= { ieee8021SpbSys 2 }

ieee8021SpbSysControlAddr OBJECT-TYPE
       SYNTAX MacAddress
       MAX-ACCESS read-write
       STATUS current
       DESCRIPTION
             "Group MAC that the ISIS control plane will use. SPB may
```

```
              use a number of different addresses for SPB Hello and
              LSP exchange. Section 27.2, 8.13.1.5 and Table 8-13 covers
              the different choices. The choices are as follows:
              01:80:C2:00:00:14 = All Level 1 Intermediate Systems
              01:80:C2:00:00:15 = All Level 2 Intermediate Systems
              09:00:2B:00:00:05 = All Intermediate Systems.
              01:80:C2:00:00:2E = All Provider Bridge Intermediate Systems.
              01:80:C2:00:00:2F = All Customer Bridge Intermediate Systems.
              This object is persistent."
       REFERENCE "12.25.1.1.2, 8.13.5.1"
       ::= { ieee8021SpbSys 3 }

ieee8021SpbSysName OBJECT-TYPE
       SYNTAX SnmpAdminString (SIZE(0..32))
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
             "Name to be used to refer to this SPB bridge. This is advertised
              in IS-IS and used for management."
       REFERENCE "12.25.1.3.3"
       ::= { ieee8021SpbSys 4 }

ieee8021SpbSysBridgePriority OBJECT-TYPE
       SYNTAX IEEE8021SpbBridgePriority
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
             "This is a 16 bit quantity which ranks this SPB Bridge
              relative to others when breaking ties. This priority
              is the high 16 bits of the Bridge Identifier. Its impact
              depends on the tie breaking algorithm. Recommend
              values 0..15 be assigned to core switches to ensure
              diversity of the ECT Algorithms."
       REFERENCE "12.25.1.3.3, 13.26.3"
       ::= { ieee8021SpbSys 5 }

ieee8021SpbmSysSPSourceId OBJECT-TYPE
       SYNTAX IEEE8021SpbmSPsourceId
       MAX-ACCESS read-write
       STATUS current
       DESCRIPTION
             "The Shortest Path Source Identifier.
              It is the high order 3 bytes for Group Address DA from this
              bridge.
              Note that only the 20 bits not including the top 4 bits are
              the SPSourceID.
              This object is persistent."
       REFERENCE "12.25.1.3.3, 3.17, 27.15"
       ::= { ieee8021SpbSys 6 }

ieee8021SpbvSysMode OBJECT-TYPE
       SYNTAX IEEE8021SpbMode
       MAX-ACCESS read-write
       STATUS current
       DESCRIPTION
             "Indication of supporting SPBV mode
              auto(=1)/manual(=2)
              auto => enable SPBV mode and auto allocate SPVIDs.
              manual => enable SPBV mode and manually assign SPVIDs.
              This object is persistent."
       REFERENCE "12.25.1.3.3, 3.20"
       DEFVAL {auto}
       ::= { ieee8021SpbSys 7 }
```

```
ieee8021SpbmSysMode OBJECT-TYPE
       SYNTAX IEEE8021SpbMode
       MAX-ACCESS read-write
       STATUS current
       DESCRIPTION
              "Indication of supporting SPBM mode
               auto(=1)/manual(=2)
               auto => enable SPBM mode and auto allocate SPsourceID.
               manual => enable SPBM mode and manually assign SPsourceID.
               This object is persistent."
       REFERENCE "12.25.1.3.3, 3.19"
       DEFVAL {auto}
       ::= { ieee8021SpbSys 8 }

ieee8021SpbSysDigestConvention OBJECT-TYPE
       SYNTAX IEEE8021SpbDigestConvention
       MAX-ACCESS read-write
       STATUS current
       DESCRIPTION
              "The Agreement Digest convention setting
               off(=1)/loopFreeBoth(=2)/loopFreeMcastOnly(=3)
               off => disable agreement digest checking in hellos
               loopFreeBoth => block unsafe group and individual
               traffic when digests disagree.
               loopFreeMcastOnly =>block unsafe group traffic when digests
               disagree.
               This object is persistent."
       REFERENCE "12.25.1.3.3, 28.4.3"
       DEFVAL {loopFreeBoth}
       ::= { ieee8021SpbSys 9 }

-- ===============================================================
-- ieee8021SpbMtidStaticTable:
-- ===============================================================
ieee8021SpbMtidStaticTable OBJECT-TYPE
       SYNTAX SEQUENCE OF Ieee8021SpbMtidStaticTableEntry
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
              "A Table of multiple logical topologies - MT."
       REFERENCE "12.25.2"
       ::= { ieee8021SpbObjects 2 }

ieee8021SpbMtidStaticTableEntry OBJECT-TYPE
       SYNTAX Ieee8021SpbMtidStaticTableEntry
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
              "This table configures a MTID to a topology index. A
               topology index uniquely identifies a specific ISIS
               topology enabling multiple instances or multiple
               MTIDs within an instance. "
       REFERENCE "12.25.2"
       INDEX {
                ieee8021SpbMtidStaticEntryMtid,
                ieee8021SpbTopIx
             }
       ::= { ieee8021SpbMtidStaticTable 1 }

Ieee8021SpbMtidStaticTableEntry ::=
       SEQUENCE {
           ieee8021SpbMtidStaticEntryMtid IEEE8021SpbMTID,
           ieee8021SpbMTidStaticEntryMtidOverload TruthValue,
           ieee8021SpbMtidStaticEntryRowStatus RowStatus,
           ieee8021SpbTopIx IEEE8021SpbMTID
       }

ieee8021SpbMtidStaticEntryMtid OBJECT-TYPE
       SYNTAX IEEE8021SpbMTID
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
              "ISIS Multi Topology Identifier MTID
               Each MTID defines logical topology and is used
               to enable multiple SPB instances within one ISIS instance."
       REFERENCE "12.25.1.3.2, 12.25.2.3.3, 28.12"
       ::= { ieee8021SpbMtidStaticTableEntry 1 }

ieee8021SpbMTidStaticEntryMtidOverload OBJECT-TYPE
       SYNTAX TruthValue
       MAX-ACCESS read-create
       STATUS current
       DESCRIPTION
              "When set for this logical topology this bridge can only
               originate or terminate traffic. It cannot transit SPB
               encapsulated traffic. This is the IS-IS overload feature
               specific to an SPB IS-IS MTID logical topology.
               This object is persistent."
       REFERENCE "12.25.2.3.3, 27.8.1"
       DEFVAL {false}
       ::= { ieee8021SpbMtidStaticTableEntry 2 }

ieee8021SpbMtidStaticEntryRowStatus 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. This object is persistent.
               This object is persistent."
       REFERENCE "12.25.2.3.3"
       ::= { ieee8021SpbMtidStaticTableEntry 3 }

ieee8021SpbTopIx OBJECT-TYPE
       SYNTAX IEEE8021SpbMTID
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
              "Unique identifier of this SPB topology
               This is index is allocated for this ISIS/MT instance.
               It is used as an index to most other SPB tables below and to
               select the exact ISIS instance and which MT instance together."
       REFERENCE "12.25.2.3.3"
       ::= { ieee8021SpbMtidStaticTableEntry 4 }

-- ===================================================================
-- ieee8021SpbTopIxDynamicTable:
-- ===================================================================
ieee8021SpbTopIxDynamicTable OBJECT-TYPE
       SYNTAX SEQUENCE OF Ieee8021SpbTopIxDynamicTableEntry
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
              "This table is for SPB dynamic information. The dynamic
               information that is sent in this bridges Hellos."
       REFERENCE "12.25.3"
       ::= { ieee8021SpbObjects 3 }
```

```
ieee8021SpbTopIxDynamicTableEntry OBJECT-TYPE
       SYNTAX Ieee8021SpbTopIxDynamicTableEntry
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
              "This table displays the digest information computed for this
bridge.
               A bridge configures this information in MTID 0 only.  "
       REFERENCE "12.25.3"
       INDEX {
                ieee8021SpbTopIxDynamicEntryTopIx
             }
       ::= { ieee8021SpbTopIxDynamicTable 1 }

Ieee8021SpbTopIxDynamicTableEntry ::=
       SEQUENCE {
           ieee8021SpbTopIxDynamicEntryTopIx IEEE8021SpbMTID,
           ieee8021SpbTopIxDynamicEntryAgreeDigest IEEE8021SpbDigest,
           ieee8021SpbTopIxDynamicEntryMCID IEEE8021SpbMCID,
           ieee8021SpbTopIxDynamicEntryAuxMCID IEEE8021SpbMCID
       }

ieee8021SpbTopIxDynamicEntryTopIx OBJECT-TYPE
       SYNTAX IEEE8021SpbMTID
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
              "ISIS-SPB Topology Index identifier
               Each Topology Index defines logical topology and is used
               to enable multiple SPB instances within several ISIS instances."
       REFERENCE "12.25.3.1.2, 28.12"
       ::= { ieee8021SpbTopIxDynamicTableEntry 1 }

ieee8021SpbTopIxDynamicEntryAgreeDigest OBJECT-TYPE
       SYNTAX IEEE8021SpbDigest
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "The topology agreement digest value. Digest of all
               topology information, as in clause 28.4."
       REFERENCE "12.25.3.1.3, 28.4"
       ::= { ieee8021SpbTopIxDynamicTableEntry 2 }

ieee8021SpbTopIxDynamicEntryMCID OBJECT-TYPE
       SYNTAX IEEE8021SpbMCID
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "The MST Identifier MCID. The MCID is a digest of the
               VID to MSTID configuration table which determines the Base VIDs
               enabled for SPBV and SPBM."
       REFERENCE "12.25.3.1.3, 13.8"
       ::= { ieee8021SpbTopIxDynamicTableEntry 3 }

ieee8021SpbTopIxDynamicEntryAuxMCID OBJECT-TYPE
       SYNTAX IEEE8021SpbMCID
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "The aux MST Identifier for migration.
               Either MCID or AuxMCID must match for adjacency to form."
       REFERENCE "12.25.3.1.3, 28.9"
       ::= { ieee8021SpbTopIxDynamicTableEntry 4 }
```

```
-- ===================================================================
-- ieee8021SpbEctStaticTable:
-- ===================================================================
ieee8021SpbEctStaticTable OBJECT-TYPE
       SYNTAX SEQUENCE OF Ieee8021SpbEctStaticTableEntry
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
             "The Equal Cost Tree (ECT)  static configuration table."
       REFERENCE "12.25.4"
       ::= { ieee8021SpbObjects 4 }

ieee8021SpbEctStaticTableEntry OBJECT-TYPE
       SYNTAX Ieee8021SpbEctStaticTableEntry
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
             "The Equal Cost Tree static configuration Table defines the
              ECT-ALGORITHM for the Base VID and if SPBV is used for the SPVID.
"
       REFERENCE "12.25.4"
       INDEX {
               ieee8021SpbEctStaticEntryTopIx,
               ieee8021SpbEctStaticEntryBaseVid
             }
       ::= { ieee8021SpbEctStaticTable 1 }

Ieee8021SpbEctStaticTableEntry ::=
       SEQUENCE {
           ieee8021SpbEctStaticEntryTopIx IEEE8021SpbMTID,
           ieee8021SpbEctStaticEntryBaseVid VlanIdOrAny,
           ieee8021SpbEctStaticEntryEctAlgorithm IEEE8021SpbEctAlgorithm,
           ieee8021SpbvEctStaticEntrySpvid VlanIdOrNone,
           ieee8021SpbEctStaticEntryRowStatus RowStatus
       }

ieee8021SpbEctStaticEntryTopIx OBJECT-TYPE
       SYNTAX IEEE8021SpbMTID
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
             "The ISIS Topology Index identifier to which this
              instance belongs. Each Topology Index defines logical topology
              and is used to enable multiple SPB instances within several
              ISIS instances."
       REFERENCE "12.25.4.2.2, 12.25.4.2.3, 28.12"
       ::= { ieee8021SpbEctStaticTableEntry 1 }

ieee8021SpbEctStaticEntryBaseVid OBJECT-TYPE
       SYNTAX VlanIdOrAny
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
             "Base VID to use for this ECT-ALGORITHM.
              Traffic B-VID (SPBM) or Management VID (SPBV).
              A Base VID value of 4095 is a wildcard for any Base VID
              assigned to SPB operation."
       REFERENCE "12.25.4.2.3, 3.3"
       ::= { ieee8021SpbEctStaticTableEntry 2 }

ieee8021SpbEctStaticEntryEctAlgorithm OBJECT-TYPE
       SYNTAX IEEE8021SpbEctAlgorithm
       MAX-ACCESS read-create
       STATUS current
       DESCRIPTION
             "This identifies the tie-breaking algorithms used in
              Shortest Path Tree computation.  Values range from
              00-80-c2-01 to 00-80-c2-16 for 802.1 for each
              the 16 ECT behaviors. The default is 00-80-c2-01
              the LowPATHID.
              This object is persistent."
       REFERENCE "12.25.4.1, 12.25.4.2.3, 3.6"
       DEFVAL {"00-80-c2-01"}
       ::= { ieee8021SpbEctStaticTableEntry 3 }

ieee8021SpbvEctStaticEntrySpvid OBJECT-TYPE
       SYNTAX VlanIdOrNone
       MAX-ACCESS read-create
       STATUS current
       DESCRIPTION
             "If SPBV mode this is the VID originating from this bridge.
              This input is ignored if ieee8021SpbvSysMode is auto(1),
              but the output always returns the SPVID in use.
              Otherwise in SPBM this is empty, should be set = 0.
              This object is persistent."
       REFERENCE "12.25.4.2.3, 3.16"
       ::= { ieee8021SpbEctStaticTableEntry 4 }

ieee8021SpbEctStaticEntryRowStatus 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.
              This object is persistent."
       REFERENCE "12.25.4.2.3"
       ::= { ieee8021SpbEctStaticTableEntry 5 }

-- ==================================================================
-- ieee8021SpbEctDynamicTable:
-- ==================================================================
ieee8021SpbEctDynamicTable OBJECT-TYPE
       SYNTAX SEQUENCE OF Ieee8021SpbEctDynamicTableEntry
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
             "A table containing Data about the ECT behavior on this bridge"
       REFERENCE "12.25.5"
       ::= { ieee8021SpbObjects 5 }

ieee8021SpbEctDynamicTableEntry OBJECT-TYPE
       SYNTAX Ieee8021SpbEctDynamicTableEntry
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
             "This table can be used to check that neighbor bridges are
              using the same ECT Algorithm. "
       REFERENCE "12.25.5"
       INDEX {
               ieee8021SpbEctDynamicEntryTopIx,
               ieee8021SpbEctDynamicEntryBaseVid
             }
       ::= { ieee8021SpbEctDynamicTable 1 }

Ieee8021SpbEctDynamicTableEntry ::=
       SEQUENCE {
           ieee8021SpbEctDynamicEntryTopIx IEEE8021SpbMTID,
           ieee8021SpbEctDynamicEntryBaseVid VlanId,
           ieee8021SpbEctDynamicEntryMode IEEE8021SpbEctMode,
           ieee8021SpbEctDynamicEntryLocalUse TruthValue,
           ieee8021SpbEctDynamicEntryRemoteUse TruthValue,
           ieee8021SpbEctDynamicEntryIngressCheckDiscards Unsigned32
       }

ieee8021SpbEctDynamicEntryTopIx OBJECT-TYPE
       SYNTAX IEEE8021SpbMTID
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
             "The ISIS Topology Index identifier to which this
              instance belongs. Each Topology Index defines logical topology
              and is used to enable multiple SPB instances within several
              ISIS instances."
       REFERENCE "12.25.5.1.2, 12.25.5.1.3, 28.12"
       ::= { ieee8021SpbEctDynamicTableEntry 1 }

ieee8021SpbEctDynamicEntryBaseVid OBJECT-TYPE
       SYNTAX VlanId
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
             "The Base VID being queried. Base VID
              define the mode in the VID to MSTID table. "
       REFERENCE "12.25.5.1.2, 12.25.5.1.3, 3.3"
       ::= { ieee8021SpbEctDynamicTableEntry 2 }

ieee8021SpbEctDynamicEntryMode OBJECT-TYPE
       SYNTAX IEEE8021SpbEctMode
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
             "The Operating mode of this Base VID.
              SPBM (=2), SPBV (=3), or disabled or none (1)."
       REFERENCE "12.25.5.1.3, 28.12.4"
       ::= { ieee8021SpbEctDynamicTableEntry 3 }

ieee8021SpbEctDynamicEntryLocalUse OBJECT-TYPE
       SYNTAX TruthValue
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
             "This value indicates the ECT is in use locally
               (True/False) for this Base Vid. ECTs may be defined before
              services are assigned. "
       REFERENCE "12.25.5.1.3, 28.12.4"
       ::= { ieee8021SpbEctDynamicTableEntry 4 }

ieee8021SpbEctDynamicEntryRemoteUse OBJECT-TYPE
       SYNTAX TruthValue
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
             "This value indicates the remote ECT is in use
               (True/False) for this Base Vid. ECTs may be defined before
              services are assigned."
       REFERENCE "12.25.5.1.3, 28.12.4"
       ::= { ieee8021SpbEctDynamicTableEntry 5 }

ieee8021SpbEctDynamicEntryIngressCheckDiscards OBJECT-TYPE
       SYNTAX Unsigned32
       MAX-ACCESS read-only
```

```
       STATUS current
       DESCRIPTION
             "The number of SA check failures on this ECT VID
              This is referred to as the ingress check and this
              counter increments whenever a packet is discarded
              for this VID because it has not come from an
                interface   which   is   on   the   shortest   path   to   its   SA.
"
       REFERENCE "12.25.5.1.3"
       ::= { ieee8021SpbEctDynamicTableEntry 6 }

-- ===================================================================
-- ieee8021SpbAdjStaticTable:
-- ===================================================================
ieee8021SpbAdjStaticTable OBJECT-TYPE
       SYNTAX SEQUENCE OF Ieee8021SpbAdjStaticTableEntry
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
             "A table containing the SPB configuration data for a neighbor"
       REFERENCE "12.25.6"
       ::= { ieee8021SpbObjects 6 }

ieee8021SpbAdjStaticTableEntry OBJECT-TYPE
       SYNTAX Ieee8021SpbAdjStaticTableEntry
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
             "This table can be used to display the interfaces and metrics
              of a neighbor bridge. "
       REFERENCE "12.25.6"
       INDEX {
               ieee8021SpbAdjStaticEntryTopIx,
               ieee8021SpbAdjStaticEntryIfIndex
             }
       ::= { ieee8021SpbAdjStaticTable 1 }

Ieee8021SpbAdjStaticTableEntry ::=
       SEQUENCE {
           ieee8021SpbAdjStaticEntryTopIx IEEE8021SpbMTID,
           ieee8021SpbAdjStaticEntryIfIndex InterfaceIndexOrZero,
           ieee8021SpbAdjStaticEntryMetric IEEE8021SpbLinkMetric,
           ieee8021SpbAdjStaticEntryIfAdminState IEEE8021SpbAdjState,
           ieee8021SpbAdjStaticEntryRowStatus RowStatus
       }

ieee8021SpbAdjStaticEntryTopIx OBJECT-TYPE
       SYNTAX IEEE8021SpbMTID
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
             "The ISIS Topology Index identifier to which this
              instance belongs. Each Topology Index defines logical topology
              and is used to enable multiple SPB instances within several
              ISIS instances."
       REFERENCE "12.25.6.1.2, 12.25.6.1.3, 28.12"
       ::= { ieee8021SpbAdjStaticTableEntry 1 }

ieee8021SpbAdjStaticEntryIfIndex OBJECT-TYPE
       SYNTAX InterfaceIndexOrZero
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
             "The System interface/index which defines this
```

```
               adjacency. A value of 0 is a wildcard for any
               interface on which SPB Operation is supported."
       REFERENCE "12.25.6.1.2, 12.25.6.1.3"
       ::= { ieee8021SpbAdjStaticTableEntry 2 }

ieee8021SpbAdjStaticEntryMetric OBJECT-TYPE
       SYNTAX IEEE8021SpbLinkMetric
       MAX-ACCESS read-create
       STATUS current
       DESCRIPTION
              "The ieee8021Spb metric (incremental cost) to this peer.
               The contribution of this link to total path cost.
               Recommended values are inversely proportional to link speed.
               Range is (1..16777215) where 16777215 (0xFFFFFF) is
               infinity; infinity signifies that the adjacency is
               UP, but is not to be used for traffic.
               This object is persistent."
       REFERENCE "12.25.6.1.2, 12.25.6.1.3, 28.12.7"
       ::= { ieee8021SpbAdjStaticTableEntry 3 }

ieee8021SpbAdjStaticEntryIfAdminState OBJECT-TYPE
       SYNTAX IEEE8021SpbAdjState
       MAX-ACCESS read-create
       STATUS current
       DESCRIPTION
              "The administrative state of this interface/port.
               Up is the default.
               This object is persistent."
       REFERENCE "12.25.6.1.2, 12.25.6.1.3"
       ::= { ieee8021SpbAdjStaticTableEntry 4 }

ieee8021SpbAdjStaticEntryRowStatus 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.
               This object is persistent."
       REFERENCE "12.25.6.1.3"
       ::= { ieee8021SpbAdjStaticTableEntry 5 }

-- ===================================================================
-- ieee8021SpbAdjDynamicTable:
-- ===================================================================
ieee8021SpbAdjDynamicTable OBJECT-TYPE
       SYNTAX SEQUENCE OF Ieee8021SpbAdjDynamicTableEntry
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
              "The SPB neighbor dynamic information table."
       REFERENCE "12.25.7"
       ::= { ieee8021SpbObjects 7 }

ieee8021SpbAdjDynamicTableEntry OBJECT-TYPE
       SYNTAX Ieee8021SpbAdjDynamicTableEntry
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
              "This table is used to determine operational values of digests
               and interfaces of neighbor bridges."
       REFERENCE "12.25.7"
       INDEX {
                ieee8021SpbAdjDynamicEntryTopIx,
              ieee8021SpbAdjDynamicEntryIfIndex,
              ieee8021SpbAdjDynamicEntryPeerSysId
            }
       ::= { ieee8021SpbAdjDynamicTable 1 }

Ieee8021SpbAdjDynamicTableEntry ::=
       SEQUENCE {
           ieee8021SpbAdjDynamicEntryTopIx IEEE8021SpbMTID,
           ieee8021SpbAdjDynamicEntryIfIndex InterfaceIndexOrZero,
           ieee8021SpbAdjDynamicEntryPeerSysId MacAddress,
           ieee8021SpbAdjDynamicEntryPort IEEE8021BridgePortNumber,
           ieee8021SpbAdjDynamicEntryIfOperState IEEE8021SpbAdjState,
           ieee8021SpbAdjDynamicEntryPeerSysName SnmpAdminString,
           ieee8021SpbAdjDynamicEntryPeerAgreeDigest IEEE8021SpbDigest,
           ieee8021SpbAdjDynamicEntryPeerMCID IEEE8021SpbMCID,
           ieee8021SpbAdjDynamicEntryPeerAuxMCID IEEE8021SpbMCID,
           ieee8021SpbAdjDynamicEntryLocalCircuitID Unsigned32,
           ieee8021SpbAdjDynamicEntryPeerLocalCircuitID Unsigned32,
           ieee8021SpbAdjDynamicEntryPortIdentifier Unsigned32,
           ieee8021SpbAdjDynamicEntryPeerPortIdentifier Unsigned32,
           ieee8021SpbAdjDynamicEntryIsisCircIndex Unsigned32
       }

ieee8021SpbAdjDynamicEntryTopIx OBJECT-TYPE
       SYNTAX IEEE8021SpbMTID
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
             "The ISIS Topology Index identifier to which this
              instance belongs. Each Topology Index defines logical topology
              and is used to enable multiple SPB instances within several
              ISIS instances."
       REFERENCE "12.25.7.1.2, 12.25.7.1.3, 28.12"
       ::= { ieee8021SpbAdjDynamicTableEntry 1 }

ieee8021SpbAdjDynamicEntryIfIndex OBJECT-TYPE
       SYNTAX InterfaceIndexOrZero
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
             "System interface/index which defines this adjacency
              A value of 0 is a wildcard for any interface
              on which SPB Operation is enabled."
       REFERENCE "12.25.7.1.2, 12.25.7.1.3"
       ::= { ieee8021SpbAdjDynamicTableEntry 2 }

ieee8021SpbAdjDynamicEntryPeerSysId OBJECT-TYPE
       SYNTAX MacAddress
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
             "The SPB System Identifier of this peer. This is used to
              identify a neighbor uniquely."
       REFERENCE "12.25.7.1.3, 3.21"
       ::= { ieee8021SpbAdjDynamicTableEntry 3 }

ieee8021SpbAdjDynamicEntryPort OBJECT-TYPE
       SYNTAX IEEE8021BridgePortNumber
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
             "The port number to reach this adjacency."
       REFERENCE "12.25.7.1.3"
       ::= { ieee8021SpbAdjDynamicTableEntry 4 }
```

```
ieee8021SpbAdjDynamicEntryIfOperState OBJECT-TYPE
       SYNTAX IEEE8021SpbAdjState
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "The operational state of this port.
               up, down or testing (in test)."
       REFERENCE "12.25.7.1.3"
       ::= { ieee8021SpbAdjDynamicTableEntry 5 }

ieee8021SpbAdjDynamicEntryPeerSysName OBJECT-TYPE
       SYNTAX SnmpAdminString (SIZE(0..32))
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "IS-IS system name of peer.
               This is the ASCII name assigned to the bridge to aid
               management. It is the same as the ieee8021SpbSysName. "
       REFERENCE "12.25.7.1.3"
       ::= { ieee8021SpbAdjDynamicTableEntry 6 }

ieee8021SpbAdjDynamicEntryPeerAgreeDigest OBJECT-TYPE
       SYNTAX IEEE8021SpbDigest
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "The peer topology agreement digest value
               (all of the elements defined in clause 28.4).
               If it does not match this bridge's digest it indicates loss of
               synchronization."
       REFERENCE "12.25.7.1.3, 28.4"
       ::= { ieee8021SpbAdjDynamicTableEntry 7 }

ieee8021SpbAdjDynamicEntryPeerMCID OBJECT-TYPE
       SYNTAX IEEE8021SpbMCID
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "The peer MST Identifier MCID. The MCID is a digest of the
               VID to MSTID configuration table which determines the Base VIDs
               enabled for SPBV and SPBM."
       REFERENCE "12.25.7.1.3, 13.8"
       ::= { ieee8021SpbAdjDynamicTableEntry 8 }

ieee8021SpbAdjDynamicEntryPeerAuxMCID OBJECT-TYPE
       SYNTAX IEEE8021SpbMCID
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "The  peer auxiliary  MST Identifier. This MCID is
               used for migration."
       REFERENCE "12.25.7.1.3, 27.4.1, 28.12.2"
       ::= { ieee8021SpbAdjDynamicTableEntry 9 }

ieee8021SpbAdjDynamicEntryLocalCircuitID OBJECT-TYPE
       SYNTAX Unsigned32
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "The value used by IS-IS to identify this adjacency locally."
       REFERENCE "12.25.7.1.3, 28.11"
       ::= { ieee8021SpbAdjDynamicTableEntry 10 }
```

```
ieee8021SpbAdjDynamicEntryPeerLocalCircuitID OBJECT-TYPE
       SYNTAX Unsigned32
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "The value used by peer IS-IS to identify this adjacency remotely."
       REFERENCE "12.25.7.1.3, 28.11"
       ::= { ieee8021SpbAdjDynamicTableEntry 11 }

ieee8021SpbAdjDynamicEntryPortIdentifier OBJECT-TYPE
       SYNTAX Unsigned32
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "The value for this bridge which has been selected by
               IS-IS to form this adjacency if there is more than 1 candidate
link."
       REFERENCE "12.25.7.1.3, 28.11"
       ::= { ieee8021SpbAdjDynamicTableEntry 12 }

ieee8021SpbAdjDynamicEntryPeerPortIdentifier OBJECT-TYPE
       SYNTAX Unsigned32
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "The value for peer port Identifier selected by IS-IS
               to form this adjacency if there is more than 1 candidate link."
       REFERENCE "12.25.7.1.3, 28.11"
       ::= { ieee8021SpbAdjDynamicTableEntry 13 }

ieee8021SpbAdjDynamicEntryIsisCircIndex OBJECT-TYPE
       SYNTAX Unsigned32
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "The isisCircTable reference. This allows cross referencing
               to an IS-IS MIB."
       REFERENCE "12.25.7.1.3"
       ::= { ieee8021SpbAdjDynamicTableEntry 14 }

-- =================================================================
-- ieee8021SpbTopNodeTable:
-- =================================================================
ieee8021SpbTopNodeTable OBJECT-TYPE
       SYNTAX SEQUENCE OF Ieee8021SpbTopNodeTableEntry
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
              "Table of network specific bridge information."
       REFERENCE "12.25.8"
       ::= { ieee8021SpbObjects 8 }

ieee8021SpbTopNodeTableEntry OBJECT-TYPE
       SYNTAX Ieee8021SpbTopNodeTableEntry
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
              "This table is used to display system level information about
               bridges in the network.   "
       REFERENCE "12.25.8"
       INDEX {
               ieee8021SpbTopNodeEntryTopIx,
               ieee8021SpbTopNodeEntrySysId
             }
       ::= { ieee8021SpbTopNodeTable 1 }

Ieee8021SpbTopNodeTableEntry ::=
       SEQUENCE {
           ieee8021SpbTopNodeEntryTopIx IEEE8021SpbMTID,
           ieee8021SpbTopNodeEntrySysId MacAddress,
           ieee8021SpbTopNodeEntryBridgePriority IEEE8021SpbBridgePriority,
           ieee8021SpbmTopNodeEntrySPsourceID IEEE8021SpbmSPsourceId,
           ieee8021SpbTopNodeEntrySysName SnmpAdminString
       }

ieee8021SpbTopNodeEntryTopIx OBJECT-TYPE
       SYNTAX IEEE8021SpbMTID
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
             "The ISIS Topology Index identifier to which this
              instance belongs. Each Topology Index defines logical topology
              and is used to enable multiple SPB instances within several
              ISIS instances."
       REFERENCE "12.25.8.1.2, 12.25.8.1.3, 28.12"
       ::= { ieee8021SpbTopNodeTableEntry 1 }

ieee8021SpbTopNodeEntrySysId OBJECT-TYPE
       SYNTAX MacAddress
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
             "The IS-IS System ID of a bridge in the SPB
              LSP database and hence the network.
              A value of 0 is a wildcard for all System identifiers."
       REFERENCE "12.25.8.1.2, 12.25.8.1.3, 3.21"
       ::= { ieee8021SpbTopNodeTableEntry 2 }

ieee8021SpbTopNodeEntryBridgePriority OBJECT-TYPE
       SYNTAX IEEE8021SpbBridgePriority
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
             "The Bridge Priority of the bridge in the LSP database.
              This is a 16 bit quantity which ranks this SPB Bridge
              relative to others when breaking ties. This priority
              is the high 16 bits of the Bridge Identifier. Its impact
              depends on the tie breaking algorithm. Recommend
              values 0..15 be assigned to core switches to ensure
              diversity of the ECT Algorithms."
       REFERENCE "12.25.8.1.3, 13.26.3"
       ::= { ieee8021SpbTopNodeTableEntry 3 }

ieee8021SpbmTopNodeEntrySPsourceID OBJECT-TYPE
       SYNTAX IEEE8021SpbmSPsourceId
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
             "The Shortest Path Source Identifier.
              It is the high order 3 bytes for Group Address DA from this
              bridge. Note that only the 20 bits not including the
              top 4 bits are the SPSourceID."
       REFERENCE "12.25.8.1.3, 3.17"
       ::= { ieee8021SpbTopNodeTableEntry 4 }

ieee8021SpbTopNodeEntrySysName OBJECT-TYPE
       SYNTAX SnmpAdminString (SIZE(0..32))
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
             "The System Name. A Human readable name of this bridge
              This is used to aid in management and is used in
              place of the System identifier in many commands and displays."
       REFERENCE "12.25.8.1.3"
       ::= { ieee8021SpbTopNodeTableEntry 5 }

-- ===================================================================
-- ieee8021SpbTopEctTable:
-- ===================================================================
ieee8021SpbTopEctTable OBJECT-TYPE
       SYNTAX SEQUENCE OF Ieee8021SpbTopEctTableEntry
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
             "Table of all ECT use in the network"
       REFERENCE "12.25.9"
       ::= { ieee8021SpbObjects 9 }

ieee8021SpbTopEctTableEntry OBJECT-TYPE
       SYNTAX Ieee8021SpbTopEctTableEntry
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
             "This table lists bridges and the ECT Algorithms configured and in
use. "
       REFERENCE "12.25.9"
       INDEX {
               ieee8021SpbTopEctEntryTopIx,
               ieee8021SpbTopEctEntrySysId,
               ieee8021SpbTopEctEntryBaseVid
             }
       ::= { ieee8021SpbTopEctTable 1 }

Ieee8021SpbTopEctTableEntry ::=
       SEQUENCE {
           ieee8021SpbTopEctEntryTopIx IEEE8021SpbMTID,
           ieee8021SpbTopEctEntrySysId MacAddress,
           ieee8021SpbTopEctEntryBaseVid VlanIdOrAny,
           ieee8021SpbTopEctEntryEctAlgorithm IEEE8021SpbEctAlgorithm,
           ieee8021SpbTopEctEntryMode IEEE8021SpbEctMode,
           ieee8021SpbvTopEctSysMode IEEE8021SpbMode,
           ieee8021SpbvTopEctEntrySpvid VlanIdOrNone,
           ieee8021SpbTopEctEntryLocalUse TruthValue
       }

ieee8021SpbTopEctEntryTopIx OBJECT-TYPE
       SYNTAX IEEE8021SpbMTID
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
             "The ISIS Topology Index identifier to which this
              instance belongs. Each Topology Index defines logical topology
              and is used to enable multiple SPB instances within several
              ISIS instances."
       REFERENCE "12.25.9.1.2, 12.25.9.1.3"
       ::= { ieee8021SpbTopEctTableEntry 1 }

ieee8021SpbTopEctEntrySysId OBJECT-TYPE
       SYNTAX MacAddress
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
```

```
              "The system ID which is using a particular ECT.
               A value of 0 is a wildcard for all System identifiers."
       REFERENCE "12.25.9.1.2, 12.25.9.1.3, 3.21"
       ::= { ieee8021SpbTopEctTableEntry 2 }

ieee8021SpbTopEctEntryBaseVid OBJECT-TYPE
       SYNTAX VlanIdOrAny
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
              "Base VID related to this algorithm
               In the case of SPBM this is the B-VID that carries
               traffic for this ECT-ALGORITHM. In the case of SPBV
               this is the Base-VID used for management.
               A Base VID value of 4095 is a wildcard for any Base VID
               assigned to SPB operation."
       REFERENCE "12.25.9.1.2, 12.25.9.1.3, 3.3"
       ::= { ieee8021SpbTopEctTableEntry 3 }

ieee8021SpbTopEctEntryEctAlgorithm OBJECT-TYPE
       SYNTAX IEEE8021SpbEctAlgorithm
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "The ECT-ALGORITHM in use.
               A 32 bit number. The ISIS Topology Index identifier to which this
               instance belongs. Each Topology Index defines logical topology
               and is used to enable multiple SPB instances within several
               ISIS instances.; the upper 24 bits are an OUI
               and the lower 8 bits are an index. This creates a
               world-wide unique identity for the computation that
               will be using the VID thus ensuring consistency."
       REFERENCE "12.25.9.1.3, 3.6"
       ::= { ieee8021SpbTopEctTableEntry 4 }

ieee8021SpbTopEctEntryMode OBJECT-TYPE
       SYNTAX IEEE8021SpbEctMode
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "Operating mode : SPBM (=2) or SPBV (=3)"
       REFERENCE "12.25.9.1.3"
       ::= { ieee8021SpbTopEctTableEntry 5 }

ieee8021SpbvTopEctSysMode OBJECT-TYPE
       SYNTAX IEEE8021SpbMode
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "Indication of supporting SPBV mode
               auto(=1)/manual(=2)
               auto => SPBV mode and auto allocate SPVIDs.
               manual => SPBV mode and manually assign SPVIDs."
       REFERENCE "12.25.9.1.3, 3.18"
       ::= { ieee8021SpbTopEctTableEntry 6 }

ieee8021SpbvTopEctEntrySpvid OBJECT-TYPE
       SYNTAX VlanIdOrNone
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "SPVID in V mode
               The VID this bridge will use to originate traffic
               using this ECT-ALGORITHM when running in SPBV mode."
       REFERENCE "12.25.9.1.3, 3.14"
       ::= { ieee8021SpbTopEctTableEntry 7 }

ieee8021SpbTopEctEntryLocalUse OBJECT-TYPE
       SYNTAX TruthValue
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "Is this ECT-ALGORITHM in use locally by advertising
               bridge :- TRUE or FALSE. This is used to help with
               disruption-free migration between ECT-ALGORITHMs.
               Changes are only allowed if this flag is FALSE."
       REFERENCE "12.25.9.1.3, 28.12.5"
       ::= { ieee8021SpbTopEctTableEntry 8 }

-- ===============================================================
-- ieee8021SpbTopEdgeTable:
-- ===============================================================
ieee8021SpbTopEdgeTable OBJECT-TYPE
       SYNTAX SEQUENCE OF Ieee8021SpbTopEdgeTableEntry
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
              "A Table of edges in network (not duplicated),
               but each link will appear as two entries, one
               ordered {near-far}, the other {far-near}."
       REFERENCE "12.25.10"
       ::= { ieee8021SpbObjects 10 }

ieee8021SpbTopEdgeTableEntry OBJECT-TYPE
       SYNTAX Ieee8021SpbTopEdgeTableEntry
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
              "The table lists information about bridge edges (links)."
       REFERENCE "12.25.10"
       INDEX {
                ieee8021SpbTopEdgeEntryTopIx,
                ieee8021SpbTopEdgeEntrySysIdNear,
                ieee8021SpbTopEdgeEntrySysIdFar
              }
       ::= { ieee8021SpbTopEdgeTable 1 }

Ieee8021SpbTopEdgeTableEntry ::=
       SEQUENCE {
           ieee8021SpbTopEdgeEntryTopIx IEEE8021SpbMTID,
           ieee8021SpbTopEdgeEntrySysIdNear MacAddress,
           ieee8021SpbTopEdgeEntrySysIdFar MacAddress,
           ieee8021SpbTopEdgeEntryMetricNear2Far IEEE8021SpbLinkMetric,
           ieee8021SpbTopEdgeEntryMetricFar2Near IEEE8021SpbLinkMetric
       }

ieee8021SpbTopEdgeEntryTopIx OBJECT-TYPE
       SYNTAX IEEE8021SpbMTID
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
              "The ISIS Topology Index identifier to which this
               instance belongs. Each Topology Index defines logical topology
               and is used to enable multiple SPB instances within several
               ISIS instances."
       REFERENCE "12.25.10.1.2, 12.25.10.1.3, 28.12"
       ::= { ieee8021SpbTopEdgeTableEntry 1 }
```

```
ieee8021SpbTopEdgeEntrySysIdNear OBJECT-TYPE
       SYNTAX MacAddress
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
              "The System ID of near bridge (the bridge
               reporting the adjacency).
               A value of 0 is a wildcard for all System identifiers."
       REFERENCE "12.25.10.1.2, 12.25.10.1.3, 3.21"
       ::= { ieee8021SpbTopEdgeTableEntry 2 }

ieee8021SpbTopEdgeEntrySysIdFar OBJECT-TYPE
       SYNTAX MacAddress
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
              "The System ID of far bridge (the neighbor
               of the bridge reporting).
               A value of 0 is a wildcard for all System identifiers."
       REFERENCE "12.25.10.1.2, 12.25.10.1.3, 3.21"
       ::= { ieee8021SpbTopEdgeTableEntry 3 }

ieee8021SpbTopEdgeEntryMetricNear2Far OBJECT-TYPE
       SYNTAX IEEE8021SpbLinkMetric
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "The metric used on this edge advertised by near end
               This is the raw value. If it is less than the
               MetricFar2Near (below), the MetricFar2Near is
               used as the SPF metric in both directions."
       REFERENCE "12.25.10.1.3, 28.12.7"
       ::= { ieee8021SpbTopEdgeTableEntry 4 }

ieee8021SpbTopEdgeEntryMetricFar2Near OBJECT-TYPE
       SYNTAX IEEE8021SpbLinkMetric
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "The metric used on this edge advertised by far end
               This is the raw value. If it is less than the
               MetricNear2Far (above), the MetricNear2Far is
               used as the SPF metric in both directions."
       REFERENCE "12.25.10.1.3, 28.12.7"
       ::= { ieee8021SpbTopEdgeTableEntry 5 }

-- =====================================================================
-- ieee8021SpbvTopSrvTable:
-- =====================================================================
ieee8021SpbmTopSrvTable OBJECT-TYPE
       SYNTAX SEQUENCE OF Ieee8021SpbmTopSrvTableEntry
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
              "All SPBM PBB encapsulated services in this network."
       REFERENCE "12.25.11"
       ::= { ieee8021SpbObjects 11 }

ieee8021SpbmTopSrvTableEntry OBJECT-TYPE
       SYNTAX Ieee8021SpbmTopSrvTableEntry
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
              "This table displays information about PBB services received
```

```
               in the LSP data base. The Service Identifier is associated with
               the MAC address and Base VID of the bridge that originates or
               terminates the service. "
       REFERENCE "12.25.11"
       INDEX {
               ieee8021SpbmTopSrvEntryTopIx,
               ieee8021SpbmTopSrvEntrySysId,
               ieee8021SpbmTopSrvEntryIsid,
               ieee8021SpbmTopSrvEntryBaseVid,
               ieee8021SpbmTopSrvEntryMac
             }
       ::= { ieee8021SpbmTopSrvTable 1 }

Ieee8021SpbmTopSrvTableEntry ::=
       SEQUENCE {
           ieee8021SpbmTopSrvEntryTopIx IEEE8021SpbMTID,
           ieee8021SpbmTopSrvEntrySysId MacAddress,
           ieee8021SpbmTopSrvEntryIsid IEEE8021SpbServiceIdentifierOrAny,
           ieee8021SpbmTopSrvEntryBaseVid VlanIdOrAny,
           ieee8021SpbmTopSrvEntryMac MacAddress,
           ieee8021SpbmTopSrvEntryIsidFlags IEEE8021PbbIngressEgress

       }

ieee8021SpbmTopSrvEntryTopIx OBJECT-TYPE
       SYNTAX IEEE8021SpbMTID
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
              "Entry of one The ISIS Topology Index identifier to which this
               instance belongs. Each Topology Index defines logical topology
               and is used to enable multiple SPB instances within several
               ISIS instances."
       REFERENCE "12.25.11.1.2, 12.25.11.1.3, 28.12"
       ::= { ieee8021SpbmTopSrvTableEntry 1 }

ieee8021SpbmTopSrvEntrySysId OBJECT-TYPE
       SYNTAX MacAddress
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
              "The System identifier this service originates/terminates on.
               A value of 0 is a wildcard for all System identifiers."
       REFERENCE "12.25.11.1.2, 12.25.11.1.3, 3.21"
       ::= { ieee8021SpbmTopSrvTableEntry 2 }

ieee8021SpbmTopSrvEntryIsid OBJECT-TYPE
       SYNTAX IEEE8021SpbServiceIdentifierOrAny
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
              "An ISID (service) originating/terminating on this bridge.
               A value of 0 is a wildcard for any ISID."
       REFERENCE "12.25.11.1.2, 12.25.11.1.3, 28.12.10"
       ::= { ieee8021SpbmTopSrvTableEntry 3 }

ieee8021SpbmTopSrvEntryBaseVid OBJECT-TYPE
       SYNTAX VlanIdOrAny
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
              "The Base VID associated with this service. The Base VID determines
               the ECT Algorithm that is associated with this service.
               A Base VID value of 4095 is a wildcard for any Base VID
              assigned to SPB operation."
       REFERENCE "12.25.11.1.2, 12.25.11.1.3, 28.12.10"
       ::= { ieee8021SpbmTopSrvTableEntry 4 }

ieee8021SpbmTopSrvEntryMac OBJECT-TYPE
       SYNTAX MacAddress
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
             "The MAC address associated with a service.
              An additional nodal MAC address by which an I-SID
              can be reached may be advertised, in which case
              traffic to this MAC follows a forwarding path identical
              to that taken to reach the corresponding SYSID (nodal) MAC.
              If no additional MAC is advertised this will be the SYSID MAC.
              A value of 0 is a wildcard for the MAC address."
       REFERENCE "12.25.11.1.3, 28.12.10"
       ::= { ieee8021SpbmTopSrvTableEntry 5 }

ieee8021SpbmTopSrvEntryIsidFlags OBJECT-TYPE
       SYNTAX IEEE8021PbbIngressEgress
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
             "A pair of flags defining the attributes of this
              service.  These specify independently whether
              ingress frames to the SPBM region should be
              transmitted within it, and whether frames
              received from the SPBM region are required
              egress it."
       REFERENCE "12.25.11.1.2, 12.25.11.1.3, 28.12.10"
       ::= { ieee8021SpbmTopSrvTableEntry 6 }


ieee8021SpbvTopSrvTable OBJECT-TYPE
       SYNTAX SEQUENCE OF Ieee8021SpbvTopSrvTableEntry
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
             "The SPBV group services in this network"
       REFERENCE "12.25.12"
       ::= { ieee8021SpbObjects 12 }

ieee8021SpbvTopSrvTableEntry OBJECT-TYPE
       SYNTAX Ieee8021SpbvTopSrvTableEntry
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
             "This table displays information about SPBV group address.
              The group address is a associated with MAC address and Base
              VID of the bridge that originates or terminates the service."
       REFERENCE "12.25.12"
       INDEX {
               ieee8021SpbvTopSrvEntryTopIx,
               ieee8021SpbvTopSrvEntrySysId,
               ieee8021SpbvTopSrvEntryMMac
             }
       ::= { ieee8021SpbvTopSrvTable 1 }

Ieee8021SpbvTopSrvTableEntry ::=
       SEQUENCE {
           ieee8021SpbvTopSrvEntryTopIx IEEE8021SpbMTID,
           ieee8021SpbvTopSrvEntrySysId MacAddress,
           ieee8021SpbvTopSrvEntryMMac MacAddress,
           ieee8021SpbvTopSrvEntryBaseVid VlanId,
           ieee8021SpbvTopSrvEntryMMacFlags IEEE8021PbbIngressEgress
       }

ieee8021SpbvTopSrvEntryTopIx OBJECT-TYPE
       SYNTAX IEEE8021SpbMTID
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
             "The ISIS Topology Index identifier to which this
              instance belongs. Each Topology Index defines logical topology
              and is used to enable multiple SPB instances within several
              ISIS instances."
       REFERENCE "12.25.12.1.2, 12.25.12.1.3, 28.12"
       ::= { ieee8021SpbvTopSrvTableEntry 1 }

ieee8021SpbvTopSrvEntrySysId OBJECT-TYPE
       SYNTAX MacAddress
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
             "The System identifier advertising this group address.
              A value of 0 is a wildcard for all System identifiers."
       REFERENCE "12.25.12.1.2, 12.25.12.1.3, 3.21"
       ::= { ieee8021SpbvTopSrvTableEntry 2 }

ieee8021SpbvTopSrvEntryMMac OBJECT-TYPE
       SYNTAX MacAddress
       MAX-ACCESS not-accessible
       STATUS current
       DESCRIPTION
             "This Group MAC address entry.
              A value of 0 is a wildcard for any Group MAC address. "
       REFERENCE "12.25.12.1.2, 12.25.12.1.3, 28.12.9"
       ::= { ieee8021SpbvTopSrvTableEntry 3 }


ieee8021SpbvTopSrvEntryBaseVid OBJECT-TYPE
       SYNTAX VlanId
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
             "The Base VID associated with this service. The Base VID determines
              the ECT Algorithm that is associated with this service."
       REFERENCE "12.25.12.1.3, 3.3"
       ::= { ieee8021SpbvTopSrvTableEntry 4 }

ieee8021SpbvTopSrvEntryMMacFlags OBJECT-TYPE
       SYNTAX IEEE8021PbbIngressEgress
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
             "A pair of {ingress, egress} flags for this
              Group Address, defining transmit/receive or both. This enables
              filtering of Group addresses to interwork with MMRP."
       REFERENCE "12.25.12.1.3, 28.12.9"
       ::= { ieee8021SpbvTopSrvTableEntry 5 }


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

ieee8021SpbConformance OBJECT IDENTIFIER ::= { ieee8021SpbMib 2}
```

```
ieee8021SpbGroups      OBJECT IDENTIFIER ::= { ieee8021SpbConformance 1}
ieee8021SpbCompliances OBJECT IDENTIFIER ::= { ieee8021SpbConformance 2}

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

ieee8021SpbSysGroupSPBM OBJECT-GROUP
       OBJECTS {
              ieee8021SpbSysAreaAddress,
              ieee8021SpbSysId,
              ieee8021SpbSysControlAddr,
              ieee8021SpbSysName,
              ieee8021SpbSysBridgePriority,
              ieee8021SpbmSysSPSourceId,
              ieee8021SpbmSysMode,
              ieee8021SpbSysDigestConvention
       }

       STATUS current
       DESCRIPTION
       "The collection of objects used to represent ieee8021SpbSys"
       ::= { ieee8021SpbGroups 1 }

ieee8021SpbMtidStaticTableGroupSPBM OBJECT-GROUP
       OBJECTS {
              ieee8021SpbMTidStaticEntryMtidOverload,
              ieee8021SpbMtidStaticEntryRowStatus
       }

       STATUS current
       DESCRIPTION
       "The collection of objects used to represent ieee8021SpbMtidStaticTable"
       ::= { ieee8021SpbGroups 2 }

ieee8021SpbTopIxDynamicTableGroupSPBM OBJECT-GROUP
       OBJECTS {
              ieee8021SpbTopIxDynamicEntryAgreeDigest,
              ieee8021SpbTopIxDynamicEntryMCID,
              ieee8021SpbTopIxDynamicEntryAuxMCID
       }

       STATUS current
       DESCRIPTION
       "The collection of objects used to represent ieee8021SpbTopIxDynamicTable"
       ::= { ieee8021SpbGroups 3 }

ieee8021SpbEctStaticTableGroupSPBM OBJECT-GROUP
       OBJECTS {
              ieee8021SpbEctStaticEntryEctAlgorithm,
              ieee8021SpbEctStaticEntryRowStatus
       }

       STATUS current
       DESCRIPTION
       "The collection of objects used to represent ieee8021SpbEctStaticTable"
       ::= { ieee8021SpbGroups 4 }

ieee8021SpbEctDynamicTableGroupSPBM OBJECT-GROUP
       OBJECTS {
              ieee8021SpbEctDynamicEntryMode,
              ieee8021SpbEctDynamicEntryLocalUse,
              ieee8021SpbEctDynamicEntryRemoteUse,
              ieee8021SpbEctDynamicEntryIngressCheckDiscards
       }

       STATUS current
       DESCRIPTION
       "The collection of objects used to represent ieee8021SpbEctDynamicTable"
       ::= { ieee8021SpbGroups 5 }

ieee8021SpbAdjStaticTableGroupSPBM OBJECT-GROUP
       OBJECTS {
              ieee8021SpbAdjStaticEntryMetric,
              ieee8021SpbAdjStaticEntryIfAdminState,
              ieee8021SpbAdjStaticEntryRowStatus
       }

       STATUS current
       DESCRIPTION
       "The collection of objects used to represent ieee8021SpbAdjStaticTable"
       ::= { ieee8021SpbGroups 6 }

ieee8021SpbAdjDynamicTableGroupSPBM OBJECT-GROUP
       OBJECTS {
              ieee8021SpbAdjDynamicEntryPort,
              ieee8021SpbAdjDynamicEntryIfOperState,
              ieee8021SpbAdjDynamicEntryPeerSysName,
              ieee8021SpbAdjDynamicEntryPeerAgreeDigest,
              ieee8021SpbAdjDynamicEntryPeerMCID,
              ieee8021SpbAdjDynamicEntryPeerAuxMCID,
              ieee8021SpbAdjDynamicEntryLocalCircuitID,
              ieee8021SpbAdjDynamicEntryPeerLocalCircuitID,
              ieee8021SpbAdjDynamicEntryPortIdentifier,
              ieee8021SpbAdjDynamicEntryPeerPortIdentifier,
              ieee8021SpbAdjDynamicEntryIsisCircIndex
       }

       STATUS current
       DESCRIPTION
       "The collection of objects used to represent ieee8021SpbAdjDynamicTable"
       ::= { ieee8021SpbGroups 7 }

ieee8021SpbTopNodeTableGroupSPBM OBJECT-GROUP
       OBJECTS {
              ieee8021SpbTopNodeEntryBridgePriority,
              ieee8021SpbmTopNodeEntrySPsourceID,
              ieee8021SpbTopNodeEntrySysName
       }

       STATUS current
       DESCRIPTION
       "The collection of objects used to represent ieee8021SpbTopNodeTable"
       ::= { ieee8021SpbGroups 8 }

ieee8021SpbTopEctTableGroupSPBM OBJECT-GROUP
       OBJECTS {
              ieee8021SpbTopEctEntryEctAlgorithm,
              ieee8021SpbTopEctEntryMode,
              ieee8021SpbTopEctEntryLocalUse
       }

       STATUS current
       DESCRIPTION
       "The collection of objects used to represent ieee8021SpbTopEctTable"
       ::= { ieee8021SpbGroups 9 }

ieee8021SpbTopEdgeTableGroupSPBM OBJECT-GROUP
```

```
       OBJECTS {
              ieee8021SpbTopEdgeEntryMetricNear2Far,
              ieee8021SpbTopEdgeEntryMetricFar2Near
       }

       STATUS current
       DESCRIPTION
       "The collection of objects used to represent ieee8021SpbTopEdgeTable"
       ::= { ieee8021SpbGroups 10 }

ieee8021SpbmTopSrvTableGroupSPBM OBJECT-GROUP
       OBJECTS {
              ieee8021SpbmTopSrvEntryIsidFlags
       }

       STATUS current
       DESCRIPTION
       "The collection of objects used to represent ieee8021SpbmTopSrvTable"
       ::= { ieee8021SpbGroups 11 }

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

ieee8021SpbSysGroupSPBV OBJECT-GROUP
       OBJECTS {
              ieee8021SpbSysAreaAddress,
              ieee8021SpbSysId,
              ieee8021SpbSysControlAddr,
              ieee8021SpbSysName,
              ieee8021SpbSysBridgePriority,
              ieee8021SpbvSysMode,
              ieee8021SpbSysDigestConvention
       }

       STATUS current
       DESCRIPTION
       "The collection of objects used to represent ieee8021SpbSys"
       ::= { ieee8021SpbGroups 12 }

ieee8021SpbMtidStaticTableGroupSPBV OBJECT-GROUP
       OBJECTS {
              ieee8021SpbMTidStaticEntryMtidOverload,
              ieee8021SpbMtidStaticEntryRowStatus
       }

       STATUS current
       DESCRIPTION
       "The collection of objects used to represent ieee8021SpbMtidStaticTable"
       ::= { ieee8021SpbGroups 13 }

ieee8021SpbTopIxDynamicTableGroupSPBV OBJECT-GROUP
       OBJECTS {
              ieee8021SpbTopIxDynamicEntryAgreeDigest,
              ieee8021SpbTopIxDynamicEntryMCID,
              ieee8021SpbTopIxDynamicEntryAuxMCID
       }

       STATUS current
       DESCRIPTION
       "The collection of objects used to represent ieee8021SpbTopIxDynamicTable"
       ::= { ieee8021SpbGroups 14 }

ieee8021SpbEctStaticTableGroupSPBV OBJECT-GROUP
```

```
       OBJECTS {
              ieee8021SpbEctStaticEntryEctAlgorithm,
              ieee8021SpbvEctStaticEntrySpvid,
              ieee8021SpbEctStaticEntryRowStatus
       }

       STATUS current
       DESCRIPTION
       "The collection of objects used to represent ieee8021SpbEctStaticTable"
       ::= { ieee8021SpbGroups 15 }

ieee8021SpbEctDynamicTableGroupSPBV OBJECT-GROUP
       OBJECTS {
              ieee8021SpbEctDynamicEntryMode,
              ieee8021SpbEctDynamicEntryLocalUse,
              ieee8021SpbEctDynamicEntryRemoteUse,
              ieee8021SpbEctDynamicEntryIngressCheckDiscards
       }

       STATUS current
       DESCRIPTION
       "The collection of objects used to represent ieee8021SpbEctDynamicTable"
       ::= { ieee8021SpbGroups 16 }

ieee8021SpbAdjStaticTableGroupSPBV OBJECT-GROUP
       OBJECTS {
              ieee8021SpbAdjStaticEntryMetric,
              ieee8021SpbAdjStaticEntryIfAdminState,
              ieee8021SpbAdjStaticEntryRowStatus
       }

       STATUS current
       DESCRIPTION
       "The collection of objects used to represent ieee8021SpbAdjStaticTable"
       ::= { ieee8021SpbGroups 17 }

ieee8021SpbAdjDynamicTableGroupSPBV OBJECT-GROUP
       OBJECTS {
              ieee8021SpbAdjDynamicEntryPort,
              ieee8021SpbAdjDynamicEntryIfOperState,
              ieee8021SpbAdjDynamicEntryPeerSysName,
              ieee8021SpbAdjDynamicEntryPeerAgreeDigest,
              ieee8021SpbAdjDynamicEntryPeerMCID,
              ieee8021SpbAdjDynamicEntryPeerAuxMCID,
              ieee8021SpbAdjDynamicEntryLocalCircuitID,
              ieee8021SpbAdjDynamicEntryPeerLocalCircuitID,
              ieee8021SpbAdjDynamicEntryPortIdentifier,
              ieee8021SpbAdjDynamicEntryPeerPortIdentifier,
              ieee8021SpbAdjDynamicEntryIsisCircIndex
       }

       STATUS current
       DESCRIPTION
       "The collection of objects used to represent ieee8021SpbAdjDynamicTable"
       ::= { ieee8021SpbGroups 18 }

ieee8021SpbTopNodeTableGroupSPBV OBJECT-GROUP
       OBJECTS {
              ieee8021SpbTopNodeEntryBridgePriority,
              ieee8021SpbTopNodeEntrySysName
       }

       STATUS current
       DESCRIPTION
```

```
        "The collection of objects used to represent ieee8021SpbTopNodeTable"
        ::= { ieee8021SpbGroups 19 }

ieee8021SpbTopEctTableGroupSPBV OBJECT-GROUP
        OBJECTS {
                ieee8021SpbTopEctEntryEctAlgorithm,
                ieee8021SpbTopEctEntryMode,
                ieee8021SpbvTopEctSysMode,
                ieee8021SpbvTopEctEntrySpvid,
                ieee8021SpbTopEctEntryLocalUse
        }

        STATUS current
        DESCRIPTION
        "The collection of objects used to represent ieee8021SpbTopEctTable"
        ::= { ieee8021SpbGroups 20 }

ieee8021SpbTopEdgeTableGroupSPBV OBJECT-GROUP
        OBJECTS {
                ieee8021SpbTopEdgeEntryMetricNear2Far,
                ieee8021SpbTopEdgeEntryMetricFar2Near
        }

        STATUS current
        DESCRIPTION
        "The collection of objects used to represent ieee8021SpbTopEdgeTable"
        ::= { ieee8021SpbGroups 21 }

ieee8021SpbvTopSrvTableGroupSPBV OBJECT-GROUP
        OBJECTS {
                ieee8021SpbvTopSrvEntryBaseVid,
                ieee8021SpbvTopSrvEntryMMacFlags
        }

        STATUS current
        DESCRIPTION
        "The collection of objects used to represent ieee8021SpbvTopSrvTable"
        ::= { ieee8021SpbGroups 22 }

-- =============================================================
-- Compliance statements SPBM
-- =============================================================

ieee8021SpbComplianceSPBM MODULE-COMPLIANCE
        STATUS current
        DESCRIPTION
                "Compliance to IEEE 802.1 SPBM mode"
        MODULE
                MANDATORY-GROUPS {
                        ieee8021SpbSysGroupSPBM ,
                        ieee8021SpbMtidStaticTableGroupSPBM ,
                        ieee8021SpbTopIxDynamicTableGroupSPBM ,
                        ieee8021SpbEctStaticTableGroupSPBM ,
                        ieee8021SpbEctDynamicTableGroupSPBM ,
                        ieee8021SpbAdjStaticTableGroupSPBM ,
                        ieee8021SpbAdjDynamicTableGroupSPBM ,
                        ieee8021SpbTopNodeTableGroupSPBM ,
                        ieee8021SpbTopEctTableGroupSPBM ,
                        ieee8021SpbTopEdgeTableGroupSPBM ,
                        ieee8021SpbmTopSrvTableGroupSPBM
                }
        ::= { ieee8021SpbCompliances 1 }

-- =============================================================
-- Compliance statements SPBV
-- ================================================================

ieee8021SpbComplianceSPBV MODULE-COMPLIANCE
       STATUS current
       DESCRIPTION
              "Compliance to IEEE 802.1 SPBV mode"
       MODULE
              MANDATORY-GROUPS {
                     ieee8021SpbSysGroupSPBV ,
                     ieee8021SpbMtidStaticTableGroupSPBV ,
                     ieee8021SpbTopIxDynamicTableGroupSPBV ,
                     ieee8021SpbEctStaticTableGroupSPBV ,
                     ieee8021SpbEctDynamicTableGroupSPBV ,
                     ieee8021SpbAdjStaticTableGroupSPBV ,
                     ieee8021SpbAdjDynamicTableGroupSPBV ,
                     ieee8021SpbTopNodeTableGroupSPBV ,
                     ieee8021SpbTopEctTableGroupSPBV ,
                     ieee8021SpbTopEdgeTableGroupSPBV ,
                     ieee8021SpbvTopSrvTableGroupSPBV
              }
       ::= { ieee8021SpbCompliances 2 }

END
```

20. Connectivity Fault Management protocols

20.2.2 Loopback Message reception and Loopback Reply transmission

Change the following list items in 20.2.2 as shown:

c) If the LBM was received at a PBB-TE MHF and contains a PBB-TE MIP TLV then the Vlan_identifier in the LBR is set to the value in the Reverse VID field; otherwise the vlan_identifier in the LBR is set to the value of the Primary VID associated with the replying MP [item d) in 12.14.7.1.3, item a) in 12.14.3.2.2, item c) in 12.14.5.3.2].

d) ~~c)~~ The OpCode field is changed from LBM to LBR.

20.28.2 xmitLBR()

Change the following list item in 20.28.2 as shown:

c) ~~(Only for PBB-TE MAs) The vlan_identifier for the LBR is the Primary VID [item d) in 12.14.7.1.3] associated with the replying PBB-TE MEP; otherwise if the received LBM contains a PBB-TE MIP TLV, the value in the Reverse VID field is used as the vlan_identifier parameter;~~ If the LBM was received at a PBB-TE MHF and contains a PBB-TE MIP TLV then the Vlan_identifier in the LBR is set to the value in the Reverse VID field; otherwise the vlan_identifier in the LBR is set to the value of the Primary VID associated with the replying MP [item d) in 12.14.7.1.3, item a) in 12.14.3.2.2, item c) in 12.14.5.3.2];

Insert the following text, Clause 27, after Clause 26:

27. Shortest Path Bridging (SPB)

Shortest path bridging (SPB) provides shortest path communication for user data frames in Shortest Path Tree (SPT) regions (Clause 3). ISIS-SPB interoperates with MSTP and RSTP (Clause 13) and STP (Clause 8 of IEEE Std 802.1D, 1998 Edition), exchanging BPDUs to determine the extent of each SPT Region and to configure full, simple, and symmetric connectivity throughout a network comprising arbitrarily interconnected bridges and individual point-to-point LANs (13.6, 13.7). IS-IS supports point to point links and shared LANs however only point to point is considered for ISIS-SPB in this specification. A future revision may add shared media. The Clause 13 SPB description allows for both point to point links and shared media.

Within an SPT Region (7.3, 13.6), the active topology that supports the frames for any given Base VID can be chosen (27.4) by the network administrator to be one of the following:

- The internal spanning tree (IST), or
- A multiple spanning tree instance (MSTI), or
- A set of shortest path trees (SPT Set), one SPT rooted at each SPT Bridge and supporting frames transmitted into the region through that bridge.

Clause 28 specifies the use of ISIS-SPB to calculate the IST and multiple SPT Sets (supporting different ECTs) within an SPT Region. Clause 13 specifies the use of MSTP to calculate MSTIs, and state machines that use the results provided by ISIS-SPB for each SPT to prevent loops within and between regions while maintaining or recovering connectivity as the network reconfigures (13.12, 13.14–13.40, 27.5). The umbrella term *ISIS-SPB* refers to the coordinated use of the protocols and algorithms specified in Clause 28 and Clause 13 to support shortest path bridging.

To allow SPB to support plug-and-play operation for some VLANs, while providing the administrative controls and scalability required for large scale operations, this standard specifies two complementary modes of SPT Bridges (SPBV and SPBM) for assigning each shortest path bridged frame to the correct SPT. All the bridges in a given SPT Region agree on the method to be used to support any given VLAN. Subclause 27.18 describes network scenarios that use one or both of these methods.

If a VLAN is supported by an SPT Bridge using SPBV mode, Shortest Path VIDs (SPVIDs) are automatically allocated by ISIS-SPB and used in VLAN tags within an SPT Region. Each SPVID identifies both the VLAN and the SPT supporting transmission of a tagged frame, and Dynamic VLAN Registration Entries are created to provide connectivity while preventing loops (13.38.2). The SPVIDs for a given VLAN use a single FID that is associated with the Base VID for that VLAN (Shared VLAN Learning, 7.5, 8.8.8). VID translation (6.9) at the region boundary (7.3, 13.5, 13.12) maps SPVIDs to and from the associated Base VID.

If a VLAN is supported by an SPT Bridge in SPBM mode, the Base VID is used in each frame's VLAN tag within the SPT Region and identifies the VLAN and the SPT Set. Each bridge mitigates and prevents loops (6.5.4, 13.38.2) by using the source and destination MAC Address of each frame with Dynamic Filtering Entries for the Base VID, MAC Address tuples. The source MAC Address identifies the particular SPT from the SPT Set, and supports active topology enforcement (13.38.2, 8.6.1). When forwarding unicast frames it is not necessary to consider the particular SPT since all the SPTs in a given set transiting a given bridge en route to a destination take the same path from that given bridge to the destination (in other words they are all reverse path congruent with the SPT rooted at that destination). Therefore for unicast frames the destination MAC address is used for egress filtering as normal. SPT Bridging using SPBM mode requires the use of SPT specific destination Group MAC Addresses, so loop-free multicast forwarding is also supported without requiring the source MAC Address or source Bridge Port to be part of the Filtering Database query.

ISIS-SPB automatically allocates SPSourceIDs (27.10) that are used as part of each Group Address. Since an SPT Bridge using SPBM mode inspects source MAC Addresses as a loop mitigation mechanism, those addresses have to be known before forwarding can take place and are advertised using ISIS-SPB. Learning of B-MAC addresses is not enabled when SPBM mode is used, and frames destined for unknown addresses are discarded and not flooded. The Individual MAC Addresses used are those of the SPT Bridges themselves, or are associated with functions provided by Provider Bridges or Backbone Edge Bridge (BEB) functions that include an SPT Bridge component. SPBM group multicast addresses are local to the SPT Bridge domain [using the Locally administered (U/L) address MAC address flag (IEEE-802 2001)], so the use of SPT specific Group MAC addresses does not intrude on the well known global multicast space.

ISIS-SPB uses the same mechanism to allocate SPVIDs (for SPBV) and SPSourceIDs (for SPBM) (27.10). The MST Configuration Identifier (MCID) (8.9, 13.8) is used to ensure that all the bridges in an SPT Region agree on the method used to support each shortest path VLAN and on the pool of SPVIDs, as well as on the other uses of VIDs (27.4). An SPSourceID is allocated to each bridge in an SPT Region, independently of SPBM mode VLANs, while an SPVID for a given SPBV VLAN is only allocated to the SPT rooted at a given bridge if VLAN Registration Entries and controls allow that bridge to transmit frames assigned to that VLAN into the region (8.8, 13.6).

An SPT Bridge using SPBV mode is typically used to support a C-VLAN or S-VLAN for a single customer. An SPT Bridge using SPBM mode is typically used to support B-VLANs in Provider Backbone Bridged Networks (Clause 25). Each B-VLAN can support many backbone service instances, each requiring connectivity between a subset of the BEBs participating in the B-VLAN, and each identified by an I-SID that is not processed by Backbone Core Bridges (BCBs). An SPT Bridge using SPBM mode does not use source address learning, so unicast B-MAC frames conveying customer data are never flooded throughout the B-VLAN. Multicast frames and flooding of unknown unicast frames for a given backbone service instance are confined to paths that reach the BEBs supporting that service instance by including the I-SID in the SPT specific destination Group MAC Addresses (Figure 27-3). The destination address filtering in each BCB prunes transmission of the encapsulated customer multicast frames as well as preventing loops.

MVRP (11.2) is not used within SPT Regions, as ISIS-SPB communicates VLAN registration information for VLANs supported by SPB from the SPT Bridges that provide ingress to and egress from the region to the other bridges, thus allowing SPT Bridges using SPBV and SPBM to configure Dynamic VLAN Registration Entries for Bridge Ports within the region to provide correct VLAN connectivity when SPTs are recalculated—instead of waiting for registration information to propagate on those SPTs. SPT Bridges can use the registration information communicated within the SPT Region to participate in MVRP with bridges outside the region (27.13) by supporting MVRP interfaces on the Region boundaries.

This clause specifies the following:

a) Protocol design and support requirements (27.1, 27.2) and design goals (27.3).

This clause further specifies how:

b) Each bridge's configuration mechanisms specify the supporting method (SPBV, SPBM, MSTI, CIST, or PBB-TE) and SPT Set (for SPBV and SPBM) for each VLAN (27.4).
c) The boundaries of each SPT Region and ISIS-SPB adjacencies are determined dynamically by ISIS-SPB (Clause 28).
d) The information exchanged by ISIS-SPB is used (27.5), including CIST calculation (27.6, 27.7), SPT calculations (27.8), allocating SPVIDs and SPSourceIDs (27.10), and controlling the connectivity supporting each VLAN (27.13) and Backbone Service Instance (27.15, 27.16).
e) The results of ISIS-SPB's CIST and SPT calculations are used by the Clause 13 state machines to determine when to discard frames to ensure the calculated trees are loop-free (27.9).
f) SPVIDs are allocated to FIDs (27.11) and translated to and from Base VIDs at the boundary of an SPT Region (6.9, 13.6).

g) VLAN Registration Entries are used to forward or discard frames for each VLAN (27.13).
h) Filtering Database (FDB) entries for Individual Addresses are used to prevent or mitigate loops for VLANs supported by SPBM (27.14).
i) SPBM Group Addresses are assigned, and FDB entries for those addresses used to forward or discard frames for each backbone service instance for B-VLANs supported by SPBM (27.15, 27.16).
j) The SPT calculations choose between equal cost trees (ECTs), making different choices for different SPT Sets, each associated with a different VLAN, enabling load spreading (27.17, 28.9.2).

This clause also:

k) Provides examples of when and how SPT Bridges using SPBV and SPBM modes can be used (27.18).

27.1 Protocol design requirements

ISIS-SPB configures entries in each SPT Bridge's Filtering Database (8.8) to meet the following requirements:

a) The active topology will fully connect all physically connected LANs and bridges, and stabilize (with high probability) within a short, known bounded interval after any change in the physical topology, maximising service availability (8.4).
b) The active topology for any given frame remains simply connected at all times (6.5.3, 6.5.4).
c) The same symmetric active topology is used, in a stable network, for all frames for the same VLAN, i.e., between any two individual LANs those frames are forwarded through the same Bridge Ports.
d) The active topology supporting a given VLAN within an SPT Region can be chosen by the network administrator to be shortest path, the IST, or an MSTI.
e) Each active topology is predictable and reproducible, and may be influenced by management, thus allowing the application of Configuration Management following traffic analysis, to meet the goals of Performance Management (6.5 and 6.5.10).
f) The configured network can support VLAN-unaware end stations, such that they are unaware of their attachment to a single LAN or a bridged network, or their use of a VLAN (6.3).
g) The SPB mode, SPBV or SPBM, can be chosen by the network administrator independently for each VLAN using shortest paths.

SPBV mode meets the following requirements:

h) No additional constraint is placed on the values of the Individual or Group MAC Addresses used, and those addresses do not have to be known before being used in data frames.
i) SPVIDs are allocated to every SPT Bridge that is a potential source of frames on VLANs supported by SPBV.
j) Filtering database entries for a given SPVID will only be populated if they are on the shortest path to a participating boundary node.

SPBM mode meets the following requirements:

k) Multicast flooding for each backbone service instance supported by a PBBN is restricted to paths necessary to reach BEBs supporting that service instance.

Additionally, SPB algorithms and protocols meet the following goals, which limit the complexity of Bridges and their configuration:

l) The memory requirements associated with each Bridge Port are either a constant or a linear function of the number of Bridges and LANs in the network.
m) The communications bandwidth consumed by ISIS-SPB on any particular LAN is always a small fraction of the total available bandwidth (6.5.10).
n) Bridges do not have to be individually configured before being added to a network, other than having their MAC Addresses assigned through normal procedures.

27.2 Protocol support

ISIS-SPB is configured to use one of the Group addresses from Table 8-14 to establish adjacencies and exchange PDUs. SPB running on C-VLAN components use the Customer Bridge address. SPB running on S-VLAN components use the Provider Bridge address. SPB running on B-VLAN components may use the Provider Bridge Address or one of the existing IS-IS addresses.

The use of the ISIS-SPB adjacency between bridges is contingent on the bridges inclusion within the same SPT Region, and thus requires they have an MCID and Auxiliary MCID (13.8, 28.12.2) where at least one matches on every adjacency in the Region (8.9.4, 13.8). MCID and Auxiliary MCID enable migration of the MCID definition allowing these bridges to operate as one SPT Region under certain small changes. The operational set of Base VLANs and SPVIDs must be consistent during migration. This standard specifies default configurations resulting in a MCID that enables plug-and-play support of both the CIST and shortest path bridging using SPT bridges using SPBV mode. No default recommendations are made for SPT Bridges using SPBM mode, since it is only expected to be used in managed environments.

This standard recommends defaults for Bridge and Bridge Port priorities and path costs to support the calculation of active topologies without further configuration. To allow management of those active topologies, a means of assigning values to the following are required:

a) The relative priority of each Bridge in the network (13.26.3).
b) The relative priority of each Port of a Bridge (13.27.47).
c) A Port Path Cost for each Port (13.27.25, 13.27.33).

Management of shortest path tree topologies and of the use of SPBV or SPBM mode to support particular VLANs, also requires the following:

d) Management of the SPB objects (12.25).
e) Administrative agreement on the MCID Configuration Name and Revision Level (Clause 28).

27.3 Protocol design goals

It is highly desirable that the operation of ISIS-SPB allows the following:

a) Addition of bridges to an existing SPT Region without disrupting existing communication.
b) SPB support for additional VLANs without disrupting existing communication.
c) Enabling or disabling SPB support for individual VLANs with minimum frame loss.

27.4 ISIS-SPB VLAN configuration

The configuration mechanisms used by MST Bridges to allocate each VLAN to a specified MSTI or the IST within an MST Region and used to allocate VIDs for PBB-TE (*25.10*) can be summarized as follows:

a) The VID to FID allocation table (8.8.8, 12.12.2) is used to allocate each VID to a FID.
b) The FID to MSTID Allocation Table is used to associate an MSTID with a FID (8.9.3, 12.12.2):
  1) The IST is identified by the reserved MSTID value 0.
  2) The use of PBB-TE is identified by the reserved MSTID value TE-MSTID (0xFFE)
  3) Each MSTID in the MSTI List identifies an MSTI.
  The reserved MSTID values 0 and TE-MSTID, SPBV-MSTID, SPBM-MSTID and 0xFFF are never used in the MSTI List.

SPT Bridges extend these configuration mechanisms as follows:

c) The VID to FID allocation table is used to allocate Base VIDs to FIDs, and to allocate VIDs for use as SPVIDs. SPVIDs are allocated to the reserved FID value 0xFFF. SPVIDs in use are reported by management as allocated to a FID supported by the implementation (27.11).
d) The reserved FID value 0xFFF is allocated to the reserved MSTID value 0xFFF.
e) The following MSTID values identify the SPT Bridge mode used by SPB:
  1) 0xFFC—SPBM.
  2) 0xFFD—SPBV.

This standard recommends default VID to FID and FID to MSTID allocations (27.2, 8.9.3) for SPT Bridges that enable plug-and-play support of both the CIST and shortest path bridging using SPBV mode.

Table 27-1 shows the VID to FID and the FID to MSTID allocation tables in a single table for the configuration example shown in Figure 27-1 for an SPT Region.

Table 27-1—Allocation of VIDs to FIDs and FIDs to MSTIDs in an SPT Region (example)

VID	FID	MSTID
0x001	0x001	0x000
0x002	0x002	0xFFC
0x003	0x003	0xFFC
0x004		
0x005	0x005	0xFFD
0x006	0x006	0xFFD
0x007		
0x100–0x1FF	0xFFF	0xFFF

Base VID 1 is allocated to the IST and VIDs from 0x100 to 0x1FF form the SPVID Pool. Base VIDs 0x002, 0x003 and 0x004 are configured to operate in SPBM mode. Base VIDs 0x005, 0x006 and 0x007 operate in SPBV mode in the example.

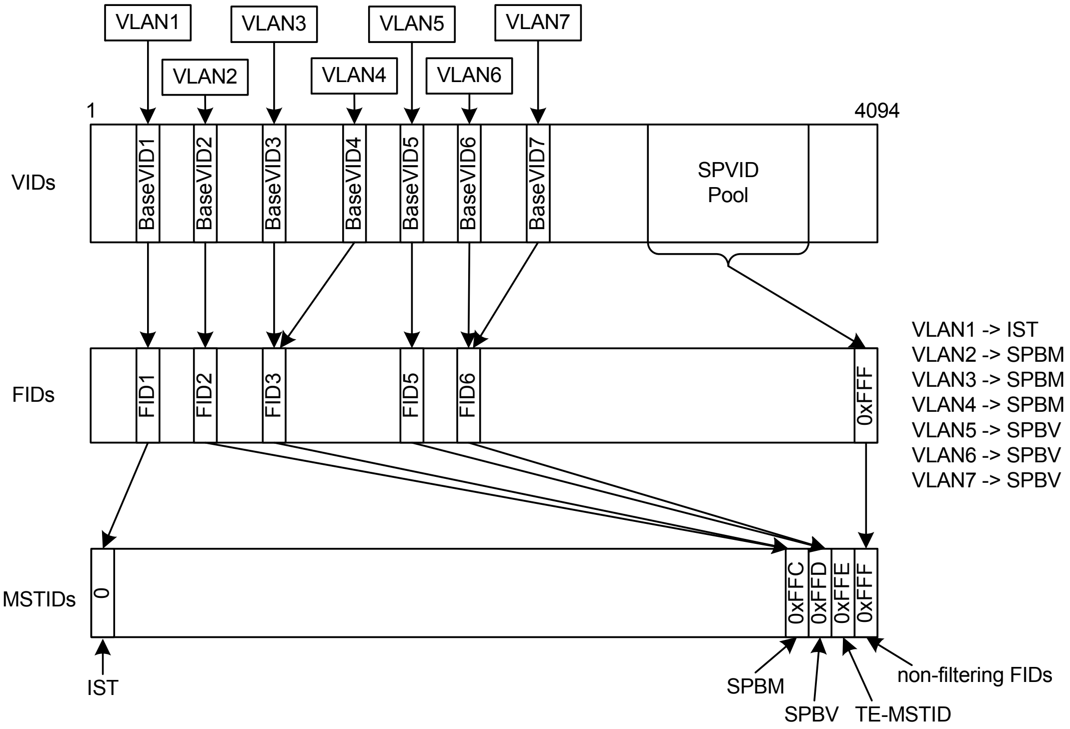


Figure 27-1—Configuring VLAN support in an SPT Region (example)

27.4.1 SPT Region and ISIS-SPB adjacency determination

It is essential that all bridges within an SPT Region agree on the allocation of VIDs to spanning trees prior to transmission and reception of frames tagged with those VIDs. This configuration requirement is common to both SPB (13.6) and MSTP operation (13.5), and MSTP can coexist with ISIS-SPB within an SPT Region, supporting different VLANs with different VIDs. ISIS-SPB uses and extends MSTP's configuration mechanisms and use of the MST Configuration Identifier (MCID, 8.9, 13.5, 13.8). Note the MCID (13.8) includes a digest derived from the mapping of VIDs to MSTIDs. To allow dynamic manipulation of SPVIDs without impacting the MCIDs, the mapping of SPVIDs to the FID associated with the supported Base VID is not captured in this table.

SPB configuration is highly automated. An SPT Region is created by configuring at least two adjacent bridges with compatible information. The region grows dynamically as more SPT Bridges with the same region configuration are introduced. Once a region has been created, other VLANs can also be created. ISIS-SPB will populate forwarding only between bridges that support the same services. The SPT region is an interconnected set of bridges supporting the Region configuration [MCID (13.8), ISIS-SPB Configuration (27.4)] but the forwarding for any one VLAN is simply the bridges configured to support services on that VLAN or the bridges that lie on the shortest path between those bridges.

The same is true of SPT Bridges operating in SPBM mode which can be added to support new VLANs on a bridge by bridge basis. One common operation for SPT bridges operating in SPBM mode is to create a new ECT for a new VLAN. To support a graceful migration of services to a new Base VID, SPB allows the configuration of the Base VID before configuring services on those Base VIDs. The new Base VIDs in SPT Bridges need to be configured on all bridges where the services reside. The U-bit (28.12.5) and Use-flag (28.12.4) are used to identify configured Base VIDs that do not have active services associated with them.

In the cases where the allocation of Base VIDs is insufficient a graceful way to support an increase in the number of Base VIDs is supported by a second (auxiliary) MCID (13.26.8, 28.12.2). Two such bridges consider themselves to be in the same MST Region provided that at least one of their MCIDs matches with one of their SPB neighbor's MCIDs. Non critical changes in the MCID, can be made while maintaining forwarding on the previous configuration. This is done by configuring a bridge with a revised VID allocation that is compatible with the current allocation in use for SPB forwarding. The bridge advertises this new MCID and continues to advertise the previous MCID as the Auxiliary MCID, maintaining its adjacencies.

When an adjacent bridge is similarly configured, it also advertises the new MCID as its MCID to its adjacencies. The Auxiliary MCID is always the outgoing MCID. Bridges that have not been updated will continue to match their updated neighbors' Auxiliary MCID until they have been updated, whereupon their MCID values will match again. When a bridge sees that all its adjacencies are advertising the value of its own new MCID as their MCID, then the bridge shall start advertising the value of its new MCID as its Auxiliary MCID also. When all bridges have been verified to have installed the new configuration, the transfer of services to use the new VLAN allocations may be administratively initiated.

NOTE 1—The MCID verifies that each bridge in the region agrees how each and every Base VID is to be assigned to the IST, to an MSTI, for use by PBB-TE, for support by SPBV mode, for use as an SPVID, or for support by SPBM mode. The MCID does not specify the SPT Set for VLANs supported by SPB, or preallocate SPVIDs to SPT Sets as disagreement on the former or temporary differences in the latter will not cause permanent loss of connectivity, loops, or protracted flooding. ISIS-SPB verifies that SPT Bridges agree on the SPT Set to be used to support any given SPB supported VLAN, and the Agreement Digest ensures that new connectivity is not created until that agreement has been reached.

NOTE 2—The MCID and the Agreement Digest (13.8, 28.4) can both influence the connectivity provided, and their respective roles need to be clear. MCID differences identify configuration mismatches that could cause network malfunction (e.g., data loops) if connectivity were to be created, and that are not amenable to timely resolution by normal protocol operation. MCID differences act as a fail-safe mechanism, forcing lowest common denominator connectivity (to a single spanning tree, the CST) so at least some connectivity is provided, albeit with lower throughput. The fallback connectivity can be used to investigate or correct configuration issues. The Agreement Digest and related mechanisms simply verify that delays in propagating and using protocol information do not cause loops in permitted connectivity.

SPT Regions are similar to MST Regions, and can also be modeled as single bridges (13.5.3). They do not overlap, superset, or subset other regions but participate in the same active topology solely through the CST. Each region can be identified by its Master Port, i.e., the Bridge Port that provides connectivity from the region to a CIST Root that lies outside the region, or, in the case where the CIST Root lies within the region, by the CIST Root Bridge itself. The Master Port is uniquely and globally identified by its Bridge Identifier and Port Identifier on an SPT Bridge, and is communicated by the CIST Regional Root Identifier and its associated CIST Port Identifier fields in SPT BPDUs.

IS-IS adjacencies are formed according to IS-IS protocol rules independent of ISIS-SPB. ISIS-SPB forwarding is only allowed on a more strict basis when the MCID, Base VID, and ECT Algorithm information lines up in the SPB Hello PDUs.

When two SPT Bridge Ports attached to the same LAN have no match between their primary or Auxiliary MCIDs in transmitted SPT BPDUs, each is a Boundary Port (13.12) for its SPT Region, and forwards user data frames only if it is a CST Root Port or Designated Port. Such a network configuration is usually temporary and infrequent, a result of as yet incomplete administrative action, or simply a configuration error (13.6). To minimize network disruption during such administrative episodes, ISIS–SPB communicates information throughout an SPT Domain comprising a transitive closure of SPT Bridges attached to individual LANs.

Each SPT Bridge Port uses BPDUs and/or SPB Hello PDUs to transmit and receive MCIDs and Agreements. BPDUs also carry currently calculated CIST Information, so that all the bridges (not just SPT

Bridges) attached to each individual LAN can be detected and can participate in the information exchange required at SPT Region boundaries. SPT BPDUs or SPB Hello PDUs also convey the CIST Agreement and Agreement Digest information that SPB uses to prevent temporary loops (28.4). This Agreement information can only be sent after the Bridge's ISIS-SPB calculations have been used to modify its Filtering Database entries, and is thus not naturally tied to the transmission of the ISIS-SPB Link State PDUs (LSP)s that convey the adjacency information that makes those calculations possible (LSPs can be disseminated throughout the SPT Domain without waiting for any single bridge to complete its calculations—each proceeding in parallel). Including the Agreement information in the SPT BPDUs helps determine the SPT Region's extent, ensuring backwards compatibility to non SPT Bridges. Exchanging SPB Hello PDUs (to establish adjacencies) and transmitting ISIS-SPB LSPs (to propagate information for Shortest Path Tree Identifier allocation and SPT calculation throughout the SPT Domain) as soon each Bridge Port is MAC_Operational is supported in parallel with BPDUs. Bridges that are not SPB capable will not support SPB Hello PDUs and it is up to the SPT BPDU logic to determine what type of STP protocol the non-SPT Bridges support. In most networks the SPT Region's boundary will coincide with that of the SPT Domain, in other cases LSPs for those Bridge Ports that bound a region but lie within the domain might have to be transmitted.

NOTE 3—Clause 13 specifies the state machines and procedures that determine each SPT's region boundaries, including interoperability with bridges implementing STP (Clause 8 of IEEE Std 802.1D, 1998 Edition), RSTP or MSTP as well as the use of SPT BPDUs to address other error conditions that can occur (e.g., disputes with other bridges due to one-way connectivity).

27.5 ISIS-SPB information

ISIS-SPB can be viewed primarily as a means of sharing information between nodes in a network, providing each node with a copy of a common database of information contributed by each and every node, and ensuring (with high probability) that the copies are kept up to date though not necessarily in lock step synchronization. Each node can then execute its own independent copy of one or more previously agreed algorithms that use that node's copy of the database to compute results that are common to all the nodes, or that are part of a common or complementary set. The result is a set of forwarding tables for each node. ISIS-SPB shares information throughout an SPT Domain so that any given SPT Bridge in the domain can calculate or identify the following:

a) The CIST connectivity provided to each SPT Bridge in the domain (27.6, 27.7). This calculation identifies each of the SPT Regions within the domain (if more than one, 8.9, 13.6), the bridges in each region, and the CST connectivity between regions.
b) The SPB connectivity provided by each SPT Set between bridges in the given bridge's region (27.8).
c) The SPVIDs (for SPBV mode) and SPSourceIDs (for SPBM mode) used to support SPB connectivity (27.10).
d) The VLAN connectivity provided for each Base VID over the calculated tree(s) (27.13).
e) The connectivity provided for each I-SID by the SPT Bridge over the calculated tree(s) (27.16).

27.6 Calculating CIST connectivity

ISIS-SPB calculates CIST connectivity (13.4.1, 13.5.3) using the following information from each bridge in the SPT Domain:

a) The Bridge Identifier and, if the bridge has currently selected its CST Root Port using information received from a bridge that is not in the SPT Domain:
    1) The CIST Root Identifier and CIST External Root Path Cost.
b) The Port Identifier and the following information for each of the Bridge's Ports:
    1) Its IS-IS adjacencies, i.e., a list of the Bridge Identifier, Port Identifier tuples that identify each of the other SPT Bridges attached to the same LAN.

   2) Whether the Bridge Port is or is not a Boundary Port (13.12).

c) For each of the Bridge's Ports that is not a Boundary Port:
   1) The Internal Port Path Cost.
d) For each of the Bridge's Ports that is a Boundary Port:
   1) The External Port Path Cost.

If the restrictedDomainRole (13.27.63) parameter is set for an SPT Bridge Port, then that port will not be selected as a CIST Root Port if the Designated Bridge for the attached LAN is not in the SPT Domain. Used arbitrarily restrictedDomainRole and restrictedRole (13.27.64) can cause lack of connectivity. If used to support the network design these parameters can speed network convergence and protect against disruption if inappropriately configured bridges are attached. If configured for all SPT Bridge Ports, restrictedDomainRole ensures that the CIST Root and all connectivity to and from that CIST Root will be provided solely by bridges within the SPT Domain, thus simplifying CIST calculation. An SPT Bridge implementation may choose to always set restrictedDomainRole, managing it as a read only parameter. This choice restricts network design, but can include all scenarios intended for the implementation (27.18).

The CIST calculation uses the Dijskstra algorithm to compute the same active topology that would be produced by the distributed distance vector calculation specified in 13.10. Computation begins with the best priority CIST bridge priority vector from those that represent each of the bridges within the domain as a potential root and from those reported as received from a bridge outside the domain [item a)1) above]. If the External Path Cost accumulated during the computation would exceed that for a bridge selecting its CST Root Port using information from a bridge outside the domain, then that bridge is also marked as having been reached in spreading computation and can be the best path to further bridges, indicating that the CST path between SPT Regions lies outside the domain (not always a temporary or undesirable result, 27.18).

NOTE 1—The purpose of defining *priority vectors* (13.9) is to allow all the components that contribute to the choice of one path or another to reach a given bridge or LAN from the CIST Root to form part of a single metric.

Each bridge in the SPT Domain performs the CIST calculation independently, computing and retaining the following information for use by the state machines specified in Clause 13 (13.10, 13.17, 13.36, 13.29.34).

e) The CIST root path priority vector for every Bridge Port on the Bridge and the CIST root priority vector for the bridge
f) The IST Port Role for each of the bridge's Bridge Ports that is not a Boundary Port.
g) The CIST designated priority vector for each of the bridge's Bridge Ports.

Each bridge's CIST calculation also identifies:

h) The SPT Region for itself, and for each of the other bridges in the domain
i) The CST Port Role for each Bridge Port that is a Boundary Port for a region.

NOTE 2—Each SPT Region can be identified by the region's Master Port. Connectivity from all bridges in a given region, R_1 say, passes through the same Boundary Port BP_{12}, to reach any bridge in a region R_2. Recording the region and boundary port information during CIST calculation facilitates identification of the boundary bridges.

27.7 Connectivity between regions in the same domain

If the SPT Domain has been found to comprise more than one region then, prior to computing shortest paths, each bridge uses the SPT Region, and CST Port Role information [item h) and item i) in 27.6] computed for each Boundary Port together with the adjacency information for those ports to determine:

a) For each of the regions (R_2, R_3, ... say) other than the computing bridge's own region (R_1, say), the Boundary Bridge (B_{12}, B_{13}, ...) within R_1 that provides CST connectivity to that region.

If the connectivity between regions lies outside the domain and either has (or has IST connectivity to a region that has) more than one bridge that provides connectivity to a bridge outside the domain, then ISIS-SPB does not have the information necessary to select a particular Boundary Bridge. Address learning (for SPBV) or registration protocols can provide the necessary information. If more than one possibility remains, an SPT bridge using SPBM mode cannot provide connectivity between the regions.

NOTE—SPBM mode will often be configured to permit connectivity only if the entire path lies within the domain.

27.8 Calculating SPT connectivity

Each SPT Bridge uses ISIS-SPB to calculate the shortest paths from each of the bridges (27.6(h)) in its own SPT Region including itself, to each of the other bridges in that region. The information used by ISIS-SPB to perform the necessary Dijkstra calculations is a subset of that used for the CIST calculation [items a), b1) c1) in 27.6]. The following information is computed for each SPT and retained for use by the state machines specified in Clause 13 and 27.9, 13.17, 13.36, and 13.29.34:

a) The root priority vector for the bridge;

and for each of the bridge's ports that is not a Boundary Port:

b) The Port Role;
c) The designated priority vector;

and for each of the bridge's ports that is not a Boundary Port and has a Port Role of Designated Port.

d) The best root path priority vector out of those that would be calculated, using the same ISIS-SPB information, by the other SPT Bridges.

There can be several equal cost shortest paths between any given pair of bridges but only one path between the pair can be used in a given SPT Set. Ties are broken using an algorithm (27.17*,* 28.5) that is independent of the order of SPT computation, that ensures that the SPTs in each SPT Set are symmetric, that allows path selection to be managed using the same variables that are used to manage active topologies in the absence of multiple paths, and that is likely to select different equal cost paths for different SPT Sets if configuration defaults are used.

When a VLAN is supported by an SPT Bridge using SPBM mode and loop mitigation (6.5.4.2, 27.14) is being used, each of the FDB entries that control forwarding for unicast frames can be created or modified as soon as the port providing the shortest path to the bridge or end station using that address is known, without the need to use additional protocol or to complete the SPT computation. In that case, restoration of unicast connectivity after a physical topology failure is most rapidly achieved by beginning the SPT (re)computation with the SPT rooted at the calculating bridge.

27.8.1 ISIS-SPB overload

IS-IS has an overload capability which applies to a node that does not allow transit traffic through this node for various operational reasons. This capability has also been added to ISIS-SPB [item c in 28.12.1] TLVs. Just as in normal IS-IS Overload modifies the shortest path results by not allowing bridges with the overload bit set to be eligible to forward transit traffic. For the purpose of computing paths, any paths through bridges with overload set are not considered shortest paths and the computation uses the remaining topology and the next shortest paths. Use of the overload condition is typically reserved for edge devices that would not benefit the network if they were to become transit switches.

In ISIS-SPB if overload were set in the only bridge connecting an SPT Region this would result in a region split where the overload bridge was connected to both halves of the region but traffic could not transit through the overload bridge. Each section of the SPT Region that is split would behave normally but there would be incomplete connectivity between the split sections. ISIS-SPB would have a complete view of the region and this condition is relatively easy to detect.

27.9 Loop prevention

The state machines in Clause 13 use the calculated priority vectors and Port Roles for each tree (27.6, 27.8) to determine the tree's Port State (i.e., the setting of the per port learn (13.27.34) and forward (13.27.29) variables), and exchange SPB BPDUs or SPB Hello PDUs to convey Agreements (13.17) that allow these variables to become set for Root Ports, Designated Ports, and Master Ports without the risk of temporary loops.

The Port State Transition state machine (13.38) signals changes to the learn and forward variables for a port for the CIST through calling disableLearning(), disableForwarding(), enableLearning(), and enableForwarding() procedures. These implementation dependent procedures prompt changes to the mechanisms that perform forwarding and/or learning for frames assigned to the CIST.

Similarly changes to forward for a port for an SPT prompt changes in the Filtering Database entries that allow (or prevent) the ingress or egress of frames assigned to that SPT (27.13, 27.14, 27.16). In turn these prompt changes to the implementation dependent mechanisms that use the entries to control forwarding.

These mechanisms can take time to respond to changes in learn or forward, and report their current state to the state machines (by setting the learning (13.27.35) and forwarding (13.27.30) variables) so that new connectivity on some ports can be made contingent on other ports not forwarding frames. The forwarding variable (for any given tree) is only cleared if frames assigned to that tree neither ingress nor egress the port.

27.10 SPVID and SPSourceID allocation

Each SPT Bridge uses ISIS-SPB to support the distributed allocation of SPVIDs (used by SPBV mode) and the 20-bit SPSourceIDs (used by SPBM mode), ensuring that all bridges in the SPT Domain agree on the allocated values. The allocation protocol for these two types of identifier differs slightly, as a consequence of the differences in the number of values available and the number required for each bridge.

Candidate values for these identifiers may be pre-configured, in some or all of the bridges in the domain, but by default they are allocated dynamically. Each bridge attempts to allocate the following values:

a) An SPSourceID for the bridge.
b) An SPVID for each VLAN that is supported by an SPT Bridge using SPBV mode, and for which:
    1) the bridge is a potential source of frames, either from a protocol entity within the bridge or by forwarding from a port on the bridge that is a Boundary Port; or
    2) the bridge has a pre-configured allocation, even if conflict resolution means that this allocated value is not used.

NOTE 1—Each SPT Bridge allocates an SPSourceID even if no VLANs are to be supported by SPBM mode, or the bridge is configured not to transmit frames for those VLANs. The number of available SPSourceIDs is much greater than the number to be allocated, so there is no need to complicate allocation scenarios by attempting to optimize some cases however frequent.

NOTE 2—If the SPT bridge does not have a pre-configured SPVID for a VLAN supported by SPBV mode, and has no need to transmit frames for that VLAN (possibly because there is no other bridge in the domain that wishes to receive frames for the VLAN or because a VLAN Registration Entry does not allow the bridge to transmit frames for that

VLAN), then an SPVID is not allocated. These conditions can arise even if the Base VID to FID to MSTID configuration for the bridge, and the resulting MCID, identifies the VLAN as supported by SPBV mode.

The allocation protocol resolves conflicting attempts by different bridges. Pre-configured or dynamic allocations are communicated by the setting of the Configuration Flags in the ISIS-SPB TLV used for allocation. Conflicts for allocations of a given type are resolved by granting the allocation to the bridge with the best priority (numerically least) Bridge Identifier (14.2.5). When the conflict occurs on an auto allocated value, the other bridges attempt to satisfy their need for a value by changing their allocation to a new dynamically allocated value. In the case that two configured values conflict, these must be resolved by operator action. All bridges are aware of these conflict resolution rules, so there is no need for additional protocol to confirm or announce the allocation.

Candidate values for dynamic allocation are chosen using a pseudo-random function of each bridge's Bridge Identifier (thus reducing the chance of only one identifier being allocated each time all the bridges transmit fresh allocation information). Values that the allocating bridge knows to be already in use are excluded (thus ensuring that an allocation that has been in existence long enough to be included in the IS-IS database of all bridges in the domain will not be disrupted). Assuming that the unallocated values are arranged as an ordered list, the bridge performs a modulo operation on its Bridge Identifier by the number of free values, and selects the indicated value from the list. This operation is repeated to select the next candidate value.

NOTE 3—When a bridge changes its allocations, i.e., allocates new values to itself as required for (a) or (b) above or releases existing values, ISIS-SPB will communicate the updated information to the other bridges in the domain. However the detailed transmission timing and the other contents of IS-IS PDUs is under the control of ISIS-SPB itself, and is not directly controlled by the allocation protocol. A single PDU will usually convey a number of allocation changes.

NOTE 4—SPT Bridges can use different pseudo-random candidate selection functions, without imperiling the eventual success of the protocol. Nonetheless, SPVID allocation conflicts are likely to be fewer, and more quickly resolved, if the bridges in a domain use the same algorithm.

On first discovering an ISIS-SPB adjacency (27.2, Clause 3) an SPT Bridge does not perform any allocation before acquiring the IS-IS database. The SPT bridge becomes aware of existing allocation through normal IS-IS operation. While acquiring the full IS-IS Database, the bridge advertises SPSourceID = 0 in its LSPs, to indicate to other bridges that it shall not be included in SPF calculations. As soon as the bridge has successfully allocated itself an SPSourceID, this allocated value shall be advertized in ISIS-SPB. If the bridge is aware of dynamic allocations that it has been granted in the past, it may begin by retrying those allocations rather than performing a pseudo-random selection for each that is free.

Each SPT Bridge ages out the allocations, both configured and dynamic, made by other bridges exactly as the allocation information ages out or is replaced in the IS-IS database.

Each SPT Bridge's SPSourceID can be any one of the possible 2^{20} values, of which a small number (less than 2^{10}) are expected to be used, and no exceptional provision is made for the case of a bridge not being able to allocate an SPSourceID. If an SPSourceID is not available, frames for VLANs supported by SPBM mode are discarded.

The pool of SPVIDs are those available for allocation and reserved in the MST Configuration Table. The MCID (13.8) communicated in SPT BPDUs ensures that each bridge in an SPT Region has agreed to the same pool of SPVIDs. A quantity of 400 SPVIDs are pre-configured for auto allocation (13.8) for small SPT networks running SPBV. Additional SPVID allocations can be made out of the remaining free VIDs. An MST Configuration Table change that extends the available SPVID pool does not disturb existing allocations, but will affect the MCID—so the connectivity between adjacent bridges will revert to the CST representing a region boundary, unless the procedures for hitless migration of MCIDs using Auxiliary MCID (27.4.1) are followed. If a bridge is added to a region or a new SPBV VLAN configured and there are not enough SPVIDs to satisfy the demand, any SPBV VLAN that does not have a full allocation of SPVIDs falls back to CST operation using its Base VID. When enough SPVIDs become available, either by being freed from other SPBV VLANs or by increasing the size of the SPVID pool, and an SPBV VLAN obtains a full allocation of SPVIDs it advances to SPBV operation using the SPVID set. During periods when there are insufficient SPVIDs to satisfy all SPBV VLANs, bridges that have successfully obtained allocations of SPVIDs to SPBV VLANs must maintain them to ensure stable behavior.

27.11 Allocation of VIDs to FIDs

The allocation of SPVIDs to FIDs (8.8.8) for SPBV is a local decision, subject to the following constraints.

a) All SPVIDs (if any) supporting a given VLAN (identified by Base VID V_1, say) shall be allocated to the same FID (F_{SV1}, say).
b) SPVIDs shall be allocated to FIDs (F_{SV1}, F_{SV2}, ..) that are different from those allocated to any VLAN's Base VID (F_{V1}, F_{VN} ..).
c) VLANs supported by different SPT Sets always use different FIDs (both for SPBV and SPBM modes).
d) If the VID to FID allocation table (8.8.8) has been configured to allocate two different Base VIDs to different FIDs (F_{V1}, F_{V2}), then the SPVIDs used to support each of the VLANs identified by those Base VIDs shall be allocated to different FIDs (F_{SV1} != F_{SV2}).
e) If the VID to FID allocation table (8.8.8) configuration allocates two different Base VIDs to the same FID ($F_{V3} = F_{V4}$), then the SPVIDs shall also be allocated to the same FID ($F_{SV3} = F_{SV4}$). This is shared VLANs where two or more VLANs can share the same topology.
f) SPVIDs allocated to FID in this manner do not show up in the MCID as changed.

NOTE 1—Frames for VLANs supported by SPBM mode are always assigned to an SPT Set within an SPT Region, never to the IST, and are always identified by each VLAN's Base VID.

NOTE 2—Constraint (b) allows the active topology for any SPBV supported VLAN to be changed from the IST to an SPT Set without requiring topology change signalling.

These constraints, and the fact that the MST Configuration Table's entries for VLANs supported by SPB are not configured directly but are a consequence of configuring the relationship between Base VIDs and FIDs (8.8.8) and between FIDs and MSTIDs (8.9.3), also ensure the following:

g) SPT Bridges using SPBV and SPBM modes always use different FIDs, so SPBV learning cannot disrupt SPBM operation.
h) VLANs with shared learning constraints (expressed by allocation of a common FID) are assigned to the same SPT Set.

If all the MAC Addresses for a VLAN supported by SPBM mode are those of Backbone Core Bridges and Backbone Edge Bridges then the use of different FIDs for B-VLANs supported by the same SPT Set can seem unnecessary. However the constraints could support additional capabilities and shall be followed.

27.12 SPBV SPVID translation

An SPT Bridge Port that is a Boundary Port (13.12) translates, on both ingress and egress, the VID assigned to a VLAN that is supported by SPBV. On reception, the Boundary Port uses the VID Translation Table (6.9) to translate the Base VID received in the tag to the SPVID for the SPT rooted at that bridge. On transmission, the Boundary Port uses the Egress VID Translation Table (6.9) to map all of the SPVIDs for that VLAN to one Base VID. The Boundary Port's VID Translation Table is also configured to prevent any local VID from being translated to a relay VID within the SPVID range.

NOTE 1—One way to prevent local VIDs within the SPVID range from being translated to relay VIDs in the SPVID range is to change the default relay VID value for these entries to the reserved value 0xFFF, which would cause these frames to be discarded.

If it has not been possible to allocate SPVIDs to support a given VLAN, no translation is performed and the frame is carried on the IST within the region using the unmodified Base VID.

If a Boundary Port's untagged or priority tagged frames belong to a VLAN supported by SPBV, the PVID for that port is set to the SPVID for the SPT rooted at that bridge, and the port should belong to the untagged set for each of the SPVIDs for that VLAN.

Frames received on ports internal to the region are not translated, whether they carry SPVIDs or Base VIDs.

NOTE 2—A frame within any bridge within an SPT Region is considered as being forwarded within that region. Consequently a frame that is forwarded between two Designated Ports at the boundary of a region can have its VID translated on reception, from the VLAN's Base VID to the SPVID for the VLAN and SPT rooted at the Bridge, and on transmission, back to the Base VID. Translation is not dependent on forwarding, so the frame could have been equally forwarded through a port within the region.

27.13 VLAN topology management

An SPT Bridge using SPBV mode uses Dynamic VLAN Registration entries, for each VLAN it supports, to create a VLAN topology that is the pruned subset of the active topology that connects participants in that VLAN. The VLAN registration entries thus serve to prevent both loops and unnecessary flooding of frames through Bridge Ports and LANs that cannot provide connectivity to their destinations.

Each SPT Bridge uses ISIS-SPB with the information calculated for the CIST and each SPT (27.6, 27.7, 27.8), and the following information from each bridge in the SPT Domain with one or more Boundary Ports:

a) VLAN membership registration for each VLAN registered on a Boundary Port, subject to the registration controls defined for MVRP (11.2.3.2.3).

For each of its Bridge Ports that is not a Boundary Port, and for each VLAN supported by SPBV mode, the SPT Bridge computes the following potential values of control in VLAN registration entries:

b) Forwarded or Filtered, for the IST, for each port that is a Root Port or a Designated Port.
c) Filtered, for the IST, for each port that is an Alternate Port or Backup Port.
d) Forwarded or Filtered, for each SPT, for each port that is a Designated Port.
e) Forwarded, for each SPT, for each port that is a Root Port.
f) Filtered, for each SPT, for each port that is an Alternate Port or Backup Port.

The potential control element for the CIST specifies that frames for the VLAN are to be Forwarded, if the port provides CIST connectivity to a Boundary Port that has registered the VLAN. Similarly the potential control element for an SPT Designated Port specifies Forwarded if that port provides the shortest path to a bridge with a Boundary Port that has registered the VLAN, either directly (if that bridge is in the same region) or via a Boundary Port that provides CST connectivity to the bridge in another region in the domain.

VLAN Registration Entries support ingress filtering, so the potential control element always specifies Forwarded for each SPT's Root Port, thus supporting Open Host Groups (*10.10.1*). However an SPVID is only allocated (27.10), to support a given VLAN over a given SPT, if frames for that VLAN can enter the SPT Region at the root of that SPT, so the potential control element might not be used.

Each SPT Bridge creates Dynamic VLAN Registration Entries using the potential control elements and the Port State variable forward (27.9) for each tree for each VLAN supported by SPBV as follows:

g) An entry for the Base VID specifies Registered for a port if:
  1) forward is True for the IST for the port, and
  2) the potential control element for the port for the IST and the VLAN specifies Forwarded;

  and specifies Not Registered for the port otherwise.

h) An entry for each SPVID specifies Registered for a port if:
  1) forward is True for the port for the SPT identified by the SPVID, and
  2) the potential control element for the port for the SPT and the VLAN specifies Forwarded;

  and specifies Not Registered for the port otherwise.

These Dynamic VLAN Registration control elements for each of the SPT Bridge's ports are subject to restrictions imposed by Static VLAN Registration Entries. Static VLAN Registration Entries shall not be administratively created for any SPVID, so the controls applied to shortest path forwarding are solely those determined at the boundaries of each SPT Region.

VLAN topology management for an SPT Bridge using SPBV, as specified in this clause (27.13), makes use of Static and Dynamic VLAN Registration Entries that can be used with MVRP. However their use in this clause is independent of whether MVRP is implemented or used, and does not require the bridges in an SPT Domain to agree on its use. If MVRP is implemented and used on Boundary Ports, then MVRP PDUs are transmitted and received on those ports (as permitted by the controls specified for MVRP) to exchange registration information with bridges or end stations outside the SPT Domain.

Since SPBM mode only forwards frames whose destination and source addresses are known to ISIS-SPB, the filtering entries installed for these addresses can be used to control all forwarding for each SPBM supported VLAN. SPBM mode creates a Dynamic VLAN Registration Entry for the VLAN used by the SPT Bridge. Operators should not create Static VLAN Registration Entries for the VLAN used by SPBM mode.

27.14 Individual addresses and SPBM

Each SPT Bridge that supports SPBM mode may be configured to use loop prevention (6.5.4.1) and/or loop mitigation (6.5.4.2) for unicast frames assigned to VLANs supported by the SPT Bridge. Loop prevention is always used for multicast frames. Both loop prevention and loop mitigation use Dynamic Filtering Entries for Individual MAC Addresses for each VLAN supported by SPBM mode. These filtering entries serve both to implement ingress checking, effectively assigning each received frame to an SPT and ensuring that the frame is only forwarded if received on the Root Port of that tree, and to implement egress checking, ensuring that the frame is only forwarded through the port providing the shortest path to its destination.

SPBM mode uses VLAN membership configuration information distributed by ISIS-SPB to reduce the number of individual address filtering entries, i.e., {VID, Individual MAC Address} tuples that need to be made in each SPT Bridge. Entries for a given VID are made only for ports that are in the pruned subset of the active topology that connects participants in the VLAN with that Base VID.

NOTE—The reduction in individual address entries is most effective in very large SPBM capable networks, using hundreds (or thousands) of bridges, and hundreds of B-VLANs each covering a small subset of the bridges, each subset supporting a number of backbone service instances. Using a large number of B-VLANs in such a network could significantly reduce the scope of the control flows used to create and maintain backbone service instances.

27.14.1 Loop mitigation

If loop mitigation is selected, each SPT Bridge can create or modify an individual address filtering entry (for a given VLAN) once the calculation of the SPT rooted at that bridge (and in the SPT Set for that VLAN) has associated the address with one of the bridge's ports, subject to satisfying constraints that prevent multicast loops. These constraints are those that govern the Port State transition of a Root Port to Forwarding (i.e., on setting the Port State variable forward true for SPT's Root Port, as specified in Clause 13). This subclause (27.14.1) specifies a sufficient, though not necessary way to meet those constraints. If any discrepancy exists between this summary and the Clause 13 specification, the latter takes precedence.

The Port Role Transition machine ensures that there are no disputes with the adjacent bridge(s) with respect to the active topology, such as might arise from partial or one-way connectivity (13.21). Otherwise the Dynamic Filtering Entry can be created or modified provided:

a) The entry was previously unused, or the operation of changing the entry ensures that no two ports can be receiving frames from or sending frames to the same {VID, Individual Address} tuple at the same time; and
b) The Static Filtering Entries that limit the propagation of the SPBM SPT specific Group Addresses for that SPT have been deleted or specify Filter on all ports (27.15).

NOTE—Where a large number of SPT specific Group Addresses are used, the effectiveness of loop mitigation as a strategy (for more rapidly restoring unicast connectivity after a component failure than might be achieved by using loop prevention) depends on the ability of the implementation to rapidly suspend multicast connectivity, as required by (b) above, possibly by distinguishing unicast and multicast forwarding.

27.14.2 Loop prevention

If loop prevention is selected, Dynamic Filtering Entries for Individual MAC Addresses for VLANs supported by SPBM mode reflect SPT Port States for both Root Ports and Designated Ports, for different SPTs in the SPT Set for each VLAN. The state machines specified in Clause 13 attempt to minimize the changes to filtering entries necessary following a network component failure by identifying conditions when the change in active topology constitutes a loop-free alternative. Otherwise Dynamic Filtering Entries are removed as necessary as part of making the forwarding variable False, so that an Agreement Digest can be transmitted to adjacent bridges and full connectivity restored.

NOTE—The Clause 13 state machines specify necessary conditions for creating new connectivity. These conditions can be satisfied in a number of ways, by processes that can execute in parallel in adjacent bridges and in the same bridge. The most effective way (time taken, resource expended) is implementation dependent, and is not specified as a set of fixed steps. For example, Static VLAN Registration Entries (or an implementation dependent equivalent) could be used to disable forwarding to allow a fresh Agreement Digest to be sent at the earliest opportunity. Alternatively, where SPBM mode uses a large number of SPT specific Group MAC Addresses to support many backbone service instances between rather fewer SPT Bridges, removing (or disabling) Dynamic Filtering Entries for Individual Addresses entries being used as an ingress check may be more expedient.

27.15 SPBM group addressing

When a B-VLAN is supported by an SPT Bridge using SPBM, the individual destination and source MAC addresses of each frame are those used by entities within SPT Bridges or within systems that include an SPT Bridge Component, such as Backbone Edge Bridges (BEB, Clause 3, Clause 25). SPBM mode assigns Group Addresses of local significance to the SPT Domain, each including an SPSourceID (27.10), for use by those entities. Taken together each frame’s B-VID and SPSourceID identify the SPT used to forward the frame, while ISIS-SPB’s knowledge of the addressed group allows it to create a MAC Address Registration Entry (8.8.4) for the address, restricting propagation to the pruned subset of the SPT that reaches the group’s members.

In particular, a distinct Group MAC Address is assigned to each Virtual Instance Port (VIP, Clause 3) for use as the B-DA of the multicast frames that it transmits into an SPT Domain. Each VIP supports a single backbone service instance, so the B-DA identifies the I–SID for each customer frame that is multicast and its transmission is confined to LANs providing connectivity between BEBs supporting that service instance.

Figure 27-2 illustrates the general format of SPBM Group MAC Addresses. The octet and bit ordering and identification conventions used in this figure follow those of IEEE Std 802-2001 Figure 8. The conventions used to encode parameters with numeric values, such as the SPSourceID or I–SID, are those used elsewhere in this standard (14.1.1): the number is encoded as an unsigned binary numeral with bit positions in lower octet numbers having more significance. The least significant and the next to least significant bits of the first octet of the address, the Individual/Group and Universally/Locally administered bits, are both set denoting a locally administered group address. The two next most significant bits compose an SPBM address type, permitting alternative uses of the remaining bits of the address. This standard encodes the I–SID in the last three octets, and the value 0 in the address type field (Figure 27-3). Other values (1, 2, and 3) are reserved for future standardization.

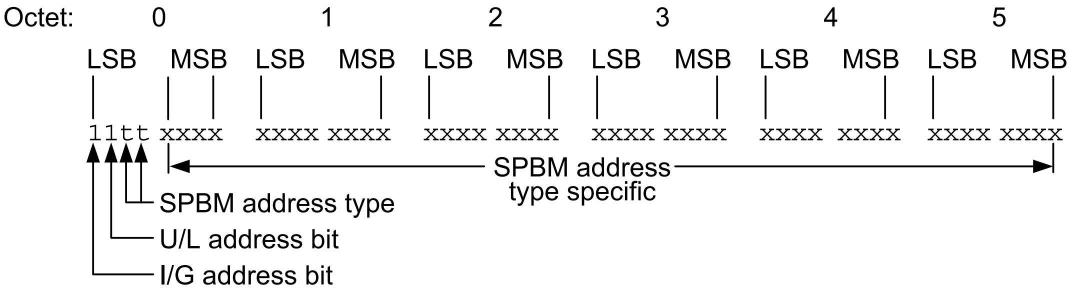

Octet: 0 1 2 3 4 5
LSB MSB LSB MSB LSB MSB LSB MSB LSB MSB LSB MSB
11tt xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx
SPBM address type specific
SPBM address type
U/L address bit
I/G address bit

Figure 27-2—SPBM Group MAC Address—general format

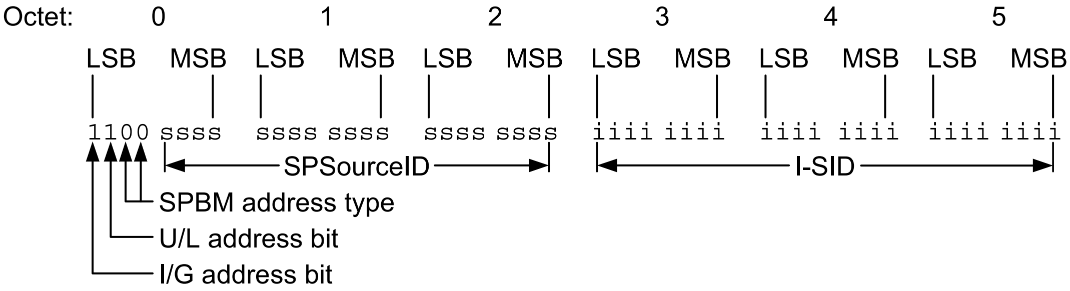

Octet: 0 1 2 3 4 5
LSB MSB LSB MSB LSB MSB LSB MSB LSB MSB LSB MSB
1100 ssss ssss ssss ssss ssss iiii iiii iiii iiii iiii iiii
SPSourceID
I-SID
SPBM address type
U/L address bit
I/G address bit

Figure 27-3—SPBM Group MAC Addresses used by this standard

The use of locally assigned Group MAC Addresses by SPBM mode does not affect the use of the same or different Group MAC Addresses for other VLANs. In addition to creating and managing the required Filtering Database entries for each specific group address (GA_{SPBM}) for any given Base VID (V_{SPBM}, say) that identifies an SPBM mode supported VLAN, an SPT Bridge using SPBM mode also manages the following entries:

- If the Filtering Database contains a static entry specifying Forward (for any Bridge Port) for the wildcard VID and any specific Group MAC Address not required by this clause (27.15) then an entry specifying Filter (for all Bridge Ports) is also created for that Group Address and each Base VID supported by SPBM mode.
- A Static Filtering Entry for All Group Addresses (for which no more specific Static Filtering Entry exists, (8.8.1) is created for each Base VID supported by SPBM mode, specifying that those addresses are to be filtered.

27.16 Backbone service instance topology management

An SPT bridge using SPBM mode uses Dynamic Filtering Entries to explicitly permit unicast communication between stations, thus restricting that communication to the shortest path between those stations. SPBM mode also uses Static Filtering Entries, for each VLAN it supports, to create a multicast topology that is the pruned subset of the active topology that connects BEBs supporting the same backbone service instance using that VLAN.

Each SPT Bridge uses ISIS-SPB with the information calculated for the CIST and each SPT (27.6, 27.7, 27.8), and the following information from each BEB in the SPT Domain:

a) I-SID membership registration for each VLAN supported by SPBM mode.

For each of its Bridge Ports, for each I-SID and each SPT supporting the VLAN supporting that I-SID, the SPT Bridge computes the potential filtering control element values, Forward or Filter, for each port that is a Designated Port or Root Port. The potential control element specifies Forwarded if that port provides the shortest path to a BEB that has registered the VLAN, either directly (if that bridge is in the same region) or via a Boundary Port (if the bridge is in another region in the domain). A value of Filter is associated with each port that is an Alternate Port or Backup Port for the SPT.

The SPT Bridge creates a Static Filtering Entry for the VLAN's B-VID and SPBM Group Address (27.15) for the I-SID and SPT, with a control element that specifies Forward if both the potential control element specifies Forward and the Port State variable forward (27.9) is True for the SPT for the port.

27.17 Equal cost shortest paths, ECTs, and load spreading

There can be several equal cost shortest paths between any given pair of bridges, and the paths used by the SPTs in a given SPT Set are required to be symmetric (13.1), so the local tie-breaker [the lowest ISIS Local Circuit ID on the bridge with the best (numerically least) SPB System Identifier (Clause 3)] is only used to select between links providing alternative connectivity between immediately adjacent bridges. Equal cost tie breaking between paths that traverse different sets of bridges is performed by calculating a unique identifier for each path that is independent of order used to compute trees. See 28.5 for a description of the Algorithm and the Tie Breakers.

The capability to support multiple ECTs allows a network administrator to choose different equal cost paths for different services.

A set of ECT Algorithms (28.8) is specified, each using a different tie breaker. The tie breaking rules have been selected to provide a reasonable probability that different ECT Algorithms will choose different paths from a set of equal cost paths. ECT path selection can be controlled by selection of ECT Algorithm and by adjustment of Bridge Priorities.

SPB makes use of all nodes in the SPT region for forwarding.

27.18 Using SPBV and SPBM

This subclause (27.18) provides one example topology of SPT Bridges using SPBV mode and one example topology of SPT Bridges using SPBM mode.

27.18.1 Shortest Path Bridging—VID

The primary advantage SPT bridges operating in SPBV mode bring over Spanning Tree Protocols is the more flexible shortest path forwarding. SPBV accomplishes this by using SPVIDs and VID translation (6.9) at SPT Region boundaries. All traffic within an SPT Region always takes a shortest path. Loops are prevented by never allowing traffic to be accepted on a non symmetric shortest path and enforced with loop prevention (13.17, 6.5.4.1).

An example campus network using SPBV is illustrated by Figure 27-4. The SPT Region contains eight SPT Bridges with two RST bridges connected to the SPT Region. All VLANs within the SPT Region are supported by the SPT Bridges. The dashed lines represent the shortest path tree for the SPVID of a Base VID (VID10) for which Access Bridge 1 is source (root of the SPT). The solid lines to the RST bridges indicates Spanning tree forwarding. VID translation from and to the respective SPVIDs occurs on the region boundary ports SPT Bridge D port 5 and SPT bridge E port 3. SPT forwarding makes better use of a mesh network however it only will use a single path per Base VID. For example the bridge D port 2 to bridge B port 4 link is not used by access bridge 1 because the ECT Algorithm has chosen paths through bridge C. If another VLAN is supported in this network the recommendation is to use a different ECT Algorithm that could choose D-B over C-B by using an appropriate tie breaker (28.8).

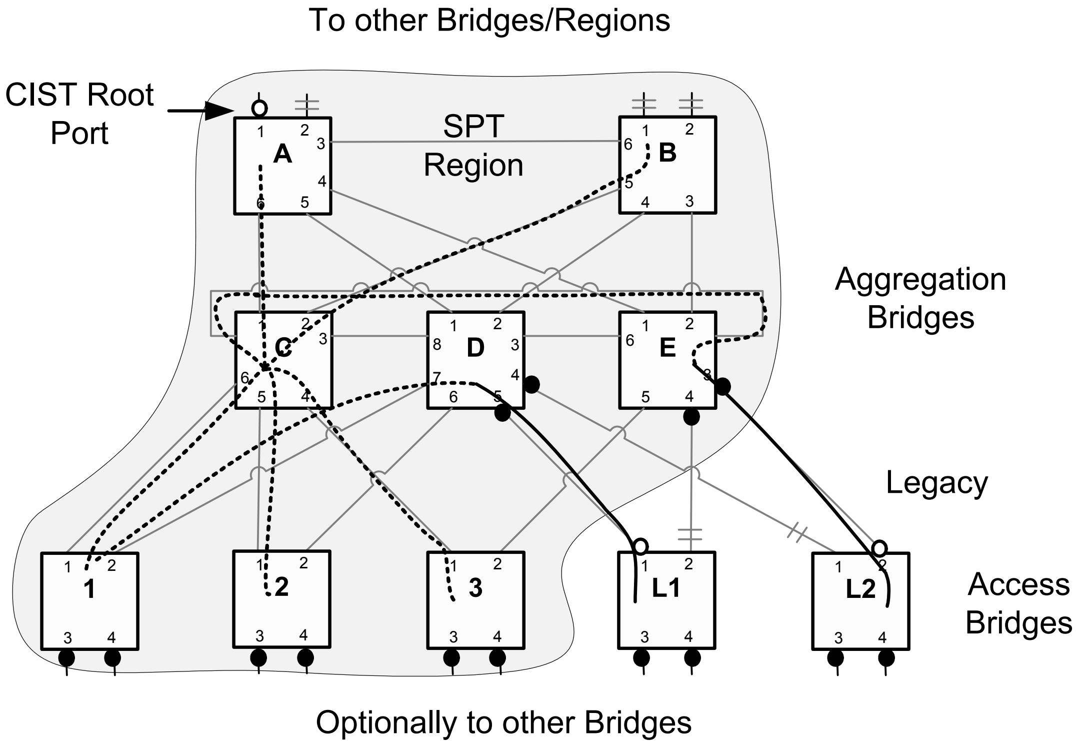


Figure 27-4—SPBV campus network example

ISIS-SPB uses SPB Hellos to establish adjacencies within the SPT region and BPDUs to establish peering with RST bridges. BPDUs from legacy bridges L1 and L2 will indicate that they are able to use RSTP. Therefore RSTP would be enabled and SPB Hello would be transmitted periodically but dropped. The ports connected to the legacy bridges are enabled and block according to RSTP logic.

SPT Bridges using SPBV mode use the same principles as MSTP to form a logical bridge with a CIST root port on Bridge A, which has the best (numerically least) Bridge Identifier in the SPT Region. The CIST in Figure 27-4 is a Spanning tree rooted on bridge A (not illustrated).

a) The eight bridges in the SPT Region have the same MCID (13.8) or Auxiliary MCID (28.9).
b) The CIST is computed in the SPT Region and maintains the Spanning Tree root costs (13.17).
c) SPT BPDUs are exchanged by the SPT Bridges within the SPT Region (13.17).
d) VLANs identified by other than the Base VIDs are supported by the CIST (13.17).

Legacy bridges L1 and L2 can be aware of the SPB operation of bridges but they only support RSTP and interface to the aggregation bridges D and E using RST BPDUs. L1 and L2 learn from the received BPDUs that aggregation bridge A is the Designated Bridge for VLAN 10 while a bridge somewhere above the SPT region is the CIST Root for the network. Aggregation bridge D receives traffic from L1 but uses an SPVID and the corresponding SPT from D to reach all nodes within the region and beyond. Normal learning of MAC addresses occurs.

Dynamic VLAN Registration capabilities are supported natively for Base VIDs by utilizing the link state to auto enable forwarding on intermediate bridges. Referring to Figure 27-4, if access bridge 1 and aggregation bridge A were the only bridges with VID 10 registered on their boundary ports forwarding would automatically be setup on aggregation bridge C to complete the data path between bridge 1 and bridge A. Only bridges configured with the Base VID, bridge 1 and bridge A require SPVIDs. MVRP for other VLANs is supported on the CIST.

MAC Address Registration (8.8.4) is supported for the Base VIDs as outlined in 28.10. Group addresses for Base VIDs are distributed in ISIS-SPB. At the region boundaries MMRP is required to propagate registration to and from the SPT Region.

Since the SPT Region follows very similar design rules as the MST Region, RSTP and STP interwork with both. SPB interworks with MSTP and other SPT Regions with MSTP.

27.18.2 Shortest Path Bridging—MAC

An example SPT Bridge network using SPBM mode is illustrated by Figure 27-5. The SPBM mode example network is divided into two major areas. These are: the Customer Equipment area, which is owned by the “customer” and is connected to the SPT Backbone Edge Bridges using C-tagged or S-tagged interfaces; and the SPT Region, which is owned by the backbone provider and interconnects to the customer equipment and possibly to other SPT Bridges. There is no visibility of either backbone topology or B-MACs outside the SPT Region, and consequently connections to other SPB or PBB areas will be identical to those to customer equipment. While this is primarily leveraging the capabilities of the Provider Backbone Bridged Network the emphasis is that the separation of the address spaces proved by the PBB encapsulation provided useful capabilities of scaling even larger enterprise or data center networks where bridging and delivery of optimized unicast and multicast frames on a Service VLAN basis is desired.

The interface between an IB BEB and a customer bridge is a Customer Network Port (CNP) of an I-Component as specified in 6.10. The interfaces between BEBs and Backbone Core Bridges (BCBs) and between BCBs and other BCBs are Provider Network Ports (PNPs). In Figure 27-5, these interfaces are over LANs labeled S and B. All exterior interfaces from a SPT Bridge network using SPBM mode are over S, or C LANs; whereas all interior interfaces within a SPT Bridge network support appropriate segments of SPB shortest path trees (SPTs), each set of which is identified by one Base VID per forwarding topology. Each individual SPT within a set is identified by the B-MAC of its source bridge. The S and C VLANs (exterior interfaces) are at the boundary between the SPT Bridge network and the attached customer networks.

Within the network example BEBs, Backbone Core Bridges, and the SPTs for each Base VID are secured so that only the network administrator operating the SPT Bridge network can manage the reception, transmission, and relay of frames between BEBs and BCBs. The arbitrary physical network topology of the SPT Bridge network and the connectivity that it provides to support segregated instances of the MAC Service is designed and managed by the network administrator to meet bandwidth and service availability requirements at the Customer Backbone Ports (CBPs) and PIPs. Application of the service instance ingress and egress rules at these Ports in support of service instance selection and identification ensures that frames cannot be transmitted or received on any backbone service instance by any customer’s equipment without prior agreement with the administrator of the SPT Bridge network.

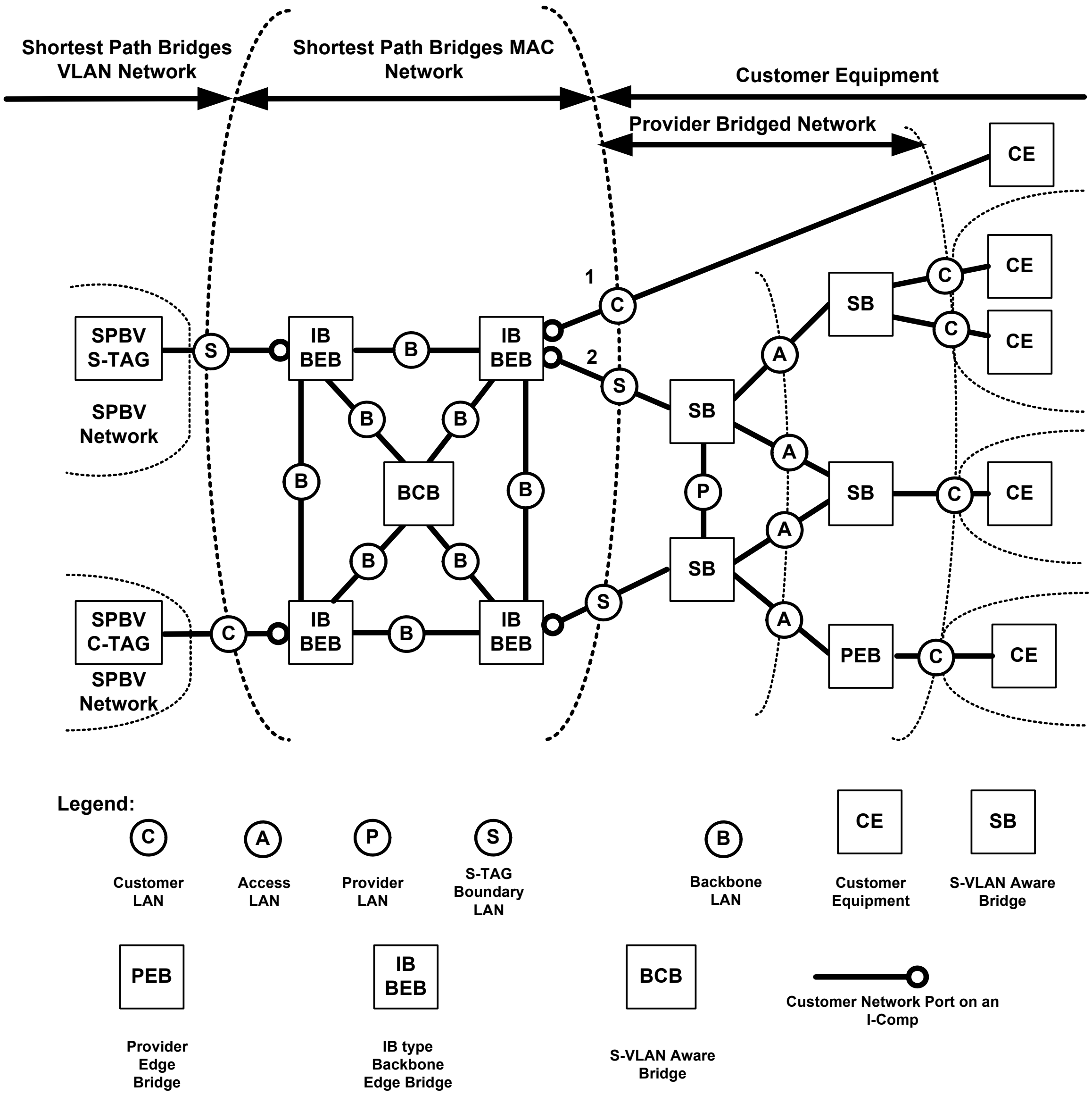


Figure 27-5—SPT Bridge Network using SPBM example

The active topology of the SPT Bridge network is separated from the active topology of the customer equipment area. This is accomplished by isolating the BPDUs for each customer network from the SPT Bridge network at the BEBs that surround the perimeter of the SPT Region. The ISIS-SPB link state packets and SPT BPDUs, that support the active topology of the SPT Bridge network, are only exchanged on Provider Network Ports and hence are not leaked into the customer bridged area.

Backbone MAC addresses are used to identify the destination BEB's PIP. These backbone MAC addresses are advertized by each BEB to all other BEBs and BCBs in the SPT Bridge area in link state packets as part of the IS-IS routing exchanges. I-SID instances associated with each PIP are also advertized using the same mechanism. To perform the encapsulation and de-encapsulation of service frames, BEBs use the connection identifier stored in the filtering database (8.8) to correlate Customer MAC Addresses to backbone MAC addresses. At startup, after convergence of ISIS-SPB, the BEBs are aware of the location of B-MACs in the topology, but have not learned the C-MAC addresses yet. When the C-MAC address is unknown, the BEB encapsulates the service frames using an SPBM Group address which is analogous to the Backbone Service Instance Group address (26.4). This address is then multicast to the set of registered receivers that are those BEBs where that I-SID has been configured. The same holds true for the service frames having a group or broadcast destination address. An individual B-DA is used for forwarding unknown or group frames when the service is point-to-point and where the far end individual address is known. C-MAC addresses are learned by the provider BEB against the B-MAC associated with the service interface on the BEB which sourced them (see Clause 25).

27.19 Security considerations

This clause contains no exceptional provisions for securing the operation of ISIS-SPB. ISIS has a number of optional capabilities which may be utilized. For example, ISIS-SPB does not prevent the use of the IS-IS Authentication TLV. If stronger security is required IEEE Std 802.1AE MAC Security should be deployed on all SPT Bridge Ports, and the configuration of the controls provided by this standard to regulate the acceptance of spanning tree protocol information [e.g., restrictedRole and restrictedTcn (13.27.64, 13.27.65)] should be set to reflect the degree of trust (i.e., authorization) accorded to the mutually authenticated securely communicating peer bridges. SPT BPDUs include a flag that indicates that the attached LAN is not considered to be a part of the transmitting bridge’s SPT Region: this avoids one way participation in a region if one bridge considers the authorization of another inadequate. Use of MAC Security in this way can ensure that only authenticated and authorized (i.e., trusted) bridges participate in a region, as well as preventing unauthorized transmission of ISIS-SPB information or user data on LANs within the region.

Insert the following text, Clause 28, after Clause 27:

28. ISIS-SPB Link State Protocol

IS-IS (Intermediate System to Intermediate System) is the link state routing protocol used for SPB (Clause 27). IS-IS is easily extended to carry the required Ethernet addresses, VLAN and Service membership information introduced by SPB. IS-IS is defined in ISO/IEC 10589. When used for SPB, IS-IS has no IP dependencies. IS-IS also supports Multi-Topology (MT) and logical instances for easy virtualization of the link state protocol. This section specifies extensions to IS-IS for Shortest Path Bridging. IS-IS extended for SPB will be referred to as ISIS-SPB. The SPB extensions are additive to IS-IS and do not preclude the IS-IS protocol being used for other purposes at the same time.

In VLANs that are configured to use SPB, ISIS-SPB controls forwarding (Clause 8) of individual and group addressed MAC frames. The Agreement Protocol (13.17), utilizes the ISIS-SPB link state database along with a synchronization mechanism carried in SPT BPDUs or SPB Hello PDUs. Locally on each bridge, the Agreement Protocol logic is used to control the ISIS-SPB FDB updates and implement loop prevention logic. ISIS-SPB uses normal IS-IS procedures to update the link state database.

IS-IS uses SPB Hello PDUs (Protocol Data Units) and Link State PDUs to communicate between SPT bridges. This section describes the relevant TLVs (type, length, value) in the ISIS-SPB PDUs. Note that ISIS-SPB does not add any new PDU's to IS-IS it extends the currently defined PDUs.

Figure 28-2 through Figure 28-11 illustrates the new TLVs and sub-TLVs for SPBV and SPBM. Note that SPBM, since it derives from PBB has the capability to indicate Service instance membership using the 24-bit I-SIDs. A default I-SID (I-SID 0x0000ff) maps to all SPT bridges operating in SPBM mode allowing a default broadcast and group address capability for ISIS-SPB. The default I-SID is assigned to the default ECT Algorithm (LowPATHID, 28.7). I-SID group registrations via ISIS-SPB are similar to the MMRP capability but because they are delivered by IS-IS they involve no additional messaging. Group addresses are constructed by SPBM for each service instance/source and therefore do not require explicit advertisement.

The I-SID value of 0 is reserved by SPBM and is used as a deleted or free I-SID value in the TLV's. This reserved value may be used to avoid re-organizing Link State PDUs (LSPs) when only one or a few I-SIDs are changed. This provides an optimization by allowing actual deletion (shuffling of the LSP contents) to be decoupled from the ISIS-SPB updates triggered by topology changes. LSPs may be reorganized during periodic updates thereby minimizing impacts.

SPBV and SPBM will operate either within an IS-IS level 1, or an ISIS level 2. As a result the TLVs specified here may propagate either in level 1 OR level 2 LSPs. IS-IS SPBM implementations shall support level 1 and may support level 2 operations. Multi level ISIS-SPB is however for further study.

NOTE—IS-IS supports point to point links and shared LANs however only point to point is considered for ISIS-SPB in this specification. A future revision may add shared media. Clause 13 SPB description allow both point to point links and shared media.

28.1 ISIS-SPB control plane MAC

ISIS-SPB makes no changes to the IS-IS control plane addressing principles. However, ISIS-SPB defines two further addresses for the propagation of LSPs besides the three addresses already reserved by ISO (see Table 8-14). ISIS-SPB differentiates itself from IS-IS for Ipv4 etc. by the contents of the SPB Hello PDUs. The most important differentiation is the NLPID value for SPB (0xC1) [Annex M] which is carried in the SPB Hello protocols supported TLV and used to decide if a given adjacency should support SPB. ISIS-SPB carries this value in the Protocols Supported TLV (129) of IS-IS.

SPB operates either stand alone within an IS-IS instance, or shares an IS-IS instance and its adjacencies with other protocols as distinguished by the appropriate Network Layer Protocol Identifiers (NLPIDs). In order to accomplish this, ISIS-SPB may as one option use the same control plane MAC addressing as the other NLPIDs.

SPB does not advertise its NLPID on shared media interfaces.

28.2 Formation and maintenance of ISIS-SPB adjacencies

ISIS-SPB uses the IS-IS three-way handshake for IS-IS point-to-point adjacencies described in IETF RFC 5303 to form adjacencies when they can be formed. This subclause specifies the conditions under which they may be formed, and the conditions under which the link may be used for forwarding after an adjacency has formed.

a) The initial exchange of SPB Hello PDUs between a pair of IS-IS instances advertises the NLPID value(s) supported by each bridge. If any NLPID values match, it is anticipated that an adjacency will be established to support the matched Network Layer Protocol(s).

This standard specifies subsequent behavior only for adjacencies which support SPB (NLPID value 0xC1). ISIS-SPB supports multiple topologies (MT), and if more than one topology is instantiated, the procedures below are applied independently to each topology instance.

b) The link supporting the adjacency is initialized to “SPB Operationally down.” This is signalled by setting the SPB Link Metric in the Link Metric Sub TLV (28.12.7) to the value (2^{24} – 1). All SPB Link Metric values < (2^{24} – 1) signify “SPB Operationally up,” and the value advertised is that to be used in path computation.
c) The SPB Hello PDUs exchange MCID and Auxiliary MCID (28.4) values to determine whether VLANs are consistently allocated to an operational mode. If a bridge detects that neither its MCID nor its Auxiliary MCID match the adjacency’s MCID or Auxiliary MCID, then the adjacency cannot be put into service, and no further tests are performed until an acceptable match has been achieved.
d) After an acceptable MCID match is achieved, configuration consistency for each Base VID in the topology is verified. The SPB Base VLAN-Identifiers sub-TLV (28.12.4) carried in SPB Hello PDUs carries, for each Base VID supported:
    - The Base VID value;
    - The ECT-ALGORITHM identifier (OUI/Index) for that Base VID;
    - The SPB Hello Use-Flag (1-bit), set if this bridge or any bridge in the SPT Region is using this Base VID for traffic;
    - The M-Bit (1-bit), which indicates SPBM mode or SPBV mode.

    Consistency is determined as follows:
    - If either the local end or the far end SPB Hello Use-Flag of the adjacency are set, then only if the respective Base VID, ECT-ALGORITHM and Mode bits match exactly is “matched” set for that Base VID.
    - If both local end and far end SPB Hello Use-Flags are clear, then “matched” is set (to allow maintenance of the adjacency during ECT Algorithm and mode reassignment).

Only if “matched” is true for all Base VIDs in the topology is the link supporting the adjacency set to “SPB Operationally up,” and usable for forwarding.

e) Once the link supporting the adjacency has advanced to “SPB Operationally up” as above, a bridge shall only modify the ECT Algorithm or Mode bit associated with any Base VID when both the local end and far end SPB Hello Use-Flags are clear. A bridge shall not modify the ECT Algorithm or Mode bit associated with any Base VID when either local or far end SPB Hello Use-Flag is set. To do so is indicative of a serious error condition.

f) If a bridge detects that an adjacent bridge has altered its advertisement of the ECT Algorithm or Mode bit associated with any Base VID while either the local or far end SPB Hello Use-Flags is set, it shall declare the link supporting the adjacency “SPB Operationally down.”

NOTE—If the error condition arises because of aberrant behavior by a single bridge, the condition described will be detected by all its adjacencies, and the behavior specified will cause the aberrant bridge to be isolated from the SPT Region. This is the least damaging mitigation of such a condition.

28.3 Loop prevention

The Group Address FDB entries for SPBV and SPBM are under the control of the ISIS-SPB database exchange. Group address entries can be assured to be loop-free when two neighboring bridges agree never to install potentially loop-forming state derived from an unsynchronized topology in the FDB (13.17, 6.5.4.1). This occurs by allowing ISIS-SPB to exchange link state messages and remove any older FDB entries that are no longer valid to any destination. Native IS-IS does not have a method for synchronizing the removal of state from the forwarding but it does have a method for synchronizing databases. ISIS-SPB augments this by using a digest of the database and adding the FDB condition. ISIS-SPB needs two modifications from regular IS-IS.

a) ISIS-SPB needs to remove any potential loop causing forwarding entries if its link state database is not synchronized with its neighbor bridges.
b) ISIS-SPB needs to synchronize with its neighbors bridges before installing Group Address FDB entries to prevent any potential loops.

The Agreement Protocol (13.17) specifies how this applies to ISIS-SPB.

28.4 The Agreement Digest

The Agreement Digest specified in this clause includes a Computed Topology Digest used to determine whether or not a neighboring bridge is operating with identical network topology information and thereby determine whether frames may be safely forwarded to the neighbor.

	Octet	Length
Agreement Digest Format Identifier	1	4 bits
Agreement Digest Format Capabilities	1	4 bits
Agreement Digest Convention Identifier	2	4 bits
Agreement Digest Convention Capabilities	2	4 bits
Agreement Digest Edge Count	3-4	2
Reserved (set to 0)	5-12	8
Computed Topology Digest	13-32	20

Figure 28-1—Agreement Digest field format

The Agreement Digest field comprises six elements:

- The Agreement Digest Format Identifier
- The Agreement Digest Format Capabilities
- The Agreement Digest Convention Identifier
- The Agreement Digest Convention Capabilities
- The Agreement Digest Edge Count
- The Computed Topology Digest

The Agreement Digest is carried as a structured 32-byte field in the ISIS-SPB Digest sub-TLV. Byte 1 is the most significant, Byte 32 is the least significant byte.These and all parameters carried in ISIS-SPB TLVs and sub-TLVs are encoded using the "big-endian" convention, in which lower numbered bits within each byte (those to the left) have higher significance.

28.4.1 Agreement Digest Format Identifier

The Agreement Digest Format Identifier (4 bits) is carried in the four most significant bits of Byte 1 of the Agreement Digest field. Its purpose is to allow alternative digests to be defined at some time in the future. This release of the standard defines Digest Format 0. All future upgrades to the standard protocol will require recipients to be able to receive and process digests of this format.

The Agreement Protocol (13.17) will not declare a topology match if the recipient does not support the Format value of this field, but otherwise it will operate as normal, so that the transmitter will receive an indication of the Digest Format Identifier and Capabilities of the receiver as a result of normal protocol operation.

28.4.2 Agreement Digest Format Capabilities

The Agreement Digest Format Capabilities (4 bits) is carried in the four least significant bits of Byte 1 of the Agreement Digest field. Its purpose is to allow alternative digest Formats to be defined and advertised at some time in the future. In this version of the standard it is transmitted as value 0 and ignored on receipt.

28.4.3 Agreement Digest Convention Identifier

The Agreement Digest Convention Identifier (4 bits) is carried in the four most significant bits of Byte 2 of the Agreement Digest field. Its purpose is to advertise the set of loop-free forwarding rules, for which the link state database information identified by the Digest acts as input, which are currently in use by the transmitter (13.17). The following rules are defined in this version of the standard:

— 1. indicates no digest match. Digests are exchanged but match is not required. Configured as off (Clause 17).
— 2. means the transmitter will continue loop-free forwarding of both multicast and unicast traffic either: only after any change which occurs has been reflected in matched digests (strict agreement); or up to the limits of change specified by the rules in this version of the standard. This is the default. Configured as loopFreeBoth (Clause 17).
— 3. means the transmitter will continue loop-free forwarding of multicast traffic up to the limits of change specified by the rules in this version of the standard, and will continue forwarding of unicast traffic unconditionally. Configured as loopFreeMCastOnly (Clause 17).

NOTE—While it is recommended that all bridges use the same convention, bridges operate independently and the conventions can be different on individual bridges. For loopFreeBoth and loopFreeMCastOnly as indicated, a bridge may block all the respective traffic on change or allow "safe" traffic during change. Individual bridges can have different definitions of "safety" applied to the forwarding allowed during digest mismatch.

28.4.4 Agreement Digest Convention Capabilities

The Agreement Digest Convention Capabilities (4 bits) is carried in the four least significant bits of Byte 2 of the Agreement Digest field. Its purpose is to advertise the set of loop-free forwarding rule conventions which are understood by the transmitter. The value 0 declares that the transmitter understands conventions 1 to 3 (above). Bridges conforming to this version of the standard transmit the value 0 and ignore this field on receipt.

28.4.5 Agreement Digest Edge Count

The Agreement Digest Edge Count (16 bits) is an unsigned integer carried in Byte 3 and Byte 4 of the Agreement Digest field. Its purpose is to provide one component of the Agreement Digest, which is simple to compute and powerful in detecting many simple topology mismatches.

This value is the sum modulo 2^{16} of all Edges in the SPB topology. Each point-to-point physical link is counted as two Edges, corresponding to its advertisement by ISIS-SPB in an LSP flooded from either end of the link.

If more that one ISIS-SPB topology is configured, the Agreement Digest Edge Count accumulates all Edges in all configured topologies. If Edges are absent in a particular topology, they have the value 0 for this summation.

NOTE—Multi-Topology SPB allows different Link Metrics to be used in different topologies. The Agreement Digest format specified here can only achieve a match between topologies when exactly the same set of bridges are present in all topologies but the requirement that MCIDs match for all bridges in an SPT Region means that this is not an additional constraint.

28.4.6 The Computed Topology Digest

The overall procedure for constructing the Computed Topology Digest is to:

— Form a signature of each Edge in the topology by computing the MD5 hash (IETF RFC 1321) of the significant parameters of the Edge, as defined below.
— Compute the Digest as the arithmetic sum of all Edges in the topology.

If more than one topology is declared in ISIS-SPB, the Computed Topology Digest covers all topologies declared, and this single value is advertised in the ISIS-SPB Digest sub-TLV. The SPB Digest sub-TLV is carried within the MT-Port-Cap TLV [IETF RFC 6165] with the MTID value of 0, which in turn is carried in an SPB Hello PDU.

NOTE 1—MD5 is widely reported to be cryptographically weak. This is not relevant in this application. What is required is a function exhibiting good avalanche properties such that signatures with potentially very similar input parameters have an infinitesimal probability of collision.

NOTE 2—This strategy allows the Digest to be incrementally computed when the topology changes, by subtracting the signatures of vanished Edges from the Digest and adding the signatures of new Edges. In general, the signature of an Edge therefore needs only to be computed once, when it is first advertised.

The input message to the MD5 hash for each Edge is constructed by concatenating the following fields in order, with the first field being the beginning of the message.

a) The Bridge Identifier (13.26.2) of the bridge advertising the Edge with the greater Bridge Identifier value (8 bytes).
b) The Bridge Identifier of the bridge advertising the Edge with the lesser Bridge Identifier value (8 bytes).
c) A variable number of 8-byte 3-tuples, one 3-tuple for each MTID (Clause 3) declared in ISIS-SPB. The 3-tuples are declared in descending order of MTID value, with the largest MTID declared first.

Each 3-tuple is constructed by concatenating the following fields in the order below:

d) A 2-byte field containing the 12-bit MTID value [IETF RFC 5120] in the least significant bits, with the 4 most significant bits of the field set to zero.

e) The SPB Link Metric [item c) in 12.25.6.1.2] for the Edge in this topology that has been advertised by the bridge with the greater Bridge Identifier value (3 bytes)
f) The SPB Link Metric for the Edge in this topology that has been advertised by the bridge with the lesser Bridge Identifier value (3 bytes)

If an Edge is not present in a topology, its SPB Link Metric is set to zero in that topology.

The value of the Computed Topology Digest is the arithmetic sum of all of the signatures returned by presenting every Edge message to MD5, treating the signature as an unsigned 16-byte integer and accumulating into a 20-byte integer. Every physical link is seen as two Edges, one advertised by each bridge comprising the adjacency, and formally the Computed Topology Digest includes both.

NOTE—Although the Computed Topology Digest contains signatures for both Edges associated with each link, the construction defined above means that these are always identical. An implementation may choose to compute a single signature per link, and then double it before accumulating it into the Computed Topology Digest.

The value of the Computed Topology Digest is placed in Bytes 13 to 32 of the Agreement Digest field for transmission in the ISIS-SPB Digest sub-TLV.

Bytes 5 to 12 of the Agreement Digest field are unused in this version of this standard. They are set to zero on transmission and ignored on reception.

28.5 Symmetric shortest path tie breaking

When performing shortest paths computation there may be several equal cost shortest paths between the same bridge pairs (27.17). In order to meet the congruency requirements, SPB requires that the computation produces a deterministic unique shortest path per SPT set independent of order or direction of computation.

The SPB symmetric (Clause 3) shortest path tie breaking algorithm needs to be accurately specified or different versions of SPB would compute different SPTs. SPB requires symmetry and congruence in its unicast and multicast routing so this section specifies unambiguously how the algorithms defined for the base specification operate and how they may be extended in the future.

NOTE—Exploration of path computation that produces various types of ECTs is beyond the scope of this standard.

SPB is designed to support many tie breaking algorithms for ECTs, and SPBM can assign traffic by I-SID to different B-VIDs assigned to different ECTs. SPB defines one required tie breaking algorithm and an open framework for the creation of a very large number of additional algorithms. As part of the base standard, a number pre-defined tie breaking algorithms are described using this framework with extensions for expansion in the future.

SPB accomplishes basic equal cost tie breaking by logically assigning a computation-order independent identifier to each possible equal cost shortest path that it computes. This tie breaking is symmetric because it is independent of which direction the selection is done and it preserves reverse path congruency. This is referred to as the Path Identifier (PATHID). A PATHID is defined as the sorted list (ascending lexicographic order) of Bridge IDs that the path traverses including the endpoints.

When there are different equal cost shortest paths output by the SPF computation, each of them will have a unique PATHID. The PATHID for each distinct path will be the same whichever bridge computes that PATHID. The PATHID will be the same for both the forward and reverse of a path (because they are sorted lists of Bridge-IDs). PATHIDs may be ranked and compared by successively comparing the sorted list of Bridge-IDs they contain. PATHID A is ranked “less than” PATHID B if it contains fewer Bridge-IDs:

E.g.: $\{9,15,22\} < \{7,8,9,10,22\}$

PATHID A is ranked “less than” PATHID B (of equal length) if its i’th Bridge-ID is less than B’s i’th Bridge-ID and all other Bridge-IDs up to the i’th are equal (lexicographic order).

E.g.: {9,15,22,99} < {9,15,22,100}

Both SPBV’s and SPBM’s default tie breaking algorithm picks the equal cost path with the lowest PATHID as defined in the above ranking.

The above algorithm can be easily implemented efficiently by simply back tracking the two or more competing equal cost paths and picking the path which has the minimum PATHID between the fork and join points of the competing paths. A path selected in this manner will also be part of any longer shortest paths which transit the fork and join points, and so intermediate resolution of shortest path permutations can be performed and the state for discarded permutations need not be carried forward.

The Bridge Priority by default is the middle value equal to 8 of the 4-bit priority in the most significant part of the 2-byte field [8 * 4096 = 32768 middle of the priority range] on all Bridges, and so does not affect the tie breaking results. Bridge Priority can be tuned by the operator and will therefore affect the pre-defined tie breaking algorithm(s) in a deterministic manner.

28.6 Symmetric ECT framework

Each tie breaking method is uniquely identified by an ECT Algorithm. The ECT Algorithm is a 32-bit number which contains an OUI and an index. This document specifies an initial set of SPB ECT Algorithms together with a framework for a large number of other algorithms. The OUI allows organizations to specify and manage their own algorithms and behaviors and to document them independently either through the IEEE, or through other SDOs, or to keep them proprietary/experimental should they desire. The different SPB ECT Algorithms defined in this document use the IEEE 802.1 OUI (00-80-C2) and Index values 1 through 16 while 1 additional non ECT spanning tree algorithm is defined with Index value 0.

The ECT Algorithm is unique for an SPT Set. One or more Base VIDs may be associated with an ECT Algorithm and this association is advertised both in link state packets to all other bridges in the SPT region, and also is exchanged in the SPB Hellos. A Base VID should not be used within the SPT region until all bridges agree on the relationship. Temporary disagreement between the Base VIDs assigned is expected and permitted between bridges as Base VIDs with new ECT Algorithms are configured. Disagreements are allowed when the Base VIDs is not in use. Base VIDs may share the same ECT Algorithm regardless of mode.

In SPBM, an I-SID is associated with a Base VID and thus by inference with an ECT Algorithm as well. This permits an operator to assign traffic to a certain ECT-ALGORITHM and to move it as required to achieve a degree of traffic balancing. In SPBM mode this Base VID is the B-VID directly.

In SPBV mode the Base VID shall be used to lookup and find the proper SPVID for the source under consideration during the setting of VLAN Registration Entries. A VLAN supported by SPBV can be forced to use the Spanning tree of the IST in cases where the bridges do not agree on the configuration or are short on SPVID resources.

Bridges must consider all SPT sets and populate forwarding even if they do not define a VID. An ECT Algorithm can be defined to operate using only data already advertised by ISIS-SPB (for example the Low and High PathID algorithms), or, it may require additional tuning parameters that cannot be foreseen. To accommodate a future ECT Algorithm need for tuning parameters SPB supports the concept of Opaque ECT data.

To allow for nodal and link based ECT behavior in future requirement the Opaque ECT data may be advertised under both SPB nodal and/or link data. The Opaque ECT data begins with the 32-bit ECT-ALGORITHM identifier but the remainder of the data format is specific to future ECT Algorithms and is therefore opaque to this document.

An example use of this framework could be as follows. Consider the tie breaking that can be done by using a minimum sum over a secondary link metric and a hash-based but deterministic assignment of such secondary metrics to every link in the network. Further consider these secondary link metrics as tunable. Initially a new value for this ECT Algorithm is assigned and registering it with the appropriate numbering authority for the OUI. The hash functions that create the secondary link metrics based for example on the two Bridge-IDs at each end of the link, could be defined. Then a format for a 32-bit opaque secondary metric could be defined which would be advertised as opaque data under the link data and prefixed with the ECT-ALGORITHM index for the new tie breaker. Migration to this new tie breaker would proceed in a manner consistent with any ECT Algorithm (28.9).

28.7 Symmetric ECT

In the case of multiple equal cost paths, multiple equal cost trees may be computed, in fact most efficient shortest path computations like Dijkstra actually produce all shortest paths from a root, i.e., an SPF tree, and different unique trees emerge depending on the tie breaking employed at the branch points. These different trees are distinguished in the forwarding plane by using SPVIDs (SPBV) or B-VIDs and source B-MACs (SPBM) for each "active topology". In order to be repeatable a unique tie breaker (ECT Algorithm) is chosen for each SPT set as described in (28.5). Typically a small number of B-VIDs would satisfy most requirements for equal cost routes in the case of SPBM.

SPBV and SPBM support a set of symmetric (Clause 3) equal cost paths between any pair of bridges for a given SPB instance/MTID. The algorithms are identified by the ECT Algorithm using the IEEE 802.1 OUI=00-80-C2 and with Index values 0..16. Index value 0 is somewhat special in that it relates the VID used for the CIST and is not a shortest path algorithm but is instead the spanning tree algorithm. The remaining algorithms are shortest path algorithms: The LowPATHID algorithm (index = 1) is the default SPT path computation tie breaker. SPB uses LowPATHID as the default SPT tie breaking algorithm. SPB can use any alternate tie breaking algorithm for another ECT when it is configured. The other defined algorithms use a computed shuffle of the LowPATHID algorithm. For example the HighPATHID ECT-ALGORITHM=<OUI=00-80-C2:index = 2> is just a rank inversion, which ones-complements the Bridge-IDs prior to doing the same comparisons as the LowPATHID algorithm. The remaining 14 pre-defined algorithms have indexes 3..16 and are defined in terms of a bit mask that they XOR the Bridge-IDs with prior to finding the minimum PATHID. Since they XOR over all 8 bytes, which includes the Bridge-Priority and the SPB System Identifier, these algorithms can be tuned in deterministic ways by adjusting the Bridge-Priority. SPBM may advertise a Base VID for each of these unique symmetric shortest paths through the ECT SPB Instance sub-TLV (28.12.5).

SPB assigns a unique ECT Algorithm to a Base VID in the Base VID sub-TLV (28.12.4) and the SPB Instance sub-TLV (28.12.5). To allow migration between different ECT Algorithms, further ECT Algorithm and Base VID bindings may be advertised, unused, without interfering with the operation of currently used ECT Algorithms and Base VIDs. Once bridges are configured to support the new ECT Algorithm and share a consistent Base VID for this algorithm, the algorithm may then be used by mapping I-SIDs to the Base VID for the desired ECT Algorithm.

SPBV also assigns a unique Base VID to each ECT Algorithm however the actual data path VID used depends on the source and therefore an SPVID shall be specified in the SPVID field in conjunction with the Base VID and ECT Algorithm in the SPB Instance sub-TLV. A similar procedure can be used with SPBV to introduce a Base VID that would not carry traffic until fully configured. Then services could be switched to the new VLAN. This is no different than introducing a new VLAN on RSTP for example.

28.8 Predefined ECT Algorithm details

Each of the predefined ECT-ALGORITHMs is formed using the OUI=00-80-C2 and the Index=1..16. An Algorithm Mask is specified for each Index, and is exclusive-ORed with each octet of each of the Bridge Identifiers (2 octet Bridge Priority concatenated with the 6 octet SPB System Identifier) to generate Masked Bridge Identifiers for each of the candidate paths. Given any two paths, the one chosen for a given ECT Algorithm has the lowest Path ID formed from the ranked masked Bridge Identifiers, which will necessarily include a lowest masked Bridge Priority, excluding those common to both paths. This comparison gives the greatest weight to the (masked) Bridge Priority of each Bridge Identifier, allowing the path selection to be managed/tuned without creating additional identifiers or identifier components for each bridge. The ECT-ALGORITHMs and corresponding Masks and their effects, in terms of prioritizing path selection by Bridge Priority are as follows:

Table 28-1—Bridge Priority Masking

ECT-ALGORITHM	Algorithm MASK	Effect of using MASK on priority nibble
00-80-C2-01	0x00	The selected path includes bridge with best (numerically lowest) Bridge Identifier after masking. In this case the masking 0x00 no effect and original administered bridge priority values apply. When bridge priority value is equal for two bridge identifiers the lower system identifier determines the priority. (0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15).
00-80-C2-02	0xFF	The selected path includes bridge with best priority (numerically lowest) Bridge Identifier after masking. In this case the masking 0xFF reverses the original administered bridge priority values. When bridge priority value is equal for two bridge identifiers the lower system identifier determines the priority. (15,14,13,12,11,10,9,8,7,6,5,4,3,2,1,0). Read vector as administered priority 0 equals 15, administered priority 15 equals 0 and so on.
00-80-C2-03	0x88	Bridges with Bridge Priority 8 will be treated as best priority for path selection, followed by Bridge Priorities of 9, 10, 11, and so on. (8,9,10,11,12,13,14,15,0,1,2,3,4,5,6,7). Read vector as administered priority 0 equals 8, administered priority 8 equals 0.
00-80-C2-04	0x77	Bridges with Bridge Priority 7 will be treated as best priority for path selection, followed by Bridge Priorities of 6, 5, 4, and so on. (7,6,5,4,3,2,1,0,15,14,13,12,11,10,9,8). Read vector as administered priority 0 equals 7, administered priority 7 equals 0.
00-80-C2-05	0x44	(4,5,6,7,0,1,2,3,12,13,14,15,8,9,10,11)
00-80-C2-06	0x33	(3,2,1,0,7,6,5,4,11,10,9,8,15,14,13,12)
00-80-C2-07	0xCC	(12,13,14,15,8,9,10,11,4,5,6,7,0,1,2,3)
00-80-C2-08	0xBB	(11,10,9,8,15,14,13,12,3,2,1,0,7,6,5,4,)
00-80-C2-09	0x22	(2,3,0,1,6,7,4,5,10,11,8,9,14,15,12,13)
00-80-C2-0A	0x11	(1,0,3,2,5,4,7,6,9,8,11,10,13,12,15,14)
00-80-C2-0B	0x66	(6,7,4,5,2,3,0,1,14,15,12,13,10,11,8,9)
00-80-C2-0C	0x55	(5,4,7,6,1,0,3,2,13,12,15,14,9,8,11,10)
00-80-C2-0D	0xAA	(10,11,8,9,14,15,12,13,2,3,0,1,6,7,4,5)
00-80-C2-0E	0x99	(9,8,11,10,13,12,15,14,1,0,3,2,5,4,7,6)
00-80-C2-0F	0xDD	(13,12,15,14,9,8,11,10,5,4,7,6,1,0,3,2)
00-80-C2-10	0xEE	(14,15,12,13,10,11,8,9,6,7,4,5,2,3,0,1)

NOTE 1—In these algorithms, the Algorithm Mask is applied to all octets of each Bridge Identifier. Thus if two alternative paths include only bridges with one or a few Bridge Priorities, the Algorithm Mask will still reorder their priorities for inclusion on the path as it acts on the rest of the SPB System Identifier component of the Bridge Identifier field.

NOTE 2—If the two priority bytes in the 8-byte Bridge Identifier are all equal, then all bridges have equal priority and ECT-ALGORITHMs 00-80-C2-01 and 00-80-C2-02 are exactly Low and High PATHID as described in 28.5.

28.9 ECT Migration

For orderly migration, each Base VID to ECT Algorithm assignment also has associated flags indicating active usage of the Base VID by services. The Use-Flag in the SPB Base VLAN-Identifiers sub-TLV (28.12.4) indicates a bridge's awareness of use of the Base VID somewhere in the SPT Region, and the U-bit in the SPB Instance sub-TLV (28.12.5) indicates local use of this Base VID. The Use-Flag in the SPB Base VLAN-Identifiers sub-TLV is carried in the SPB Hello PDU, and is set once the bridge advertising the algorithm has started either using or seeing the use of that ECT-ALGORITHM. ISIS-SPB enforces the following rules with respect to ECT Algorithms and their usage in order to protect the SPB network from outages created by incorrect configuration. If neighbors disagree on a Base VID to ECT-ALGORITHM assignment, then use of the adjacency for SPB traffic is permitted (i.e., the SPB Link Metric may be set less than $2^{24} – 1$) only if both neighbors' Use-Flags associated with the Base VID are cleared (28.2).

If a new ECT Algorithm is implemented in all bridges of an SPT Domain and intended to be used by a Base VID, then the assignment of the Base VID to the new ECT-ALGORITHM with Use-Flag cleared is first configured in the SPT Domain on a bridge by bridge basis. As soon as this configuration is complete, services (I-SIDs in SPBM or SPVIDs in SPBV) may be assigned to the given Base VID, which causes the local U-bit to be set, which will in turn cause the Use-Flag to be set for the given Base VID.

SPBV and SPBM have different migration capabilities after the introduction of a new ECT Algorithm. The following subclauses describe how SPBV and SPBM use a newly introduced ECT Algorithm.

28.9.1 Use of a new ECT Algorithm in SPBV

A new ECT Algorithm, which is implemented by each bridge of an SPT Domain, gets to be used if it is configured to support a VLAN. In SPBV mode, the application of a new ECT Algorithm typically coincides with the introduction of a new VLAN supported by SPBV. That is, the Base VID of the new VLAN is assigned to the new ECT-ALGORITHM in the SPB Hello PDU Base VID and SPB Instance sub-TLVs (28.9). ISIS-SPB then allocates (27.10) the SPVIDs necessary for the support of the new VLAN. To support in-service upgrade from one set of VID / ECT Algorithm allocations to another, SPT Bridges allow graceful configuration of those services. Provided there is an operating SPBV or SPBM VLAN in a region, SPT bridges can be configured within the Region for a new VLAN on a bridge by bridge basis. As long as the SPT Region is connected the bridges supporting the new VLAN do not need to be directly connected. Intermediate bridges are capable of populating forwarding without terminating services allowing them to participate in the VLAN.

The migration of a VLAN from an old ECT Algorithm to a new ECT Algorithm can be performed by the reassignment of the Base VID of the VLAN. That is, the Base VID is assigned to the new ECT-ALGORITHM instead of the old ECT-ALGORITHM on a bridge by bridge basis. In SPBV the forcing of one of the bridges Base-VIDs to use the spanning tree algorithm as opposed to another algorithm forces all of the bridges in the SPT region using that Base VID to Spanning tree. Migration can then proceed to the new Base VID assignment until the last bridge is configured to use the new ECT-ALGORITHM. The new assignment is then propagated in SPB Hello PDU Base VID and SPB Instance sub-TLVs (28.9) by ISIS-SPB within the SPT Domain. As soon as each SPT Bridge is configured with the Base VID assignment to the new ECT-ALGORITHM, the Spanning tree assignment on the last bridge is cleared.

NOTE 1—ISIS-SPB automatically performs the allocation and release of SPVIDs during the migration. The SPVIDs allocated for the Base VID old ECT-ALGORITHM assignment are released as soon as the assignment has been broken. These SPVIDs might be then allocated for the Base VID new ECT-ALGORITHM assignment. That is, there is no need for additional SPVID allocations if a Base VID supported by SPBV is migrated from an old ECT-ALGORITHM to a new one.

NOTE 2—SPBV does not support hitless migration. That is, the migration of a VLAN from an old ECT-ALGORITHM to a new ECT-ALGORITHM may cause traffic outage.

28.9.2 Use of a new ECT Algorithm in SPBM

SPBM offers the ability to map I-SIDs to Base VIDs as a load balancing option. In topologies with multiple equal cost paths the ability to spread traffic on a per I-SID basis comes from using the Base VIDs with different algorithms as specified in this clause.

When an I-SID is configured to be associated with a Base VID, that Base VID shall be consistently associated with the same ECT-ALGORITHM throughout the SPT Domain. Once at least one I-SID has been assigned to a Base VID the U-bit is set and advertised in subsequent LSPs, which also causes the Use-Flag to be set in SPB Hello PDUs.

SPBM supports in-service migration of an I-SID from an old ECT Algorithm to a new ECT Algorithm supported by each bridge of an SPT Domain. In order to accomplish this, two complete Base VIDs must be configured in the SPT Domain. Then an I-SID under migration can be assigned to two Base VIDs at the same time: the old Base VID (old ECT-ALGORITHM), and the new Base VID (new ECT-ALGORITHM). The U-bit (in the SPB Instance sub-TLV, 28.12.5) is set by the advertising bridge when there is at least one service using this VID. This propagates throughout the SPT domain, and any bridge seeing at least one U-bit set in received LSPs (signifying that the indicated Base VID is being used by a service somewhere) sets the Use-Flag in its SPB Base VLAN-Identifiers sub-TLV (28.12.4).

I-SIDs will have originally been configured to be associated with the old Base VID (28.12.10). No filtering entry is set for the new Base VID in the FDBs as there is no Transmitter set on the new Base VID yet. I-SIDs can be configured to be associated with the new Base VID. Care must be taken to ensure the new Base VID is configured in all the required places. When a particular I-SID has been added to a new Base VID, ISIS-SPB detects that the I-SID is now associated with two Base VIDs and at each location that I-SID may receive traffic on either Base-VID. During this period individual addresses are available under both Base VIDs. The setting of the R and T bits still determines whether the group addresses are populated for each bridge supporting that I-SID in either Base VID. In other words the same rules for a single Base VID are followed but two base VIDs are temporarily in use for received traffic. Traffic is “rolled” from the old Base VID to the new one by administrative reassignment of the I-SID at each service end-point from the old to the new B-VID. This can be an atomic action, and so at any instant only one Base VID will be used by any service originating frames. The association of the I-SID to the old Base VID is removed once all I-SIDs are configured to one Base VID. Furthermore, if this is the last I-SID removed from a Base VID the U-bit is cleared (in the SPB Instance sub-TLV) for the old Base VID to ECT-ALGORITHM assignment. Eventually the Use-Flag carried in the SPB Hello PDU will be cleared when all the bridges have been configured.

28.10 MAC Address registration

ISIS-SPB creates MAC Address Registration Entries (8.8.4) for Group Addresses in order to reach the members of a group. The Group Addresses are locally administered in case of SPBM (27.16), thus, they are only significant inside the SPT Region.

ISIS-SPB also installs MAC Address Registration Entries for universally administered Group Addresses as they might be used by SPBV. Furthermore, ISIS-SPB provides interworking with MMRP at the Boundary Ports of an SPT Region. That is, ISIS-SPB interprets MMRPDUs received at a Boundary Port and assembles the proper ISIS-SPB TLVs in order to advertise the corresponding MAC address registration inside the SPT Region. Furthermore, ISIS-SPB assembles and issues MMRPDUs on Boundary Ports if MAC address registration needs to be propagated outside of the SPT Region.

When an MMRPDU is received on a Boundary Port, ISIS-SPB disseminates the registration by sending LSPs with an SPBV MAC Address sub-TLV (28.12.9) filled according to the registration received from the MMRPDU. The MAC Address and Base VID values shall be copied from the MMRPDU, with the R bit set and T bit cleared. The SR bits (28.12.9) are set according to the Group service requirement information carried in the MMRPDU. When processing received SPBV MAC Address sub-TLVs which have the R bit set, SPT Bridges at the boundary of an SPT Region shall check whether they should filter or forward the frames received on their Boundary Port in the VLAN for which the registration applies. If the option is to forward frames, then they also send the SPBV MAC Address sub-TLV into the SPT Region for the MAC Address under registration; but set the T bit, clear the R bit and for the VID value use the SPVID allocated to the given VLAN. ISIS-SPB then creates MAC Address Registration Entries inside the SPT Region based on the SPBV MAC Address sub-TLVs. Furthermore, if the Boundary Port is in the given VLAN's member set, then ISIS-SPB assembles the proper MMRPDU and issues it on the Boundary Port.

Besides interworking with MMRP at Boundary Ports, ISIS-SPB also takes into account the Registrar Administrative Control parameters (10.7.2).

NOTE—Creation of unnecessary MAC Address Registration Entries can be prevented by using the "Registration Forbidden" parameter for the proper Ports and VLANs.

28.11 Circuit IDs and Port Identifiers

IS-IS allows the formation of multiple adjacencies between two routers. ISIS-SPB supports only a single logical adjacency per MTID. Link aggregation allows a single ISIS-SPB adjacency to cover multiple links, and this is the only method defined in this version of this standard by which multiple links between neighbors are supported. In cases where multiple potential adjacencies exist, each pair of peers using ISIS-SPB should choose the link with the lowest Local Circuit ID on the bridge with the better (numerically lower) SPB System Identifier as the link over which to send and receive frames.

Procedures for using multiple adjacencies are not specified by this standard.

28.12 ISIS-SPB TLVs

All SPB TLVs and sub-TLVs are described together in this section. SPBV and SPBM share most of the same TLVs. Multi-Topology (MT) [Annex M] is introduced as well, but multiple SPT instances, each supporting a different VLAN, and each using either SPBV or SPBM can be described in a single topology instance. Multiple topology instances allow different topologies to be advertised, for example if there is a desire to use different metrics since there is only a single metric scheme within a topology.

SPB adds one new TLV and several new sub-TLVs that can be added to IS-IS PDUs, as described below. The new ISIS-SPB data have been organized to support multiple topologies within a single IS-IS instance. Each of the new sub-TLV's can be included in a TLV containing a topology instance identifier (MTID) value (as per [MT]) so that the computations can operate only on sub-TLV's specific to their topology instance. The intent of MT is to support the use of multiple metric sets when required. The construction of the MCID and the Agreement Digest (13.17, 28.4) requires that all SPT bridges in an SPT region appear in all Multi-Topology instances.

All parameters carried in the ISIS-SPB (sub-)TLVs defined below are encoded using the "big-endian" convention, in which higher significance bytes and bits within each byte appear to the left of and above bytes and bits of lower significance.

28.12.1 MT-Capability TLV

The Multi-Topology Capability (MT-Capability) TLV [Annex M] is a top level TLV for Link State PDUs that provides Multi-Topology context. It identifies the MTID for sub-TLVs in LSPs.

	Octet	Length
Type (144)	1	1
Length (2)	2	1
Overload Bit	3	1 bit
reserved	3	3 bits
MTID	3-4	12 bits

Figure 28-2—MT-Capability TLV

a) Type (8 bits) Value 144

b) Length (8 bits)

Total number of bytes contained in the value field.

c) Overload Bit (1 bit)

The overload bit is an ISIS-SPB MT instance specific overload bit.

d) MTID (12 bits)

This is used to provide the ISIS-SPB Multi-Topology context for SPB sub-TLVs. This should be 0 when there is only one topology instance.

28.12.2 SPB MCID sub-TLV

This sub-TLV shall be included in an MT-Port-Cap TLV [IETF RFC 6165] with MTID equal to 0 in SPB Hello PDUs to announce the bridge's MST Configuration Identifier (MCID) (13.8). This information should be the same on all bridges in the topology being controlled by this ISIS-SPB instance. The MCID is controlled solely by configuration and is a digest of the protocol assignments for all possible VIDs. Two MCIDs are carried to allow transitions between configurations when the changes are non-critical. If neither MCID advertised by a neighbor matches either MCID on this bridge the adjacency is a Region boundary.

	Octet	Length
Type (4)	1	1
Length (102)	2	1
MCID	3-53	51
Aux MCID	54-104	51

Figure 28-3—SPB MCID sub-TLV

a) Type (8 bits) Value 4

b) Length (8 bits)

Total number of bytes contained in the value field.

c) MCID (51 bytes)

The complete MCID defined in (13.8) that identifies an SPT Region.

d) AUX-MCID (51 bytes)

In the case of migration and where the MCID will change but in a non-critical way, this MCID contains the previous advertized MCID. The neighbor will not drop an adjacency if at least one of these MCIDs matches either of its MCIDs.

28.12.3 SPB Digest sub-TLV

This sub-TLV shall be included in an MT-Port-Cap TLV with MTID 0 in SPB Hello PDUs. The ISIS-SPB Agreement Digest (13.26.1, 28.4) is computed based on current topology and it changes when significant topology changes. The Agreement Digest is a key input into the Agreement Protocol (13.17), which provides multicast loop prevention. During the propagation of LSPs the Agreement Digest will vary between neighbors until the LSPs are common. The digest is a summarized means of determining agreement between nodes on the distance to all multicast roots, hence is essential for loop prevention. For each shortest path tree where it has been determined the distance to the root has changed, unsafe multicast forwarding is blocked until the exchanged digests match.

	Octet	Length
Type (5)	1	1
Length (33)	2	1
reserved	3	3 bits
V	3	1 bit
A	3	2 bits
D	3	2 bits
Agreement Digest	4-35	32

Figure 28-4—SPB Digest sub-TLV

a) Type (8 bits) Value 5

b) Length (8 bits)
Total number of bytes contained in the value field.

c) V (1 bit)
The Agreed Digest Valid Bit (13.27.7, 13.27.9). This bit is populated and read from the Agreement Logic.

d) A (2 bits)
The agreement number 0-3, which aligns with BPDUs agreement number concept (13.27.11). When the Agreement Digest for this node changes this number is updated and sent in the SPB Hello.

e) D (2 bits)
The discarded agreement number 0-3, which aligns with BPDUs agreement number concept (13.27.12). When the Agreement Digest for this node changes this number is updated. Once an Agreement has been sent it is considered outstanding until a matching or more recent Discarded Agreement Number is received.

f) Agreement Digest (32 bytes)
An Agreement Digest as described in 28.4.

28.12.4 SPB Base VLAN-Identifiers sub-TLV

This sub-TLV is included in an MT-Port-Cap TLV in an SPB Hello PDU to indicate the ECT Algorithms for the Base VIDs (and by implication the VID(s) used on the forwarding path for each SPT Set for a VLAN identified by a Base VID) that are in use. This information should be the same on all bridges in the topology identified by the MT-Port-Cap TLV in which it is carried. Discrepancies between neighbors with respect to this sub-TLV are temporarily allowed during reconfiguration but at all times the active Base-VIDs (as determined by the state of the Use-Flag carried in this sub-TLV) shall agree and use the same ECT-ALGORITHM.

In the case of SPBM, the Base VID is the B-VID used to forward packets. In the case of SPBV, each source uses a different SPVID and a Base VID is used for frames transmitted on the IST. One or more Base VIDs is associated with an ECT Algorithm. This structure supports multiple SPT sets within an IS-IS topology instance for both SPBV and SPBM.

		Octet	Length
	Type (6)	1	1
	Length (6n)	2	1
ECT-VID Tuple 1	ECT Algorithm	3-6	4
	Base VID	7-8	12 bits
	U	8	1 bit
	M	8	1 bit
	reserved	8	2 bits
	...		
ECT-VID Tuple n	ECT Algorithm	(6n-3)-6n	4
	Base VID	(6n+1)-(6n+2)	12 bits
	U	6n+2	1 bit
	M	6n+2	1 bit
	reserved	6n+2	2 bits

Figure 28-5—SPB Base VLAN-Identifiers sub-TLV

a) Type (8 bits) Value 6

b) Length (8 bits)

The size of the value is ECT-VID Tuples*6 bytes. Each 6-byte part of the ECT-VID tuple is formatted as described below.

c) ECT-ALGORITHM (4 bytes)

The ECT-ALGORITHM is advertised when the bridge supports a given ECT-ALGORITHM (by OUI/Index) on a given Base VID.

d) Base VID (12 bits)

The Base VID that is associate with the SPT Set.

e) Use-Flag (1 bit)

The Use-Flag is set if this bridge, or any bridge in the SPT Region is currently using this ECT-ALGORITHM and Base VID. The Use-Flag is calculated as the logical OR of the U-bit values of all bridges in the region, as found in their SPB Instance sub-TLV (28.12.5) in the link state database. This definition includes the U-bit value of this bridge.

f) M-Bit (1 bit)

The M-bit indicates if this is SPBM when set to 1 or SPBV mode when set to 0.

g) 2 Reserved Bits set to 0.

28.12.5 SPB Instance sub-TLV

This sub-TLV must be carried within an MT-Capability TLV in the fragment ZERO LSP. It identifies the bridge uniquely and identifies the ECT-ALGORITHMs supported by the bridge and the Base VIDs and SPVIDs assigned to those algorithms. For SPBM, only the Base VID is valid and the SPVID is set to zero. In the case of SPBV, the Base VID is associated with the SPVID used for forwarding by the bridge originating the TLV. There may be multiple ECT-ALGORITHMs specifying a number of ECTs.

		Octet	Length
	Type (1)	1	1
	Length	2	1
	CIST Root Identifier	3-10	8
	CIST External Root Path Cost	11-14	4
	Bridge Priority	15-16	2
	reserved	17-18	11 bits
	V	18	1 bit
	SPSourceID	18-20	20 bits
	Number of Trees	21	1
VLAN ID Tuple 1	U	22	1 bit
	M	22	1 bit
	A	22	1 bit
	reserved	22	5 bits
	ECT Algorithm	23-26	4
	Base VID	27-28	12 bits
	SPVID	28-29	12 bits
	...		
VLAN ID Tuple n	U	8n+14	1 bit
	M	8n+14	1 bit
	A	8n+14	1 bit
	reserved	8n+14	5 bits
	ECT Algorithm	(8n+15)- (8n+18)	4
	Base VID	(8n+19)- (8n+20)	12 bits
	SPVID	(8n+20)- (8n+21)	12 bits

Figure 28-6—SPB Instance sub-TLV

a) Type (8 bits) Value 1

b) Length (8 bits)

Total number of bytes contained in the value field.

c) CIST Root Identifier (64 bits)

The CIST Root Identifier is for SPB interworking with RSTP and MSTP at SPT Region Boundaries. This is an imported value from a Spanning tree.

d) CIST External Root Path Cost (32 bits)

The CIST External Root Path Cost is the cost from the Spanning tree algorithm to the Root.

e) Bridge Priority (16 bits)

Bridge priority is the 16 bits that together with the low 6 bytes of the SPB System Identifier form the Bridge Identifier. The Bridge Identifier is the Spanning tree compatible Bridge identifier. This is configured exactly as specified in IEEE Std 802 (IEEE Std 802.1D). This allows SPB to build a compatible Spanning tree using link state by combining the Bridge

Priority and the SPB System Identifier to form the 8-byte Bridge Identifier. The 8-byte Bridge Identifier is also the input to the 16 pre defined ECT tie breaker algorithms.

f) V bit (1 bit)

The V bit (SPBM) indicates this SPSourceID is auto allocated (27.10). If the V bit is clear the SPSourceID has been configured and shall be unique. When a bridge joining an SPT Region has formed at least one SPB adjacency, it can discover the previously allocated SPsourceIDs and it will allocate a SPSourceID according to the allocation logic (27.10).

g) SPSourceID (20 bits)

The SPSourceID (SPBM) is a 20 bit value used to construct Group MAC Address (27.15) for multicast packets originating from the origin (SPT bridge) of the link state packet (LSP) that contains this sub-TLV. SPSourceID may be 0 if it has not been allocated (27.10) or if there is no SPBM service configured. When SPSourceID is 0 all information pertaining to this MTID SPB instance will be distributed but not acted upon until a valid SPSourceID is populated. If the SPSourceID conflicts with another SPSourceID from another Bridge the tie breaker rules determine the winning bridge (27.10).

h) Number of Trees (8 bits)

The Number of Trees is set to the number of VLAN ID tuples that follow (minimum one). The following seven fields make up a VLAN ID tuple. A sequence of VLAN ID tuples can occur in any order.

1) U-Bit (1 bit)

The U-bit is set if this bridge is currently using this ECT-ALGORITHM for I-SIDs it sources or sinks or the SPVIDs it sources. This is importantly different from the Use-Flag found in the SPB Hello, which is set if a bridge sees other nodal U-bits are set OR it sources or sinks itself.

2) M-Bit (1 bit)

The M-bit indicates if this is SPBM when set to 1 or SPBV mode when set to 0.

3) A bit (1 bit)

The A bit (SPB) when set declares this is an SPVID with auto allocation (27.10). If the SPVID value is zero, VID will be allocated once the bridge has synchronized the IS-IS LSPs. Neighbor bridges can distribute the LSPs but shall not populate filtering databases (forwarding) for traffic from a bridge that has an SPVID of 0. When the bridge allocating is synchronized with the IS-IS adjacency, it will allocate one or more SPVIDs according to the allocation logic (27.10).

4) Reserved (5 bits)

Five bits reserved for future use shall be set to 0.

5) ECT-ALGORITHM (4-bytes)

ECT-ALGORITHM is advertised when the bridge supports a given ECT-ALGORITHM (by OUI/Index) on a given VID. This declaration shall match the declaration in the SPB Hello PDU originating from the same bridge. When migrating from one algorithm to another discrepancies between algorithm and Base VID on different bridges are allowed if the Use-Flag is clear.

6) Base VID (12 bits)

The Base VID that associated the SPT Set via the ECT-ALGORITHM.

7) SPVID (12 bits)

The SPVID is the Shortest Path VID when using SPBV mode. In SPBM mode it is not used and should be set to 0.

28.12.6 SPB Instance Opaque ECT Algorithm sub-TLV

There are multiple ECT Algorithms defined for SPB, and in the future additional algorithms may be defined. These algorithms may use this optional sub-TLV to carry new algorithm parameters, for example tie breaking data. There are two broad classes of ECT Algorithm: one which uses nodal data to break ties and one which uses link data to break ties. This sub-TLV can associate opaque data with a node. This sub-TLV is carried in an MT-Capability TLV in a Link State PDU (along with a valid SPB Instance sub-TLV). Multiple copies of this sub-TLV may be carried for different ECT-ALGORITHMs related to the node.

	Octet	Length
Type (2)	1	1
Length	2	1
ECT-ALGORITHM	3-6	4
ECT Information	7-(Length+2)	variable

Figure 28-7—SPB Instance Opaque ECT-ALGORITHM sub-TLV

a) Type (8 bits) Value 2

b) Length (8 bits)
Total number of bytes contained in the value field.

c) ECT-ALGORITHM (4 bytes)
ECT-ALGORITHM is advertised when the bridge supports a given ECT-ALGORITHM (by OUI/Index) on a given VID.

d) ECT Information (variable)
ECT-ALGORITHM Information of variable length.

28.12.7 SPB Link Metric sub-TLV

This sub-TLV is included in an Extended IS Reachability TLV (type 22) [IETF RFC 5305] or MT Intermediate Systems TLV (type 222) [IETF RFC 5120]. If this sub-TLV is not present for an IS-IS adjacency then that adjacency must not carry SPB traffic for the given topology instance. The maximum metric value ($2^{24} – 1$) signifies that the link shall not be used for SPB traffic (it is to treated as “SPB operationally down”).

	Octet	Length
Type (29)	1	1
Length (6)	2	1
SPB Link Metric	3-5	3
Number PORTs	6	1
Port Identifier	7-8	2

Figure 28-8—SPB Link Metric sub-TLV

a) Type (8 bits) Value 29

b) Length (8 bits)
Total number of bytes contained in the value field.

c) SPB Link Metric (24 bits)
The indicates SPB Link Metric the administrative cost or weight of using this link as a 24 bit unsigned number. Smaller numbers indicate lower weights and are more likely to carry SPB traffic. Only one metric is allowed per SPB instance per link. If multiple metrics are required multiple SPB instances are required, either within IS-IS or within several independent IS-IS instances. If there are multiple links the metric for all links are the same.

d) Number PORTs (8 bits)
Number PORTs is the number of Port identifiers associated with this link.

e) Port Identifier (16 bits)
Port Identifier is the unique Port identifier comprising two parts, the Port Number and the Port Priority field (13.27.47 of this standard, 17.18.16 of IEEE Std 802.1D).

28.12.8 SPB Adjacency Opaque ECT Algorithm sub-TLV

There are multiple ECT Algorithms defined for SPB, and in the future additional algorithms may be defined. The SPB Adjacency Opaque ECT Algorithm sub-TLV may be included in an Extended IS Reachability TLV (type 22) or MT Intermediate System TLV (type 222) to carry new algorithm parameters associated with an adjacency. Multiple copies of this sub-TLV may be carried for different ECT Algorithms related to the adjacency.

	Octet	Length
Type (30)	1	1
Length	2	1
ECT-ALGORITHM	3-6	4
ECT Information	7-(Length+2)	variable

Figure 28-9—SPB Adjacency Opaque ECT-ALGORITHM sub-TLV

a) Type (8 bits) Value 30

b) Length (8 bits)
Total number of bytes contained in the value field.

c) ECT-ALGORITHM (4 bytes)
ECT-ALGORITHM is advertised when the bridge supports a given ECT-ALGORITHM (by OUI/Index) on a given VID.

d) ECT Information (variable)
ECT-ALGORITHM Information of variable length.

28.12.9 SPBV MAC address sub-TLV

The SPBV MAC Address sub-TLV is carried in an MT-Capability TLV in a Link State PDU. It should be used for advertisement of Group MAC Addresses in SPBV mode. Individual MAC addresses will normally be distributed by reverse path learning, but carrying them in this sub-TLV is not precluded. It has the following format:

		Octet	Length
	Type (4)	1	1
	Length	2	1
	reserved	3	2 bits
	S-R	3	2 bits
	SPVID	3-4	12 bits
MAC Address Tuple 1	T	5	1 bit
	R	5	1 bit
	reserved	5	6 bits
	MAC Address	6-11	6
	...		
MAC Address Tuple n	T	(7n-2)	1 bit
	R	(7n-2)	1 bit
	reserved	(7n-2)	6 bits
	MAC Address	(7n-1)- (7n+4)	6

Figure 28-10—SPBV MAC Address sub-TLV

a) Type (8 bits) Value 4

b) Length (8 bits)

Total number of bytes contained in the value field.

c) SR bits (2 bits)

The SR bits are the service requirement parameter from MMRP. The service requirement parameters have the value 0 (Forward all Groups) (10.12.1.7) and 1 (Forward All Unregistered Groups) defined. However this attribute may also be missing. So the SR bits are defined as 0 not declared, 1 Forward all Groups and 2 Forward All Unregistered Groups.

d) SPVID (12 bits)

This is the SPVID and by association Base VID and the ECT-ALGORITHM and SPT Set that the MAC addresses defined below will use. If the SPVID is not allocated the SPVID Value is 0.

NOTE—If the ECT Algorithm in use is Spanning Tree Algorithm this value should be populated with the Base VID.

e) T Bit (1 bit)

This is the Transmit allowed Bit for the following group MAC address. This is an indication that SPBV Group MAC Address with SPVID of source should be populated (for the bridge advertising this Group MAC), and installed in the FDB of transit bridges, when the bridge computing the trees is on the corresponding ECT-ALGORITHM shortest path between the bridge advertising this Group MAC address with the T bit set, and any receiver of this Group MAC Address. A bridge that does not advertise this bit set for an Group MAC Address should have no forwarding state installed for traffic originating from that bridge on other transit bridges in the network.

f) R Bit (1 bit)

This is the Receive allowed Bit for the following Group MAC Address. This is an indication that SPBV Group MAC Addresses as receiver should be populated (for bridges advertising this Group MAC Address with the T bit set) and installed when the bridge computing the trees lies on the corresponding shortest path for this ECT-ALGORITHM between this receiver and any transmitter on this Group MAC Address. An entry that does not have this bit set for an Group MAC Address is prevented from receiving on this Group MAC Address because transit bridges will not install multicast forwarding state towards it in their FDBs or the traffic is explicitly filtered.

g) MAC Address (48 bits)

The MAC Address is either a group address or an individual address. When the MAC address is a group address it declares this bridge as part of the multicast interest for this destination MAC address. Multicast trees can be efficiently constructed for destination by populating Group Address FDB entries for the subset of the shortest path tree that connects the bridges supporting the multicast address. This replaces the function of MMRP for SPTs. The T and R bits above have meaning if this is a group address. Individual addresses are populated only as if the R bit was set. (Note that individual addresses could also be learned normally.)

28.12.10 SPBM Service Identifier and Unicast Address (ISID-ADDR) sub-TLV

This sub-TLV declares an individual B-MAC address and maps I-SIDs in the context of a B-VID to that B-MAC, allowing automatic creation of efficient group trees that are subsets of the SPT. In an ECT environment, the I-SIDs are mapped to B-VID, which is associated with an ECT-ALGORITHM specifying the SPT Set. It is carried in an MT-Capability TLV in a Link State PDU.

		Octet	Length
	Type (3)	1	1
	Length	2	1
	B-MAC Address	3-8	6
	reserved	9	4 bits
	Base VID	9-10	12 bits
I-SID Tuple 1	T	11	1 bit
	R	11	1 bit
	reserved	11	6 bits
	I-SID	12-14	3
	...		
I-SID Tuple n	T	(4n+7)	1 bit
	R	(4n+7)	1 bit
	reserved	(4n+7)	6 bits
	I-SID	(4n+8)- (4n+10)	3

Figure 28-11—SPBM Service Identifier and Unicast Address sub-TLV

a) Type (8 bits) Value 3

b) Length (8 bits)

Total number of bytes contained in the value field.

c) B-MAC Address (48 bits)

B-MAC ADDRESS is an Individual MAC Address on this bridge. It may be either the single nodal address, or may address a port or any other level of granularity relative to the bridge. In the case where the bridge only has one B-MAC address this may be the same as the MAC address portion SPB System Identifier of the bridge. To add multiple B-MACs this sub-TLV shall be repeated for each additional B-MAC.

Each Individual MAC Address has one or more I-SIDs that are associated with this address. In PBB terms this corresponds to a PIP B-MAC. All B-MACs that are used for Individual MAC addresses and for the Sources of Group MACs shall be associated with at least one I-SID. This sub-TLV is initiated by IB-BEBs.

d) Base VID (12 bits)

The Base VID identifies the B-VID and by association the ECT-ALGORITHM and SPT Set that the I-SIDs defined below will use.

e) T Bit (1 bit)

This is the Transmit allowed Bit for the following I-SID. This is an indication that Group MAC Addresses for this I-SID as source should be constructed and installed (for the bridge advertising this I-SID), and installed in the FDB of transit bridges, when the bridge computing the trees is on the corresponding ECT-ALGORITHM shortest path between the bridge advertising this I-SID with the T bit set, and any receiver of this I-SID. A bridge that does not advertise this bit set for an I-SID should have no forwarding state installed for traffic originating from that bridge on other transit bridges in the network.

f) R Bit (1 bit)

This is the Receive allowed Bit for the following I-SID. This is an indication that Group MAC Addresses for this I-SID as receiver should be constructed and installed (for bridges advertising this I-SID with the T bit set) and installed when the bridge computing the trees lies on the corresponding shortest path for this ECT-ALGORITHM between this receiver and any transmitter on this I-SID. An entry that does not have this bit set for an I-SID is prevented from receiving on this Group MAC Address because transit bridges will not install multicast forwarding state towards it in their FDBs.

g) I-SID (24 bits)

I-SID is the 24 bit group service membership identifier. If two bridges have an I-SID in common, intermediate bridges on the unique shortest path between them will create forwarding state for the related B-MAC addresses. They will also construct multicast forwarding state using the I-SID and the bridge's SPSourceID to construct a multicast DA. Each I-SID has a Transmit (T) and Receive (R) bit which indicates if the membership is as a Transmitter/ Receiver or both (with both bits set). In the case where the Transmit (T) and Receive (R) bits are both zero, the I-SID is ignored (this forces a point to point capability that does not require Group address installation, but the advertised individual B-MAC address shall be processed and installed if required on the SPT). If more I-SIDs are associated with a particular B-MAC than can fit in a single sub-TLV, this sub-TLV can be repeated with the same B-MAC and (MTID) but with additional I-SID values. I-SID values can be in any order. Duplicate I-SIDs for a B-MAC are allowed (within a sub-TLV, across two or more sub-TLVs, across two or more LSP fragments, or even across two or more LSPs). If the duplicates are in different B-VIDs they are processed, otherwise the T and R bits processed are the logical- OR of the T and R bits of the two entries and it is processed as a single entry. (I-SID=0 is an exception that is always ignored, and allows a fast method to remove an I-SID without restructuring all the LSPs). Repacking an LSP (when I-SID=0 entries should be collapsed) can be done by moving entries from the high end into the empty I-SID=0 entries. To avoid outages temporary duplicates are allowed during this repacking and the entry further up in the LSP may later be removed without impact. The I-SID value 0x0000ff is reserved for SPB broadcast to all bridges.

Annex A

(normative)

PICS proforma—Bridge implementations[5]

A.5 Major capabilities

Insert the following row at the end of A.5:

Item	Feature	Status	References	Support
SPB	Is Shortest Path Bridging supported?	O.1	5.4.5, 27	Yes [ ] No [ ]

A.14 Bridge management

In A.14, change MGT-72, MGT-84, and MGT-84 as shown, insert two new items (MGT-86 and MGT-87) in numeric order, renumber the subsequent items, and insert one new item (MGT-225) at the end of the table (note that the entire table is not shown here):

Item	Feature	Status	References	Support
MGT-72	Does the implementation support configuration of VID to FID allocations via management?	MGT:O	5.4.1, ~~8.8.8.1~~8.8.8, 12.10.3	Yes [ ] No [ ]
MGT-84	Read VID Translation Table Entry	MGT AND VLAN-31:M	~~12.13.2.1~~ 12.10.1.1	Yes [ ] No [ ] ~~N/A [ ]~~
MGT-85	Configure VID Translation Table Entry	MGT AND VLAN-31:M	~~12.13.2.2~~ 12.10.1.8	Yes [ ] No [ ] ~~N/A [ ]~~
MGT-86	Read Egress VID Translation Table Entry	MGT AND VLAN-32:M	12.10.1.9	Yes [ ] No [ ]
MGT-87	Configure Egress VID Translation Table Entry	MGT AND VLAN-32:M	12.10.1.8.2	Yes [ ] No [ ]
MGT-225	Does the implementation support the SPB managed objects?	MGT AND SPB:M	5.4.5, 12.25	Yes [ ] No [ ]

A.19 VLAN support

In A.19, change VLAN-24, VLAN-25, and VLAN-31 as shown, insert one new item (VLAN-32) in numeric order, and renumber the subsequent items in the table (note that the entire table is not shown here):

[5] *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.

Item	Feature	Status	References	Support
VLAN-24	Does the implementation allocate VIDs to FIDs in accordance with the specification in 8.8.8?	M	8.8.8~~,~~ ~~8.8.8.2~~	Yes [ ]
VLAN-25	Does the implementation correctly detect Learning Constraint violations?	M	8.8.8~~.3~~	Yes [ ]
VLAN-31	Is the VID translation table supported?	~~CB:X~~ ~~PB:M~~ ~~BEB:O~~ ¬SPBV:O SPBV:M	~~5.5, 5.6.1,~~ 5.4.1, 5.4.5, 6.9	Yes [ ] No [ ] ~~N/A [ ]~~
VLAN-32	Is the Egress VID translation table supported?	¬VLAN-31:X ¬SPBV AND VLAN-31:O SPBV:M	5.4.1, 5.4.5, 6.9	Yes [ ] No [ ]

A.24 Management Information Base (MIB)

Insert the following item at the end of A.24:

Item	Feature	Status	References	Support
MIB-38	Is the IEEE8021-AQ-MIB module fully supported (per its MODULE-COMPLIANCE)?	MIB AND SPB:O	5.4.5, 17.7.19	Yes [ ] No [ ] N/A [ ]

Insert the following subclause, A.37, after A.36:

A.37 Shortest Path Bridging

Item	Feature	Status	References	Support
SPB-1	Does the Bridge support IS-IS Link State Protocol with procedures to ensure Loop Prevention?	SPB:M	5.4.5, 28	Yes [ ] No [ ]
SPB-2	Encode, decode, and validate SPT BPDUs for the Agreement Protocol (AP) and support AP logic in IS-IS?	SPB:M	5.4.5, 28	Yes [ ] No [ ]
SPB-3	State the maximum number of FIDs supported. Minimum is three for SPB	SPB:M	5.4.5	________FIDs
SPB-4	Support VLAN Registration and filtering of frames with unregistered VIDs	SPB:O	5.4.5	Yes [ ] No [ ]
SPBV	Is Shortest Path Bridging VID mode supported?	SPB: O.6	5.4.5	Yes [ ] No [ ]
SPBM	Is Shortest Path Bridging MAC mode supported?	SPB: O.6	5.4.5	Yes [ ] No [ ]

Annex F

(informative)

Shared and Independent VLAN Learning

F.1 Requirements for Shared and Independent Learning

F.1.3 Asymmetric VLANs and Rooted-Multipoint connectivity

F.1.3.2 Rooted-Multipoint

Insert the following text, including Figure F-6, at the end of F.1.3.2:

As noted in bullet a) above, the configurations described above apply to scenarios where the ingress and egress frames at a Root or Leaf Port on a Customer Bridge or S-VLAN Bridge are untagged. Customer Bridges and S-VLAN Bridges that support both the VID Translation Table and Egress VID Translation Table (6.9) can support tagged ingress and egress frames at a Root or Leaf Port. Figure F-6 shows the configuration where the server(s) and subscribers transmit and receive tagged frames. The server uses VID R for frames associated with the rooted-multipoint service, and the subscribers use VIDs L1 and L2. The values of R, L1, and L2 could be, but are not necessarily, the same. The configuration of the Bridge(s) with Root and Leaf Ports are shown in the diagram. All Root and Leaf Ports are included in the member set of the Trunk and Branch VIDs, but not in the untagged set (including the Leaf Ports in the member set of the Branch VID allows ingress VLAN filtering to be enabled at these ports if desired). The VID translation at a Root Port assures that ingress frames go to the Trunk VID, and egress frames come from both the Trunk and Branch VIDs. The VID translation at a Leaf Port assures that ingress frames go to the Branch VID, and egress frames come from the Trunk VID (translating the Branch VID to 0xFFF on egress frames at Leaf Ports causes these frames to be discarded).

NOTE—The example has been deliberately simplified; in practical applications, the central Bridge would likely be replaced by a number of VLAN-aware Bridges, interconnected with links that would carry the traffic between subscribers and servers as frames tagged with either the Trunk or Branch VID. VLAN translation occurs only at the Root and Leaf Ports of the network.

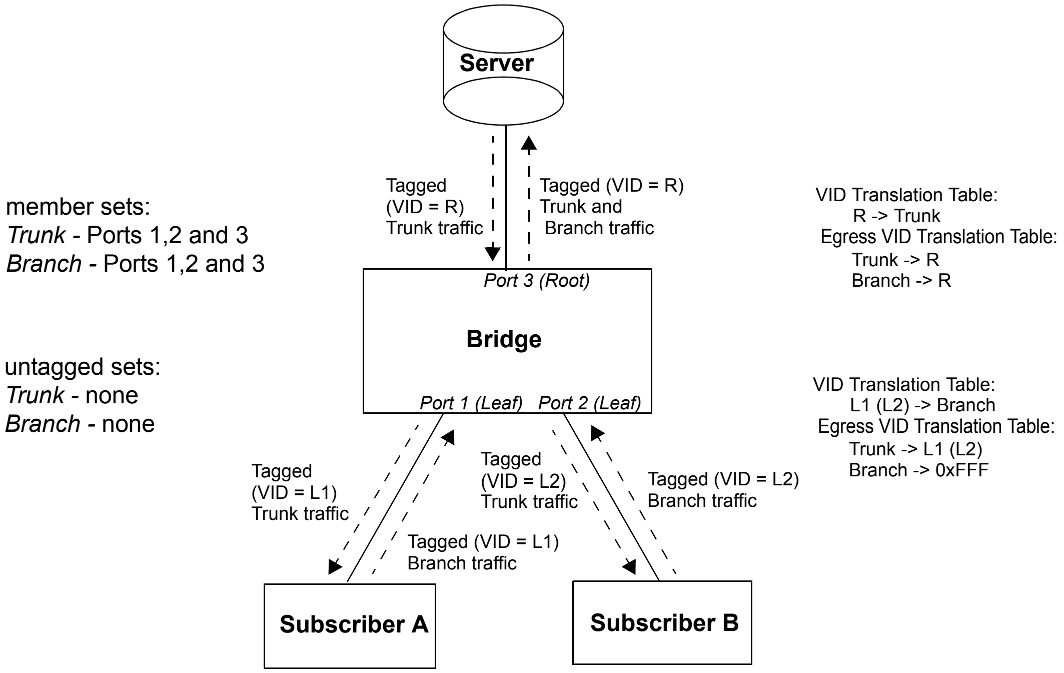

Server
member sets:
Trunk - Ports 1,2 and 3
Branch - Ports 1,2 and 3
Tagged (VID = R) Trunk traffic
Tagged (VID = R) Trunk and Branch traffic
VID Translation Table:
R -> Trunk
Egress VID Translation Table:
Trunk -> R
Branch -> R
Port 3 (Root)
Bridge
untagged sets:
Trunk - none
Branch - none
Port 1 (Leaf)
Port 2 (Leaf)
VID Translation Table:
L1 (L2) -> Branch
Egress VID Translation Table:
Trunk -> L1 (L2)
Branch -> 0xFFF
Tagged (VID = L1) Trunk traffic
Tagged (VID = L2) Trunk traffic
Tagged (VID = L2) Branch traffic
Tagged (VID = L1) Branch traffic
Subscriber A
Subscriber B

Figure F-6—Rooted-Multipoint with tagged interfaces

Insert the following subclause, F.1.4 (including Figure F-7), after F.1.3, and renumber the subsequent subclause accordingly:

F.1.4 Shared Learning and Shortest Path Bridging VID Mode

Shortest Path Bridging VID Mode (SPBV) uses shared learning among the set of SPVIDs that support a given SPBV VLAN. Each SPT Bridge supporting an SPBV VLAN is assigned a unique VID (SPVID) for transmitting VLAN frames into the SPT Region. This SPVID is registered by the ISIS-SPB control plane along a shortest path tree (SPT) rooted at the SPT Bridge to which it is assigned. The SPTs determined by ISIS-SPB provide symmetric bidirectional paths (i.e., congruent paths) between any pair of SPT Bridges within an SPT Region. The congruency of the paths is essential to allow shared learning. Shared learning is required because frames forwarded from a given SPT Bridge will contain a different SPVID from frames being forwarded to that SPT Bridge.

Figure F-7 shows an example of a simple SPBV VLAN operating in an SPT Region comprising three bridges A, B, and C. Each of the bridges is assigned an SPVID, in this example using the same letter designation as the bridge to which it is assigned. ISIS-SPB maintains the VID member sets such that the SPVID of an SPT Bridge follows an SPT rooted at that particular SPT Bridge. The SPT includes all boundary ports participating in the SPBV VLAN (in this example the ports connected to stations A, B, and C).

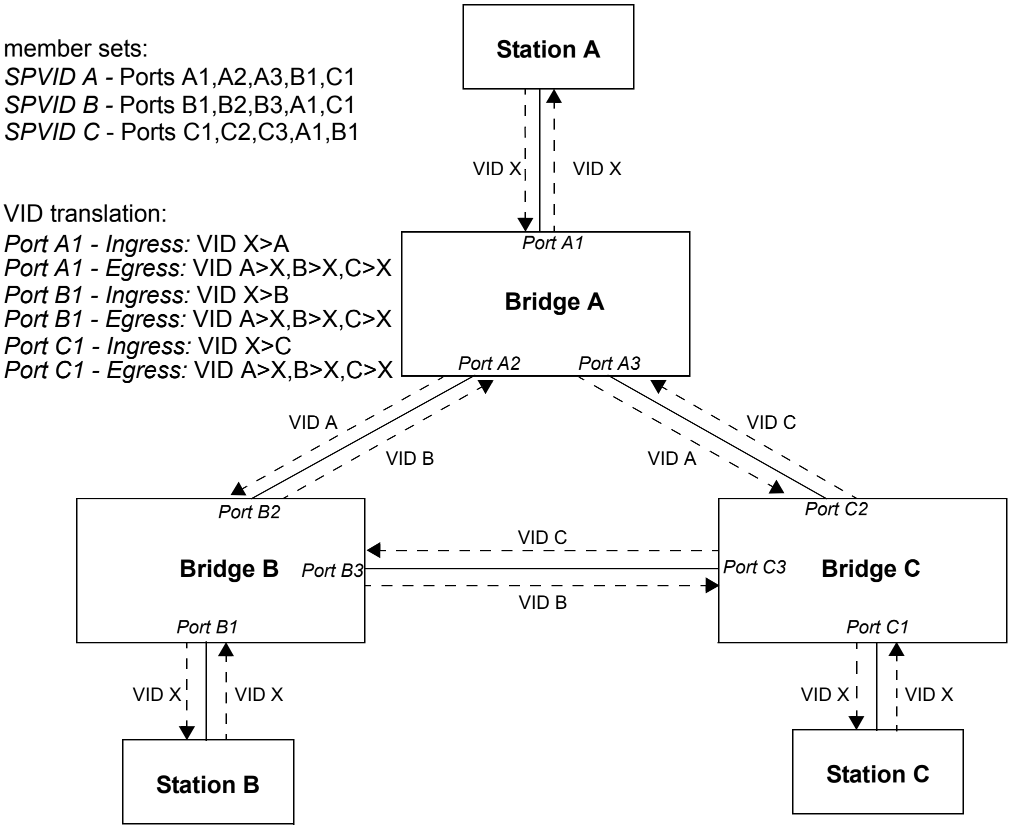

member sets:
SPVID A - Ports A1,A2,A3,B1,C1
SPVID B - Ports B1,B2,B3,A1,C1
SPVID C - Ports C1,C2,C3,A1,B1
VID translation:
Port A1 - Ingress: VID X>A
Port A1 - Egress: VID A>X,B>X,C>X
Port B1 - Ingress: VID X>B
Port B1 - Egress: VID A>X,B>X,C>X
Port C1 - Ingress: VID X>C
Port C1 - Egress: VID A>X,B>X,C>X
Station A
VID X
VID X
Port A1
Bridge A
Port A2
Port A3
VID A
VID B
VID A
VID C
Port B2
Port C2
VID C
Bridge B
Port B3
Port C3
Bridge C
VID B
Port B1
Port C1
VID X
VID X
VID X
VID X
Station B
Station C

Figure F-7—SPBV VLAN Shared Learning and VID Translation

Shared learning is used among all the SPVIDs belonging to the SPBV VLAN. This enables the filtering information learned from frames transmitted from a bridge (with that bridge's SPVID) to be applied to frames being transmitted to that bridge from other bridges using the other bridges' SPVIDs. For example, a frame sent from station A to station B will be transmitted by bridge A using SPVID A. At bridge B the frame's source address will be learned at port B2. When station B later sends a frame to station A, bridge B will assign the frame SPVID B. Shared learning enables bridge B to use the MAC address previously learned from the frame with SPVID A to filter the frame with SPVID B to transmit only on port B2.

Enhanced VID translation capabilities allow a single VID to be used for VLANs on boundary ports connected to an SPT Region even though many SPVIDs with shared learning are used within the SPT Region. In this example stations A, B and C all use VID X. At each boundary port ingress translation is set to translate the external VID (usually the Base VID) to the SPVID for the bridge on which the boundary port resides. Egress VID translation is set to translate all the SPVIDs belonging to the SPBV VLAN to the external VID.

F.2 Configuring the Global VLAN Learning Constraints

In F.2, change the "8.8.8.2" and "8.8.8.3" cross references to "8.8.8."

Annex M

(informative)

Bibliography

Insert the following references in alphanumeric order in Annex M, and renumber the subsequent references in the annex accordingly:

[Bx] IETF RFC 1321, The MD5 Message-Digest Algorithm (http://tools.ietf.org/html/rfc1321).

[Bx] IETF RFC 5306, Restart Signalling for IS-IS (Proposed Standard) (http://tools.ietf.org/html/rfc5306).

[Bx] IETF RFC 6328, IANA Considerations for Network Layer Protocol Identifiers (Proposed Standard) (http/tools.ietf.org/html/rfc6328).

[Bx] IETF RFC 6329, IS-IS Extensions Supporting IEEE 802.1aq Shortest Path Bridging: Pre-publication (http://www.rfc-editor.org/authors/rfc6329.txt).