IEEE STANDARDS ASSOCIATION

IEEE Standard for
Local and metropolitan area networks—

Bridges and Bridged Networks

Amendment 24: Path Control and Reservation

IEEE Computer Society

Sponsored by the
LAN/MAN Standards Committee

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

IEEE Std 802.1Qca™-2015
(Amendment to
IEEE Std 802.1Q™-2014
as amended by
IEEE Std 802.1Qcd™-2015 and
IEEE Std 802.1Q-2014/Cor 1-2015)

IEEE Std 802.1Qca™-2015
(Amendment to
IEEE Std 802.1Q™-2014
as amended by
IEEE Std 802.1Qcd™-2015 and
IEEE Std 802.1Q-2014/Cor 1-2015)

IEEE Standard for
Local and metropolitan area networks—

Bridges and Bridged Networks

Amendment 24: Path Control and Reservation

Sponsor

**LAN/MAN Standards Committee
of the
IEEE Computer Society**

Approved 3 September 2015

IEEE-SA Standards Board

Abstract: Explicit path control, bandwidth reservation, and redundancy (protection, restoration) for data flows are specified in this amendment to IEEE Std 802.1Q-2014.

Keywords: Bridge, Bridged Local Area Network, IEEE 802®, IEEE 802.1Q™, IEEE 802.1Qac™, LAN, local area network, metropolitan area network, Shortest Path Bridging, SPB, Virtual Bridged Local Area Network, virtual LAN

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

PDF: **ISBN 978-1-5044-0772-4** **STD20844**
Print: **ISBN 978-1-5044-0773-1** **STDPD20844**

Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page, appear in all standards and may be found under the heading "Important Notice" or "Important Notices and Disclaimers Concerning IEEE Standards Documents."

Notice and Disclaimer of Liability Concerning the Use of IEEE Standards Documents

IEEE Standards documents (standards, recommended practices, and guides), both full-use and trial-use, are developed within IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association ("IEEE-SA") Standards Board. IEEE ("the Institute") develops its standards through a consensus development process, approved by the American National Standards Institute ("ANSI"), which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and participate without compensation from IEEE. 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.

IEEE does not warrant or represent the accuracy or content of the material contained in its standards, and expressly disclaims all warranties (express, implied and statutory) not included in this or any other document relating to the standard, including, but not limited to, the warranties of: merchantability; fitness for a particular purpose; non-infringement; and quality, accuracy, effectiveness, currency, or completeness of material. In addition, IEEE disclaims any and all conditions relating to: results; and workmanlike effort. IEEE standards documents are supplied "AS IS" and "WITH ALL FAULTS."

Use of an IEEE standard is wholly voluntary. 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.

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.

IN NO EVENT SHALL IEEE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE PUBLICATION, USE OF, OR RELIANCE UPON ANY STANDARD, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

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 or inferred to be the official position of IEEE or any of its committees and shall not be considered to be, or be relied on 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 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. For the same reason, IEEE does not respond to interpretation requests. Any person who would like to participate in 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

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

IEEE draft and approved standards are copyrighted by IEEE under U.S. and international copyright laws. They are made available by IEEE and are adopted 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 these documents available for use and adoption by public authorities and private users, IEEE does not waive any rights in copyright to the documents.

Photocopies

Subject to payment of the appropriate fee, IEEE will grant users a limited, non-exclusive license to photocopy portions of any individual standard for company or organizational internal use or individual, non-commercial use only. To arrange for payment of licensing fees, 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.

Updating of IEEE Standards 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.

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 order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit the IEEE-SA Website at http://ieeexplore.ieee.org/xpl/standards.jsp or contact IEEE at the address listed previously. For more information about the IEEE-SA or IEEE's standards development process, visit the IEEE-SA Website at http://standards.ieee.org.

Errata

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

Patents

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

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

Participants

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

Glenn Parsons, *Chair*
John Messenger, *Vice Chair*
János Farkas, *Editor*
Stephen Haddock, *Chair, Interworking Task Group*

Ting Ao
Christian Boiger
Paul Bottorff
David Chen
Feng Chen
Weiying Cheng
Diego Crupnicoff
Rodney Cummings
Patrick Diamond
Aboubacar Kader Diarra
Norman Finn
Geoffrey Garner
Anoop Ghanwani
Mark Gravel
Eric W. Gray
Craig Gunther
Hitoshi Hayakawa
Jeremy Hitt
Rahil Hussain
Michael Johas Teener
Peter Jones
Hal Keen
Marcel Kiessling
Yongbum Kim
Philippe Klein
Jouni Korhonen
Jeff Lynch
Ben Mack-Crane
Christophe Mangin
James McIntosh
Eric Multanen
Donald Pannell
Karen Randall
Maximilian Riegel
Dan Romascanu
Jessy V. Rouyer
Panagiotis Saltsidis
Behcet Sarikaya
Michael Seaman
Daniel Sexton
Johannes Specht
Kevin B. Stanton
Wilfried Steiner
Vahid Tabatabaee
Patricia Thaler
Jeremy Touve
Karl Weber
Yuehua Wei
Brian Weis
Jordon Woods
Juan-Carlos Zuniga

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

Iwan Adhicandra
Thomas Alexander
Richard Alfvin
Butch Anton
Gennaro Boggia
Christian Boiger
Nancy Bravin
Ashley Butterworth
William Byrd
Radhakrishna Canchi
Juan Carreon
Charles Cook
Rodney Cummings
Yezid Donoso
Sourav Dutta
Richard Edgar
Liu Fangfang
János Farkas
Michael Fischer
Yukihiro Fujimoto
Devon Gayle
Joel Goergen
Eric W. Gray
David Gregson
Randall Groves
Craig Gunther
Stephen Haddock
Marco Hernandez
Werner Hoelzl
David Hunter
Noriyuki Ikeuchi
Sergiu Iordanescu
Akio Iso
Atsushi Ito
Raj Jain
Anthony Jeffree
Michael Johas Teener
Peter Jones
Adri Jovin
Shinkyo Kaku
Piotr Karocki
Stuart Kerry
Yongbum Kim
Jeff Koftinoff
Bruce Kraemer
Hyeong Ho Lee
Arthur H. Light
Elvis Maculuba
James Marin
Roger Marks
Jonathon Mclendon
Charles Moorwood
Jose Morales
Ronald Murias
Michael Newman
Nick S. A. Nikjoo
Satoshi Obara
Alon Regev
Maximilian Riegel
Robert Robinson
Benjamin Rolfe
Dan Romascanu
Jessy V. Rouyer
Osman Sakr
Panagiotis Saltsidis
Bartien Sayogo
Michael Seaman
Kapil Sood
Thomas Starai
Eugene Stoudenmire
Walter Struppler
Payam TorabJahromi
Mark-Rene Uchida
Lorenzo Vangelista
Dmitri Varsanofiev
Prabodh Varshney
George Vlantis
Khurram Waheed
Stephen Webb
Hung-Yu Wei
Andreas Wolf
Oren Yuen

When the IEEE-SA Standards Board approved this amendment on 3 September 2015, it had the following membership:

Introduction

This introduction is not part of IEEE Std 802.1Qca™-2015, IEEE Standard for Local and metropolitan area networks—Bridges and Bridged Networks—Amendment 24: Path Control and Reservation.

This amendment to IEEE Std 802.1Q-2014 specifies protocol extensions, procedures and managed objects for explicit path control, bandwidth reservation, and redundancy (protection, restoration) for data flows.

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

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

Contents

Figures

Tables

IEEE Standard for
Local and metropolitan area networks—

Bridges and Bridged Networks

Amendment 24: Path Control and Reservation

IMPORTANT NOTICE: IEEE Standards documents are not intended to ensure safety, security, health, or environmental protection, or ensure against interference with or from other devices or networks. Implementers of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, and interference protection practices and all applicable laws and regulations.

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

(This amendment is based on IEEE Std 802.1Q™-2014 as amended by IEEE Std 802.1Qcd™-2014.)

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

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

1. Overview

1.3 Introduction

Insert the following paragraph and list items at the end of 1.3:

This standard also specifies further protocol extensions, procedures, and managed objects to IS-IS for providing capabilities beyond Shortest Path Bridging (SPB) for Bridged Networks. These extensions involve explicit path control, bandwidth reservation, and redundancy (protection, restoration) for data flows. Thus, this standard specifies bridging on explicit paths for unicast and multicast frames, specifying protocols to determine multiple active topologies. To this end it

ca) Describes the use of explicit trees, e.g., to improve resiliency and decrease the probability of congestion.
cb) Requires that active topologies calculated by one or multiple entities external to the routing protocol are such that the characteristics of the MAC Service are provided.
cc) Supports management selection of explicit trees for support of any given VLAN within an SPT Region.
cd) Specifies Intermediate System to Intermediate System Path Control and Reservation (ISIS-PCR): the use of and extensions to the Intermediate System to Intermediate System (IS-IS) protocol to establish explicit trees.
ce) Specifies the use of ISIS-PCR for recording bandwidth assignments.
cf) Specifies redundancy for ISIS-SPB and ISIS-PCR.

2. Normative references

Insert the following references into Clause 2 in alphanumeric order:

IETF RFC 5303, Three-way Handshake for IS-IS Point-to-Point Adjacencies, October 2008.[2]

IETF RFC 5305, IS-IS Extensions for Traffic Engineering, October 2008.

IETF RFC 5307, IS-IS Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS), October 2008.

IETF RFC 7810, IS-IS Traffic Engineering (TE) Metric Extensions, 2016.

IETF RFC 7811, An Algorithm for Computing Maximally Redundant Trees for IP/LDP Fast-Reroute, 2016.

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

3. Definitions

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

3.x Almost Directed Acyclic Graph (ADAG): A directed graph that can be transformed into a Directed Acyclic Graph (DAG) by removing all arcs incoming to the ADAG Root.

3.x Bridge Local Computation Engine (BLCE): A computation engine in a Bridge that performs path and routing computations. The BLCE implements, e.g., Shortest Path First (SPF), Constrained Shortest Path First (CSPF), or the Maximally Redundant Trees Algorithm (IETF RFC 7811).

3.x constrained tree: A tree meeting a certain constraint, e.g., providing a minimal available bandwidth.

3.x constrained routing (CR): Shortest path routing on a topology pruned to only contain the links meeting a given constraint.

3.x Constrained Shortest Path First (CSPF): The phrase for the Dijkstra Shortest Path First Algorithm when it is run after pruning the links not meeting a given constraint.

3.x cut-bridge: A Bridge whose removal partitions the network.

3.x cut-link: A link whose removal partitions the network.

3.x Directed Acyclic Graph (DAG): A directed graph containing no directed cycle.

3.x Explicit Tree (ET): An explicitly defined tree, which is specified by its Edge Bridges and the paths among the Edge Bridges. If only the Edge Bridges are specified but the paths are not, then it is a loose explicit tree. If the paths are also specified, then it is a strict explicit tree.

3.x Explicit Tree Database (ETDB): A database storing explicit trees.

3.x Generalized Almost Directed Acyclic Graph (GADAG): A directed graph that has only Almost Directed Acyclic Graphs (ADAGs) as all of its topology blocks.

3.x Intermediate System to Intermediate System (IS-IS) Path Control and Reservation (ISIS-PCR): IS-IS with the Path Control and Reservation extensions specified by Clause 45 of IEEE Std 802.1Qac-2015.

3.x Loop-Free Alternate (LFA): A loop-free backup path for local repair after a failure event.

3.x Maximally Redundant Trees (MRTs): A pair of trees with a common MRT Root where the path from any leaf Bridge to the MRT Root along the first tree (MRT-Blue) and the path from the same leaf Bridge along the second tree (MRT-Red) share the minimum number of Bridges and the minimum number of links. Each such shared Bridge is a cut-bridge. Any shared links are cut-links. If more than two maximally redundant trees are needed, then they are handled as individual static explicit trees as explained in 45.3.2 of IEEE Std 802.1Qac-2015.

3.x MRT-Blue: One of the two Maximally Redundant Trees (MRTs), specifically, the increasing MRT where links in the Generalized Almost Directed Acyclic Graph (GADAG) are taken in the direction from a lower topologically ordered Bridge to a higher one.

3.x MRT-Red: One of the two Maximally Redundant Trees (MRTs), specifically, the decreasing MRT where links in the Generalized Almost Directed Acyclic Graph (GADAG) are taken in the direction from a higher topologically ordered Bridge to a lower one.

3.x MRT Root: The common root of the two Maximally Redundant Trees (MRTs), i.e., of MRT-Blue and MRT-Red.

3.x Path Control Agent (PCA): The agent that is part of the Intermediate System to Intermediate System (IS-IS) Domain and thus can perform IS-IS operations on behalf of a Path Computation Element (PCE), e.g., maintain Link State Database (LSDB) and send Link State protocol data units (PDUs) (LSPs).

3.x Path Computation Element (PCE): An entity that is capable of computing a path through a network based on a representation of the network's topology (obtained by undefined means external to the PCE). A PCE is a higher layer entity in a Bridge or an end station.

3.x Point of Local Repair (PLR): A Bridge that locally redirects the data traffic to a backup path, e.g., to a loop-free alternate path after a failure event.

3.x redundant trees: A pair of trees with a common root where the paths from any leaf Bridge to the root along the first tree and the second tree are disjoint.

3.x Shared Risk Link Group (SRLG): A set of links that share a resource whose failure affects each link.

3.x topology block: A maximally two-connected (induced) subgraph, a cut-link with the two Bridges at either end, or an isolated Bridge.

3.x Traffic Engineering Database (TED): A database storing the traffic engineering information propagated by a link state protocol, e.g., Intermediate System to Intermediate System (IS-IS).

3.x Two-connected: A topology that has no cut-bridges. This is a topology that requires at least two Bridges to be removed before it is partitioned.

4. Abbreviations

Insert the following abbreviations into Clause 4 in alphabetic order:

ADAG	Almost DAG
BLCE	Bridge Local Computation Engine
CR	constrained routing
CSPF	Constrained Shortest Path First
DAG	Directed Acyclic Graph
ET	Explicit Tree
ETDB	Explicit Tree Database
GADAG	Generalized ADAG
ISIS-PCR	IS-IS with Path Control and Reservation extensions
LFA	Loop-Free Alternate
LSDB	Link State Database
LT	Loose Tree
LTS	Loose Tree Set
MO	Managed Object
MRTs	Maximally Redundant Trees
PCE	Path Computation Element
PCA	Path Control Agent
PCR	Path Control and Reservation
PLR	Point of Local Repair
PTP	IEEE 1588™ precision time protocol
SRLG	Shared Risk Link Group
ST	Strict Tree
TAI	Temps Atomic International—International Atomic Time
TED	Traffic Engineering Database

5. Conformance

Insert the following subclause, 5.4.6, after 5.4.5.2:

5.4.6 Path Control and Reservation (PCR) (optional)

A VLAN Bridge implementation that conforms to the provisions of this standard for PCR in Clause 45 shall

a) Support the IS-IS link state protocol with the extensions for explicit path control (ISIS-PCR) as specified in 45.1;
b) Support the SPB Link Metric sub-TLV as specified in 28.12.7;
c) Support the SPB Base VLAN-Identifiers sub TLV as specified in 28.12.4;
d) Support the SPB Instance sub-TLV as specified in 28.12.5;
e) Support the SPBV MAC address sub-TLV as specified in 28.12.9;
f) Support the SPBM Service Identifier and Unicast Address sub-TLV as specified in 28.12.20;
g) Support the ST ECT Algorithm as specified in 45.1.2 and the use of the ST ECT Algorithm as specified in 45.3.2;
h) Support the Topology sub-TLV, the Hop sub-TLV, and the corresponding ISIS-PCR operations for explicit trees as specified in 45.1;
i) Support the management functionality specified in 12.28.

A VLAN Bridge implementation that conforms to the provisions of this standard for PCR in Clause 45 may

j) Support the LT ECT Algorithm as specified in 45.1.2;
k) Support the LTS ECT Algorithm as specified in 45.1.2
l) Support the MRT ECT Algorithm as specified in 45.3.3;
m) Support the MRTG ECT Algorithm as specified in 45.3.4
n) Support the Administrative Group sub-TLV and the corresponding ISIS-PCR operations as specified in 45.1.11;
o) Support the Bandwidth Constraint sub-TLV and the corresponding ISIS-PCR operations as specified in 45.1.12;
p) Support the Bandwidth Assignment sub-TLV and the corresponding ISIS-PCR operations as specified in 45.2.1;
q) Support the Timestamp sub-TLV as specified in 45.2.2;
r) Support LFA for unicast data flows as specified in 45.3.1;
s) Support the PCR Management Information Base (MIB) objects defined in 17.7.19.

7. Principles of Virtual Bridged Network operation

Change the introductory text of Clause 7 as shown:

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

a) The operation of individual VLAN Bridges (Clause 8);
b) Their participation in the Spanning Tree Algorithm and Protocol (STP) (Clause 8 of IEEE Std 802.1D, 1998 Edition [B9]), Rapid Spanning Tree Algorithm and Protocol (RSTP) (Clause 13), Multiple Spanning Tree Algorithm and Protocol (MSTP) (Clause 13), ~~and~~ Shortest Path Bridging (SPB) ~~protocols~~ (Clause 27), and Path Control and Reservation (PCR) (Clause 45) protocols;
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:

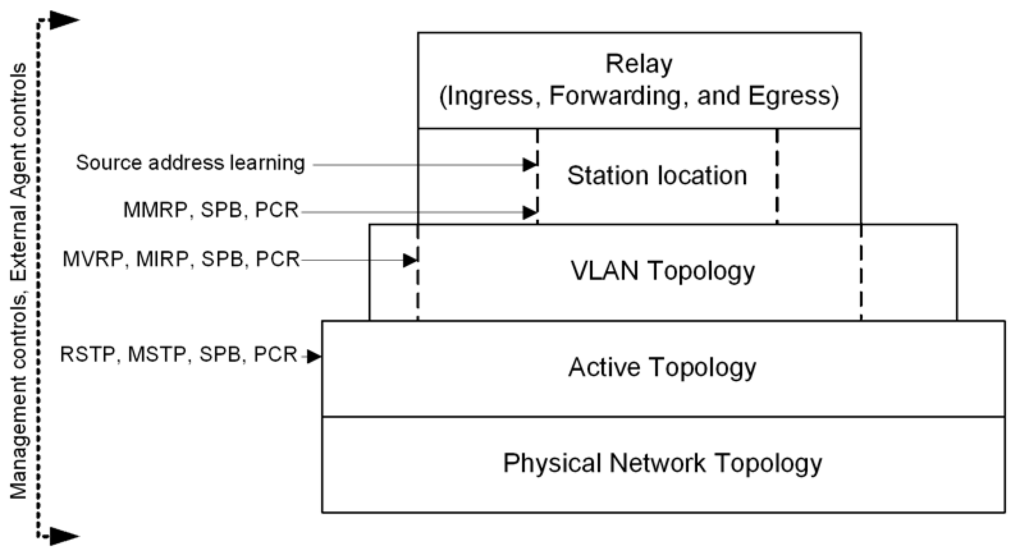

Figure 7-1—VLAN Bridging overview

7.3 Active topology

Change 7.3 as shown:

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), PCR (Clause 45)—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 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 VID to a given subtree within the region. That subtree can be an MSTI, a SPT, an Explicit Tree (ET), or the IST that is the logical continuation of the CST through the region. Interoperability between MST and SPT Bridges and 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.

8. Principles of Bridge operation

8.4 Active topologies, learning, and forwarding

Change the 12th paragraph of 8.4 as follows:

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 among 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). An individual destination address identifies the SPT that is rooted at that destination and belongs to the SPT Set identified by the VID. If relaxed ingress checking (45.3.1) is supported, then a MAC Address Registration Entry is included in the Filtering Database for that VID, individual destination address tuple if there is any source whose path to the destination in that SPT is via the given Bridge. The Port Map of those MAC Address Registration entries comprises only the ports that are upstream for a given individual destination address (45.3.1). If a B-VLAN is supported by SPBM, learning is disabled for all frames for that B-VID.

8.6.1 Active topology enforcement

Change the 11th paragraph of the introductory text of 8.6.1 as follows:

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 or a MAC Address Registration Entry for the frame's individual destination address exists and specifies Forward for that Port; and
— Forwarding is True for all other Bridge Ports.

Change the 12th paragraph in the introductory text of 8.6.1 as follows:

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

— 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-Pool-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.

8.8.9 Querying the FDB

Insert the following paragraph before the first paragraph in 8.8.9:

If a frame is assigned to a VLAN supported by SPBM, then the Filtering Database is queried upon reception of the frame by the ingress port for ingress checking and the frame is forwarded or filtered. If single port admittance ingress checking (6.5.4.2) is applied, then the frame is forwarded only if a Dynamic Filtering Entry for that VID, source address tuple specifies forward for the ingress port; otherwise, the frame is filtered. If relaxed ingress checking (45.3.1) is applied, then the frame is forwarded only if the ingress port is in the Port Map of the MAC Address Registration Entry for that VID, individual destination address tuple; otherwise, the frame is filtered.

8.9.3 ID to MSTI Allocation Table

Change 8.9.3 as follows:

The FID to MSTI Allocation Table defines, for all FIDs that the Bridge supports, the MSTID to which the FID is allocated.

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~~SPVID-Pool-MSTID 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—SPVID-Pool-MSTID: 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 SPB.

12. Bridge management

12.25 Shortest Path Bridging managed objects

Replace Figure 12-3 with the following figure:

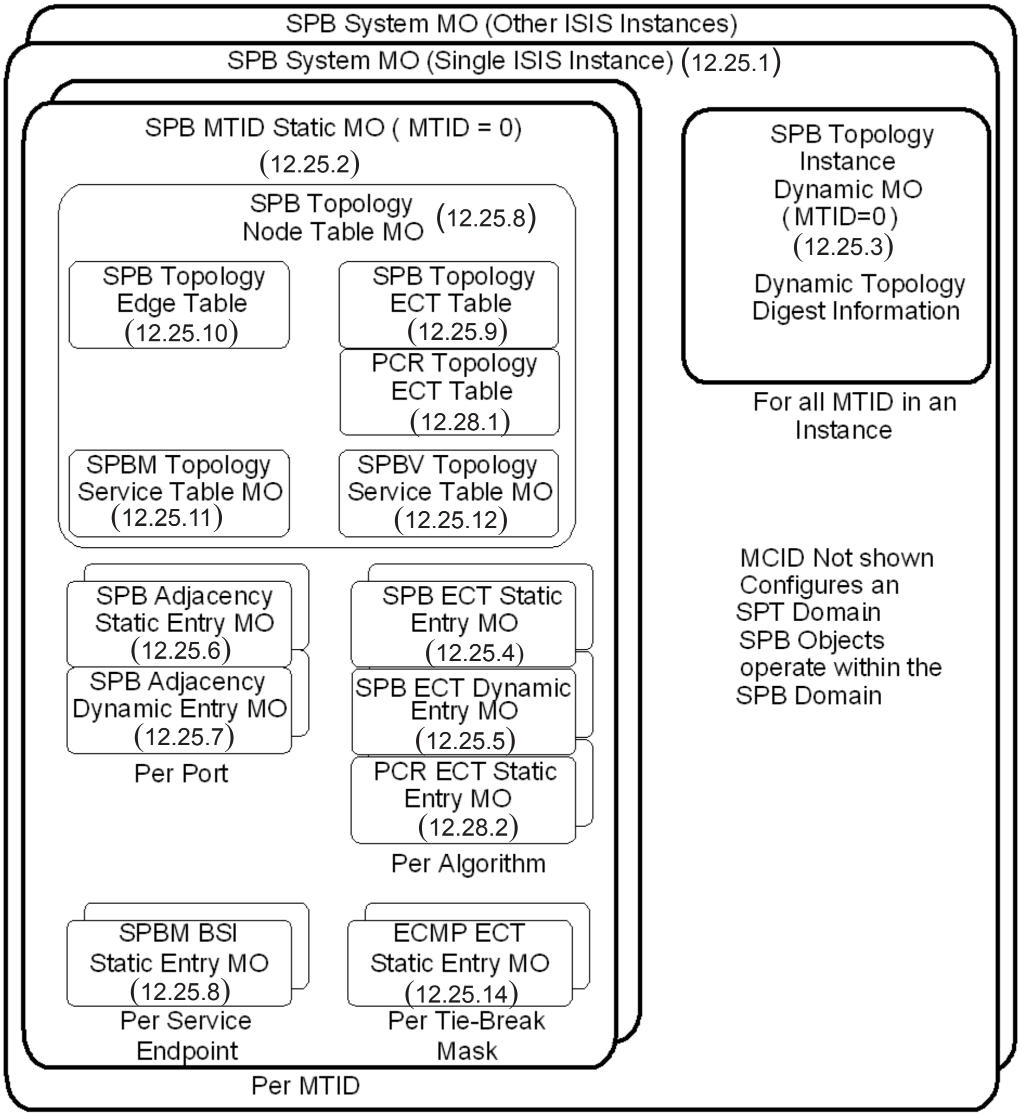

Figure 12-3—SPB managed objects (MOs)

Insert the following items at the end of the lettered list in 12.25:

o) The PCR ECT Static Entry managed object (12.28.1).
p) The PCR Topology ECT Table managed object (12.28.2).

12.25.4 The SPB ECT Static Entry managed object

12.25.4.1 Create SPB ECT Static Entry managed object

12.25.4.1.2 Inputs

Change items c) and d) of 12.25.4.1.2 as shown:

c) The identifier of the ECT Algorithm (ECT-ALGORITHM) to be used for this Base VID (Table 28-1, Table 44-1, Table 45-1). 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 [12.25.1.2.2, item c)]. This VID value must have been assigned to the ~~reserved MSTID value 0xFFF~~SPVID-Pool-MSTID (27.4) in the FID to MSTI Allocation Managed Object (12.12.2.2).

Insert the following subclause, 12.28, after 12.27.1:

12.28 Path Control and Reservation (PCR) management

The VLAN Bridges that support IS-IS implement the following SPB managed objects because they are required for IS-IS operations in a Bridged Network. Therefore, these SPB managed objects are required for explicit path control via IS-IS, i.e., for PCR operations:

— SPB System managed object (12.25.1),
— SPB MTID Static managed object (12.25.2),
— SPB Topology Instance Dynamic managed object (12.25.3),
— SPB ECT Static Entry managed object (12.25.4),
— SPB ECT Dynamic Entry managed object (12.25.5),
— SPB Adjacency Static Entry managed object (12.25.6),
— SPB Adjacency Dynamic Entry managed object (12.25.7),
— SPBM BSI Static Entry managed object (12.25.8),
— SPB Topology Node Table managed object (12.25.9),
— SPB Topology ECT Table managed object (12.25.10),
— SPB Topology Edge Table managed object (12.25.11),
— SPBM Topology Service Table managed object (12.25.12),
— SPBV Topology Service Table managed object (12.25.13).

The above listed SPB managed objects shall be used to configure PCR operations in SPT Bridges as specified in Clause 45. For instance, if a VLAN is under explicit path control via IS-IS, then the corresponding explicit ECT Algorithm value of Table 45-1 has to be used for the VLAN's Base VID when the SPB ECT Static Entry managed object (12.25.4) is created.

NOTE—The SPBM BSI Static Entry managed object (12.25.8) and the SPBM Topology Service Table managed object (12.25.12) are not required if no backbone Service Instance is present in the SPT Region.

Managed objects dedicated to PCR are needed only for Maximally Redundant Trees (MRTs). Due to the dependence on SPB managed objects, the PCR managed objects are specified as part of the SPB managed object hierarchy as illustrated in Figure 12-3. The following are the PCR managed objects in an SPT Bridge:

a) The PCR ECT Static Entry managed object (12.28.1)
b) The PCR Topology ECT Table managed object (12.28.2)

The SPVID parameter of an ISIS-SPB sub-TLV and an SPB managed object does not convey Shortest Path VID if it refers to an explicit tree (Clause 45), but the SPVID parameter is used to carry another type of VID as specified by 45.1.3, e.g., an MRT VID. These VIDs are always configured and never auto-allocated.

If the ECT Algorithm takes the value of the LTS ECT Algorithm (Table 45-1) and it is a learning VLAN (45.1.3), i.e., the VLAN's Base VID is allocated to the SPBV-MSTID (0xFFD), then the Bridge's VID [also allocated to the SPBV-MSTID, and not to the SPVID pool (SPVID-Pool-MSTID: 0xFFF) as is done for Shortest Path Bridging operation] has to be configured in place of the SPVID parameter of the SPB ECT Static Entry managed object [12.25.4.1.2, item d)].

If the ECT Algorithm is one of the MRT algorithms (Table 45-1), then MRT VID values are domain-wide for a non-learning VLAN, whereas they are Bridge-local for a learning VLAN. If it is a learning VLAN (45.1.3), i.e., the VLAN's Base VID is allocated to the SPBV-MSTID, then the Bridge's MRT VIDs (also allocated to the SPBV-MSTID) have to be configured as the MRT-Blue VID and MRT-Red VID parameters of the PCR ECT Static Entry managed object (12.28.1). MRT VIDs are configured; they cannot be auto-allocated. The Base VID's SPVID entry in the SPB ECT Static managed object (12.25.2) is ignored for learning VLANs if the MRTs protect each other. If the MRTs protect an SPT for a learning VLAN, then a Shortest Path VID (a real SPVID allocated to the SPVID-Pool-MSTID) has to be associated with the SPT Bridge, in addition to its MRT VIDs, which is provided as specified by 12.25. The Shortest Path VID is configured via the SPB ECT Static Entry managed object (see 12.25.4.1.2, item d)] and its allocation method (automatic or manual) is configured via the SPB System managed object (see 12.25.1.2.2, item c)].

12.28.1 The PCR ECT Static Entry managed object

Each Bridge has one PCR ECT Static Entry managed object per Base VID assigned with MRT operation (45.3.3, 45.3.4). The static Entry contains the MRT VID configuration parameters for 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 a PCR ECT Static Entry managed object (12.28.1.1).
b) Read a PCR ECT Static Entry managed object (12.28.1.2).
c) Write a PCR ECT Static Entry managed object (12.28.1.3).
d) Delete a PCR ECT Static Entry managed object (12.28.1.4).

12.28.1.1 Create PCR ECT Static Entry managed object

12.28.1.1.1 Purpose

To create, configure, and interrogate a PCR ECT Static Entry managed object on a Bridge.

12.28.1.1.2 Inputs

a) A Topology Index value that uniquely identifies a topology instance.
b) A Base VID. This VID has been allocated either to the SPBM-MSTID (0xFFC) for non-learning VLANs or to the SPBV-MSTID (0xFFD) for learning VLANs in the FID to MSTI Allocation table (12.12.2.2). An SPB ECT Static Entry managed object (12.25.4) has to be created for the Base VID and Topology Index prior to the creation of the PCR ECT Static Entry managed object.
c) The MRT-Blue VID assigned with the Base VID of the corresponding SPB ECT Static Entry managed object. The MRT-Blue VID has to be allocated to the same MSTID as the Base VID. If it is a non-learning VLAN, i.e., the Base VID is allocated to the SPBM-MSTID (0xFFC), and the MRTs are used to protect each other, then the value of the MRT-Blue VID has to take the value of the Base VID. An MRT-Blue VID is always assigned by configuration. In the case of a learning VLAN or if the MRTs protect SPTs of a non-learning VLAN, the values of Base VID, MRT-Blue VID, and MRT-Red VID are all different.

d) The MRT-Red VID assigned with the Base VID of the corresponding SPB ECT Static Entry managed object. The MRT-Red VID has to be allocated to the same MSTID as the Base VID. An MRT-Red VID is always assigned by configuration.

12.28.1.1.3 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 SPB ECT Static Entry managed object does not exist for the Base VID and Topology Index.
   3) Operation rejected because the requested Base VID was not assigned with MRT operation.
   4) Operation rejected due to MRT-Blue VID not allocated to the same MSTID as the Base VID.
   5) Operation rejected due to MRT-Red VID not allocated to the same MSTID as the Base VID.
   6) Operation accepted.

12.28.1.2 Read PCR ECT Static Entry managed object

12.28.1.2.1 Purpose

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

12.28.1.2.2 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 on any Base VID assigned with MRT operation is requested.

12.28.1.2.3 Outputs

a) Operation status. This takes one of the following values:
   1) Operation rejected.
   2) Operation accepted.
b) A Topology Index value that uniquely identifies this topology instance.
c) A list of <Base VID, MRT-Blue VID, MRT-Red VID> 3-tuples assigned with MRT operation. The list is null if no VLAN is assigned with MRT operation. The list comprises a single 3-tuple if the Base VID parameter is not the wildcard (12.28.1.2.2).

12.28.1.3 Write PCR ECT Static Entry managed object

12.28.1.3.1 Purpose

To configure a PCR ECT Static Entry managed object for a Base VID.

12.28.1.3.2 Inputs

a) A Topology Index value that uniquely identifies a topology instance.
b) A Base VID. This VID has been allocated either to the SPBM-MSTID (0xFFC) for non-learning VLANs, or to the SPBV-MSTID (0xFFD) for learning VLANs in the FID to MSTI Allocation table (12.12.2.2). An SPB ECT Static Entry managed object (12.25.4) has to be created for the Base VID and Topology Index prior to the creation of the PCR ECT Static Entry managed object.
c) The MRT-Blue VID assigned with the Base VID of the corresponding SPB ECT Static Entry managed object. The MRT-Blue VID has to be allocated to the same MSTID as the Base VID. If it is a non-learning VLAN, i.e., the Base VID is allocated to the SPBM-MSTID (0xFFC), and the MRTs are used to protect each other, then the value of the MRT-Blue VID has to take the value of the Base

VID. An MRT-Blue VID is always assigned by configuration. In the case of a learning VLAN or if the MRTs protect SPTs of a non-learning VLAN, the values of Base VID, MRT-Blue VID, and MRT-Red VID are all different.

d) The MRT-Red VID assigned with the Base VID of the corresponding SPB ECT Static Entry managed object. The MRT-Red VID has to be allocated to the same MSTID as the Base VID. An MRT-Red VID is always assigned by configuration.

12.28.1.3.3 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 SPB ECT Static Entry managed object does not exist for the Base VID and Topology Index.
   3) Operation rejected because the requested Base VID was not assigned with MRT operation.
   4) Operation rejected due to MRT-Blue VID not allocated to the same MSTID as the Base VID.
   5) Operation rejected due to MRT-Red VID not allocated to the same MSTID as the Base VID.
   6) Operation accepted.

12.28.1.4 Delete PCR ECT Static Entry managed object

12.28.1.4.1 Purpose

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

12.28.1.4.2 Inputs

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

12.28.1.4.3 Outputs

a) Operation status. This takes one of the following values:
   1) Operation rejected because the Bridge has services currently assigned with this Base VID.
   2) Operation rejected because the SPT Region has services currently using this Base VID.
   3) Operation rejected because PCR ECT Static Entry managed object does not exist for the Base VID.
   4) Operation accepted.

12.28.2 The PCR Topology ECT Table managed object

There is one PCR Topology ECT Table managed object per Bridge per IS-IS topology. It is populated by IS-IS 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 a PCR ECT Static Entry managed object for the given IS-IS topology. It is automatically deleted with the deletion of the last PCR ECT Static Entry managed object for the given IS-IS topology. The management operations that can be performed on the PCR Topology ECT Table managed object are as follows:

a) Read PCR Topology ECT Table managed object.

12.28.2.1 Read PCR Topology ECT Table managed object

12.28.2.1.1 Purpose

To obtain nodal information from all SPT Bridges of an SPT Region on the binding of Base VIDs to the MRT algorithms in an IS-IS topology.

12.28.2.1.2 Inputs

a) A Topology Index value that uniquely identifies a topology instance.
b) The System Identifier of a remote Bridge. The value zero is a wildcard indicating the information on all System identifiers is requested.
c) The Base VID. The value 4095 is a wildcard indicating that information on any Base VID assigned with MRT operation is requested.

12.28.2.1.3 Outputs

A list comprising one entry for each Base VID assigned to MRT operation at a Bridge within the SPT Region that the local IS-IS instance is aware. The list can be empty. Each entry contains the following information about a remote Bridge:

a) The Topology Index value that uniquely identifies this topology instance.
b) The System Identifier of the remote Bridge.
c) The Base VID.
d) The VLAN’s operational mode (learning or non-learning).
e) MRT-Blue VID.
f) MRT-Red VID.

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

17. Management Information Base (MIB)

17.2 Structure of the MIB

Change the following row in Table 17-1 as shown:

Table 17-1—Structure of the MIB modules

Module	Subclause	Defining IEEE standard	Reference	Notes
IEEE8021-SPB-MIB	17.2.19, 17.7.19	802.1aq, 802.1Qca	Clause 27, Clause 28, Clause 45	Initial version in IEEE Std 802.1aq; PCR MIB is specified as an extension to IEEE8021-SPB-MIB, initial version in IEEE Std 802.1Qca

17.2.19 Structure of the IEEE8021-SPB-MIB

Insert the following rows at the end of Table 17-25:

Table 17-25—IEEE8021-SPB-MIB structure and relationship to this standard

Clause 17 MIB table/object	Reference
ieee8021PcrEctStaticTable	12.28.1
ieee8021PcrEctStaticTableEntry	12.28.1
ieee8021PcrEctStaticEntryTopIx	12.28.1.1.2
ieee8021PcrEctStaticEntryBaseVid	12.28.1.1.2
ieee8021PcrEctStaticEntryMrtBlueVid	12.28.1.1.2, 45.3.3, 45.3.4
ieee8021PcrEctStaticEntryMrtRedVid	12.28.1.1.2, 45.3.3, 45.3.4
ieee8021PcrEctStaticEntryRowStatus	12.28.2.1.3
ieee8021PcrTopEctTable	12.28.2
ieee8021PcrTopEctTableEntry	12.28.2
ieee8021PcrTopEctEntryTopIx	12.28.2.1.2, 12.28.2.1.3
ieee8021PcrTopEctEntrySysId	12.28.2.1.2, 12.28.2.1.3
ieee8021PcrTopEctEntryBaseVid	12.28.2.1.2, 12.28.2.1.3
ieee8021PcrTopEctEntryMode	12.28.2.1.2, 12.28.2.1.3
ieee8021PcrTopEctEntryMrtBlueVid	12.28.2.1.3, 45.3.3, 45.3.4
ieee8021PcrTopEctEntryMrtRedVid	12.28.2.1.3, 45.3.3, 45.3.4

17.3 Relationship to other MIBs

Insert the following subclause, 17.3.22, after 17.3.21:

17.3.22 Relationship of the PCR MIB to other MIB modules

PCR configurations are based on the IEEE8021-SPB-MIB because PCR operations are largely based on ISIS-SPB sub-TLVs. The IEEE8021-SPB-MIB provides all that is needed for explicit path control; only MRT operations require extensions. Therefore, the PCR MIB objects and compliance are specified as part of the IEEE8021-SPB-MIB.

17.4 Security considerations

Insert the following subclause, 17.4.22, after 17.4.21:

17.4.22 Security considerations of the PCR 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 can 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.

The security considerations described for the EEE8021-SPB-MIB apply to the PCR MIB.

The following PCR MIB objects could be manipulated to interfere with the operation of SPT Bridges. This could, for example, be used to misconfigure the network to cause loss of connectivity or to misdirect traffic. The read-write and/or read-create objects in following table are vulnerable:

ieee8021PcrEctStaticTable

Some of the readable MIB objects (i.e., objects with a MAX-ACCESS other than not accessible) can 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 their respective objects in this MIB could be used by an attacker to understand the logical topology of the network:

ieee8021PcrTopEctTable

17.7 MIB modules

17.7.19 Definitions for the IEEE8021-SPB-MIB module

Replace 17.7.19 with the following text:

```
-- ===================================================================
--  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, IEEE8021PbbServiceIdentifier,
         IEEE8021PbbTeEsp
        FROM IEEE8021-TC-MIB
    ieee8021BridgeBasePort
        FROM IEEE8021-BRIDGE-MIB
    VlanId, VlanIdOrNone, VlanIdOrAny
        FROM Q-BRIDGE-MIB
    dot1agCfmMepEntry, dot1agCfmMdIndex, dot1agCfmMaIndex
        FROM IEEE8021-CFM-MIB
    InterfaceIndexOrZero
        FROM IF-MIB
    SnmpAdminString
        FROM SNMP-FRAMEWORK-MIB
    MODULE-COMPLIANCE, OBJECT-GROUP
        FROM SNMPv2-CONF;

ieee8021SpbMib MODULE-IDENTITY
    LAST-UPDATED "201506230000Z"  -- June 23, 2015
    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: IEEE 802.1 Working Group Chair
            Postal: C/O IEEE 802.1 Working Group
                    IEEE Standards Association
                    445 Hoes Lane
                    Piscataway
                    NJ 08855
                    USA
            E-mail: STDS-802-1-L@LISTSERV.IEEE.ORG"
    DESCRIPTION  "802.1 SPB MIB"

    REVISION  "201506230000Z"  -- June 23, 2015
    DESCRIPTION  "802.1Qca additions"

    REVISION  "201305130000Z"  -- May 13, 2013
    DESCRIPTION  "802.1Qbp additions and corrections"

    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 }
```

```
-- ===================================================================
-- ieee8021PcrObjects:
-- ===================================================================
ieee8021PcrObjects OBJECT IDENTIFIER ::= { ieee8021SpbMib 3 }

-- ===================================================================
-- 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 can 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 can
        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 can
        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 => auto allocate SPVIDs.
         manual => 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 the 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 has to 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 method and the algorithm used
        to determine the active topology. The standard ECT
        Algorithm values are specified by Table 28-1,
        Table 44-1, and Table 45-1.
        Table 28-1 values identify the tie-breaking algorithms
        used in Shortest Path Tree computation;
        values range from 00-80-c2-01 to 00-80-c2-10.
        Table 44-1 values (00-80-c2-11 and 00-80-c2-12)
        identify ECMP operations.
        Table 45-1 values identify explicit path control;
        values are 00-80-c2-17, 00-80-c2-18, 00-80-c2-19,
        and the range from 00-80-c2-21 to 00-80-c2-40.
        The default is 00-80-c2-01, which is the LowPATHID
        from Table 28-1.
        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 can 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 can 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 ingress 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
       or because it is not upstream for the individual DA if
       relaxed ingress checking is applied.
   REFERENCE "12.25.5.1.3, 8.4, 45.3.1"
   ::= { 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 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 }

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

```
-- ieee8021SpbmTopSrvTable:
-- ====================================================================
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 can 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:
-- ===================================================================
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 }

-- ======================================================================
-- ieee8021SpbmBsiStaticTable:
-- ======================================================================
ieee8021SpbmBsiStaticTable OBJECT-TYPE
     SYNTAX SEQUENCE OF Ieee8021SpbmBsiStaticEntry
     MAX-ACCESS not-accessible
     STATUS current
     DESCRIPTION
         "Table of BSIs configured on this system and assigned to
          an SPBM VID.
          The table is indexed by
          - ieee8021SpbTopIx from ieee8021SpbMtidStaticTable
              indicating the ISIS-SPB topology instance into
              which the BSI will be advertised,
          - ieee8021BridgeBasePort from ieee8021PbbCbpTable
              identifying the CPB on which the BSI is configured,
          - an I-SID value identifying the BSI, and
          - a VID value identifying a B-VID for which forwarding
              state is to be installed for the BSI"
     REFERENCE "12.25.8"
     ::= { ieee8021SpbObjects 13 }

ieee8021SpbmBsiStaticEntry OBJECT-TYPE
     SYNTAX Ieee8021SpbmBsiStaticEntry
     MAX-ACCESS not-accessible
     STATUS current
     DESCRIPTION
         "This table contains information about backbone services
          configured on this system to be advertised by ISIS-SPB."
     REFERENCE "12.25.8"
     INDEX {
```

```
            ieee8021SpbTopIx,
            ieee8021BridgeBasePort,
            ieee8021SpbmBsiStaticEntryIsid,
            ieee8021SpbmBsiStaticEntryBaseVid
          }
    ::= { ieee8021SpbmBsiStaticTable 1 }

Ieee8021SpbmBsiStaticEntry ::=
    SEQUENCE {
        ieee8021SpbmBsiStaticEntryIsid
                              IEEE8021PbbServiceIdentifier,
        ieee8021SpbmBsiStaticEntryBaseVid       VlanId,
        ieee8021SpbmBsiStaticEntryTBit          TruthValue,
        ieee8021SpbmBsiStaticEntryRBit          TruthValue,
        ieee8021SpbmBsiStaticEntryTsBit         TruthValue,
        ieee8021SpbmBsiStaticEntryTieBreakMask  Integer32,
        ieee8021SpbmBsiStaticEntryRowStatus     RowStatus
    }

ieee8021SpbmBsiStaticEntryIsid OBJECT-TYPE
    SYNTAX      IEEE8021PbbServiceIdentifier
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An I-SID registered on the CBP identified
         by ieee8021BridgeBasePort."
    ::= { ieee8021SpbmBsiStaticEntry 1 }

ieee8021SpbmBsiStaticEntryBaseVid OBJECT-TYPE
    SYNTAX      VlanId
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An B-VID registered on the CBP identified
         by ieee8021BridgeBasePort."
    ::= { ieee8021SpbmBsiStaticEntry 2 }

ieee8021SpbmBsiStaticEntryTBit OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "If true(1), indicates the BSI transmits multicast
         frames from this CBP.
         This object is persistent."
    ::= { ieee8021SpbmBsiStaticEntry 3 }

ieee8021SpbmBsiStaticEntryRBit OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "If true(1), indicates the BSI wishes to receive
         multicast frames at this CBP.
         This object is persistent."
    ::= { ieee8021SpbmBsiStaticEntry 4 }

ieee8021SpbmBsiStaticEntryTsBit OBJECT-TYPE
    SYNTAX      TruthValue
```

```
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "If true(1), indicates the BSI transmits multicast
         frames on a shared tree from this CBP.
         This object is persistent."
    ::= { ieee8021SpbmBsiStaticEntry 5 }

ieee8021SpbmBsiStaticEntryTieBreakMask OBJECT-TYPE
    SYNTAX      Integer32 (0..15)
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The value used to create the Tie-Break Mask
         for calculating multicast trees.
         This object is persistent."
    ::= { ieee8021SpbmBsiStaticEntry 6 }

ieee8021SpbmBsiStaticEntryRowStatus OBJECT-TYPE
    SYNTAX     RowStatus
    MAX-ACCESS read-create
    STATUS     current
    DESCRIPTION
        "This column holds the status for this row.
         When the status is active, no columns of this table can be
         modified.
         This object is persistent."
    ::= { ieee8021SpbmBsiStaticEntry 7 }

-- ======================================================================
-- SPBM MEP configurable objects
-- ======================================================================
dot1agCfmMepSpbmTable OBJECT-TYPE
    SYNTAX SEQUENCE OF Dot1agCfmMepSpbmEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "The additional objects configurable in SPBM MEPs"
    REFERENCE "27.18"
    ::= { ieee8021SpbObjects 14 }

dot1agCfmMepSpbmEntry OBJECT-TYPE
    SYNTAX      Dot1agCfmMepSpbmEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "The SPBM MEP table additions."

    AUGMENTS { dot1agCfmMepEntry }
    ::= { dot1agCfmMepSpbmTable 1 }

Dot1agCfmMepSpbmEntry ::=
    SEQUENCE {
        dot1agCfmMepTransmitLbmSpbmDA   MacAddress,
        dot1agCfmMepTransmitLtmSpbmDA   MacAddress
    }

dot1agCfmMepTransmitLbmSpbmDA OBJECT-TYPE
    SYNTAX      MacAddress
```

```
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
       "The MAC Address to be used as the LBM destination address
        in an SPBM MA: A unicast or multicast address."
    REFERENCE
       "12.14.7.3.2:g"
    ::= { dot1agCfmMepSpbmEntry 1 }

dot1agCfmMepTransmitLtmSpbmDA OBJECT-TYPE
    SYNTAX      MacAddress
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
       "The MAC Address to be used as the LTM destination address
        in an SPBM MA: A unicast or multicast address."
    REFERENCE
       "12.14.7.4.2:f"
    ::= { dot1agCfmMepSpbmEntry 2 }

-- ====================================================================
-- SPBM path MA and ECMP path MA TE-SIDs
-- ====================================================================
dot1agCfmMepSpbmEspTable OBJECT-TYPE
    SYNTAX     SEQUENCE OF Dot1agCfmMepSpbmEspEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
        "The SPBM ESP table contains path-tesid information for each
         SPBM path MA known to a system.

         This table uses three indices. The first two indices are the
         indices of the Maintenance Domain and MA tables, the reason
         being that a path-tesid is always related to an MA and
         Maintenance Domain."
    REFERENCE
        "27.18.1, 12.14.5.3.2:c"
    ::= { ieee8021SpbObjects 15 }

dot1agCfmMepSpbmEspEntry OBJECT-TYPE
    SYNTAX     Dot1agCfmMepSpbmEspEntry
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
        "The SPBM path MA ESP entry.  Each entry refers to an
         ESP by identifier and contains information about
         one of the ESPs that comprise an SPBM path MA.
         The "
    INDEX { dot1agCfmMdIndex,
            dot1agCfmMaIndex,
            dot1agCfmMepSpbmEspIndex
          }
    ::= { dot1agCfmMepSpbmEspTable 1 }

Dot1agCfmMepSpbmEspEntry ::=
    SEQUENCE {
        dot1agCfmMepSpbmEspIndex       Unsigned32,
        dot1agCfmMepSpbmEspEsp         IEEE8021PbbTeEsp,
        dot1agCfmMepSpbmEspRowStatus   RowStatus
```

```
    }

dot1agCfmMepSpbmEspIndex OBJECT-TYPE
    SYNTAX     Unsigned32 (1..4294967295)
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
        "This is an identifier, of local signifigance to a particular
         SPBM path MA which is used to index the ESPs associated
         with that MA."
    ::= { dot1agCfmMepSpbmEspEntry 1 }

dot1agCfmMepSpbmEspEsp OBJECT-TYPE
    SYNTAX     IEEE8021PbbTeEsp
    MAX-ACCESS read-create
    STATUS     current
    DESCRIPTION
        "This column holds the ESP identifier for one of the Ethernet
         Switched Paths that define the SPBM path MA.
         This object is persistent."
    REFERENCE
        "27.18.1, 12.14.5.3.2:c"
    ::= { dot1agCfmMepSpbmEspEntry 2 }

dot1agCfmMepSpbmEspRowStatus OBJECT-TYPE
    SYNTAX     RowStatus
    MAX-ACCESS read-create
    STATUS     current
    DESCRIPTION
        "This column holds the status for this row.
         When the status is active, no columns of this table can be
         modified.
         This object is persistent."
    ::= { dot1agCfmMepSpbmEspEntry 3 }

-- ======================================================================
--  PCR objects:
-- ======================================================================

-- ======================================================================
--  ieee8021PcrEctStaticTable:
-- ======================================================================
ieee8021PcrEctStaticTable OBJECT-TYPE
   SYNTAX SEQUENCE OF Ieee8021PcrEctStaticTableEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "The Path Control and Reservation (PCR)
       static configuration table."
   REFERENCE "12.28.1"
   ::= { ieee8021PcrObjects 1 }

ieee8021PcrEctStaticTableEntry OBJECT-TYPE
   SYNTAX Ieee8021PcrEctStaticTableEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
      "The PCR static configuration Table defines the
       MRT VIDs for the Base VID if MRT is used."
```

```
    REFERENCE "12.28.1"
    INDEX {
            ieee8021PcrEctStaticEntryTopIx,
            ieee8021PcrEctStaticEntryBaseVid
          }
    ::= { ieee8021PcrEctStaticTable 1 }

Ieee8021PcrEctStaticTableEntry ::=
    SEQUENCE {
        ieee8021PcrEctStaticEntryTopIx IEEE8021SpbMTID,
        ieee8021PcrEctStaticEntryBaseVid VlanIdOrAny,
        ieee8021PcrEctStaticEntryMrtBlueVid VlanIdOrNone,
        ieee8021PcrEctStaticEntryMrtRedVid VlanIdOrNone,
        ieee8021PcrEctStaticEntryRowStatus RowStatus
    }

ieee8021PcrEctStaticEntryTopIx OBJECT-TYPE
    SYNTAX IEEE8021SpbMTID
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "The IS-IS Topology Index identifier to which this
         instance belongs."
    REFERENCE "12.28.1.1.2"
    ::= { ieee8021PcrEctStaticTableEntry 1 }

ieee8021PcrEctStaticEntryBaseVid OBJECT-TYPE
    SYNTAX VlanIdOrAny
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Base VID to use for the MRT ECT-Algorithm or for the
         MRTG ECT Algorithm.
         In the case of a non-learning VLAN, this is the VID
         that carries traffic. In the case of a learning VLAN,
         this is the Base-VID used for management.
         A Base VID value of 4095 is a wildcard for any Base VID
         assigned to MRT operation."
    REFERENCE "12.28.1.1.2"
    ::= { ieee8021PcrEctStaticTableEntry 2 }

ieee8021PcrEctStaticEntryMrtBlueVid OBJECT-TYPE
    SYNTAX VlanIdOrNone
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "MRT-Blue VID.
         The VID this bridge will use to originate traffic
         on MRT-Blue for the VLAN if the VLAN is associated
         with MRT operation.
         This object is persistent."
    REFERENCE "12.28.1.1.2, 45.3.3, 45.3.4"
    ::= { ieee8021PcrEctStaticTableEntry 3 }

ieee8021PcrEctStaticEntryMrtRedVid OBJECT-TYPE
    SYNTAX VlanIdOrNone
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
```

```
        "MRT-Red VID.
         The VID this bridge will use to originate traffic
         on MRT-Red for the VLAN if the VLAN is associated
         with MRT operation.
         This object is persistent."
    REFERENCE "12.28.1.1.2, 45.3.3, 45.3.4"
    ::= { ieee8021PcrEctStaticTableEntry 4 }

ieee8021PcrEctStaticEntryRowStatus 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 "12.28.1.2.3"
    ::= { ieee8021PcrEctStaticTableEntry 5 }

-- ======================================================================
--  ieee8021PcrTopEctTable:
-- ======================================================================
ieee8021PcrTopEctTable OBJECT-TYPE
    SYNTAX SEQUENCE OF Ieee8021PcrTopEctTableEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Table of MRT use in the network."
    REFERENCE "12.28.2"
    ::= { ieee8021PcrObjects 2 }

ieee8021PcrTopEctTableEntry OBJECT-TYPE
    SYNTAX Ieee8021PcrTopEctTableEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "This table lists bridges configured to use MRT."
    REFERENCE "12.25.9"
    INDEX {
            ieee8021PcrTopEctEntryTopIx,
            ieee8021PcrTopEctEntrySysId,
            ieee8021PcrTopEctEntryBaseVid
          }
    ::= { ieee8021PcrTopEctTable 1 }

Ieee8021PcrTopEctTableEntry ::=
    SEQUENCE {
        ieee8021PcrTopEctEntryTopIx IEEE8021SpbMTID,
        ieee8021PcrTopEctEntrySysId MacAddress,
        ieee8021PcrTopEctEntryBaseVid VlanIdOrAny,
        ieee8021PcrTopEctEntryMode IEEE8021SpbEctMode,
        ieee8021PcrTopEctEntryMrtBlueVid VlanIdOrNone,
        ieee8021PcrTopEctEntryMrtRedVid VlanIdOrNone
    }

ieee8021PcrTopEctEntryTopIx OBJECT-TYPE
    SYNTAX IEEE8021SpbMTID
    MAX-ACCESS not-accessible
    STATUS current
```

```
    DESCRIPTION
        "The IS-IS Topology Index identifier to which this
         instance belongs."
    REFERENCE "12.28.2.1.2, 12.28.2.1.3"
    ::= { ieee8021PcrTopEctTableEntry 1 }

ieee8021PcrTopEctEntrySysId OBJECT-TYPE
    SYNTAX MacAddress
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "The System ID that is using MRT.
         A value of 0 is a wildcard for all System identifiers."
    REFERENCE "12.28.2.1.2, 12.28.2.1.3"
    ::= { ieee8021PcrTopEctTableEntry 2 }

ieee8021PcrTopEctEntryBaseVid OBJECT-TYPE
    SYNTAX VlanIdOrAny
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Base VID related to this algorithm.
         In the case of a non-learning VLAN, this is the VID
         that carries traffic. In the case of a learning VLAN,
         this is the Base-VID used for management.
         A Base VID value of 4095 is a wildcard for any Base VID
         assigned to MRT operation."
    REFERENCE "12.28.2.1.2, 12.28.2.1.3"
    ::= { ieee8021PcrTopEctTableEntry 3 }

ieee8021PcrTopEctEntryMode OBJECT-TYPE
    SYNTAX IEEE8021SpbEctMode
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Operating mode : non-learning (=2) or learning (=3)"
    REFERENCE "12.28.2.1.3"
    ::= { ieee8021PcrTopEctTableEntry 4 }

ieee8021PcrTopEctEntryMrtBlueVid OBJECT-TYPE
    SYNTAX VlanIdOrNone
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "MRT-Blue VID.
         The VID this bridge will use to originate traffic
         on MRT-Blue for the VLAN if the VLAN is associated
         with MRT operation."
    REFERENCE "12.28.2.1.3, 45.3.3, 45.3.4"
    ::= { ieee8021PcrTopEctTableEntry 5 }

ieee8021PcrTopEctEntryMrtRedVid OBJECT-TYPE
    SYNTAX VlanIdOrNone
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "MRT-Red VID.
         The VID this bridge will use to originate traffic
         on MRT-Red for the VLAN if the VLAN is associated
```

```
        with MRT operation."
    REFERENCE "12.28.2.1.3, 45.3.3, 45.3.4"
    ::= { ieee8021PcrTopEctTableEntry 6 }

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

ieee8021SpbConformance OBJECT IDENTIFIER ::= { ieee8021SpbMib 2}
ieee8021SpbGroups      OBJECT IDENTIFIER ::= { ieee8021SpbConformance 1}
ieee8021SpbCompliances OBJECT IDENTIFIER ::= { ieee8021SpbConformance 2}

ieee8021PcrConformance OBJECT IDENTIFIER ::= { ieee8021SpbMib 4 }
ieee8021PcrGroups      OBJECT IDENTIFIER ::= { ieee8021PcrConformance 1}
ieee8021PcrCompliances OBJECT IDENTIFIER ::= { ieee8021PcrConformance 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 }

-- See below for additional SPBM Units of conformance (after SPBV section)

-- ================================================================
-- 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 }

-- ==============================================================
-- Additional SPBM Units of conformance
-- ==============================================================

ieee8021SpbmBsiStaticTableGroupSPBM OBJECT-GROUP
    OBJECTS {
                ieee8021SpbmBsiStaticEntryTBit,
                ieee8021SpbmBsiStaticEntryRBit,
                ieee8021SpbmBsiStaticEntryTsBit,
                ieee8021SpbmBsiStaticEntryTieBreakMask,
                ieee8021SpbmBsiStaticEntryRowStatus
    }
    STATUS current
    DESCRIPTION
    "The collection of objects used to represent ieee8021SpbmBsiStaticTable"
    ::= { ieee8021SpbGroups 23 }

dot1agCfmMepSpbmTableGroupSPBM OBJECT-GROUP
    OBJECTS {
                dot1agCfmMepTransmitLbmSpbmDA,
                dot1agCfmMepTransmitLtmSpbmDA
    }
    STATUS current
    DESCRIPTION
    "The collection of objects used to represent dot1agCfmMepSpbmTable"
    ::= { ieee8021SpbGroups 24 }

dot1agCfmMepSpbmEspTableGroupSPBM OBJECT-GROUP
    OBJECTS {
                dot1agCfmMepSpbmEspEsp,
                dot1agCfmMepSpbmEspRowStatus
    }
    STATUS current
    DESCRIPTION
    "The collection of objects used to represent dot1agCfmMepSpbmEspTable"
    ::= { ieee8021SpbGroups 25 }

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

ieee8021PcrSysGroup OBJECT-GROUP
    OBJECTS {
        ieee8021SpbSysAreaAddress,
```

```
        ieee8021SpbSysId,
        ieee8021SpbSysControlAddr,
        ieee8021SpbSysName,
        ieee8021SpbSysBridgePriority,
        ieee8021SpbmSysSPSourceId,
        ieee8021SpbmSysMode,
        ieee8021SpbvSysMode,
        ieee8021SpbSysDigestConvention
    }
    STATUS current
    DESCRIPTION
    "The collection of objects used to represent ieee8021SpbSys for PCR."
    ::= { ieee8021PcrGroups 1 }

ieee8021PcrMtidStaticTableGroup OBJECT-GROUP
    OBJECTS {
        ieee8021SpbMTidStaticEntryMtidOverload,
        ieee8021SpbMtidStaticEntryRowStatus
    }
    STATUS current
    DESCRIPTION
    "The collection of objects used to represent ieee8021SpbMtidStaticTable for
PCR."
    ::= { ieee8021PcrGroups 2 }

ieee8021PcrTopIxDynamicTableGroup OBJECT-GROUP
    OBJECTS {
        ieee8021SpbTopIxDynamicEntryAgreeDigest,
        ieee8021SpbTopIxDynamicEntryMCID,
        ieee8021SpbTopIxDynamicEntryAuxMCID
    }
    STATUS current
    DESCRIPTION
    "The collection of objects used to represent ieee8021SpbTopIxDynamicTable for
PCR."
    ::= { ieee8021PcrGroups 3 }

ieee8021PcrEctStaticTableGroupMAC OBJECT-GROUP
    OBJECTS {
        ieee8021SpbEctStaticEntryEctAlgorithm,
        ieee8021SpbEctStaticEntryRowStatus
    }
    STATUS current
    DESCRIPTION
    "The collection of objects used to represent ieee8021SpbEctStaticTable for
PCR,
    for non-learning VLAN, i.e., MAC-based."
    ::= { ieee8021PcrGroups 4 }

ieee8021PcrEctStaticTableGroupVID OBJECT-GROUP
    OBJECTS {
        ieee8021SpbEctStaticEntryEctAlgorithm,
        ieee8021SpbvEctStaticEntrySpvid,
        ieee8021SpbEctStaticEntryRowStatus
    }
    STATUS current
    DESCRIPTION
    "The collection of objects used to represent ieee8021SpbEctStaticTable for
PCR,
```

```
    for learning VLAN, i.e., VID-based."
    ::= { ieee8021PcrGroups 5 }

ieee8021PcrEctStaticTableGroupMrt OBJECT-GROUP
    OBJECTS {
        ieee8021PcrEctStaticEntryMrtBlueVid,
        ieee8021PcrEctStaticEntryMrtRedVid,
        ieee8021PcrEctStaticEntryRowStatus
    }
    STATUS current
    DESCRIPTION
    "The collection of objects used to represent ieee8021PcrEctStaticTable,
    for MRT operation."
    ::= { ieee8021PcrGroups 6 }

ieee8021PcrEctDynamicTableGroup OBJECT-GROUP
    OBJECTS {
        ieee8021SpbEctDynamicEntryMode,
        ieee8021SpbEctDynamicEntryLocalUse,
        ieee8021SpbEctDynamicEntryRemoteUse,
        ieee8021SpbEctDynamicEntryIngressCheckDiscards
    }
    STATUS current
    DESCRIPTION
    "The collection of objects used to represent ieee8021SpbEctDynamicTable for
PCR."
    ::= { ieee8021PcrGroups 7 }

ieee8021PcrAdjStaticTableGroup OBJECT-GROUP
    OBJECTS {
        ieee8021SpbAdjStaticEntryMetric,
        ieee8021SpbAdjStaticEntryIfAdminState,
        ieee8021SpbAdjStaticEntryRowStatus
    }
    STATUS current
    DESCRIPTION
    "The collection of objects used to represent ieee8021SpbAdjStaticTable for
PCR."
    ::= { ieee8021PcrGroups 8 }

ieee8021PcrAdjDynamicTableGroup 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 for
PCR."
    ::= { ieee8021PcrGroups 9 }
```

```
ieee8021PcrTopNodeTableGroup OBJECT-GROUP
   OBJECTS {
      ieee8021SpbTopNodeEntryBridgePriority,
      ieee8021SpbmTopNodeEntrySPsourceID,
      ieee8021SpbTopNodeEntrySysName
   }
   STATUS current
   DESCRIPTION
   "The collection of objects used to represent ieee8021SpbTopNodeTable for
PCR."
   ::= { ieee8021PcrGroups 10 }

ieee8021PcrTopEctTableGroup OBJECT-GROUP
   OBJECTS {
      ieee8021PcrTopEctEntryMode,
      ieee8021PcrTopEctEntryMrtBlueVid,
      ieee8021PcrTopEctEntryMrtRedVid
   }
   STATUS current
   DESCRIPTION
   "The collection of objects used to represent ieee8021PcrTopEctTable."
   ::= { ieee8021PcrGroups 11 }

ieee8021PcrTopEdgeTableGroup OBJECT-GROUP
   OBJECTS {
      ieee8021SpbTopEdgeEntryMetricNear2Far,
      ieee8021SpbTopEdgeEntryMetricFar2Near
   }
   STATUS current
   DESCRIPTION
   "The collection of objects used to represent ieee8021SpbTopEdgeTable for
PCR."
   ::= { ieee8021PcrGroups 12 }

ieee8021PcrTopSrvTableGroupVid OBJECT-GROUP
   OBJECTS {
      ieee8021SpbvTopSrvEntryBaseVid,
      ieee8021SpbvTopSrvEntryMMacFlags
   }
   STATUS current
   DESCRIPTION
   "The collection of objects used to represent ieee8021SpbvTopSrvTable for PCR,
   i.e., when the service is provided by a VID."
   ::= { ieee8021PcrGroups 13 }

-- ============================================================
-- 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 ,
            ieee8021SpbmBsiStaticTableGroupSPBM
        }

                GROUP   dot1agCfmMepSpbmTableGroupSPBM
                DESCRIPTION
                    "This group is mandatory ONLY for devices supporting
                     SPBM VID MAs."


                GROUP   dot1agCfmMepSpbmEspTableGroupSPBM
                DESCRIPTION
                    "This group is mandatory ONLY for devices supporting
                     SPBM path MAs or ECMP path MAs."

    ::= { 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 }

-- ================================================================
-- Compliance statements PCR
-- ================================================================

ieee8021PcrCompliance MODULE-COMPLIANCE
    STATUS current
    DESCRIPTION
        "Compliance to IEEE 802.1 PCR"
    MODULE
        MANDATORY-GROUPS {
            ieee8021PcrSysGroup ,
```

```
          ieee8021PcrMtidStaticTableGroup ,
          ieee8021PcrTopIxDynamicTableGroup ,
          ieee8021PcrEctStaticTableGroupMAC ,
          ieee8021PcrEctStaticTableGroupVID ,
          ieee8021PcrEctStaticTableGroupMrt,
          ieee8021PcrEctDynamicTableGroup ,
          ieee8021PcrAdjStaticTableGroup ,
          ieee8021PcrAdjDynamicTableGroup ,
          ieee8021PcrTopNodeTableGroup ,
          ieee8021PcrTopEctTableGroup ,
          ieee8021PcrTopEdgeTableGroup ,
          ieee8021PcrTopSrvTableGroupVid
       }
    ::= { ieee8021PcrCompliances 1 }

END
```

27. Shortest Path Bridging (SPB)

Insert the following item at the end of the dashed list after the second paragraph in the preliminary text in Clause 27:

— A set of explicit trees (ET).

Insert the following paragraph after the third paragraph ("Clause 28 specifies the use of ISIS-SPB....") in the preliminary text of Clause 27:

Clause 45 specifies the use of IS-IS to establish explicit trees within an SPT Region. These trees can differ from the SPTs.

27.1 Protocol design requirements

Change item d) in 27.1 as shown:

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, or an explicit tree.

27.4 ISIS-SPB VLAN configuration

Change item b3) in 27.4 as shown:

b) The FID to MSTID Allocation Table is used to associate an MSTID with a FID (8.9.3, 12.12.2):
. . .
3) Each MSTID in the MSTI List identifies an MSTI.
The reserved MSTID values 0 and TE-MSTID, SPBV-MSTID, SPBM-MSTID and ~~0xFFF~~SPVID-Pool-MSTID are never used in the MSTI List.

Change item d) in 27.4 as shown:

d) The reserved FID value 0xFFF is allocated to the ~~reserved MSTID value~~SPVID-Pool-MSTID (0xFFF).

Replace Figure 27-1 with the following figure:

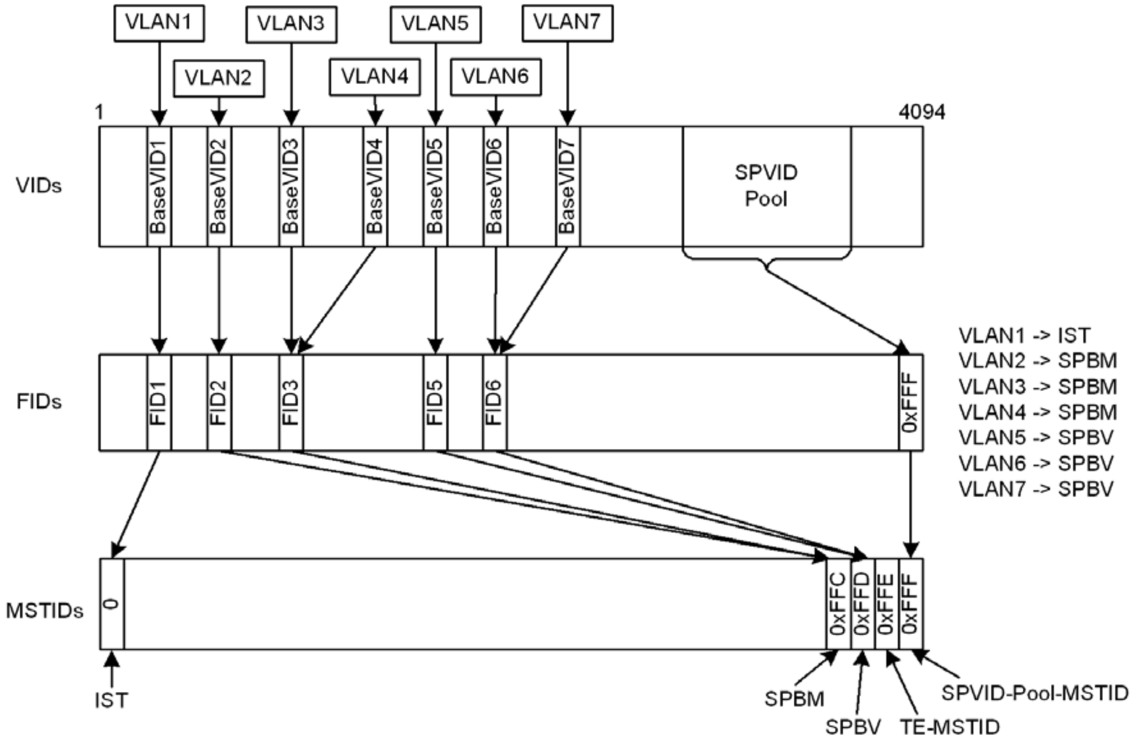

Figure 27-1—Configuring VLAN support in an SPT Region (example)

28. ISIS-SPB Link State Protocol

28.6 Symmetric ECT framework

Change the first paragraph of 28.6 as shown:

Each tie breaking method is uniquely identified by an ECT Algorithm. The ECT Algorithm is a 32-bit number that 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 or CID 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 are associated with symmetric ECT Algorithms while one additional non ECT spanning tree algorithm is defined with Index value 0.

28.7 Symmetric ECT

Change the second paragraph of 28.7 as shown:

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 symmetric shortest path 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 include the Bridge-Priority and the SPB System Identifier, these algorithms can be tuned in deterministic ways by adjusting the Bridge-Priority. SPBM ~~may~~can advertise a Base VID for each of these unique symmetric shortest paths through the ECT SPB Instance sub-TLV (28.12.5).

Change the title, the first paragraph, and the first sentence of the second paragraph of 28.8 as shown:

28.8 Symmetric ECT Algorithm details

The exact method applied for computing the active topology of a VLAN assigned to ISIS-SPB control is determined by the ECT Algorithm configured for the given VLAN. ~~This standard defines a~~A set of symmetric ECT Algorithms ~~used~~are defined to calculate SPT Sets for SPBV or SPBM VLANs~~ and one~~. Two ECMP ECT Algorithms (44.1.2) ~~used~~are defined to calculate SPTs and shared trees for ECMP (Clause 44). Five explicit ECT Algorithms (45.1) are defined for VLANs associated with Explicit Trees (Clause 45).

Each of the standard symmetric ECT-ALGORITHM values is formed using the OUI=00-80-C2 and the Index=1..16.

28.12.4 SPB Base VLAN-Identifiers sub-TLV

Change the second paragraph of 28.12.4 as shown:

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~~are associated with an ECT Algorithm. This structure supports multiple SPT ~~s~~Sets within an IS-IS topology instance for both SPBV and SPBM. It also allows ECMP (Clause 44) for SPBM and the use of Explicit Trees (Clause 45) both for SPBV and SPBM within the same IS-IS topology instance supporting the SPT Sets.

28.12.5 SPB Instance sub-TLV

Change the first paragraph of 28.12.5 as shown:

This sub-TLV (Figure 28-6) ~~must be~~is carried within an MT-Capability TLV in the fragment ZERO LSP. It identifies the Bridge uniquely and identifies the ECT-ALGORITHM values 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~~can be multiple ECT-ALGORITHM values specifying a number of ECTs. Alternatively, the ECT-ALGORITHM value can indicate that the VLAN is assigned to ECMP operation (Table 44-1) or to explicit path control (Table 45-1).

28.12.10 SPBM Service Identifier and Unicast Address (ISID-ADDR) sub-TLV

Change the first paragraph of 28.12.10 as shown:

This sub-TLV (Figure 28-12) 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 rooted at the node identified by that individual B-MAC address. In a symmetric ECT environment, the I-SIDs are mapped to a B-VID that is associated with a symmetric ECT Algorithm specifying the SPT Set. In ECMP, the I-SIDs are mapped to a B-VID that is associated with the ECMP ECT Algorithm, which may specify a source rooted SPT or a shared tree for group addressed frames. Multicast trees can be selected per I-SID for maximum diversity. In the case of explicit trees, the I-SID is mapped to a B-VID allocated to one of the explicit ECT Algorithms (Table 45-1), and the B-MAC can be either an Individual MAC address or null. The null value indicates that the Backbone Service Instance Group address (26.4) is used by all sources of multicast traffic onto the given I-SID in (*,G) mode on one simple tree as specified by 45.1.5. This sub-TLV is carried in an MT-Capability TLV in an LSP.

Insert the following text, Clause 45, after Clause 44:

45. Path Control and Reservation (PCR)

This clause specifies IS-IS extensions to provide the following:

a) Establishment of explicit trees for frame forwarding in an SPT Region (45.1),
b) Use of IS-IS to communicate bandwidth assignments made by the Path Computation Element (PCE) (45.2), and
c) Redundancy with the establishment of the corresponding trees (45.3).

The Path Control and Reservation (PCR) IS-IS extensions specified in this clause are compatible with ISIS-SPB specified in Clause 27 and Clause 28. Furthermore, IS-IS with PCR extensions (ISIS-PCR) relies on the SPB architecture and terminology; and ISIS-PCR also leverages some of the ISIS-SPB sub-TLVs (see, e.g., 5.4.6) as specified by this clause. This specification considers only point-to-point links for PCR although IS-IS also supports shared media LANs.

NOTE 1—ISIS-PCR does not require the implementation of the full ISIS-SPB protocol; but in addition to IS-IS, ISIS-PCR requires the support of the ISIS-SPB sub-TLVs listed in 5.4.6 (also listed as PCR-2–PCR-5 in A.43), whose use is specified by this clause. Nonetheless, if an SPT Bridge supports both ISIS-SPB and ISIS-PCR, then both of them are implemented by the same IS-IS Higher Layer Entity.

A VID can be associated with an explicit tree, i.e., with an explicit active topology within an SPT Region. The Base VID of the VLAN is then associated with explicit path control mode of IS-IS operation, i.e., ISIS-PCR. A VID can be associated with multiple explicit trees if the considerations explained in 45.1.4 are taken into account. VIDs controlled by ISIS-PCR do not participate in ECMP operation (Clause 44).

NOTE 2—A VLAN can provide a point-to-point, point-to-multipoint, or multipoint-to-point service using the multipoint-to-multipoint connectivity provided by an active topology. Similarly the Transmit and Receive IS-IS sub-TLV flags (28.12.10) of Edge Bridges allow an I-SID to use a multipoint-to-multipoint VLAN to provide point-to-point, point-to-multipoint, or multipoint-to-point service.

Path Computation Element (PCE) entities are external to the IS-IS protocol and fully or partially determine and describe each explicit tree to be established by ISIS-PCR throughout an SPT Region. There can be multiple PCEs in a region; nevertheless, any given explicit tree is under the control of one PCE. Explicit trees can be used, for example, for placing selected traffic on a precisely defined route, usually off the shortest path tree.

ISIS-PCR is able to record and communicate bandwidth assignments if instructed to do so by a PCE. ISIS-PCR can be used for bandwidth assignments only if MSRP is not used in the SPT Region. If ISIS-PCR communicates bandwidth assignments, then the tree descriptor assembled by the PCE also includes the details of the bandwidth assignment to be recorded by the Bridges. This mode of operation is expected to be used to divert traffic aggregates off their shortest path route, in order to avoid potential congestion on the default shortest path.

The redundancy provided by a physical topology can be leveraged by various resiliency schemes. A PCE can be used for the computation, and ISIS-PCR can be used for the establishment of the trees required for a given resiliency solution.

45.1 Explicit trees

This subclause specifies IS-IS extensions that provide explicit forwarding trees for data frames. An explicit tree is determined by a Path Computation Element (PCE) and is not required to follow the shortest path.

PCE is defined by IETF RFC 4655. A PCE is an entity that is capable of computing a topology for forwarding based on a network topology, its corresponding attributes, and potential constraints. A PCE explicitly describes a forwarding tree as specified in 45.1.9. Either a single PCE or multiple PCEs determine explicit trees for a region. Even if there are multiple PCEs in a region, each explicit tree is determined by only one PCE, which is referred to as the owner PCE of the tree. PCEs and ISIS-PCR can be used in combination with ISIS-SPB shortest path routing.

A PCE is a higher layer entity in an SPT Bridge or an end station. The PCE interacts with the active topology control protocol, i.e., with ISIS-PCR. The collaboration with ISIS-PCR can be provided by a Path Control Agent (PCA) on behalf of a PCE. Either the PCE or the corresponding PCA is part of the IS-IS Domain. If the PCE is not part of the IS-IS Domain, then the PCE has to be associated with a PCA that resides either in an SPT Bridge or in an end station directly connected to at least one Bridge of the SPT Region. The PCE or its PCA establishes IS-IS adjacency (45.1.7) in order to receive all the LSPs transmitted by the Bridges in the region. The PCE, either on its own or via its PCA, can control the establishment of explicit trees in that region by injecting an LSP conveying an explicit tree and thus instruct ISIS-PCR to set up the explicit tree determined by the PCE. Each PCE, whether located in a Bridge or end station, has access to the link state topology and resource information common throughout the region. If instructed to do so by a PCE, ISIS-PCR can also record and communicate bandwidth assignments, which can be applied only if MSRP is not used in the region. Different PCE and PCA locations are illustrated in Figure 45-1 and Figure 45-2.

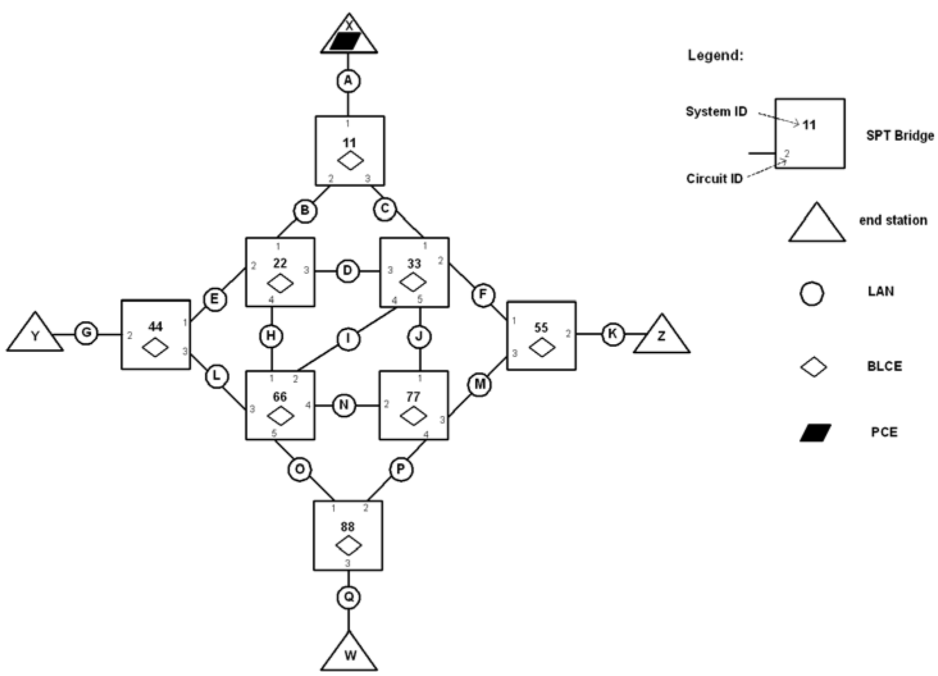

Figure 45-1—An SPT Region controlled by a single PCE

NOTE 1—The operation details of the PCE are not specified by this standard. If the PCE is part of the IS-IS Domain, then the PCE uses IS-IS PDUs to communicate with the IS-IS Domain, and the PCE has a live IS-IS LSDB (i.e., the PCE implements the PCA functions, too). A PCE can instead communicate with the IS-IS Domain via a PCA, e.g., to retrieve the Link State Database (LSDB) or instruct the creation of an explicit tree. However, the means of communication between the PCE and the PCA is not specified by this standard. A PCE could operate on a network topology retrieved by other means, e.g., configuration, instead of retrieving it from a live IS-IS LSDB; which operation mode is not specified by this standard. Having no live LSDB, the PCE instructs its PCA to flood the LSP conveying the appropriate Topology sub-TLV.

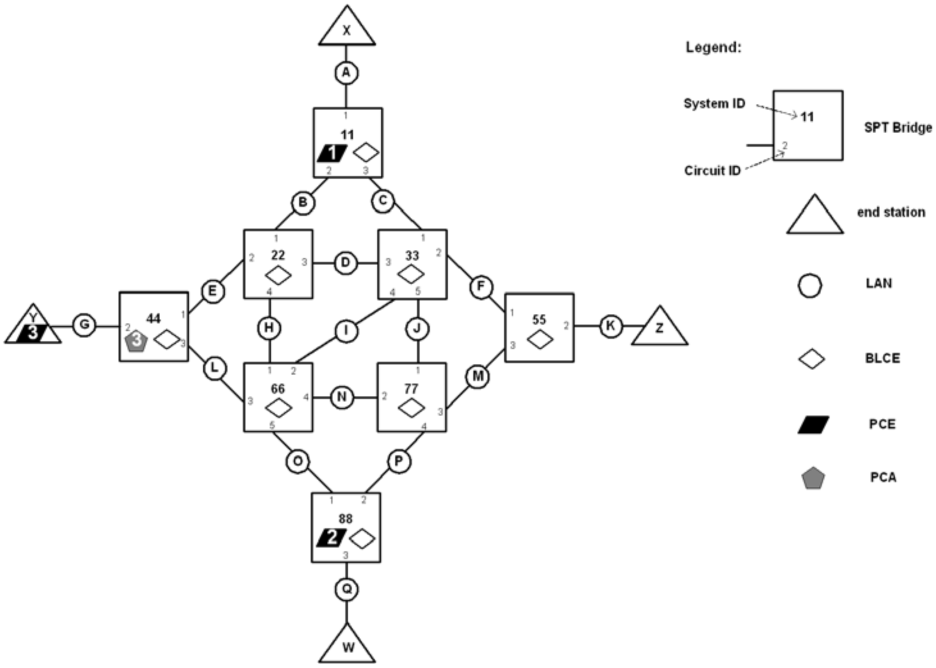

Figure 45-2—An SPT Region controlled by multiple PCEs

ISIS-PCR implements Software Defined Networking (SDN) through hosting the PCE in an external agent, e.g., an SDN Controller. In this case IS-IS is the protocol used to control the Bridges by the SDN Controller; the explicit trees are programmed in the Bridges via IS-IS.

Figure 45-1 shows a region controlled by a single PCE residing in end station X connected to SPT Bridge 11. The PCE has IS-IS adjacency established with SPT Bridge 11, i.e., the PCE is part of the IS-IS Domain.

The IS-IS Domain coincides with the SPT Region in Figure 45-1 and Figure 45-2; they comprise SPT Bridges 11, 44, 55, and 88 (Edge Bridges) and SPT Bridges 22, 33, 66, and 77 (Core Bridges). As shown in the figures, each SPT Bridge implements a Bridge Local Computation Engine (BLCE).

Figure 45-2 shows a region controlled by multiple PCEs. Some of the SPT Bridges implement a PCE in addition to their BLCE; PCE 1 and PCE 2 reside in SPT Bridge 11 and SPT Bridge 88, respectively. PCE 3 resides in End Station Y. PCE 3 is not part of the IS-IS Domain; PCE 3 uses PCA 3 to instruct ISIS-PCR for the establishment of explicit trees and to retrieve link state data from the domain.

An Explicit Tree (ET) is an undirected loop-free topology, whose use is under the control of the owner PCE by means of associating VIDs and MAC addresses with it. As it is undirected, the ET contains no assumptions about the direction of any flows that use it; it can be used in either direction as specified by the VIDs and MAC addresses associated with it. It is the responsibility of the PCE to ensure reverse path congruency (3.180) and multicast unicast congruency (3.248) if that is required, e.g., making an explicit tree symmetric (3.233) by setting it up so that it is used in both directions between a pair of Bridges.

An explicit tree is either strict or loose. A strict explicit tree specifies all Bridges and paths it comprises. A loose tree specifies only the Bridges that have a special role in the tree, e.g., an Edge Bridge, and no path or path segment is specified between the Bridges, which are therefore loose hops even if Edge Bridges are adjacent neighbors. The special role of a hop can be an Edge Bridge, a root, a leaf, a Bridge to be avoided, or a transit hop in the case of a tree with a single leaf. The path for a loose hop is determined by the BLCE of the SPT Bridges. The shortest path is used for a loose hop unless specified otherwise by the descriptor (45.1.9) of the tree or by the corresponding ECT Algorithm (45.1.2).

NOTE 2—PCE and BLCE are architecturally different entities. A BLCE is implemented in each SPT Bridge, where the BLCE implements the SPF algorithm at the minimum and can also implement more complex path computation algorithms, e.g., for determining constrained or redundant trees. For instance, the BLCE of SPT Bridges supporting ISIS-SPB at least implements an all pairs shortest path algorithm in addition to SPF. Forwarding trees are computed by the BLCEs unless they are explicitly given. Conversely, a PCE can implement more complex computation algorithms, and its main task is to determine explicit trees even if the PCE resides in a Bridge.

NOTE 3—If shortest paths are used for loose hops, then there is no need for any computation in addition to SPB as ISIS-SPB performs all pairs shortest path computation.

A loose explicit tree is constrained if the tree descriptor includes one or more constraints, e.g., the administrative group to which the links of the tree have to belong. The BLCE of the SPT Bridges then performs constrained routing (CR) to determine a loose hop instead of shortest path routing. CR relies on distributed link state operation similarly to shortest path routing. CR essentially performs shortest path routing on the topology that contains only the links meeting the constraint(s); therefore, the applied path computation algorithm is also referred to as Constrained Shortest Path First (CSPF).

NOTE 4—If a tree is a strict explicit tree, then it is fully specified; therefore, no constraint is included in the tree descriptor (45.1.9) as no further computation is needed.

An explicit tree is specified by a Topology sub-TLV (45.1.9). The Topology sub-TLV associates one or more VIDs with an explicit tree. The Topology sub-TLV includes two or more Hop sub-TLVs (45.1.10), and a hop is specified by an IS-IS System ID. A strict tree is decomposed to branches, and each branch is specified by an ordered list of Hop sub-TLVs. A loose tree is not fully specified, but the Topology sub-TLV conveys some of its hops. The Hop sub-TLV of a loose hop can include a delay constraint [45.1.10, item h)]. A Topology sub-TLV can also include further sub-TLVs to constrain (45.1.11, 45.1.12) loose hops. The Bridges involved in an explicit tree store the corresponding Topology sub-TLVs in their Explicit Tree Database (ETDB).

Explicit trees are propagated and set up by ISIS-PCR in an SPT Region. The PCE or its PCA assembles the Topology sub-TLVs (45.1.9) and then instructs ISIS-PCR to establish the tree. If the PCE resides in an SPT Bridge, then the PCE entity passes the Topology sub-TLV (45.1.9) to the ISIS-PCR entity, which shall then flood an LSP including the Topology sub-TLV throughout the SPT Domain. If the PCE resides in an end station, then either the PCE or its PCA adds the Topology sub-TLV (45.1.9) into an LSP, which is flooded throughout the SPT Domain. The Topology sub-TLV is flooded by the same techniques used for the SPB LSPs. The SPT Bridges then shall process the Topology sub-TLV (45.1.9) upon reception. If the Topology sub-TLV specifies one or more loose trees, then the path for the loose hops is determined by the BLCE of the SPT Bridges. The SPT Bridges then install the appropriate FDB entries (45.1.6) for frame forwarding along the tree described by the Topology sub-TLV (45.1.9) or the trees computed based on the Topology sub-TLV. Dynamic Filtering Entries are maintained by ISIS-PCR for the VID, MAC address tuples associated with an ET.

NOTE 5—Due to the LSP aging of IS-IS, the Topology sub-TLVs (45.1.9) have to be refreshed similar to other IS-IS TLVs in order to keep the integrity of the LSDB. The corresponding Dynamic Filtering Entries are also refreshed when a Topology sub-TLV is refreshed. Refreshing Topology sub-TLVs is the task of the entity being part of the IS-IS Domain, i.e., either the PCE or the PCA.

The owner PCE can withdraw an explicit tree by sending an updated LSP that does not include the Topology sub-TLV. If a Topology sub-TLV is removed from an LSP (or has been changed), so that (previous) Topology sub-TLV is no longer present (or has been changed) in the LSDB, then that (previous) Topology sub-TLV is implicitly withdrawn. ISIS-PCR then removes (or updates) the explicit tree.

ISIS-PCR provides precedence order among Topology sub-TLVs as specified in 45.2.3 if it is needed in a region with multiple PCEs.

45.1.1 Tree structures

Two types of tree structures can be used for explicit trees, as follows:

— Ad-hoc trees
— Template trees

Ad-hoc explicit trees do not follow any template; they can comprise any path. A particular explicit tree is defined, and a VID is assigned to the explicit tree. The administratively straightforward way to do this is to use a bidirectional VID for the explicit tree. This approach constructs a shared bidirectional (*,G) tree joining all members of a group.

NOTE 1—A single common (*,G) spanning tree can be constructed this way.

Alternatively, explicit trees can follow templates. It can be helpful to define a tree rooted on every Bridge, where each tree spans every Bridge of a domain and the complete set of trees, one rooted on each Bridge of a domain, are constructed consistently, so that collectively they form a single forwarding plane offering any-to-any connectivity within that domain. Such trees are used as a source rooted tree for multicast traffic emitted from a Bridge, and they are also used as a destination rooted tree for unicast traffic being sent to their Bridge. There is a fundamental constraint that has to be obeyed for all such structures individually in that plane. They have to be simply connected trees, with only one path from any point in the network domain to the root of the tree. If paths to the root cross at any point, or merge and subsequently diverge, a unicast forwarding inconsistency or an unwanted multicast packet replication point is created.

This connectivity style (i.e., a tree per Bridge per plane) is, for example, enforced by construction by Shortest Path Bridging. In SPB, a Base VID is associated with one of the standard symmetric ECT Algorithms (28.8), which identifies a plane. In SPBM, the individual trees within each Base VID are defined by the MAC address of the root Bridge; in SPBV each Bridge is allocated an individual SPVID to define its tree.

Any preferred algorithm can be used when performing explicit routing, and the required constraint can be replicated by maintaining the simply connected tree construct. Thus the template-tree-based approach constructs one or more explicitly routed forwarding planes with tree sets, one tree rooted on each Bridge within each plane.

In the case of a learning VLAN, a unidirectional VID can be assigned to each tree (similarly to the unidirectional SPVID model), so that each tree is independent and identifies its root Bridge. Shared VLAN Learning (3.204) takes place among the VIDs associated with the same explicit tree set, where reverse congruence (3.180) has to be enforced by the function that computes the explicit trees on a particular forwarding plane (Base VID).

In the case of a non-learning VLAN, the MAC of the root Bridge identifies the tree, and a single Base VID identifies the entire forwarding plane.

Care has to be taken when constructing multicast trees with multiple sources if the template tree approach is used. Source-specific multicast (S,G) has to be applied if multiple sources participate in a single multicast group. A separate source rooted tree is constructed for every source of the group, and the complete multicast structure is the superposition of all these separate source rooted trees. The multicast traffic of the different sources has to be distinguished either by a VID (as in SPBV) or by a source-specific Group MAC address (as in SPBM, see 27.15).

NOTE 2—There is no requirement here that the initial template construction actually generates spanning trees. If Bridges only initially require connectivity to a subset of the SPT Bridges of a region, only the relevant part of the template need initially be constructed. This leaves freedom to add hitherto unconnected Bridges by extension of trees using a preferred route at a later stage.

45.1.2 Explicit ECT Algorithms

Five explicit path control modes are specified by this standard; each of them is identified by a distinct ECT-ALGORITHM. Table 45-1 summarizes the ECT Algorithms that can be used for explicit trees. The ECT-ALGORITHMs specified by this standard for explicit trees are formed using OUI=00-80-C2 and Index values 0x17, 0x18, 0x19, and the range from 0x21 to 0x40. SPT Bridges that support PCR shall support the Strict Tree ECT Algorithm (00-80-C2-17) and may support the other ECT Algorithms of Table 45-1. VLANs under explicit path control are associated with one of the ECT Algorithms as specified by 45.1.3.

NOTE 1—The ECT-ALGORITHM is in fact the identifier of the method and the algorithm used to determine the active topology. Although the active topology specified by this clause is not an Equal Cost Tree, the terminology and the ECT Algorithm framework of Clause 27 and Clause 28 is kept. The Opaque ECT concept (28.6) is also supported for explicit path control, which can be used for further algorithms.

Table 45-1—ECT-ALGORITHM values for explicit trees

ECT-ALGORITHM	Algorithm Name	Behavior
00-80-C2-17	Strict Tree (ST ECT Algorithm)	A single strict explicit tree. No restoration or update is performed by IS-IS on its own; only the PCE can initiate the update of a strict ET.
00-80-C2-21 ... 00-80-C2-30	Loose Tree (LT ECT Algorithm)	A single loose explicit tree. The loose hops are computed by the BLCE of SPT Bridges applying constrained or shortest path routing. The loose hops are restored by IS-IS upon a topology change if loop-free paths are available. The Bridge Priority Mask to be used in the case of constrained or shortest paths is specified by Table 45-2.
00-80-C2-31 ... 00-80-C2-40	Loose Tree Set (LTS ECT Algorithm)	A set of loose explicit trees. The set comprises an individual tree for each Edge Bridge included in the descriptor of the explicit tree, i.e., they are template trees. Each tree is computed by the BLCE of SPT Bridges applying constrained routing. These trees are restored by IS-IS upon a topology change. The Bridge Priority Mask to be used in the case of constrained or shortest paths is specified by Table 45-2.
00-80-C2-18	Maximally Redundant Trees (MRT ECT Algorithm)	Maximally Redundant Trees (MRTs) are loose trees for each MRT Root, which are computed together with the corresponding GADAG by the BLCE of SPT Bridges and cautiously restored by ISIS-PCR. If the topology view is identical throughout the SPT Domain, then IS-IS restores these trees one by one, only one of each redundant set at a time (45.3.3).
00-80-C2-19	Maximally Redundant Trees with GADAG (MRTG ECT Algorithm)	Maximally Redundant Trees (MRTs) are loose trees for each MRT Root, which are computed by the BLCE of SPT Bridges based on the GADAG received from the single GADAG Computer. MRTs are cautiously restored by ISIS-PCR upon reception of a new GADAG from the GADAG Computer. IS-IS restores these trees one by one, only one of each redundant set at a time (45.3.4).

The Strict Tree (ST) ECT Algorithm is used for a strict explicit tree. A strict ET is static as no other entity can update it but the tree owner PCE. In the case of a topology change, it is the task of the owner PCE to detect the topology change, e.g., based on the changes in the LSDB, and to update the strict trees if needed. In other words, the owner PCE computes the new tree, assembles its descriptor, and then instructs ISIS-PCR to install it. The use of VIDs for strict trees is described in 45.1.4, and redundant strict trees are explained in 45.3.2.

The Loose Tree (LT) ECT Algorithm is used for a single loose explicit tree. The path for loose hops is determined by the BLCE of the SPT Bridges; therefore, the Topology sub-TLV (45.1.9) specifying the tree has to indicate which hop is the root of the tree. The loose hops are maintained by IS-IS, i.e., restored upon a topology change if a loop-free path is available. If the tree computed by the BLCE visits the same Bridge twice (implying that a loop or hairpin has been created), then that loop or hairpin has to be pruned from the tree even if it contains a hop specified by the Topology sub-TLV. Constrained routing can be applied for the loose hops based on the attributes listed in 45.1.8. If a Bridge is not to be included is also a constraint, which can be specified by the Exclude flag [45.1.10, item c6)] of a Hop sub-TLV (45.1.10) conveyed by the Topology sub-TLV specifying the tree.

The Loose Tree Set (LTS) ECT Algorithm is used if connectivity among the Edge Bridges specified by the Topology sub-TLV (45.1.9) is to be provided by a set of loose trees so that one tree is rooted at each Edge Bridge, i.e., the loose trees are template trees (45.1.1). The BLCEs of the SPT Bridges compute the loose trees, which are maintained by IS-IS, i.e., restored upon a topology change. Avoiding some bridges in these trees is a constraint, which can be specified by the Exclude flag [45.1.10, item c6)]. Further constraints can be specified by the Topology sub-TLV based on the attributes listed in 45.1.8.

NOTE 2—A Loose Tree Set is similar to a Shortest Path Tree Set with the following differences: The trees of an LTS span only the Edge Bridges specified by the Topology sub-TLV; they do not span the entire SPT Region. Furthermore, constrained routing is applied instead of shortest path routing.

In the case of the LT and LTS ECT Algorithms, the symmetric shortest path tie-breaking specified in 28.5, 28.6, 28.7, and 28.8 is used during shortest path computation to manipulate the lexicographic ordering of hops on a path after pruning the topology according to the constraints. The Bridge Priority Masks for the LT and LTS ECT Algorithms are shown in Table 45-2. The mask identification method specified for the Symmetric ECT Algorithms is used. The IEEE 802.1 OUI (00-80-C2) is used, Index values 0x21 through 0x30 are used for the LT ECT Algorithm, and Index values 0x31 through 0x40 are used for the LTS ECT Algorithm as shown in Table 45-2.

Table 45-2—Bridge Priority Masking for the LT and LTS ECT Algorithms

ECT-ALGORITHM (LT ECT Algorithm)	**ECT-ALGORITHM** (LTS ECT Algorithm)	**Algorithm MASK**
00-80-C2-21	00-80-C2-31	0x00
00-80-C2-22	00-80-C2-32	0xFF
00-80-C2-23	00-80-C2-33	0x88
00-80-C2-24	00-80-C2-34	0x77
00-80-C2-25	00-80-C2-35	0x44
00-80-C2-26	00-80-C2-36	0x33
00-80-C2-27	00-80-C2-37	0xCC
00-80-C2-28	00-80-C2-38	0xBB
00-80-C2-29	00-80-C2-39	0x22
00-80-C2-2A	00-80-C2-3A	0x11
00-80-C2-2B	00-80-C2-3B	0x66
00-80-C2-2C	00-80-C2-3C	0x55
00-80-C2-2D	00-80-C2-3D	0xAA

Table 45-2—Bridge Priority Masking for the LT and LTS ECT Algorithms *(continued)*

ECT-ALGORITHM (LT ECT Algorithm)	ECT-ALGORITHM (LTS ECT Algorithm)	Algorithm MASK
00-80-C2-2E	00-80-C2-3E	0x99
00-80-C2-2F	00-80-C2-3F	0xDD
00-80-C2-30	00-80-C2-40	0xEE

The Maximally Redundant Trees (MRT) ECT Algorithm or the Maximally Redundant Trees with Generalized Almost Directed Acyclic Graph (GADAG) (MRTG) ECT Algorithm is used if maximally redundant loose explicit trees have to be maintained together. The use of the MRT ECT Algorithm and the MRTG ECT Algorithm is described in 45.3.3 and 45.3.4, respectively.

ISIS-PCR uses the link metrics specified by the SPB Link Metric sub-TLVs (28.12.7) if the LT, the LTS, the MRT, or the MRTG ECT Algorithm is used. The SPB Link Metric sub-TLV is used as specified in 28.12.7; therefore, the maximum metric value is used in cases where the metrics advertised by adjacent Bridges for a given link are different.

A topology change can cause the need for recomputing and updating multiple loose trees. If constrained routing based on available bandwidth (45.1.12) is used for multiple loose trees, then there can be a race hazard for the same resources if they are updated at the same time. Furthermore, the computation order can influence the trees. Therefore, ISIS-PCR applies the tie-breaking method specified in 45.2.3 in order to determine the computation order among the loose trees based on their descriptor Topology sub-TLV. If a Topology sub-TLV specifies multiple loose trees, e.g., LTS or MRTs, then the computation order of these trees follows the ascending order of the IS-IS System ID of the Bridges rooting the trees.

45.1.3 ISIS-PCR VLAN configuration

A VLAN provided by an explicit tree controlled by IS-IS is associated with IS-IS by means of allocating the VLAN's Base VID to the appropriate MSTID, i.e., either to the SPBM-MSTID or to the SPBV-MSTID. The explicit path control mode is then selected by associating the Base VID with the corresponding explicit ECT Algorithm.

The Base VID of VLANs controlled by IS-IS is allocated either to the SPBM-MSTID (0xFFC) or to the SPBV-MSTID (0xFFD) as specified in 27.4, which also applies to VLANs under explicit path control via IS-IS. If multiple VIDs belong to a VLAN under explicit path control, then each VID has to be allocated to the same MSTID; otherwise, the explicit trees do not get installed for the VLAN.

The exact active topology enforcement method applied for the VLAN is determined by the ECT Algorithm (28.8) with which the VLAN is associated. Explicit path control mode is configured by means of associating the Base VID with one of the ECT Algorithms specified in 45.1.2. The association of the VLAN with the ECT Algorithm shall be provided by the SPB Base VLAN-Identifiers sub-TLV (28.12.4).

Each VID belongs only to one PCE, which is the owner PCE that has full control on the use of the VID. The VIDs of a PCE are to be configured at the PCE. If multiple PCEs try to use the same VID, then ISIS-PCR provides the precedence as specified in 45.2.3.

An IS-IS controlled VLAN is a non-learning VLAN if the VLAN's Base VID is allocated to the SPBM-MSTID (0xFFC). MAC addresses are distributed explicitly for non-learning VIDs by IS-IS. The M flag is set in the SPB Instance sub-TLV (28.12.5) for non-learning VLANs. If a non-learning VLAN is associated with a symmetric ECT Algorithm (28.8), then the VLAN is under SPBM control as specified by Clause 27 and Clause 28 even if the VLAN is not a B-VLAN. If a non-learning VLAN is associated with one of the explicit ECT Algorithms (45.1.2), then it is under explicit path control as specified by this clause.

An IS-IS controlled VLAN is a learning VLAN if the VLAN's Base VID is allocated to the SPBV-MSTID (0xFFD). If a learning VLAN is supported by multiple VIDs, then Shared VLAN Learning (3.204) takes place among the VLAN's VIDs associated with the same shortest path or explicit tree set. The M flag is cleared in the SPB Instance sub-TLV (28.12.5) for learning VLANs. If a learning VLAN is associated with a symmetric ECT Algorithm (28.8), then the VLAN is under SPBV control as specified by Clause 27 and Clause 28; correspondingly, SPVIDs can be auto-allocated to SPTs from the pool of SPVIDs, i.e., from the VIDs allocated to the SPVID-Pool-MSTID (0xFFF) as specified by 27.10. If a learning VLAN is associated with one of the explicit ECT Algorithms (45.1.2), then it is under explicit path control as specified by this clause; and each VID of the VLAN is allocated to the SPBV-MSTID (0xFFD), i.e., none of the VLAN's VIDs is allocated automatically. When a VLAN is under explicit path control, it is the responsibility of the PCE to provide reverse congruent paths (3.180) so that Shared VLAN Learning (3.204) operates correctly.

NOTE 1—The learning VIDs used for explicit trees are not allocated to the SPVID-Pool-MSTID. Therefore, they are not taken from the SPVID pool; they are not Shortest Path VIDs even if conveyed by an SPVID field of an IS-IS sub-TLV. To emphasize, a learning VID is an SPVID only if it is taken from the SPVID pool.

The Topology sub-TLV (45.1.9) conveys only the VLAN's Base-VID for each ECT Algorithm. Further VIDs, if any, are associated with the VLAN and its Base VID by ISIS-SPB sub-TLVs as specified below.

If the ST ECT Algorithm (Table 45-1) is used for either a learning or a non-learning VLAN, then the VLAN is supported only by its Base VID. The VLAN's Base VID shall be associated with the ST ECT Algorithm in the SPB Base VLAN-Identifiers sub-TLV (28.12.4) and in the SPB Instance sub-TLV (28.12.5).

If the LT ECT Algorithm (Table 45-1) is used for either a learning or a non-learning VLAN, then the VLAN is supported only by its Base VID, which is associated with the LT ECT Algorithm in the SPB Base VLAN-Identifiers sub-TLV (28.12.4) and in the SPB Instance sub-TLV (28.12.5).

If the LTS ECT Algorithm (Table 45-1) is used for a non-learning VLAN, then the VLAN is supported only by its Base VID, which is associated with the LTS ECT Algorithm in the SPB Base VLAN-Identifiers sub-TLV (28.12.4) and in the SPB Instance sub-TLV (28.12.5).

NOTE 2—The use of the LTS ECT Algorithm for a non-learning VLAN is similar to SPBM (Clause 27 and Clause 28) operations, but constrained trees are used instead of SPTs. One VID is sufficient for the support of a set of constrained trees, which is specified as a set of loose explicit trees.

However, if the LTS ECT Algorithm (Table 45-1) is used for a learning VLAN, then the VLAN is supported by multiple VIDs because each tree of an LTS is required to have its own VID. The VLAN's Base-VID is associated with the LTS ECT Algorithm in the SPB Base VLAN-Identifiers sub-TLV (28.12.4) and in the SPB Instance sub-TLV (28.12.5). In addition to the Base VID, as many VIDs are needed to support the VLAN as the number of transmitter Edge Bridges specified by the Topology sub-TLV (45.1.9). The individual VID of a transmitter Edge Bridge is configured to support the VLAN at the given SPT Bridge. The VID to be used for the loose tree of a transmitter Edge Bridge is conveyed by the SPVID field of the VLAN ID Tuple of the corresponding Base VID in the SPB Instance sub-TLV (28.12.5) of the given Edge Bridge. The A flag of the given VLAN ID Tuple is cleared to indicate that auto-allocation is not used for the SPVID parameter. If no VID is configured for the loose tree in the SPB Instance sub-TLV of a transmitter Edge Bridge, then ISIS-PCR does not install the loose tree for that Edge Bridge. If the same local VID value is configured at multiple Bridges, then ISIS-PCR does not install the loose tree for either Bridge, and the conflict has to be resolved by operator action. Although the field is called SPVID, the VID it conveys is not a Shortest Path VID but a loose tree VID as indicated by the ECT Algorithm field of the given VLAN ID Tuple and by the VID to MSTID allocation.

NOTE 3—The use of the LTS ECT Algorithm for a learning VLAN is similar to SPBV (Clause 27 and Clause 28) operations, but with loose trees, i.e., constrained trees are used instead of SPTs. Therefore, each Bridge rooting a loose tree has to have its own VID for the support of a learning VLAN on that tree. The SPB Instance sub-TLV is used to bind the VIDs of the individual root Bridges to the Base VID; the SPVID parameter conveys the VID to be used for the loose tree rooted at the given Bridge. The VID is not a Shortest Path VID because it is used for a loose explicit tree as indicated by the ECT Algorithm parameter and by the VID to MSTID allocation.

If the MRT ECT Algorithm (Table 45-1, 45.3.3) or the MRTG ECT Algorithm (Table 45-1, 45.3.4) is used for a VLAN, then a distinct VID is required for the two MRTs: MRT-Blue and MRT-Red. In the case of a non-learning VLAN, a single VID is used for MRT-Blue of all MRT Roots, and another VID is used for all MRT-Reds. Two VIDs support a non-learning VLAN if the MRTs protect each other, whereas the VID of the SPT Set is also needed as the third VID if the MRTs protect SPTs. The MRT VIDs are also allocated to the SPBM-MSTID (0xFFC) in the case of a non-learning VLAN. However, each MRT Root has to be configured with its own unique VID-pair for its MRT-Blue and MRT-Red in the case of a learning VLAN, and all the MRT VIDs are allocated to the SPBV-MSTID (0xFFD). Independent VLAN Learning (3.94) is applied among the VIDs associated with MRTs. Twice as many MRT VIDs support a learning VLAN as MRT Roots, which are all the VIDs that are needed if the MRTs protect each other. The Shortest Path VIDs (real SPVIDs) of the SPTs are also needed in addition to the MRT VIDs for the support of a learning VLAN if the MRTs protect SPTs. As the two MRT algorithms differ only in GADAG handling, the VLAN configuration is the same for these two ECT Algorithms once the appropriate ECT-ALGORITHM value is used in the ECT Algorithm field of the IS-IS sub-TLVs. The VLAN's Base VID is associated with the MRT ECT Algorithm or with the MRTG ECT Algorithm in the SPB Base VLAN identifier sub-TLV (28.12.4).

The SPB Instance sub-TLV (28.12.5) provides the VIDs for MRT-Blue and MRT-Red and also associates them with the VLAN. The same VID values are used in the corresponding SPVID fields in the SPB Instance sub-TLV of all MRT Roots in the case of a non-learning VLAN; whereas the VID values in the SPVID fields are unique for each MRT Root in the case of a learning VLAN. The SPB Instance sub-TLV also specifies whether the MRTs are used to protect each other or whether they protect an SPT.

The SPB Instance sub-TLV (28.12.5) conveys three VLAN ID Tuples for the VLAN. The ECT Algorithm parameter of the second and the third VLAN ID Tuple is either the MRT ECT Algorithm or the MRTG ECT Algorithm, accordingly. If the MRTs are used to protect SPTs, then the ECT Algorithm parameter of the first VLAN ID Tuple specifies the ECT Algorithm to be used for shortest path computation, e.g., one of the standard symmetric ECT Algorithms (Table 28-1). The first VLAN ID Tuple also specifies the VID to be used for the shortest paths, which is the Base VID for a non-learning VLAN or the Shortest Path VID (allocated to the SPVID-Pool-MSTID) conveyed by the SPVID parameter for a learning VLAN. If the MRTs are used to protect each other, then the ECT Algorithm parameter of the first VLAN ID Tuple is the same as that of the second and the third VLAN ID Tuple. The second and the third VLAN ID Tuples associate the MRT VIDs with the Base VID. The SPVID parameter of the second VLAN ID Tuple provides the VID for MRT-Blue. The SPVID parameter of the third VLAN ID Tuple provides the VID for MRT-Red. In other words, the SPVID parameters of the second and third VLAN ID tuples convey the MRT VIDs, not Shortest Path VIDs. If ISIS-PCR detects that the same local MRT VID value is configured at multiple Bridges, then ISIS-PCR does not install the MRT for either Bridge, and the conflict has to be resolved by operator action.

If the MRTs are used to protect SPTs, then the MRT VIDs differ from the Base VID and also differ from Shortest Path VID conveyed by the SPVID parameter of a learning VLAN's first VLAN ID Tuple. In the case of non-learning VLANs, if the MRTs are used to protect each other, then the VLAN's Base VID is used as the VID for MRT-Blue; consequently, the SPVID parameter of the VLAN's second VLAN ID Tuple conveys the VLAN's Base VID. If the MRTs of a learning VLAN are used to protect each other, then no Shortest Path VID is required for the VLAN; therefore, the SPVID parameter of the VLAN's first VLAN ID Tuple is not used but ignored. The use of the SPB instance sub-TLV for the MRT and MRTG ECT Algorithms is illustrated in Figure 45-3.

	Parameter	Value			
		non-learning VLAN		*learning VLAN*	
		MRTs protect SPT	*MRTs protect each other*	*MRTs protect SPT*	*MRTs protect each other*
	Type	1			
	Length	43			
	CIST Root Identifier	CIST Root Identifier (imported from RSTP or MSTP)			
	CIST External Root Path Cost	CIST External Root Path Cost (imported from RSTP or MSTP)			
	Bridge Priority	Bridge Priority			
	reserved	0			
	V	1 or 0			
	SPSourceID	SPSourceID or 0			
	Number of Trees	3			
VLAN ID Tuple 1	U	1			
	M	1	1	0	0
	A	0	0	1 or 0	0
	reserved	0			
	ECT Algorithm	Symmetric ECT Alg.	MRT or MRTG	Symmetric ECT Alg.	MRT or MRTG
	Base VID	Base VID			
	SPVID	–	–	SPVID	–
VLAN ID Tuple 2	U	1			
	M	1	1	0	0
	A	0			
	reserved	0			
	ECT Algorithm	MRT or MRTG			
	Base VID	Base VID			
	SPVID	MRT-Blue domain VID	MRT-Blue domain VID = Base VID	MRT-Blue local VID	MRT-Blue local VID
VLAN ID Tuple 3	U	1			
	M	1	1	0	0
	A	0			
	reserved	0			
	ECT Algorithm	MRT or MRTG			
	Base VID	Base VID			
	SPVID	MRT-Red domain VID	MRT-Red domain VID	MRT-Red local VID	MRT-Red local VID

Figure 45-3—The use of the SPB Instance sub-TLV for MRT

NOTE 4—The SPVID parameter conveys only a Shortest Path VID if the ECT Algorithm parameter of the given VLAN ID Tuple identifies shortest path operations and the VID is allocated to the SPVID-Pool-MSTID. The SPVID parameter is not a Shortest Path VID if the ECT Algorithm parameter identifies explicit path control, i.e., taken from Table 45-1, which is also indicated by the fact that the VID is not allocated to the SPVID-Pool-MSTID.

An I-SID of a PBBN shall be associated with an explicit tree by the SPBM Service Identifier and Unicast Address sub-TLV (28.12.10) by means of associating the I-SID with a Base VID that is allocated to one of the explicit ECT Algorithms of Table 45-1.

45.1.4 Use of VIDs for strict explicit trees

The use of a distinct VID for each explicit tree does not scale in some cases. A more flexible and scalable method of VID assignment is available for explicit trees. This subclause explains the rules to be observed and suggests two schemes for the use of VIDs for strict explicit trees, i.e., for VIDs associated with the ST ECT Algorithm (Table 45-1). Nonetheless, ensuring that VIDs associated with strict explicit trees actually follow these rules in order to provide unambiguous and loop-free frame forwarding is the responsibility of the network administrator and the owner PCE controlling the VID; this standard provides no method for policing this. As each VID belongs to a single owner PCE, a VID can be used only for multiple explicit trees that are controlled by the same PCE.

NOTE 1—Use of VIDs associated with MRTs is explained in 45.3.3.

Ensuring unambiguous filtering entries and providing forwarding to the appropriate destination are the fundamental requirements to be met when assigning VIDs with explicit trees.

For unicast frames, there has to be a single egress port for each Individual MAC, VID tuple in each SPT Bridge. In other words, different explicit trees associated with the same bidirectional VID within an SPT Domain are not allowed to have any SPT Bridges in common when MAC learning is being used. Unidirectional VIDs or bidirectional non-learning VIDs associated with different explicit trees to a particular destination can have common SPT Bridges along the merged segments of those explicit trees as long as every such tree uses a single egress port for each unicast destination in every Bridge.

NOTE 2—A VID can be unidirectional by means of asymmetric use, e.g., as explained in F.1.3 or like an SPVID (3.220). Also, a bidirectional non-learning VID is in fact used in unidirectional fashion with respect to any given destination MAC address.

For multicast frames, it has to be ensured that each member of the multicast group receives only a single copy of a particular frame, even if there are multiple potential sources within a group.

Congruency (3.180), if required, has to be enforced by the PCE. In the case of unidirectional VIDs, the same path has to be used for both directions between a source and destination pair in order to provide reverse path congruency, essential when MAC learning is employed. Unicast and multicast traffic have to be placed on the same tree in order to provide unicast multicast congruency.

The tree structures explained in 45.1.1 can be applied for strict explicit trees, and VIDs can be used on top of these strict trees as follows.

An ad-hoc explicit tree (45.1.1) with a bidirectional VID provides a shared bidirectional (*,G) tree joining all members of a group. This structure minimizes VID consumption (as a local instance-by-instance optimization) compared to cases when each member has its own VID. On the other hand, there is the constraint that trees using the same VID are not allowed to touch or cross; hence the number of explicit trees is limited by the available VID space.

The template tree (45.1.1) approach constructs one or more explicitly routed forwarding planes with tree sets, one tree rooted on each Bridge within each plane. In the case of learning VIDs, each tree is associated with a unidirectional VID, and Shared VLAN Learning (3.204) takes place among the VIDs of a tree set. In the case of non-learning VIDs, the MAC of the root Bridge identifies the tree; therefore, one VID (the Base VID) is enough for an entire plane. Furthermore, source-specific multicast (S,G) has to be applied if multiple sources participate in a single multicast group as explained in 45.1.1. The desired connectivity can be then laid for VIDs associated with strict template trees with the complete assurance that the required simply connected tree constraint is obeyed.

The efficiency of the template trees approach can be improved substantially in terms of the usage of local VIDs if Edge Bridges “inherit” the tree and VID rooted at their directly connected Core Bridge. This is possible because a loop or forwarding ambiguity cannot be created in a single Ethernet hop; to guarantee this, multi-homed Edge Bridges have to be always configured as non-transit Bridges.

45.1.5 MAC addresses and ISIS-PCR

Propagation of MAC address information by IS-IS is required for explicit trees in order to create Dynamic Filtering Entries for VID, MAC tuples, except for learning VIDs. The SPBV MAC Address sub-TLV (28.12.9) shall be used for the advertisement of both Individual and Group MAC Addresses for S-VLANs and C-VLANs associated with explicit trees. The VLAN’s VID is conveyed by Octets 3 and 4 of the SPBV MAC Address sub-TLV. This local VID (if required) and the VLAN's Base VID are allocated either to the SPBV-MSTID (for learning VIDs) or to the SPBM-MSTID (for non-learning VIDs); furthermore, the Base VID is allocated to one of the explicit ECT Algorithms (Table 45-1) as explained in 45.1.2 in more detail.

NOTE 1—Multiple SPBV MAC Address sub-TLVs are used if different MAC addresses are mapped to different VIDs at an SPT Bridge, i.e., MAC addresses associated with an SPT Bridge are not bound together.

The SPBM Service Identifier and Unicast Address sub-TLV (28.12.10) shall be used to associate Individual addresses with an I-SID; Octets 3 through 8 of the sub-TLV convey an Individual B-MAC address. Thus, the SPT Bridges populate their FDB with the Individual MAC addresses according to their T/R flags for the B-VLANs allocated to the SPBM-MSTID and associated with explicit path control.

Either a Group MAC address associated with an I-SID is source specific, or it is the Backbone Service Instance Group address (26.4). If Octets 3 through 8 of any SPBM Service Identifier and Unicast Address sub-TLV convey the null value throughout the SPT Domain, then the Backbone Service Instance Group address corresponding to the given I-SID is to be used for all multicast sources in (*,G) mode on one simple explicit tree. If no SPBM Service Identifier and Unicast Address sub-TLV with null value in its B-MAC address field is present in the SPT Domain, then the source-specific group addressing specified in 27.15 has to be applied, exactly as used for SPBM. In the source-specific case, the SPB Instance sub-TLV (28.12.5) shall be used to propagate the SPSourceID (27.10) of the Bridges that are Edge Bridges of the explicit tree and can be a source of the multicast traffic. In order to avoid forwarding anomalies, source-specific group addressing has to be used if there are multiple multicast sources for a given I-SID using the same B-VID and the template tree model (45.1.4) is followed, i.e., multiple trees are used.

NOTE 2—Multiple SPBM Service Identifier and Unicast Address sub-TLVs are used if different MAC addresses are mapped to different I-SIDs at an SPT Bridge, i.e., MAC addresses associated with an SPT Bridge are not bound together.

45.1.6 Filtering Database entries for explicit trees

The Topology sub-TLV (45.1.9) provides the information needed for ISIS-PCR to create the appropriate Dynamic VLAN Registration Entries, i.e., it provides the explicit tree, the VIDs associated with the tree, and the direction of the directed VIDs.

The per Bridge association of a MAC address with a VID is provided by the SPBV MAC Address sub-TLV (28.12.9) as described in 45.1.5. Based on the Topology sub-TLVs and SPBV MAC Address sub-TLVs, ISIS-PCR can create the VID, MAC tuple Dynamic Filtering Entries for VLANs associated with explicit trees.

The association of a B-VID with an explicit tree is provided by the Topology sub-TLV (45.1.9), and the per Bridge association of an I-SID to a B-VID is provided by the SPBM Service Identifier and Unicast Address sub-TLV (28.12.10), which also provides the association of a B-MAC to an I-SID as described in 45.1.5. The SPBM Service Identifier and Unicast Address sub-TLVs also make it clear whether source-specific group addressing is to be used. If that is the case, then the SPB Instance sub-TLVs (28.12.5) provide the

SPSourceIDs (27.10) that are needed for the creation of the automatically generated Group MAC addresses of the I-SID as described in 45.1.5. Based on these sub-TLVs, ISIS-PCR can create the B-VID, B-MAC tuple Dynamic Filtering Entries for I-SIDs associated with explicit trees.

45.1.7 ISIS-PCR support

ISIS-PCR and PCEs are configured to use one of the Group addresses from Table 8-14 to establish adjacencies and exchange PDUs. ISIS-PCR running on C-VLAN components use the Customer Bridge address. ISIS-PCR running on S-VLAN components use the Provider Bridge address. ISIS-PCR running on B-VLAN components can use the Provider Bridge Address or one of the existing IS-IS addresses.

The use of the ISIS-PCR adjacency between Bridges is contingent on the Bridges inclusion within the same SPT Region and thus requires that 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 has to be consistent during migration. The MCID and the Auxiliary MCID are exchanged in IS-IS Hello PDUs, which also include the SPB Base VLAN-Identifiers sub-TLV (28.12.4). ISIS-PCR adjacencies are formed and maintained as specified by 28.2, i.e., in exactly the same way as ISIS-SPB adjacencies. Either a PCE hosted by an end station has a PCA that is part of the IS-IS Domain, or the PCE forms and maintains an ISIS-PCR adjacency between the PCE and the SPT Bridge to which the end station is attached.

Management of explicit trees also requires the following:

a) Management of the PCR objects (12.28).
b) Administrative agreement on the MCID Configuration Name and Revision Level (Clause 28).

NOTE—The addresses used by IS-IS PDUs and the operations with respect to IS-IS adjacencies are the same for ISIS-PCR and ISIS-SPB. This subclause (45.1.7) simply reiterates provisions of Clause 27 and Clause 28 (see 27.2, 28.2).

45.1.8 Attributes for path computation

More attributes than the link metric are needed for PCEs or BLCEs in order to be able to determine paths that meet the requirements. Extended link attributes are specified by the Traffic Engineering (TE) extensions for IS-IS in IETF RFC 5305. The TE information is flooded in LSPs by IS-IS and it is stored in the Traffic Engineering Database (TED). The TED contains the topology and the resource information of the IS-IS Domain.

SPT Bridges may support the Extended IS Reachability TLV (type 22) specified in IETF RFC 5305, which provides the following link attribute IS-IS sub-TLVs:

a) Administrative Group (color, resource class) (sub-TLV type 3)
b) Maximum Link Bandwidth (sub-TLV type 9)
c) Maximum Reservable Link bandwidth (sub-TLV type 10)
d) Unreserved Bandwidth (sub-TLV type 11)
e) Traffic Engineering Default Metric (sub-TLV type 18)

SPT Bridges may support the following IS-IS TE Metric Extension link attribute sub-TLVs specified in [R-Qca-2]:

f) Unidirectional Link Delay (sub-TLV type 33)
g) Min/Max Unidirectional Link Delay (sub-TLV type 34)
h) Unidirectional Delay Variation (sub-TLV type 35)
i) Unidirectional Link Loss (sub-TLV type 36)

j) Unidirectional Residual Bandwidth (sub-TLV type 37)
k) Unidirectional Available Bandwidth (sub-TLV type 38)
l) Unidirectional Utilized Bandwidth (sub-TLV type 39)

The IS-IS TE Metric Extensions sub-TLVs are conveyed by IS-IS Extended IS Reachability TLV (type 22). The bandwidth already allocated determines the Residual Bandwidth and the Available Bandwidth. When they are used in a Bridged Network, reservations performed by MSRP or bandwidth assignments recorded by ISIS-PCR are taken into account to determine these sub-TLVs.

NOTE—The Residual Bandwidth and the Unreserved Bandwidth sub-TLVs can be used concurrently; see IETF RFC 7810.

If Shared Risk Link Group (SRLG) information is to be applied, then the SRLG TLV (type 138) of IETF RFC 5307 may be used as specified here. Figure 45-4 shows the format of the SRLG TLV.

		Octet	Length
	Type (138)	1	1
	Length	2	1
	System ID	3-8	6
	Pseudonode num	9	1
	Flag	10	1 bit
	reserved	10	7 bits
	Link Local Identifier	11-14	4
	Link Remote Identifier	15-18	4
SRLG *1*	Shared Risk Link Group Value	19-22	4
	...		
SRLG *n*	Shared Risk Link Group Value	(4*n*+15)-(4*n*+18)	4

Figure 45-4—Shared Risk Link Group (SRLG) TLV

The SRLG parameters are encoded as follows:

a) Type (8 bits). Octet 1 conveys the type of the sub-TLV, and its value is 138.
b) Length (8 bits). Octet 2 conveys the total number of bytes contained in the Value field, which is the number of SRLG Values multiplied by 4 plus 16 bytes.
c) System ID (48 bits). Octets 3 through 8 convey the 6-byte IS-IS System Identifier of the adjacent neighbor.
d) Pseudonode num (8 bits). Octet 9 conveys the pseudonode number if the neighbor is connected through a shared media.
e) Flag (1 bit). Octet 10 conveys a flag, which is encoded as a bit in a single octet. A flag is set if the bit takes the value 1; otherwise, it is reset. The Flag of Octet 10 indicates whether an IP address is used to identify the link. It is not an IP address in the case of ISIS-PCR, i.e., the link is unnumbered. Therefore, the flag is reset in an SPT Domain, i.e., each bit of Octet 10 takes value 0.
f) reserved (7 bits). Seven bits of Octet 10 are reserved for future use, transmitted as 0, and ignored on receipt.
g) Link Local Identifier (32 bits). Octets 11 through 14 convey the local identifier of the interface, which is an Extended Local Circuit ID as specified by IETF RFC 5303.
h) Link Remote Identifier (32 bits). Octets 15 through 18 convey the identifier of the remote interface, which is a Neighbor Extended Local Circuit ID as specified by IETF RFC 5303.
i) Shared Risk Link Group Value (32 bits). Octets $4n$+15 through $4n$+18 convey the value assigned to SRLG *n* to which the link belongs. The same SRLG value is used for all links that share a resource whose failure affects each link of the group. A link can belong to multiple SRLGs; therefore, multiple Shared Risk Link Group Values can be present in the same SRLG TLV.

45.1.9 Topology sub-TLV

The variable-length Topology sub-TLV shall be used to describe an explicit tree. The Topology sub-TLV may be also used for describing a Generalized Almost Directed Acyclic Graph (GADAG) as explained in 45.3.4 in detail. The Topology sub-TLV is carried in an MT-Capability TLV (type 144) in a Link State PDU. A Topology sub-TLV specifying an explicit tree conveys one or more Base VIDs and two or more Hop sub-TLVs (45.1.10) and can convey further sub-TLVs (45.1.11, 45.1.12) to constrain loose hops. The Topology sub-TLV also specifies the algorithm to be used for constrained routing. Figure 45-5 shows the format of the Topology sub-TLV.

		Octet	Length	
	Type (21)	1	1	
	Length	2	1	
	Number of Base VIDs	3	1	
Base VID Tuple *1*	reserved	4	4 bits	0 or 16 bits
	Base VID	4-5	12 bits	
	...			
Base VID Tuple *n*	reserved	(2*n*+2)	4 bits	0 or 16 bits
	Base VID	(2*n*+2) - (2*n*+3)	12 bits	
sub-TLV *1*	sub-TLV			
	...			
sub-TLV *m*	sub-TLV			

Figure 45-5—Topology sub-TLV

The parameters of explicit trees are encoded by the Topology sub-TLV as follows:

a) Type (8 bits). Octet 1 conveys the type of the sub-TLV, and its value is 21.
b) Length (8 bits). Octet 2 conveys the total number of bytes contained in the Value field.
c) Number of Base VIDs (8 bits). Octet 3 conveys the number of Base VIDs carried in the Topology sub-TLV. Its minimum value is 1 if the Topology sub-TLV specifies an active topology. Its value can be 0 if the Topology sub-TLV specifies only a GADAG.
d) reserved (4 bits). Four bits of Octet $2n+2$ are reserved for future use, transmitted as 0, and ignored on receipt.
e) Base VID (12 bits). Octet $2n+2$ and Octet $2n+3$ convey the nth Base VID parameter. The Base VID parameter provides the Base VID of the VLAN that is associated with the explicit tree. Multiple Base VIDs can be associated with the same explicit tree.

   Some of the explicit ECT Algorithms (45.1.2) require further VIDs, in addition to the Base VID, which are associated with the VLAN by using the SPB Instance sub-TLV as specified in 45.1.3.

   By default, each Edge Bridge is both transmitter and receiver for all VIDs associated with an explicit tree. Different behavior can be specified in the Hop sub-TLV (45.1.10) of the Edge Bridge.

   If VIDs of a learning VLAN associated with the LTS ECT Algorithm (Table 45-1) are conveyed by the Topology sub-TLV, then the default behavior for each Edge Bridge is that it is the only transmitter on its own VID and a receiver on the VIDs of all the other Edge Bridges. For such a learning VLAN, a per Bridge local VID is announced by a transmitter Edge Bridge via the SPB Instance sub-TLV (28.12.5) as specified in 45.1.3. If no VID is allocated to the Edge Bridge, then it is not a transmitter but only a receiver.

   A Topology sub-TLV specifying a GADAG can have zero Base VID parameters. In this case, the given GADAG has to be applied for each VLAN associated with the MRTG ECT Algorithm.

f) sub TLVs. The rest of the octets convey further sub-TLVs that specify the hops of the topology and can also specify constraints, bandwidth assignment, and timestamp as specified in the following subclauses.

A topology is specified by a list of Hop sub-TLVs (45.1.10), and a hop is specified by an IS-IS System ID. An ill-formed Topology sub-TLV, e.g., specifying an invalid strict tree or having a conflict in its specification, is ignored; no tree is installed but a management report is generated.

The Topology sub-TLV specifies a strict tree by decomposing the tree to branches. Each branch is a point-to-point path specified by an ordered list of hops where the end of each branch is a leaf. Each element of a branch is the direct link between adjacent neighbor Bridges whose Hop sub-TLV is next to each other in the Topology sub-TLV. The first hop of the Topology sub-TLV is the root; hence, the first branch originates from the root. The rest of the branches fork from another branch. The first hop of a branch is a Bridge that is already part of a former branch, and the last hop is a leaf Bridge. Therefore, the hop after a leaf hop is the beginning of a new branch, if any. A hop of a branch is created if and only if the Bridge specified for that hop is directly connected to the preceding Bridge of the same branch. The order of the branches does not matter within the Topology sub-TLV, but the first branch begins with the root. Figure 45-6 shows an example for the specification of a strict tree.

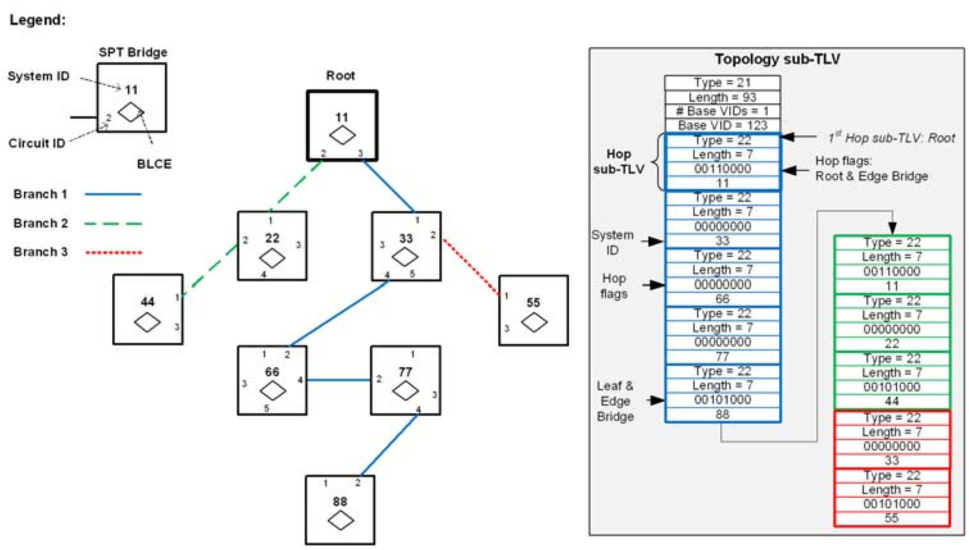

Figure 45-6—A strict tree and its descriptor Topology sub-TLV

The Topology sub-TLV of a loose tree does not provide any path or path segment, but the hops that are to participate. The root is the first hop. The leaves of a single loose tree are also specified. Hop sub-TLVs can be included in a Topology sub-TLV to specify Bridges that have to be avoided. If the Topology sub-TLV specifies only a single leaf, then one or more transit hops can be specified by the Topology sub-TLV to direct the path along a sequence of Bridges, specified by the order of hops. If Bridges whose respective Hop sub-TLVs are adjacent to each other in the Topology sub-TLV but are not topology neighbors, then it is a loose hop. If a Topology sub-TLV conveys one or more loose hops, then that sub-TLV defines a loose explicit tree, and each hop is considered as a loose hop. If a loose tree specified by a Topology sub-TLV would create a loop or hairpin, then the loop is pruned, and the remaining hops are installed. An example for the specification of a single loose tree is shown in Figure 45-7, where Bridge 77 is the root; Bridges 11, 44, 55, and 88 are leaves; and Bridge 22 is excluded from the tree. Bridges 11, 44, 55, and 88 are the Edge Bridges. Bridges 11, 44, 55, 77, and 88 are part of the tree because each of them has its special role as specified by the Topology sub-TLV. Conversely, the role of Bridges 33 and 66 is determined by the BLCE of the SPT Bridges only when the tree is computed after the reception of the Topology sub-TLV.

If the Base VIDs of the Topology sub-TLV are associated with the LTS ECT Algorithm or the MRT ECT Algorithm, then the Hop sub-TLVs conveyed by the Topology sub-TLV belong to Edge Bridges or Bridges to be excluded [45.1.10, item c6)]. The BLCEs compute the loose trees, e.g., MRTs, so that they span the Edge Bridges and are rooted at an Edge Bridge.

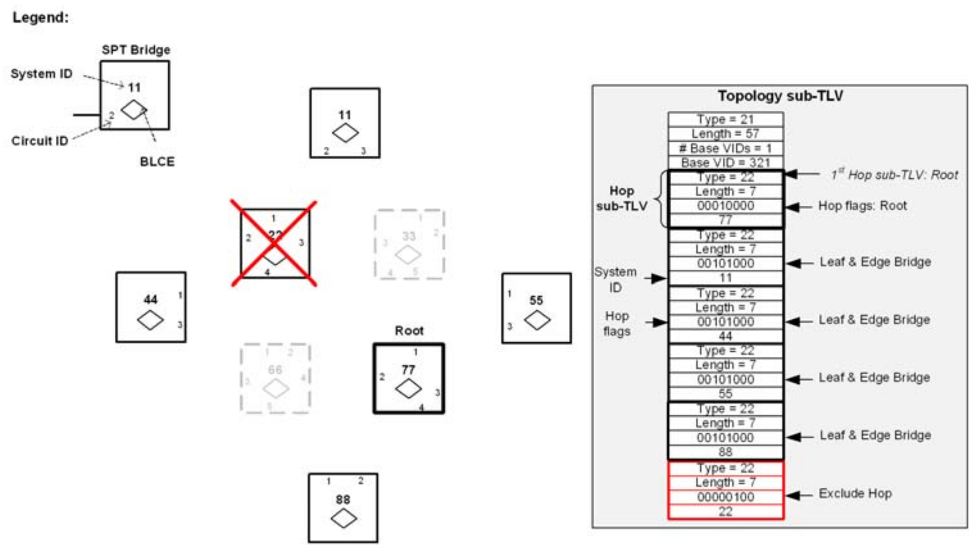

Figure 45-7—Topology sub-TLV of a loose tree

The Topology sub-TLV specifies a GADAG if the Base VIDs conveyed by the Topology sub-TLV are associated with the MRTG ECT Algorithm (Table 45-1). The Topology sub-TLV specifies a GADAG by directed ear decomposition. A directed ear is a directed point-to-point path whose end points can coincide, but no other element of the path is repeated in the ear. Each ear is specified by an ordered list of hops, and that order of hops is according to the direction of the arcs in the GADAG. There are no leaves in a GADAG; hence, the Leaf flag is used to ease the parsing of the Topology sub-TLV and to mark the end of a topology block (3.x). The sequence of ears in the Topology sub-TLV is such that the end points of an ear belong to former ears. The GADAG Root is not marked by any flag, but the GADAG Root is the first hop in the Topology sub-TLV; correspondingly the first ear starts and ends with the GADAG Root. MRT Roots are marked by the Root flag, and all other Edge Bridges are leaves of the MRTs. If no MRT Root is specified, then each SPT Root is also an MRT Root. An example GADAG is shown in Figure 45-13.

Each Edge Bridge of an explicit tree is always specified in the Topology sub-TLV by the inclusion of the Hop sub-TLVs corresponding to the Edge Bridges. The Edge Bridges of a tree are identified by setting the Edge Bridge flag [45.1.10, c) 3)] in the appropriate Hop sub-TLVs.

If the explicit tree is loose, then Administrative Group (45.1.11) and Bandwidth Constraint (45.1.12) sub-TLVs can be also conveyed by the Topology sub-TLV. In that case, each path of the tree has to meet the constraints specified by these sub-TLVs.

If ISIS-PCR is used for recording bandwidth assignment, then the Topology sub-TLV conveys the Bandwidth Assignment sub-TLV (45.2.1), and it can also convey a Timestamp sub-TLV (45.2.2). If the assignment of the bandwidth indicated by the Bandwidth Assignment sub-TLV of the Topology sub-TLV would result in overbooking any link of the explicit tree, then bandwidth assignment is not performed ,and a management report is generated. If such a Topology sub-TLV describes a new valid explicit tree, then the paths of the tree are installed, but no bandwidth assignment is performed by ISIS-PCR.

NOTE 1—The LSP ID can be also considered as an identifier of an explicit tree if an LSP describes only a single tree, i.e., includes a single Topology sub-TLV.

NOTE 2—Allocation of VIDs to individual Aggregation Links between adjacent IS-IS Systems has to be done by the explicit configuration of conversations in that LAG at the two Systems in accordance with IEEE Std 802.1AX™-2014 [B7a].

45.1.10 Hop sub-TLV

The Hop sub-TLV shall be used to specify a hop of a topology. Each Hop sub-TLV conveys an IS-IS System ID, which specifies a hop. A Hop sub-TLV is conveyed by a Topology sub-TLV (45.1.9). A strict explicit tree is decomposed to branches where each branch is a point-to-point path specified by an ordered list of Hop sub-TLVs as specified in 45.1.9 and illustrated in Figure 45-6. A hop of a branch is created if and only if the Bridge specified for that hop is directly connected to the preceding Bridge in the path. In other words, a point-to-point LAN is identified by the two Bridges it interconnects; and the LAN is part of the strict tree if and only if the Hop sub-TLVs of the two Bridges are next to each other in the Topology sub-TLV. A Hop sub-TLV can convey a Circuit ID in order to distinguish multiple links between adjacent neighbor Bridges. A Hop sub-TLV also specifies the role of a Bridge, e.g., if it is the root or an Edge Bridge. The Topology sub-TLV of a loose tree comprises only the Hop sub-TLVs of the Bridges that have special roles in the tree. The Hop sub-TLV may also specify a delay budget for a loose hop.

By default, the Edge Bridges both transmit and receive with respect to each VID associated with an explicit tree, except for an LTS associated with a learning VLAN [45.1.3 and 45.1.9, item e)], which uses a unidirectional VID per Bridge. The Hop sub-TLV allows different configuration by means of the VID's Transmit (T) and Receive (R) flags conveyed in the sub-TLV. The VID and its T/R flags are present in the Hop sub-TLV only if the behavior of the Edge Bridges differs from the default (defined in 45.1.9, item e). The T/R flags of an I-SID associated with an explicit tree in a PBBN are set in the SPBM Service Identifier and Unicast Address sub-TLV (28.12.10), which also provides the association of the I-SID with the B-VID.

Figure 45-8 shows the format of the variable-length Hop sub-TLV, which shall be conveyed by a Topology sub-TLV (45.1.9).

		Octet	Length	
	Type (22)	1	1	
	Length	2	1	
Hop Flags	Circuit	3	1 bit	8 bits
	VID	3	1 bit	
	Edge Bridge	3	1 bit	
	Root	3	1 bit	
	Leaf	3	1 bit	
	Exclude	3	1 bit	
	reserved	3	2 bits	
	System ID	4-9	6	
	Extended Local Circuit ID	10-13	0 or 4	
	Number of VIDs	10 or 14	0 or 1	
VID Tuple *1*	T	11 or 15	1 bit	0 or 16 bits
	R	11 or 15	1 bit	
	reserved	11 or 15	2 bits	
	VID	11-12 or 15-16	12 bits	
	...			
VID Tuple *n*	T	$(2n+9)$ or $(2n+13)$	1 bit	0 or 16 bits
	R	$(2n+9)$ or $(2n+13)$	1 bit	
	reserved	$(2n+9)$ or $(2n+13)$	2 bits	
	VID	$(2n+9)$-$(2n+10)$ or $(2n+13)$-$(2n+14)$	12 bits	
	Delay Constraint	10-15 or 14-19 or $(2n+13)$-$(2n+16)$ or $(2n+17)$-$(2n+20)$	0 or 6	

Figure 45-8—Hop sub-TLV

The parameters of a hop are encoded as follows:

a) Type (8 bits). Octet 1 conveys the type of the sub-TLV, and its value is 22.
b) Length (8 bits). Octet 2 conveys the total number of bytes contained in the Value field.
c) Hop Flags (8 bits). Octet 3 conveys the Hop flags. The Circuit and the VID flags influence the length of the Hop sub-TLV. Table 45-3 summarizes the use of the flags in the case of the different explicit ECT Algorithms.

Table 45-3—Hop sub-TLV flags

		Flag			
		Edge Bridge	**Root**	**Leaf**	**Exclude**
ECT Algorithm	**Strict Tree**	The Bridge is an Edge Bridge.	The Bridge is the root of the tree. (first hop)	The Bridge is a leaf of the tree. (always an Edge Bridge)	—
	Loose Tree	The Bridge is an Edge Bridge. (set for every hop of multi-point trees except for exclude hops)	The Bridge is the root of the tree. (first hop)	The Bridge is a leaf of the tree. (always an Edge Bridge)	The Bridge has to be excluded from the tree.
	Loose Tree Set	The Bridge is an Edge Bridge. (set for every hop of multi-point trees except for exclude hops)	The Bridge roots a tree. (every Bridge that is Edge Bridge)	—	The Bridge has to be excluded from each tree.
	MRT	The Bridge is an Edge Bridge.	The Bridge is an MRT Root. (always an Edge Bridge)	—	The Bridge has to be excluded from the MRTs.
	MRT with GADAG	The Bridge is an Edge Bridge.	The Bridge is an MRT Root. (always an Edge Bridge)	The Bridge is the end of a topology block.	—

1) Circuit flag (1 bit). The Circuit flag indicates whether the Extended Local Circuit ID parameter is present. If the flag is set, then an Extended Local Circuit ID is also included in the Hop sub-TLV.
2) VID flag (1 bit). The VID flag indicates whether one or more VIDs are conveyed by the Hop sub-TLV. If the flag is set, then the Number of VIDs parameter is present and indicates how many VIDs are conveyed by the Hop sub-TLV. If the VID flag is reset, then neither the Number of VIDs parameter nor VIDs are present in the Hop sub-TLV.
3) Edge Bridge flag (1 bit). The Edge Bridge flag indicates whether the given System is an Edge Bridge, i.e., transmitter and/or receiver. If the System is an Edge Bridge, then the Edge Bridge flag is set. The Edge Bridge flag indicates that FDB entries have to be installed for the given hop as specified by its SPBV MAC address sub-TLV or SPBM Service Identifier and Unicast Address sub-TLV.
4) Root flag (1 bit). The Root flag indicates whether the given System is a root of the explicit tree specified by the Topology sub-TLV. If the System is a root of a tree, then the Root flag is set.

   If the Topology sub-TLV specifies a single tree, i.e., the Base VIDs conveyed by the Topology sub-TLV are associated with either the ST ECT Algorithm or the LT ECT Algorithm, then the Root flag is set only for one of the Systems conveyed by the Topology sub-TLV. Furthermore, the first Hop sub-TLV of the Topology sub-TLV conveys the System that is the root of the tree. If the Topology sub-TLV specifies a Loose Tree Set, i.e., the Base VIDs conveyed by the

Topology sub-TLV are associated with the LTS ECT Algorithm (Table 45-1), then the Root flag is set for each Edge Bridge as each of them roots a tree. The first Hop sub-TLV of the Topology sub-TLV conveys a root.

If the Topology sub-TLV is used for MRT operations, i.e., the Base VIDs conveyed by the Topology sub-TLV are associated with either the MRT ECT Algorithm or the MRTG ECT Algorithm (Table 45-1), then the Root flag is set for each MRT Root. If no MRT Root is specified by a Topology sub-TLV specifying a GADAG, then each SPT Root is an MRT Root as well.

If the Base VIDs conveyed by the Topology sub-TLV are associated with the MRTG ECT Algorithm (Table 45-1), then the Topology sub-TLV specifies a GADAG, and the very first Hop sub-TLV specifies the GADAG Root. There is no flag for indicating the GADAG Root.

5) Leaf flag (1 bit). The Leaf flag indicates whether the given System is a leaf of the explicit tree specified by the Topology sub-TLV. If the System is a leaf, then the Leaf flag is set.

   The Leaf flag is used to mark a leaf of a tree only if the Topology sub-TLV specifies a single tree, i.e., the Base VIDs are associated with either the ST ECT Algorithm or the LT ECT Algorithm (Table 45-1).

   The Leaf flag is not used for marking leaves if the ECT Algorithm produces multiple trees because a Bridge that is leaf in one tree can be the root of another tree.

   The Leaf flag is used to indicate the end of a topology block (3.x) if the Topology sub-TLV specifies a GADAG; see 45.1.9, item f) and 45.3.4.

6) Exclude flag (1 bit). The Exclude flag indicates whether the given System has to be excluded from the topology. The Exclude flag and the Root flag cannot be set for a given hop at the same time.
7) reserved (2 bits). Two bits of Octet 3 are reserved for future use, transmitted as 0, and ignored on receipt.

d) System ID (48 bits). Octets 4 through 9 convey the 6-byte IS-IS System Identifier of the Bridge to which the Hop sub-TLV refers.
e) Extended Local Circuit ID (32 bits). Octets 10 through 13 convey the Extended Local Circuit ID as specified by IETF RFC 5303 if Extended Local Circuit ID is present in the Hop sub-TLV, as indicated by the Circuit flag.

   Parallel links corresponding to different IS-IS adjacencies between a pair of neighbor SPT Bridges can be distinguished by means of the Extended Local Circuit ID. The Extended Local Circuit ID is conveyed by the Hop sub-TLV specifying the Bridge nearer to the root of the tree and identifies a circuit that attaches the given Bridge to its neighbor cited by the next Hop sub-TLV of the Topology sub-TLV. The Extended Local Circuit ID can be used only in strict trees.

f) Number of VIDs (8 bits). A single octet conveys the Number of VIDs parameter, which is Octet 10 if the Extended Local Circuit ID parameter is absent whereas it is Octet 14 if the Extended Local Circuit ID parameter is present. If the Hop sub-TLV does not convey VIDs, then the Number of VIDs parameter is not present, as indicated by the VID flag.
g) VID and its T/R flags (14 bits). Two octets convey the VID and its T/R flags parameters. The first one is Octet $2n$+9 and the second one is Octet $2n$+10 if the Extended Local Circuit ID parameter is absent whereas the first one is Octet $2n$+13 and the second one is Octet $2n$+14 if the Extended Local Circuit ID parameter is present.

   The VID and its T/R flags are present in the Hop sub-TLV only if the given Bridge is an Edge Bridge and it behaves differently from the default with respect to that particular VID. (The default behavior is defined in item e of the Topology sub-TLV (45.1.9).)

   1) T flag (1 bit). This bit is the Transmit allowed flag for the VID following the flag.
   2) R flag (1 bit). This bit is the Receive allowed flag, i.e., R flag for the VID following the flag.
   3) reserved (2 bits). Two bits of the first Octet are reserved for future use, transmitted as 0, and ignored on receipt.

4) VID (12 bits). The VID can be a Base VID, an MRT VID, or a VID in support of a learning VLAN associated with the LTS ECT Algorithm.

h) Delay Constraint (48 bits). The last six octets specify a delay constraint if they convey a Unidirectional Link Delay sub-TLV [45.1.8, item f)]. The delay constraint can be used in a Topology sub-TLV that specifies a single loose tree, i.e., the Base VIDs are associated with the LT ECT Algorithm (Table 45-1). If delay constraint is applied, then the loose hop has to fit in the delay budget specified by the Delay parameter of the Unidirectional Link Delay sub-TLV [45.1.8, item f)] conveyed by the Hop sub-TLV. If the Topology sub-TLV specifies a single leaf, then the path between the preceding Hop sub-TLV and the current Hop sub-TLV has to meet the delay budget. If the Topology sub-TLV specifies multiple leaves, then the path between the root and the current Hop sub-TLV has to meet the delay budget. If the tree is used as a reverse congruent tree, then the delay constraint applies in both directions. If the tree is used as a directed tree, then the delay constraint applies in the direction of the tree.

NOTE 1—The T and R flags can be used to establish asymmetric VLANs, e.g., for rooted-multipoint connectivity.

NOTE 2—The Root and Leaf flags specify the role of the Bridges only in the explicit tree, i.e., the active topology. Roots and leaves with respect to data traffic are specified by the T/R flags of VIDs and MAC addresses associated with the Edge Bridges. Roots and leaves of the active topology and data flows do not necessarily coincide, e.g., see Figure 45-7.

45.1.11 Administrative Group sub-TLV

The Administrative Group sub-TLV may be included in a Topology sub-TLV (45.1.9) in order to put a constraint on loose hops. In this case, each LAN included in a loose hop has to be a member of the Administrative Group(s) indicated by the Administrative Group sub-TLV. The format of the Administrative Group sub-TLV is specified by IETF RFC 5305 as shown in Figure 45-9. In other words, the same format is used for the constraint as for the TE attribute; therefore, their comparison is easy.

	Octet	Length
Type (3)	1	1
Length (4)	2	1
Administrative Group	3-6	4

Figure 45-9—Administrative Group sub-TLV

The administrative groups are encoded as follows:

a) Type (8 bits). Octet 1 conveys the type of the sub-TLV, and its value is 3.
b) Length (8 bits). Octet 2 conveys the total number of bytes contained in the Value field. The value of the Length field is 4.
c) Administrative Group (32 bits). Octets 3 through 6 convey the Administrative Group parameter. It comprises 32 flags, and each flag belongs to a distinct group. Group membership is indicated by the flags set.

45.1.12 Bandwidth Constraint sub-TLV

The Bandwidth Constraint sub-TLV may be included in a Topology sub-TLV (45.1.9) in order to specify how much available bandwidth is to be provided by the tree. Each loose hop has to meet the bandwidth constraint. The bandwidth value of the constraint is a total value, or it refers only to a single PCP as specified by the sub-TLV. Figure 45-10 shows the format of the Bandwidth Constraint sub-TLV.

	Octet	Length
Type (23)	1	1
Length (5)	2	1
PCP	3	3 bits
DEI	3	1 bit
PCP flag	3	1 bit
reserved	3	3 bits
Available Bandwidth	4-7	4

Figure 45-10—Bandwidth Constraint sub-TLV

The parameters of the bandwidth constraint are encoded as follows:

a) Type (8 bits). Octet 1 conveys the type of the sub-TLV, and its value is 23.
b) Length (8 bits). Octet 2 conveys the total number of bytes contained in the Value field, The value of the Length field is 5 bytes.
c) PCP (3 bits). Octet 3 conveys the PCP parameter, which determines the traffic class to which the Available Bandwidth parameter refers.
d) DEI (1 bit). Octet 3 also conveys the DEI parameter.

   If the DEI parameter is clear, then the bandwidth constraint refers to committed information rate. If the DEI parameter is set, then the bandwidth constraint refers to peak information rate.

e) PCP flag (1 bit). Octet 3 also conveys the PCP flag. If it is set, then the PCP parameter is taken into account.
f) reserved (3 bits). Three bits of Octet 3 are reserved for future use, transmitted as 0, and ignored on receipt.
g) Available Bandwidth (32 bits). Octets 4 through 7 convey the Available Bandwidth. The Available Bandwidth is specific to the traffic class identified by the PCP parameter if the PCP flag is set; otherwise, it is total bandwidth. In line with the bandwidth parameters specified in IETF RFC 5305, the Available Bandwidth is encoded as a 32-bit IEEE floating point number, and the units are bytes (not bits!) per second. Thus, the Available Bandwidth constraint applied for a traffic class is easily comparable with the Unreserved Bandwidth stored in the TED for the given traffic class (see sub-TLV 11 specified by IETF RFC 5305).

   The bandwidth constraint applies for both directions in the case of symmetric (3.233) explicit trees. Nevertheless, a VID associated with an explicit tree can be made unidirectional by means of the T/R flags belonging to the VID in the Hop sub-TLV (45.1.10) of the Edge Bridges. If all the VIDs of the Topology sub-TLV (45.1.9) are unidirectional and all belong to the traffic class identified by the PCP parameter of the Bandwidth Constraint sub-TLV, then it is enough to meet the bandwidth constraint in the direction applied for those VIDs.

NOTE—The peak information rate is the committed information rate plus the excess information rate.

45.2 Reservation

ISIS-PCR may be used for recording bandwidth assignment for explicitly placed data traffic in an SPT Region if MSRP is not used within the region. If MSRP is used in an SPT Region, then only MSRP performs reservations and IS-IS does not.

45.2.1 Bandwidth Assignment sub-TLV

ISIS-PCR can record bandwidth assignment for explicitly placed data traffic. ISIS-PCR can be used for recording bandwidth assignment only within an SPT Region where MSRP is not used.

If precedence order has to be determined among bandwidth assignments in an SPT Region with multiple PCEs, then ISIS-PCR provides it as specified in 45.2.3.

The Bandwidth Assignment sub-TLV can be used to define the amount of bandwidth whose assignment is to be recorded by ISIS-PCR at each hop of the explicit tree described by the corresponding Topology sub-TLV (45.1.9). The Bandwidth Assignment sub-TLV is conveyed by a Topology sub-TLV (45.1.9).

The Bandwidth Assignment sub-TLV is used by ISIS-PCR for the recording of bandwidth assignment for a traffic class. Figure 45-11 shows the format of the Bandwidth Assignment sub-TLV.

	Octet	Length
Type (24)	1	1
Length (5)	2	1
PCP	3	3 bits
DEI	3	1 bit
Importance	3	3 bits
reserved	3	1 bit
Bandwidth	4-7	4

Figure 45-11—Bandwidth Assignment sub-TLV

The parameters for bandwidth assignment recorded by ISIS-PCR are encoded as follows:

a) Type (8 bits). Octet 1 conveys the type of the sub-TLV, and its value is 24.
b) Length (8 bits). Octet 2 conveys the total number of bytes contained in the Value field. The value of the Length field is 5.
c) PCP (3 bits). Octet 3 conveys the PCP parameter, which specifies the traffic class for which the bandwidth is to be assigned.
d) DEI (1 bit). Octet 3 also conveys the DEI parameter.

   If the DEI parameter is clear, then the bandwidth assignment is performed to provide committed information rate.

   If the DEI parameter is set, then the bandwidth assignment is performed to provide peak information rate.
e) Importance (3 bits). Octet 3 also conveys the Importance parameter for determining precedence order among bandwidth assignments within a PCP as specified in 45.2.3. A lower numerical value indicates a more important bandwidth assignment within a PCP. The default value of the Importance parameter is 7.
f) reserved (1 bit). One bit of Octet 3 is reserved for future use, transmitted as 0, and ignored on receipt.
g) Bandwidth (32 bits). Octets 4 through 7 convey the amount of bandwidth to be assigned for the traffic class identified by the PCP parameter. In line with the bandwidth values specified in IETF RFC 5305, the Bandwidth parameter is encoded as a 32-bit IEEE floating point number, and the units are bytes (not bits!) per second.

   The bandwidth assignment applies for both directions in the case of symmetric (3.233) explicit trees. Nevertheless, a VID associated with an explicit tree can be made unidirectional by means of the T/R flags belonging to the VID in the Hop sub-TLV (45.1.10) of the Edge Bridges. If all the VIDs having the same PCP in the Topology sub-TLV (45.1.9) are unidirectional, then the bandwidth is assigned only in the direction applied for those VIDs.

If bandwidth is to be assigned both for committed information rate and for peak information rate, then the Topology sub-TLV conveys two Bandwidth Assignment sub-TLVs: one with DEI cleared and another with DEI set.

45.2.2 Timestamp sub-TLV

The Timestamp sub-TLV may be included in a Topology sub-TLV (45.1.9) in order to provide precedence order among equally important bandwidth assignments within a PCP as specified in 45.2.3. Figure 45-12 shows the format of the Timestamp sub-TLV.

Field	Octet	Length
Type (25)	1	1
Length (4)	2	1
Time	3-6	4

Figure 45-12—Timestamp sub-TLV

The timestamp represents a positive time with respect to the Precision Time Protocol (PTP) epoch, and it is encoded as follows:

a) Type (8 bits). Octet 1 conveys the type of the sub-TLV, and its value is 25.
b) Length (8 bits). Octet 2 conveys the total number of bytes contained in the Value field. The value of the Length field is 4.
c) Time (32 bits). Octets 3 through 6 convey the time in units of seconds with respect to the PTP epoch.

NOTE—The Timestamp sub-TLV carries the seconds portion of PTP as specified by IEEE Std 1588. The epoch is 1970-01-01 00:00:00 TAI (i.e., the PTP time does not include leap seconds).

45.2.3 Precedence ordering

The PCEs are collectively responsible for making a consistent set of bandwidth assignments when ISIS-PCR is used for recording bandwidth allocations. Despite that responsibility, if precedence ordering is required among bandwidth assignments, then ordering based on the following parameters has to be applied:

1) PCP parameter of Bandwidth Assignment Sub-TLV,
2) Importance parameter of Bandwidth Assignment Sub-TLV,
3) Timestamp sub-TLV (if present in the Topology sub-TLV).

A bandwidth assignment takes precedence if it has higher PCP, or higher Importance within a PCP, or earlier timestamp in the case of equal Importance within a PCP. A bandwidth assignment associated with a timestamp takes precedence over a bandwidth assignment without timestamp when PCP and Importance of different bandwidth assignments are both equal.

If resolution is not possible based on the above parameters or they are not available, e.g., each bandwidth assignment lacks timestamp or the precedence order has to be determined for the use of a VID, then the item is granted to the PCE whose LSP has the numerically least LSP ID.

45.3 Redundancy

This subclause specifies IS-IS extensions in support of redundancy schemes, i.e., protection and restoration for data flows.

Local protection for unicast data flows based on loop-free alternates is described in 45.3.1.

Protection schemes are often based on two or more redundant trees, or if that is not possible, then on maximally redundant trees. Maximally Redundant Trees (MRTs) can be used either in a static fashion or with cautious restoration for ISIS-PCR. This subclause specifies how MRTs are installed and maintained leveraging the tools provided by 45.1. The use of static MRTs is described in 45.3.2. Setting up and maintaining MRTs with cautious restoration is described in 45.3.3 and 45.3.4.

45.3.1 Loop-free alternates for unicast data flows

Downstream Loop-Free Alternates (LFAs) provide a simple redundancy scheme, which may be supported by ISIS-SPB for unicast flows of an SPBM VLAN and for SPBV VLANs. A path toward an individual destination MAC address is downstream if it is ensured that the distance to the destination decreases at each hop along the path. Downstream paths are always loop-free alternates to each other with respect to an individual destination MAC address. Therefore, a downstream alternate path can be safely used for local protection as a loop-free backup path.

NOTE—The Alternate Ports of RSTP and MSTP leverage the loop-free alternate paths provided by the physical topology to the Root Bridge. In the case of spanning trees, ‘up’ or ‘upward’ to the Root is used to refer to the same direction as ‘downstream’ for destination rooted trees.

LFA can be provided for SPBV by SPTs (13.17, Clause 27, Clause 28). The LFA mechanism is the same for an SPT as for an MSTI where Alternate Ports leverage LFA paths.

Loop mitigation (6.5.4.2) may be relaxed in order to allow LFAs for unicast SPBM traffic. Ingress checking is modified and relaxed in order to allow the reception of a unicast frame from any SPT Bridge that is upstream with respect to the given individual destination MAC address. Thus, relaxed ingress checking implements a split horizon for loop mitigation by ensuring that a frame is admitted only if it is received from an SPT Bridge that is farther from the destination than the receiving SPT Bridge, which determines the valid set of ingress ports.

In the case of a failure at an SPT Bridge’s Port or LAN on the shortest path toward individual MAC addresses, the SPT Bridge acts as a Point of Local Repair (PLR) and forwards frames through ports providing downstream LFA paths toward the given individual MAC addresses, if any.

An SPT Bridge updates its relaxed ingress checking after detecting a mismatch in the Agreement Digest. An ingress port is removed from the valid set of ingress ports with respect to a particular individual destination MAC address if a formerly upstream SPT Bridge is no longer farther from the destination than the given SPT Bridge. An LFA path is removed if the corresponding formerly downstream neighbor is not closer any more to the destination than the given SPT Bridge.

An SPT Bridge updates its LFA paths and relaxed ingress checking after finishing the updates of multicast states. If LFA paths with respect to an individual MAC address have been changed, then the new LFA paths are added. The valid set of ingress ports is extended with the new upstream Bridges with respect to an individual MAC address, if any.

45.3.2 Static redundant trees

The trees associated with the ST ECT Algorithm (00-80-C2-17, Table 45-1) are static in the sense that no entity other than the owner PCE can update them. In other words, ISIS-PCR only installs the strict trees and does not take any further action on them; i.e., ISIS-PCR neither restores them nor maintains them. In the case of a topology change, it is the task of the owner PCE to detect the topology change, e.g., based on the changes in its LSDB, and also update the strict trees if needed. This requires the owner PCE to compute the new tree and to assemble the corresponding new Topology sub-TLV (45.1.9). The PCE then sends the updated LSP conveying the new Topology sub-TLV, and ISIS-PCR updates the strict explicit tree. The owner PCE can withdraw an explicit tree by sending an updated LSP that does not include the Topology sub-TLV. If a Topology sub-TLV is removed from an LSP (or has been changed), so that (previous) Topology sub-TLV is no longer present (or has been changed) in the LSDB, then that (previous) Topology sub-TLV is implicitly withdrawn. ISIS-PCR then removes (or updates) the explicit tree.

NOTE 1—The PCE preferably does not initiate the update of explicit trees if there is any reason to suspect that the SPT Bridges of the SPT Region do not share a common view on network topology.

NOTE 2—The PCE can have a hot standby backup PCE in order to avoid a single point of failure.

Strict explicit trees can be used for Maximally Redundant Trees (MRTs), especially if more than two redundant trees are needed. Each tree is described by a distinct Topology sub-TLV (45.1.9). The VLAN configuration of strict trees is specified in 45.1.3.

The PCE or its PCA assembles and floods the corresponding Topology sub-TLVs (45.1.9); thus ISIS-PCR installs the explicit trees throughout the SPT Region as specified in 45.1. All of the maximally redundant trees are installed independently of each other, i.e., ISIS-PCR is not aware of any relationship between them. Maintaining the association among the redundant strict trees is not the responsibility of ISIS-PCR, but the entity that requests and uses them, e.g., the owner PCE and the management entity responsible for configuring the protection scheme in use. Therefore, no parameters for association of static explicit trees are present in the ISIS-PCR sub-TLVs.

NOTE 3—For instance, two static redundant trees can be used for CFM-based protection switching as specified in 26.10, where the association and the use of the paths are provided. Static redundant trees can be also used for 1+1 types of protection schemes, where a copy of each data frame is forwarded on each of the redundant paths.

The algorithm used by the owner PCE for the computation of static redundant trees is not specified by this standard. Number K, the number of trees to be computed, is an input for the owner PCE. The Topology sub-TLVs (45.1.9) are the output of the owner PCE. Other outputs can include the number of common links and their identity.

45.3.3 Maximally Redundant Trees (MRTs)

SPT Bridges may support MRTs. The BLCE of SPT Bridges is also involved in the computation of MRTs. The MRTs can be, for example, used for point-to-multipoint protection if the MRTs are source rooted or for multipoint-to-point protection if the MRTs are destination rooted. The MRTs can be used to protect an SPT rooted at the MRT Root. The MRTs can be also used so that they protect each other, e.g., in a 1+1 type of protection scheme where data frames are sent on both MRTs. An MRT provides an undirected active topology just like an SPT or a spanning tree instance (which is also a shortest path tree rooted at the spanning tree root). Correspondingly, MRTs can be used for bidirectional or unidirectional traffic, as determined by the T/R parameters of the data traffic carried on the tree.

The MRT ECT Algorithm (00-80-C2-18, Table 45-1) is used for the establishment and maintenance of MRTs in a distributed fashion as specified by this subclause. The MRT Lowpoint Algorithm specified by [R-Qca-3] has to be used for the computation of MRTs that span more than two Edge Bridges. If only two Edge Bridges are to be connected, then the algorithm specified in this subclause has to be used for path computation. The MRT Lowpoint Algorithm produces two MRTs for an MRT Root: MRT-Blue and MRT-Red. If the MRTs are used in a primary and backup fashion, then MRT-Blue is the primary and MRT-Red is the backup. The MRTs are cautiously restored after a topology change as described in this subclause.

NOTE 1—If more than two maximally redundant trees are required, then static redundant trees (45.3.2) have to be used.

When the MRT Lowpoint Algorithm is used by ISIS-PCR, then the input parameters of the Interface_Compare function are as follows: The Interface_compare_metric is the SPB Link Metric (28.12.7), and the maximum metric value is used in cases where the metrics advertised by adjacent Bridges for a given link are different. The Interface_compare_unique_node_identifier is the 8-byte Bridge Identifier, which comprises a 6-byte IS-IS System Identifier and a 2-byte Bridge Priority.

The MRT Lowpoint Algorithm first computes the GADAG, and the computation requires the selection of the GADAG Root. The Bridge with the best Bridge Identifier is selected as the GADAG Root, where the numerically lower value indicates the better identifier. The manageable Bridge Priority component of the Bridge Identifier allows the configuration of the GADAG Root. The Bridge Priority is conveyed by Octets 15 and 16 of the SPB Instance sub-TLV (28.12.5). Based on its LSDB, each SPT Bridge can locally determine which Bridge is the GADAG Root and then compute the GADAG. The MRTs are then computed based on the GADAG. Examples for a GADAG and two corresponding MRTs are shown in Figure 45-13 and Figure 45-14, respectively. If tie-breaking is required among equal cost paths while the MRTs are computed, then the tie-breaking specified by the LowPATHID algorithm (28.7) has to be used. If MRTs support a VLAN having multiple multicast sources, then the BLCEs of the Bridges have to compute the MRTs of other Bridges in addition to their own MRTs due to the need to apply source-specific (S,G) multicast. This is similar to the application of All Pairs Shortest Path computation in ISIS-SPB. (See 45.1.1 for further details on source-specific (S,G) multicast.)

The MRT Lowpoint Algorithm is specified in terms of determining next hop FDB entries toward a particular unicast destination. This generates a tree directed to the root, which is the destination. By inverting the direction of this destination rooted tree, a source rooted tree can be retrieved.

If MRTs are required for a VLAN so that each Bridge performs all the MRT computation steps on its own including GADAG computation, i.e., the MRT computation is fully distributed, then the VLAN has to be associated with the MRT ECT Algorithm (Table 45-1) as specified in 45.1.3.

Just using the MRT ECT Algorithm creates MRT-Blue and MRT-Red for each SPT Root, i.e., each SPT Root is also an MRT Root, and IS-IS maintains all the MRTs. Furthermore, the MRTs are spanning trees in this case, i.e., each Bridge of the SPT Region is included in both MRT-Blue and MRT-Red. This operation mode is selected by providing the VLAN configuration in the SPB Base VLAN Identifiers sub-TLV (28.12.4) and in the SPB Instance sub-TLV (28.12.5) as specified in 45.1.3. The Topology sub-TLV (45.1.9) is not used in this case, i.e., only ISIS-SPB sub-TLVs are used.

If the level of redundancy provided by each SPT Root also being an MRT Root is not required, then the MRT Roots can be specified by a Topology sub-TLV (45.1.9); this approach can be also applied when MRTs protect each other and SPTs are not used. The Topology sub-TLV describes loose explicit trees in this case, i.e., it specifies only the Edge Bridges that the MRTs are to span, and each Edge Bridge is a loose hop. An MRT Root is specified by setting the Root flag [45.1.10, item c4)] in the corresponding Hop sub-TLV (45.1.10). An MRT Root is typically an Edge Bridge too. MRTs provide a path from each Edge Bridge to the MRT Root, i.e., each Edge Bridge is a leaf except for the MRT Root. If a Topology sub-TLV specifies multiple MRT Roots, then an Edge Bridge has different roles in the MRTs rooted at different Bridges. Therefore, the Leaf flag [45.1.10, item c5)] is not used to mark leaves of MRTs. The Topology sub-TLV can include a hop with the Exclude flag [45.1.10, item c6)] set for a Bridge that is to be avoided by the MRTs, in which case the Bridge is excluded from the GADAG, too. The Topology sub-TLV can convey one or more Base VIDs that are associated with the MRT ECT Algorithm. Whether the MRTs are unidirectional or bidirectional is determined by the T/R flags corresponding to the VIDs, I-SIDs, and MAC addresses associated with the given MRTs.

NOTE 2—For instance, if a VLAN is supported by ISIS-SPB in SPBM mode (Clause 27, Clause 28) and MRTs are used for the protection of SPTs, then only three VIDs are required to support the VLAN. The Base VID is used for all the SPTs. Another VID is used for all MRT-Blues, and a third VID is used for all MRT-Reds. The MRTs are then used as destination rooted trees for unicast traffic and as source rooted trees for multicast traffic just like the SPTs. The T/R parameters for the MRT VIDs are exactly the same as for the Base VID; therefore, IS-IS populates the FDBs for the MRTs the same way as for the SPTs, i.e., just uses the corresponding MRT VID instead of the Base VID.

If the Topology sub-TLV specifies only two Edge Bridges, then the following algorithm is used for the computation of the two maximally redundant paths instead of the MRT Lowpoint Algorithm. As they interconnect only two Edge Bridges, the MRTs are reverse path congruent maximally redundant paths in this case. According to the algorithm below, one of the two maximally redundant paths is the shortest path. The two paths are used to protect each other. A VLAN associated with maximally redundant paths has to be configured as specified in 45.1.3 for the MRT ECT Algorithm. The VID of MRT-Blue is used for the shortest path, and the VID of MRT-Red is used for the other path. The establishment of the redundant paths is initiated by the flooding of a Topology sub-TLV (45.1.9). The paths are then determined by the BLCE of the SPT Bridges and then installed by ISIS-PCR. The Topology sub-TLV can convey constraint sub-TLVs, which have to be also taken into account when determining the paths. The BLCEs perform the following path computation algorithm:

1) Prune the topology according to the constraints of the Topology sub-TLV (45.1.9), if any. The rest of the computation steps are performed on the pruned topology. If pruning splits the SPT Region, then the outcome can be that no path is available.
2) Run Dijkstra to get the (constrained) shortest path, which is the primary path.
3) Install the FDB entries corresponding to the VID of MRT-Blue along the primary path.
4) Update th network graph by the appropriate method as follows:
   a) Prune the links of the primary path if that does not partition the SPT Region.
   b) Otherwise, multiply by 1000 the link metric of the links taking part in the primary path.
5) Run Dijkstra to get the backup path.
6) Install the FDB entries corresponding to the VID of MRT-Red along the backup path.

It is recommended to use distinct VIDs for MRT-Blue and MRT-Red, which are loose trees and therefore computed by multiple entities. As a strict tree is controlled by a single PCE, it is easier to ensure unambiguous forwarding and allow the use of the same VID for different strict trees as described in 45.1.4 than to do so for loose trees.

MRTs have to be restored cautiously after a topology change. The restoration should not be initiated if there is any reason to suspect that the SPT Bridges of the SPT Region do not share a common view on the network topology. BLCEs then perform path computation, and MRTs are updated as follows.

If MRTs are used for SPT protection, then MRTs are restored only after the restoration of the SPTs. Upon a failure event, Point of Local Repair (PLR) Bridges redirect data traffic to the MRT that is not affected by the failure, if any. As the topology change is announced by the corresponding LSPs, the SPT Bridges compute and install the new SPTs. If each SPT Bridge has installed the new SPT, then data traffic is directed back to the SPTs. After that, the new MRTs are computed and installed by the Bridges.

If MRTs support a non-learning VLAN, then the PLRs can redirect traffic by replacing the FDB entry belonging to the tree of normal operation (i.e., the SPT or an MRT) with the FDB entry belonging the protection tree (i.e., with MRT-Blue or MRT-Red) upon a failure event. If MRTs support a learning VLAN, then the PLRs can redirect traffic by updating VID translation at the Ingress Port from the VID of normal operation to the VID of the protection tree (i.e., to MRT-Blue or MRT-Red) upon a failure event.

If MRTs protect each other, then only one of the MRTs is restored at a time. Upon a failure event, PLR Bridges direct all data traffic to the MRT (e.g., MRT-Red) that is not affected by the failure. If a 1+1 type of protection scheme is used, then there is no need to react to a failure event as data frames are forwarded on both MRTs, i.e., there are no PLRs. When each SPT Bridge has the same view on the network topology again, then the SPT Bridges compute the new MRTs. The broken MRT (e.g., MRT-Blue) then is restored while the other MRT (e.g., MRT-Red) is not touched. If each SPT Bridge has installed the new MRT (e.g., the new MRT-Blue), then data traffic is directed to this newly restored MRT. The SPT Bridges then install the other MRT (e.g., the new MRT-Red). The same restoration procedure is followed if both MRTs are affected by the failure so that MRT-Blue is restored first.

Loop prevention (6.5.4.1, 27.9) has to be applied for MRTs during the period when they are being restored by ISIS-PCR.

There is a non-zero probability of out-of-order frame delivery while ISIS-PCR restores loose explicit trees, e.g., MRTs. This probability is similar to the probability of out-of-order frame delivery while ISIS-SPB restores SPTs.

If there are two maximally redundant paths between a pair of Edge Bridges and only the backup path went down, then just Steps 4-6 of the above algorithm are performed again. If only the primary path went down, then data traffic takes the backup path, i.e., only the backup path is in use by the applied protection mechanism. The primary path is then updated according to Steps 1-3. No further steps are taken until the primary path is up again. The protection scheme can then revert to the primary path, and the backup path can be updated according to Steps 4-6. If both paths went down, then Steps 1-6 are performed again.

NOTE 3—The method to be used for verifying whether the SPT Bridges have a common view on the topology and whether SPTs and MRTs have been restored is left to the implementer and the network operator. A common method is to wait some time after the reception of the last LSP indicating a topology change. Alternatively, OAM can be used for instance.

45.3.4 MRTs with centralized GADAG computation

The GADAG is identical for all the MRTs within a network domain, as a consequence of the use of the MRT Lowpoint Algorithm (IETF RFC 7811). The GADAG is computed by a single entity for VLANs associated with the MRTG ECT Algorithm (Table 45-1). This entity is referred to as the GADAG Computer and is either a PCE or the BLCE of the GADAG Root.

SPT Bridges may support the MRTG ECT Algorithm (00-80-C2-19, Table 45-1). MRTs are then computed by the SPT Bridges based on the GADAG provided by the GADAG Computer in a Topology sub-TLV (45.1.9). The GADAG and the MRTs are computed as specified by the MRT Lowpoint Algorithm (IETF RFC 7811). The MRTs produced by the MRTG ECT Algorithm are cautiously restored after a failure event as described in this subclause. A VLAN is associated with the MRTG ECT Algorithm as specified in 45.1.3 if MRTs with central GADAG computation are to be used for that VLAN.

The GADAG Root is determined based on the Bridge Identifier as described in 45.3.4, which also determines the sole SPT Bridge computing the GADAG if the GADAG Root is the GADAG Computer. Otherwise, the GADAG Computer PCE determines the GADAG Root, which also allows direct configuration of the GADAG Root.

The GADAG Computer performs the GADAG computation as specified by the MRT Lowpoint Algorithm (IETF RFC 7811). The GADAG Computer then encodes the GADAG in a Topology sub-TLV (45.1.9), which is then flooded throughout the domain. A GADAG is encoded in a Topology sub-TLV by means of directed ear decomposition as follows: A directed ear is a directed point-to-point path whose end points can coincide, but no other element of the path is repeated in the ear. Each ear is specified by an ordered list of hops, and that order of hops is according to the direction of the arcs in the GADAG. There are no leaves in a GADAG; hence, the Leaf flag [45.1.10, item c5)] is used to ease the parsing of the Topology sub-TLV and to mark the end of a topology block (3.x). The sequence of ears in the Topology sub-TLV is such that the end points of an ear belong to former ears. The GADAG Root is not marked by any flag, but the GADAG Root is the first hop in the Topology sub-TLV; correspondingly the first ear starts and ends with the GADAG Root. MRT Roots are marked by the Root flag [45.1.10, item c4)], and all other Edge Bridges are leaves of the given MRTs. If no MRT Root is specified, then each SPT Root is also an MRT Root. If a Bridge has to be excluded from the MRTs, then it is not part of the GADAG.

Figure 45-13 shows the GADAG for the network topology depicted in Figure 45-1 and Figure 45-2 when SPT Bridge 11 is the GADAG Root. The graph shown in Figure 45-13 is in fact an ADAG, and removing the arc going from Bridge 33 to Bridge 11 (the GADAG Root) makes it a DAG. Figure 45-13 also shows the Topology sub-TLV that specifies the GADAG. The first directed ear starts and ends at the GADAG Root, i.e., Bridge 11. The second ear (Ear 2) starts at Bridge 22 and ends at Bridge 33. The third ear (Ear 3) comprises Bridges 66, 88, and 77. Each of the remaining arcs is an individual ear added by a pair of Hop sub-TLVs according to the direction of the arc. Bridges 11, 44, 55, and 88 are Edge Bridges. Bridge 55 is the only MRT Root.

The MRTs are then computed by the SPT Bridges as specified by the MRT Lowpoint Algorithm (IETF RFC 7811) based on the GADAG descriptor received from the GADAG Computer. Whether the MRTs are unidirectional or bidirectional is determined by the T/R flags corresponding to the VIDs, I-SIDs, and MAC addresses associated with the given MRTs. Figure 45-14 shows MRT-Blue and MRT-Red if the GADAG descriptor Topology sub-TLV of Figure 45-13 is used for MRT computation, i.e., SPT Bridge 55 is the only MRT Root. Figure 45-14 also shows that an Edge Bridge is not necessarily a leaf; see Bridge 11.

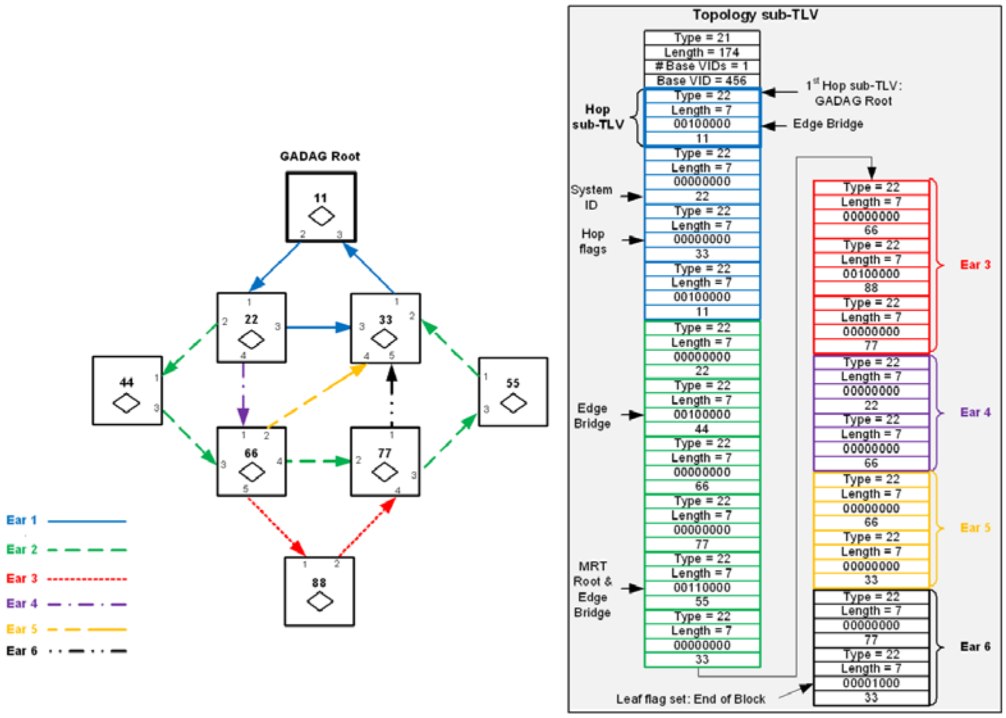

Figure 45-13—A GADAG and its descriptor Topology sub-TLV

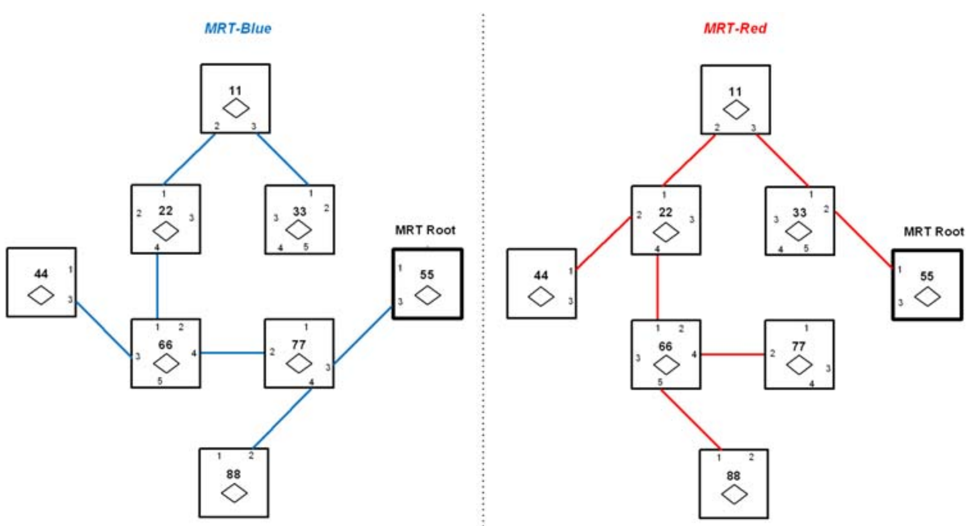

Figure 45-14—MRT-Blue and MRT-Red for MRT Root 55

The Topology sub-TLV specifying the GADAG can convey Base VIDs associated with the MRTG ECT Algorithm, which specifies the VLANs for which the GADAG is relevant. If there is no Base VID parameter in the Topology sub-TLV specifying a GADAG, then the given GADAG has to be used for all VLANs whose Base VID is associated with the MRTG ECT Algorithm.

A network topology can comprise multiple topology blocks (3.x), like the one illustrated in Figure 45-15, which comprises four blocks because each cut-link with the two Bridges at either end is a block. A GADAG is also shown in Figure 45-15. Note that two arcs with opposite direction represent a cut-link in a GADAG, e.g., see the cut-link between Bridges 75 and 77. The encoding starts with the block (ADAG) involving the GADAG Root as illustrated in Figure 45-15. The first hop in the Topology sub-TLV is the GADAG Root (Bridge 11 in this example). The ADAG of the first block is then described using the ear decomposition, as described above. In this example, the first block has been completely traversed at the second occurrence of Bridge 11 in the GADAG descriptor. The end of a block is indicated by setting the Leaf flag for the last hop of the block, e.g., for the second occurrence of Bridge 11 in the example GADAG descriptor. The next Bridge that appears in the GADAG descriptor (Bridge 44 in this case) is the localroot for the Bridges in the next block. Continuing this process, the Leaf flag is set for the third occurrence of Bridge 44, the third occurrence of Bridge 75, and the third occurrence of Bridge 77, each indicating the end of a block.

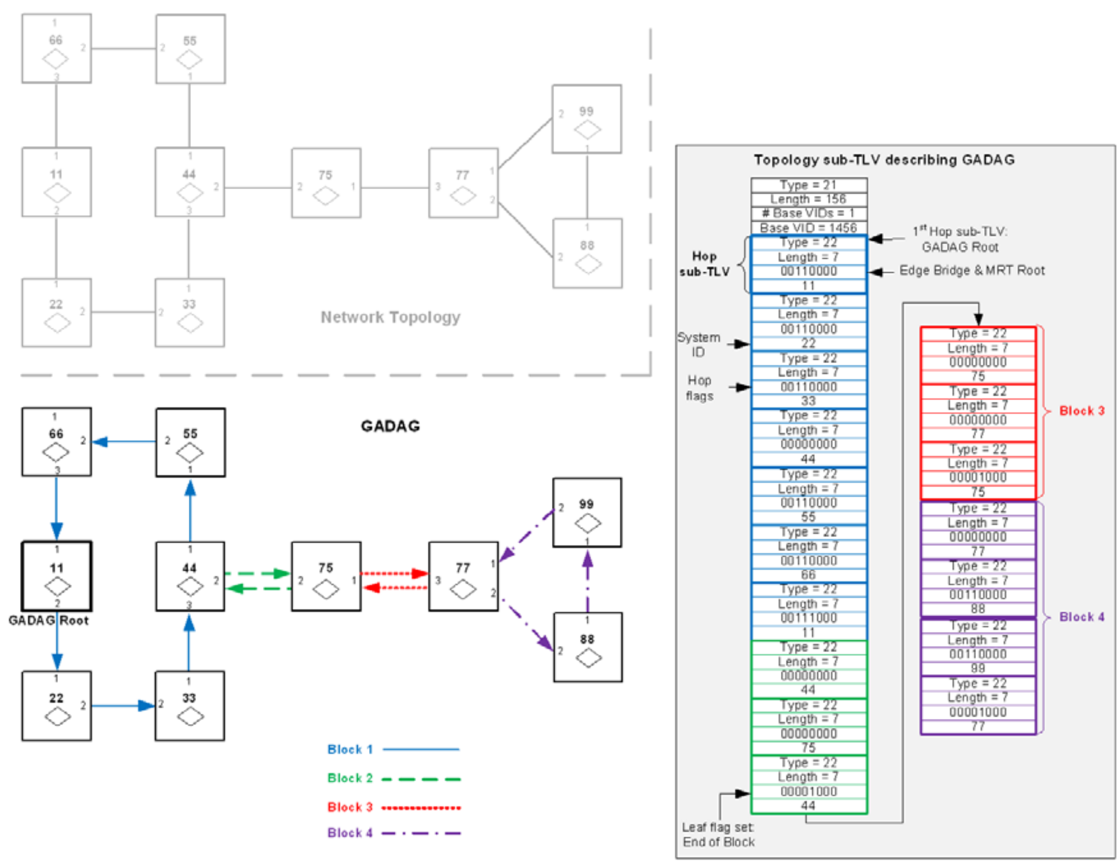

Figure 45-15—A GADAG for a topology with multiple blocks

The first hop of the first block is the GADAG Root; the first hop in the rest of the blocks is the localroot. The position of the set Leaf flags helps to determine the localroot, which is the next hop. In the example GADAG descriptor, one can determine that Bridge 11 is the localroot for Bridges 22, 33, 44, 55, and 66 (and Bridge 11 is the GADAG Root). Bridge 44 is the localroot for Bridge 75. Bridge 75 is the localroot for Bridge 77. Bridge 77 is the localroot for Bridges 88 and 99. The GADAG Root is assigned a localroot of None.

Block IDs are reconstructed while parsing a Topology sub-TLV specifying a GADAG. The current Block ID starts at 0 and is assigned to the GADAG Root. A Bridge appearing in the GADAG descriptor without a previously assigned Block ID value is assigned the current Block ID. The current Block ID is incremented by 1 after processing the localroot of a block. Note that the localroot of a block will keep the Block ID of the firsts block in which it is assigned a Block ID. In the example shown in Figure 45-15, Bridge 11 has Block ID=0. Bridges 22, 33, 44, 55, and 66 have Block ID=1. Bridge 75 has Block ID=2. Bridge 77 has Block ID=3. Bridges 88 and 99 have Block ID=4.

MRTs have to be restored cautiously after a topology change. The restoration should not be initiated if there is any reason to suspect that the SPT Bridges of the SPT Region do not share a common view on the network topology.

If MRTs are used for SPT protection, then MRTs are restored only after the restoration of the SPTs. Upon a failure event, PLR Bridges redirect data traffic to the MRT that is not affected by the failure, if any. As the topology change is announced by the corresponding LSPs, SPT Bridges and the GADAG Computer become aware of the change. The SPT Bridges then compute and install the new SPTs. When each SPT Bridge has installed the new SPT, then data traffic is directed back to the SPTs. The GADAG Computer then computes the new GADAG and floods its descriptor. The SPT Bridges then compute and install the new MRTs if each SPT Bridge has already directed traffic back to the SPTs.

If MRTs protect each other, then only one of the MRTs is restored at a time. Upon a failure event, PLR Bridges direct all data traffic to the MRT that is not affected by the failure (e.g., MRT-Red). If a 1+1 type of protection scheme is used, then there is no need to react to a failure event as data frames are forwarded on both MRTs, i.e., there are no PLRs. When the GADAG Computer and the SPT Bridges have the same view on the network topology again, then the GADAG Computer computes the new GADAG and floods its descriptor. Upon reception of the Topology sub-TLV describing the new GADAG, the SPT Bridges compute the new MRTs. The broken MRT (e.g., MRT-Blue) then is restored while the other MRT (e.g., MRT-Red) is not touched. If each SPT Bridge has installed the new MRT (e.g., the new MRT-Blue), then data traffic is directed to this newly restored MRT. The SPT Bridges then install the other MRT (e.g., the new MRT-Red). The same restoration procedure is followed if both MRTs are affected by the failure so that MRT-Blue is restored first.

NOTE—The method to be used for verifying whether the SPT Bridges have a common view on the topology and whether SPTs and MRTs have been restored is left to the implementer and the network operator. A common method is to wait some time after the reception of the last LSP indicating a topology change. Alternatively, OAM can be used, for instance.

Annex A

(normative)

PICS proforma—Bridge implementations[3]

Insert the following row at the end of the table in A.5:

A.5 Major capabilities

Item	Feature	Status	References	Support
PCR	Is Path Control and Reservation supported?	O	45	Yes [ ] No [ ]

Insert the following rows at the end of the table in A.14:

A.14 Bridge management

Item	Feature	Status	References	Support
MGT-220	Does the implementation support the managed objects required for PCR? If item PCR is not supported in Table A.5, mark N/A and continue at MGT-248	MGT AND PCR:M	5.4.6, 12.28	Yes [ ] No [ ] N/A [ ]
MGT-221	Create SPB System managed object	PCR:M	12.25.1.1	Yes [ ]
MGT-222	Write SPB System managed object	PCR:M	12.25.1.2	Yes [ ]
MGT-223	Read SPB System managed object	PCR:M	12.25.1.3	Yes [ ]
MGT-224	Delete SPB System managed object	PCR:M	12.25.1.4	Yes [ ]
MGT-225	Create SPB MTID Static managed object	PCR:M	12.25.2.1	Yes [ ]
MGT-226	Write SPB MTID Static managed object	PCR:M	12.25.2.2	Yes [ ]
MGT-227	Read SPB MTID Static managed object	PCR:M	12.25.2.3	Yes [ ]
MGT-228	Delete SPB MTID Static managed object	PCR:M	12.25.2.4	Yes [ ]
MGT-229	Read SPB Topology Instance Dynamic managed object	PCR:M	12.25.3.1	Yes [ ]
MGT-230	Create SPB ECT Static Entry managed object	PCR:M	12.25.4.1	Yes [ ]
MGT-231	Read SPB ECT Static Entry managed object	PCR:M	12.25.4.2	Yes [ ]
MGT-232	Delete SPB ECT Static Entry managed object	PCR:M	12.25.4.3	Yes [ ]

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

A.14 Bridge management *(continued)*

Item	Feature	Status	References	Support
MGT-233	Read SPB ECT Dynamic Entry managed object	PCR:M	12.25.5.1	Yes [ ]
MGT-234	Write SPB Adjacency Static Entry managed object	PCR:M	12.25.6.1	Yes [ ]
MGT-235	Read SPB Adjacency Static Entry managed object	PCR:M	12.25.6.2	Yes [ ]
MGT-236	Read SPB Adjacency Dynamic Entry managed object	PCR:M	12.25.7.1	Yes [ ]
MGT-237	Read SPBM BSI Static Entry managed object	BEB AND PCR:M	12.25.8.2	Yes [ ]
MGT-238	Read SPBM Topology Service Table managed object	BEB AND PCR:M	12.25.12.1	Yes [ ]
MGT-239	Read SPB Topology Node Table managed object	PCR:M	12.25.9.1	Yes [ ]
MGT-240	Read SPB Topology ECT Table managed object	PCR:M	12.25.10.1	Yes [ ]
MGT-241	Read SPB Topology Edge Table managed object	PCR:M	12.25.11.1	Yes [ ]
MGT-242	Read SPBM Topology Service Table managed object	BEB AND PCR:M	12.25.12.1	Yes [ ]
MGT-243	Read SPBV Topology Service Table managed object	PCR:M	12.25.13.1	Yes [ ]
MGT-244	Create a PCR ECT Static Entry managed object	PCR:O	12.28.1.1	Yes [ ] No [ ]
MGT-245	Read a PCR ECT Static Entry managed object	PCR:O	12.28.1.2	Yes [ ] No [ ]
MGT-246	Delete a PCR ECT Static Entry managed object	PCR:O	12.28.1.4	Yes [ ] No [ ]
MGT-247	Read PCR Topology ECT Table managed object	PCR:O	12.28.2.1	Yes [ ] No [ ]

Insert the following row at the end of the table in A.24:

A.24 Management Information Base (MIB)

Item	Feature	Status	References	Support
MIB-41	Are PCR MIB objects fully supported (per PCR MIB COMPLIANCE)?	MIB AND PCR:O	5.4.6, 17.7.19	Yes [ ] No [ ] N/A [ ]

Insert the following subclause, A.43, after A.42:

A.43 Path Control and Reservation

Item	Feature	Status	Reference	Support
	If Path Control and Reservation (PCR in Table A.5) is not supported, mark N/A and ignore the remainder of this table.			N/A [ ]
PCR-1	Does the Bridge support the ISIS-PCR protocol?	PCR:M	45	Yes [ ]
PCR-2	Does the Bridge support the SPB Link Metric sub-TLV?	PCR:M	28.12.7	Yes [ ]
PCR-3	Does the Bridge support the SPB Base VLAN-Identifiers sub TLV?	PCR:M	28.12.4	Yes [ ]
PCR-4	Does the Bridge support the SPB Instance sub-TLV?	PCR:M	28.12.5	Yes [ ]
PCR-5	Does the Bridge support the SPBV MAC address sub-TLV?	PCR:M	28.12.9	Yes [ ]
PCR-6	Does the Bridge support the SPBM Service Identifier and Unicast Address sub-TLV?	BEB AND PCR:M	28.12.20	Yes [ ]
PCR-7	Does the Bridge support the Topology sub-TLV?	PCR:M	45.1.9	Yes [ ]
PCR-8	Does the Bridge support the Hop sub-TLV?	PCR:M	45.1.10	Yes [ ]
PCR-9	Does the Bridge support the ST ECT Algorithm?	PCR:M	45.1.2	Yes [ ]
PCR-10	Does the Bridge support the LT ECT Algorithm?	PCR:O	45.1.2	Yes [ ] No [ ]
PCR-11	Does the Bridge support the LTS ECT Algorithm?	PCR:O	45.1.2	Yes [ ] No [ ]
PCR-12	Does the Bridge support the MRT ECT Algorithm?	PCR:O	45.1.2, 45.3.3	Yes [ ] No [ ]
PCR-13	Does the Bridge support the MRTG ECT Algorithm?	PCR:O	45.1.2, 45.3.4	Yes [ ] No [ ]
PCR-14	Does the Bridge support the Extended IS Reachability TLV?	PCR:O	45.1.8, IETF RFC 5305	Yes [ ] No [ ]
PCR-15	Does the Bridge support the Unidirectional Link Delay sub-TLV?	PCR:O	45.1.8, [R-Qca-02]	Yes [ ] No [ ]
PCR-16	Does the Bridge support the Min/ Max Unidirectional Link Delay sub-TLV?	PCR:O	45.1.8, [R-Qca-02]	Yes [ ] No [ ]
PCR-17	Does the Bridge support the Unidirectional Delay Variation sub-TLV?	PCR:O	45.1.8, [R-Qca-02]	Yes [ ] No [ ]

A.43 Path Control and Reservation *(continued)*

Item	Feature	Status	Reference	Support
PCR-18	Does the Bridge support the Unidirectional Link Loss sub-TLV?	PCR:O	45.1.8, [R-Qca-02]	Yes [ ] No [ ]
PCR-19	Does the Bridge support the Unidirectional Residual Bandwidth sub-TLV?	PCR:O	45.1.8, [R-Qca-02]	Yes [ ] No [ ]
PCR-20	Does the Bridge support the Unidirectional Available Bandwidth sub-TLV?	PCR:O	45.1.8, [R-Qca-02]	Yes [ ] No [ ]
PCR-21	Does the Bridge support the Unidirectional Utilized Bandwidth sub-TLV?	PCR:O	45.1.8, [R-Qca-02]	Yes [ ] No [ ]
PCR-22	Does the Bridge support the Shared Risk Link Group TLV?	PCR:O	45.1.8, IETF RFC 5307	Yes [ ] No [ ]
PCR-23	Does the Bridge support the Administrative Group sub-TLV?	PCR:O	45.1.11	Yes [ ] No [ ]
PCR-24	Does the Bridge support the Bandwidth Constraint sub-TLV?	PCR:O	45.1.12	Yes [ ] No [ ]
PCR-25	Does the Bridge support the Bandwidth Assignment sub-TLV?	PCR:O	45.2.1	Yes [ ] No [ ]
PCR-26	Does the Bridge support the Timestamp sub-TLV?	PCR:O	45.2.2	Yes [ ] No [ ]
PCR-27	Does the Bridge support Loop-Free Alternates for unicast data flows supported by SPBM?	SPBM:O PCR:O	45.3.1	Yes [ ] No [ ]
PCR-28	Does the Bridge support relaxed ingress checking?	SPBM:O PCR:O	8.4, 8.6.1, 8.8.9, 45.3.1	Yes [ ] No [ ]

Annex Q

(informative)

Bibliography

Insert the following references in alphanumeric order in Annex Q, and renumber the subsequent references in the annex accordingly:

[B7a] IEEE Std 802.1AX™-2014, IEEE Standard for Local and metropolitan area networks—Link Aggregation.[4]

[B12a] IEEE Std 1588™-2008, IEEE Standard for Precision Clock Synchronization Protocol for Networked Measurements and Control Systems.

[B31a] IETF RFC 4655, A Path Computation Element (PCE)-Based Architecture (Informational RFC), August 2006.[5]

[B37a] IETF RFC 7813, IS-IS Path Control and Reservation, 2016.

[4]IEEE publications are available from The Institute of Electrical and Electronic Engineers (http://standards.ieee.org).
[5]IETF documents (i.e., RFCs) are available for download at http://www.rfc-archive.org/.

Consensus

WE BUILD IT.

Connect with us on:

Facebook: https://www.facebook.com/ieeesa

Twitter: @ieeesa

LinkedIn: http://www.linkedin.com/groups/IEEESA-Official-IEEE-Standards-Association-1791118

IEEE-SA Standards Insight blog: http://standardsinsight.com

YouTube: IEEE-SA Channel

IEEE
standards.ieee.org
Phone: +1 732 981 0060   Fax: +1 732 562 1571