IEEE Standards

IEEE Std 802.1Q™, 2003 Edition
(Incorporates IEEE Std 802.1Q-1998, IEEE Std 802.1u™-2001,
IEEE Std 802.1v™-2001, and IEEE Std 802.1s™-2002)

802.1Q™

IEEE Standards for
Local and metropolitan area networks

Virtual Bridged Local Area Networks

IEEE Computer Society

Sponsored by the
LAN/MAN Standards Committee

IEEE

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

7 May 2003

Print: SH95124
PDF: SS95124

IEEE Std 802.1Q™, 2003 Edition

(Incorporates IEEE Std 802.1Q-1998, IEEE Std 802.1u™-2001,
IEEE Std 802.1v™-2001, and IEEE Std 802.1s™-2002)

IEEE Standards for
Local and metropolitan area networks—

Virtual Bridged Local Area Networks

Sponsor

LAN/MAN Standards Committee
of the
IEEE Computer Society

Abstract: This standard defines an architecture for Virtual Bridged LANs, the services provided in Virtual Bridged LANs, and the protocols and algorithms involved in the provision of those services.

Keywords: local area networks, MAC Bridge management, media access control bridges, virtual LANs

The Institute of Electrical and Electronics Engineers, Inc.
345 East 47th Street, New York, NY 10017-2394, USA

Print: ISBN 0-7381-3662-X SH95124
PDF: ISBN 0-7381-3663-8 SS95124

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

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

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

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

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

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Comments on standards and requests for interpretations should be addressed to:

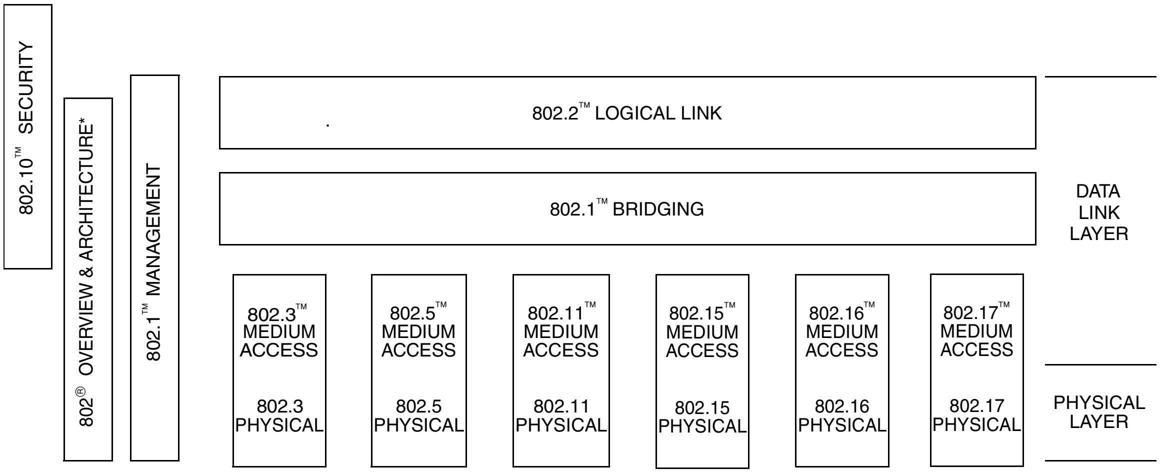
Secretary, IEEE-SA Standards Board
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
USA

Note: 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 with respect to the existence or validity of any patent rights in connection therewith. The IEEE shall not be responsible for identifying patents for which a license may be required by an IEEE standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention. A patent holder has filed a statement of assurance that it will grant licenses under these rights without compensation or under reasonable rates and nondiscriminatory, reasonable terms and conditions to all applicants desiring to obtain such licenses. The IEEE makes no representation as to the reasonableness of rates and/or terms and conditions of the license agreements offered by patent holders. Further information may be obtained from the IEEE Standards Department.

Introduction to IEEE Std 802.1Q™, 2003 Edition

(This introduction is not part of IEEE Std 802.1Q, 2003 Edition, IEEE Standards for Local and metropolitan area networks— Virtual Bridged Local Area Networks.)

This standard is part of a family of standards for local and metropolitan area networks. The relationship between the standard and other members of the family is shown below. (The numbers in the figure refer to IEEE standard designations.[1])



* Formerly IEEE Std 802.1A™.

This family of standards deals with the Physical and Data Link Layers as defined by the International Organization for Standardization (ISO) Open Systems Interconnection Basic Reference Model (ISO/IEC 7498-1:1994). The access standards define several types of medium access technologies and associated physical media, each appropriate for particular applications or system objectives. Other types are under investigation.

The standards defining the technologies noted above are as follows:

- IEEE Std 802™:[2]

  Overview and Architecture. This standard provides an overview to the family of IEEE 802 Standards. This document forms part of the IEEE Std 802.1 scope of work.

- IEEE Std 802.1B™ and 802.1K™ [ISO/IEC 15802-2]:

  LAN/MAN Management. Defines an Open Systems Interconnection (OSI) management-compatible architecture, and services and protocol elements for use in a LAN/MAN environment for performing remote management.

[1] The IEEE Standards referred to in the above figure and list are trademarks owned by the Institute of Electrical and Electronics Engineers, Incorporated.

[2] The IEEE 802 Overview and Architecture Specification, originally known as IEEE Std 802.1A, has been renumbered as IEEE Std 802. This has been done to accommodate recognition of the base standard in a family of standards. References to IEEE Std 802.1A should be considered as references to IEEE Std 802.

- IEEE Std 802.1D™ *Media Access Control (MAC) Bridges*. Specifies an architecture and protocol for the [ISO/IEC 15802-3]: interconnection of IEEE 802 LANs below the MAC service boundary.

- IEEE Std 802.1E™ [ISO/IEC 15802-4]: System Load Protocol. Specifies a set of services and protocol for those aspects of management concerned with the loading of systems on IEEE 802 LANs.

- IEEE Std 802.1F™ *Common Definitions and Procedures for IEEE 802 Management Information.*

- IEEE Std 802.1G™ [ISO/IEC 15802-5]: *Remote Media Access Control (MAC) Bridging*. Specifies extensions for the interconnection, using non-LAN systems communication technologies, of geographically separated IEEE 802 LANs below the level of the logical link control protocol.

- IEEE Std 802.1H™ [ISO/IEC TR 11802-5] *Recommended Practice for Media Access Control (MAC) Bridging of Ethernet V2.0 in IEEE 802 Local Area Networks.*

- IEEE Std 802.1Q™ *Virtual Bridged Local Area Networks*. Defines an architecture for Virtual Bridged LANs, the services provided in Virtual Bridged LANs, and the protocols and algorithms involved in the provision of those services.

- IEEE Std 802.2™ [ISO/IEC 8802-2]: *Logical Link Control.*

- IEEE Std 802.3™ [ISO/IEC 8802-3]: *CSMA/CD Access Method and Physical Layer Specifications.*

- IEEE Std 802.5™ [ISO/IEC 8802-5]: *Token Ring Access Method and Physical Layer Specifications.*

- IEEE Std 802.10™: *Standard for Interoperable LAN Security (SILS).* Currently approved: Secure Data Exchange (SDE).

- IEEE Std 802.11™: [ISO/IEC 8802-11] *Wireless LAN Medium Access Control (MAC) Sublayer and Physical Layer Specifications.*

- IEEE Std 802.15™: *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for: Wireless Personal Area Networks.*

- IEEE Std 802.16™: *Standard Air Interface for Fixed Broadband Wireless Access Systems.*

- IEEE P802.17™: *Resilient Packet Ring Access Method and Physical Layer Specifications.* Currently under development as a draft IEEE Standard.

The reader of this standard is urged to become familiar with the complete family of standards.

Conformance test methodology

An additional standards series, identified by the number 1802™, has been established to identify the conformance test methodology documents for the IEEE 802 family of standards. Thus the conformance test documents for IEEE 802.3 are numbered 1802.3™, the conformance test documents for IEEE 802.5™ will be 1802.5™, and so on. Similarly, ISO will use 18802 to number conformance test standards for 8802 standards.

IEEE Std 802.1Q, 2003 Edition

The MAC Bridge standardization activities that resulted in the development of IEEE Std 802.1D-1993 introduced the concept of Filtering Services in Bridged LANs, and mechanisms whereby filtering information in such LANs may be acquired and held in a Filtering Database.

IEEE Std 802.1D, 1998 Edition, an amended version of IEEE Std 802.1D-1993, extends this concept of Filtering Services in order to define additional capabilities in Bridged LANs aimed at

a) The provision of expedited traffic capabilities, to support the transmission of time-critical information in a LAN environment;

b) The use of signalled user priority information as the basis for identifying expedited classes of traffic;

c) The provision of filtering services that support the dynamic definition and establishment of Groups in a LAN environment, and the filtering of frames by Bridges such that frames addressed to a particular Group are forwarded only on those LAN segments that are required in order to reach members of that Group;

d) The provision of a Generic Attribute Registration Protocol (GARP) that is used to support the mechanism for providing Group filtering capability, and is also made available for use in other attribute registration applications.

This standard, first published as IEEE Std 802.1Q-1998, makes use of the concepts and mechanisms of LAN Bridging that were introduced by IEEE Std 802.1D, 1998 Edition, and defines additional mechanisms that allow the implementation of Virtual Bridged LANs. The following are described:

e) Virtual LAN Services in Bridged LANs;

f) The operation of the Forwarding Process that is required in order to support Virtual Bridged LANs;

g) The structure of the Filtering Database that is required in order to support Virtual Bridged LANs;

h) The nature of the protocols and procedures that are required in order to provide Virtual LAN services, including the definition of the frame formats used to represent VLAN identification information, and the procedures used in order to insert and remove VLAN identifiers and the headers in which they are carried;

i) The ability to support end-to-end signalling of user priority information regardless of the intrinsic ability of the underlying MAC protocols to signal user priority information;

j) The GARP VLAN Registration Protocol (GVRP) that allows distribution and registration of VLAN membership information (the protocol described makes use of the GARP protocol defined in ISO/IEC 15802-3);

k) The management services and operations that are required in order to configure and administer Virtual Bridged LANs.

The 2003 Edition of this standard incorporated three amendments, IEEE Std 802.1u-2001, IEEE Std 802.1v-2001, and IEEE Std 802.1s-2002, into the text of IEEE Std 802.1Q-1998. These amendments describe enhancements to the standard to allow:

l) Dynamic Group and VLAN registration to be restricted, based on the contents of static filtering entries;

m) VLAN classification according to link layer protocol type;

n) Support for VLANs carried over multiple Spanning Tree instances.

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
P.O. Box 1331
Piscataway, NJ 08855-1331
USA

Participants

The following is a list of participants in the Interworking activities of the IEEE 802.1 Working Group during the development of IEEE Std 802.1Q-1998 . Voting members at the time of publication are marked with an asterisk (*).

William P. Lidinsky, *Chair**
Mick Seaman, *Chair, Interworking Task Group**
Editing Team:
Tony Jeffree*, *Coordinating Editor*
Anil Rijsinghani*, **Richard Hausman***, **Michele Wright***, **Paul Langille***, **P. J. Singh***

Steve Adams*
Stephen Ades
Ken Alonge
Floyd Backes*
John Bartlett*
Les Bell*
Avner Ben-Dor
Michael Berger*
James S. Binder*
David Brady
Martin Brewer
Bill Bunch*
Bob Cardinal
Paul Carroll*
Jeffrey Catlin*
Dennis Cave
Alan Chambers*
Steve Chan
David W. Chang*
Ken Chapman
Hon Wah Chin*
Chi Chong
Chris Christ*
Paul Congdon*
Glenn Connery*
David Cullerot*
Ted Davies*
Andy Davis
David Delaney*
Prakash Desai
Jeffrey Dietz*
Kurt Dobbins
Peter Ecclesine*
J. J. Ekstrom*
Norman W. Finn*
Yishai Fraenkel
Paul Frantz
Lars Henrik Frederiksen*
Anoop Ghanwani*
John Grinham
Steve Haddock
Sharam Hakimi*
John Hart*
Scott Harvell
Wayne Hathaway

Vic Hayes
David Head*
Gaby Hecht
Deepak Hegde*
Ariel Hendel
John Hickey
David Hollender
Steve Horowitz*
Michelle Hsiung
Rita Hunt
David Husak
Altaf Hussain*
Vipin K. Jain*
Neil Jarvis
Allen Kasey
Toyoyuki Kato*
Hal Keen*
Kevin Ketchum*
Keith Klamm*
Bruce Kling*
Walter Knitl
Dan Krent*
Paul Kummer
Paul Lachapelle*
Bill Lane
Johann Lindmeyr*
Gary Littleton
Robert D. Love
Andy Luque
Peter Martini
Keith McCloghrie
Martin McNealis
Milan Merhar*
John Messenger*
Colin Mick
Amol Mitra
Yaron Nachman*
Krishna Narayanaswamy*
Paul Nikolich
Lawrence Ng*
Henry Ngai*
Eugene O'Neil
Satoshi Obara*
Toshio Ooka*
Jörg Ottensmeyer*

Luc Pariseau*
Yonadav Perry
John Pickens*
Gideon Prat
Kirk Preiss
Steve Ramberg*
Shlomo Reches*
Dick Reohr
James Richmond*
Doug Ruby
Ray Samora
Ayman Sayed*
Rich Seifert
Lee Sendelbach*
Himanshu Shah*
Phil Simmons*
K. Karl Shimada
Fred Shu
Rosemary V. Slager*
Alexander Smith*
Andrew Smith*
Larry Stefani*
Stuart Soloway*
Sundar Subramaniam*
Richard Sweatt
Robin Tasker*
Fouad Tobagi
Naoki Tsukutari
Dhadesugoor Vaman
Steve Van Seters*
Dono van-Mierop*
John Wakerly*
Peter Wang*
Philip Wang
Y. C. Wang*
Trevor Warwick*
Bob Watson
Alan Weissberger
Glenn Wenig
Keith Willette*
Michael Witkowski*
Edward Wong*
Michael D. Wright*
Allen Yu*
Wayne Zakowski*

The following persons were on the balloting committee of IEEE Std 802.1Q. Balloters may have voted for approval, disapproval, or abstention.

Corey Anderson
Kit Athul
Thomas W. Bailey
Peter K. Campbell
James T. Carlo
David E. Carlson
Alan M. Chambers
R. Allan Cobb
Robert S. Crowder
Thomas J. Dineen
Peter Ecclesine
Philip H. Enslow
Changxin Fan
John W. Fendrich
Michael A. Fischer
Harvey A. Freeman
Gautam Garai
Harry Gold
Julio Gonzalez-Sanz
Stephen R. Haddock
Allen W. Hathaway
Donald N. Heirman
Raj Jain
Neil A. Jarvis
Anthony A. Jeffree
Robert W. Klessig
Stephen Barton Kruger
William G. Lane
David J. Law
Lanse M. Leach
Randolph S. Little
Peter Martini
Milan Merhar
John L. Messenger
Bennett Meyer
David S. Millman
John E. Montague
Wayne D. Moyers
Shimon Muller
Paul Nikolich
Charles Oestereicher
Roger Pandanda
John R. Pickens
Vikram Punj
Edouard Y. Rocher
James W. Romlein
Floyd E. Ross
Michael Salzman
Norman Schneidewind
Mick Seaman
Rich Seifert
Leo Sintonen
Michael A. Smith
Patricia Thaler
Geoffrey O. Thompson
Mark-Rene Uchida
Oren Yuen

When the IEEE 802.1 Working Group approved Amendment 802.1u, it had the following membership:

Tony Jeffree, *Chair and Editor*

Neil Jarvis, *Vice Chair*
Mick Seaman, *Chair, Interworking Task Group*

Floyd Backes
Les Bell
Alan Chambers
Marc Cochran
Paul Congdon
Hesham El Bakoury
Norman W. Finn
Sharam Hakimi
Bob Hott
Toyayuki Kato
Hal Keen
Daniel Kelley
Keith Klamm
Bill Lidinsky
Yaron Nachman
Satoshi Obara
Luc Pariseau
Anil Rijsinghani
John J. Roese
Ted Schroeder
Benjamin Schultz
Rosemary V. Slager
Andrew Smith
Michel Soerensen
Robin Tasker
Manoj Wadekar
Robert Williams

The following members of the balloting committee voted on Amendment 802.1u. Balloters may have voted for approval, disapproval, or abstention.

Jacob Ben Ary
James T. Carlo
Keith Chow
Guru Dutt Dhingra
Thomas J. Dineen
Christos Douligeris
Sourav K. Dutta
Philip H. Enslow
Changxin Fan
John W. Fendrich
Michael A. Fischer
Richard A. Froke
Robert J. Gagliano
Gautam Garai
Alireza Ghazizahedi
Tim Godfrey
Robert M. Grow
Chris G. Guy
Simon Harrison
Raj Jain
Kamran Jamal
Neil A. Jarvis
Anthony A. Jeffree
Stuart J. Kerry
Daniel R. Krent
Stephen Barton Kruger
David J. Law
William Lidinsky
Randolph S. Little
Peter Martini
Bennett Meyer
David S. Millman
John E. Montague
Robert Mortonson
Robert O'Hara
Satoshi Obara
Roger Pandanda
Vikram Punj
Gary S. Robinson
Edouard Y. Rocher
James W. Romlein
Floyd E. Ross
Jaideep Roy
Rich Seifert
Leo Sintonen
Joseph S. Skorupa
Fred J. Strauss
Jonathan R. Thatcher
Mark-Rene Uchida
Scott A. Valcourt
John Viaplana
Paul A. Willis
Oren Yuen

When the IEEE 802.1 Working Group approved Amendment 802.1v, it had the following membership:

Tony Jeffree, *Chair and Editor*
Neil Jarvis, *Vice Chair*
Mick Seaman, *Chair, Interworking Task Group*
David Delaney, *Editor*
Andrew Smith, *Editor*

Floyd Backes
Les Bell
Alan Chambers
Marc Cochran
Paul Congdon
Hesham El Bakoury
Norman W. Finn
Sharam Hakimi
Bob Hott
Toyayuki Kato
Hal Keen
Daniel Kelley
Keith Klamm
Bill Lidinsky
Yaron Nachman
Satoshi Obara
Luc Pariseau
Anil Rijsinghani
John J. Roese
Ted Schroeder
Benjamin Schultz
Rosemary V. Slager
Andrew Smith
Michel Soerensen
Robin Tasker
Manoj Wadekar
Robert Williams

The following members of the balloting committee voted on Amendment 802.1v. Balloters may have voted for approval, disapproval, or abstention.

Terry S. Arnold
James T. Carlo
Keith Chow
Robert S. Crowder
Guru Dutt Dhingra
Thomas J. Dineen
Christos Douligeris
Sourav K. Dutta
Philip H. Enslow
Changxin Fan
John W. Fendrich
Michael A. Fischer
Richard A. Froke
Robert J. Gagliano
Gautam Garai
Alireza Ghazizahedi
Tim Godfrey
Robert M. Grow
Chris G. Guy
Simon Harrison
Osamu Ishida
Raj Jain
Kamran Jamal
Neil A. Jarvis
Anthony A. Jeffree
Jack R. Johnson
Stuart J. Kerry
Daniel R. Krent
Stephen Barton Kruger
Joseph Kubler
David J. Law
William Lidinsky
Randolph S. Little
Ronald Mahany
Peter Martini
Bennett Meyer
David S. Millman
James F. Mollenauer
John E. Montague
Robert Mortonson
Robert O'Hara
Satoshi Obara
Roger Pandanda
Vikram Punj
Gary S. Robinson
Edouard Y. Rocher
James W. Romlein
Floyd E. Ross
Jaideep Roy
Rich Seifert
Leo Sintonen
Joseph S. Skorupa
Fred J. Strauss
Jonathan R. Thatcher
Mark-Rene Uchida
Scott A. Valcourt
John Viaplana
Paul A. Willis
Oren Yuen

When the IEEE 802.1 Working Group approved Amendment 802.1s, it had the following membership:

Tony Jeffree, *Chair and Editor*
Neil Jarvis, *Vice Chair*
Mick Seaman, *Chair, Interworking Task Group and Editor*
Norm Finn, *Editor*

Les Bell
Paul Congdon
Hesham Elbakoury
Robert W. Hott
Ran Ish-Shalom
Shyam Kaluve
Hal Keen
Loren Larsen
Frank Reichstein
Curtis Simonson
Michel Thorsen
Michael D. Wright

The following members of the balloting committee voted on Amendment 802.1s. Balloters may have voted for approval, disapproval, or abstention.

Vishal Anand
Linda Cheng
Keith Chow
Dr. Guru Dutt Dhingra
Thomas Dineen
Michael Eckhoff
Mike Geipel
Stuart Holoman
Raj Jain

Tony Jeffree
Pranav Koushik
Jeffrey Lynch
Kyle Maus
George Miao
Paul Nikolich
Ellis Nolley
Satoshi Obara
Subbu Ponnuswamy

Vikram Punj
Maximilian Riegel
Thomas Ruf
Mick Seaman
Rich Seifert
Pat Thaler
Dmitri Varsanofiev
Cassio Vinhal
Oren Yuen

When the IEEE-SA Standards Board approved this standard on 8 December 1998, it had the following membership:

Richard J. Holleman, *Chair*
Donald N. Heirman, *Vice Chair*
Judith Gorman, *Secretary*

Satish K. Aggarwal
Clyde R. Camp
James T. Carlo
Gary R. Engmann
Harold E. Epstein
Jay Forster*
Thomas F. Garrity
Ruben D. Garzon

James H. Gurney
Jim D. Isaak
Lowell G. Johnson
Robert Kennelly
E. G. "Al" Kiener
Joseph L. Koepfinger*
Stephen R. Lambert
Jim Logothetis
Donald C. Loughry

L. Bruce McClung
Louis-François Pau
Ronald C. Petersen
Gerald H. Peterson
John B. Posey
Gary S. Robinson
Hans E. Weinrich
Donald W. Zipse

*Member Emeritus

Kristin M. Dittmann
IEEE Standards Project Editor,
IEEE Std 802.1Q-1998

Jennifer McClain Longman
IEEE Standards Project Editor,
IEEE Std 802.1u-2001
IEEE Std 802.1v-2001

Andrew Ickowicz
IEEE Standards Project Editor,
IEEE Std 802.1s-2002
IEEE Std 802.1Q, 2003 Edition

Contents

IEEE Standards for
Local and metropolitan area networks—

Virtual Bridged Local Area Networks

1. Overview

IEEE 802™ Local Area Networks (LANs) of all types can be connected together with Media Access Control (MAC) Bridges, as specified in IEEE Std 802.1D™, 1998 Edition[1]. This standard defines the operation of Virtual LAN (VLAN; see 3.38) Bridges that permit the definition, operation, and administration of VLAN topologies within a Bridged LAN (see 3.35) infrastructure.

1.1 Scope

For the purpose of compatible interconnection of information technology equipment using the IEEE 802 MAC Service supported by interconnected IEEE 802 standard LANs using different or identical MAC methods, this standard specifies a general method for the operation of MAC Bridges that support the construction of VLANs (see 3.38). To this end it

a) Positions the function of VLANs within an architectural description of the MAC Sublayer;

b) Defines enhancements to the Support of the MAC Service, as described and defined in IEEE Std 802.1D, 1998 Edition, for the purposes of VLAN Bridging;

c) Specifies an Enhanced Internal Sublayer Service provided to the Media Access Independent functions that provide frame relay (3.7) in the VLAN Bridge;

d) Specifies the operation of the functions that provide frame relay in the VLAN Bridge;

e) Defines the structure, encoding, and interpretation of the VLAN control information carried in tagged frames (3.33) in a VLAN;

f) Specifies the rules that govern the insertion and removal of VLAN control information in frames;

g) Specifies the rules that govern the ability to carry data in Canonical format and Non-canonical format using different LAN MAC methods;

NOTE—The meanings of the terms *Canonical format* and *Non-canonical format* are discussed in Annex F.

h) Establishes the requirements for, and specifies the means of, automatic configuration of VLAN topology information;

i) Defines the management functionality that may be provided in a VLAN Bridge in order to facilitate administrative control over VLAN operation;

j) Defines the operation of the Multiple Spanning Tree algorithm and protocol (MSTP);

k) Defines the enhancements made to the Rapid Spanning Tree Algorithm and Protocol in order to create MSTP;

l) Describes the protocols and procedures necessary to support interoperation between MST and SST Bridges in the same Bridged LAN;

m) Specifies the requirements to be satisfied by equipment claiming conformance to this standard.

[1] Information about references can be found in Clause 2.

1.2 VLAN aims and benefits

VLANs aim to offer the following benefits:

a) VLANs are supported over all IEEE 802 LAN MAC protocols, and over shared media LANs as well as point-to-point LANs.
b) VLANs facilitate easy administration of logical groups of stations that can communicate as if they were on the same LAN. They also facilitate easier administration of moves, adds, and changes in members of these groups.
c) Traffic between VLANs is restricted. Bridges forward unicast, multicast, and broadcast traffic only on LAN segments that serve the VLAN to which the traffic belongs.
d) As far as possible, VLANs maintain compatibility with existing bridges and end stations.
e) If all Bridge Ports are configured to transmit and receive untagged frames (3.36), bridges will work in plug-and-play IEEE Std 802.1D mode. End stations will be able to communicate throughout the Bridged LAN.

NOTE—Whether a VLAN Bridge will operate in IEEE Std 802.1D mode depends upon the configuration of the various Port parameters (8.4) and the Filtering Database (8.11). A VLAN Bridge in its default configuration is transparent to untagged frames (3.36) but is not transparent to tagged frames (3.33), so the operation of such Bridges in the presence of tagged traffic differs from that of an IEEE Std 802.1D Bridge. If the configuration settings of VLAN Bridges are changed from the default values defined in this standard, then transparency with respect to untagged frames may also be affected.

1.3 Relationship with IEEE Std 802.1D

This standard makes use of specific aspects of the MAC Bridge specification contained in IEEE Std 802.1D; those aspects therefore become provisions of this standard. Table 1-1 shows how the relevant clauses of IEEE Std 802.1D, 1998 Edition are incorporated into this standard.

Table 1-1—Relationship between this standard and IEEE Std 802.1D

IEEE Std 802.1D clause	Use in this standard
5. Conformance	Provision of this standard, as extended by Clause 5
6. Support of the MAC Service	Provision of this standard, as modified by Clause 6
7. Principles of operation	Replaced by Clause 8
8. The spanning tree algorithm and protocol	Provision of this standard
9. Encoding of bridge protocol data units	Provision of this standard
10. GARP Multicast Registration Protocol (GMRP)	Provision of this standard, as modified by Clause 11
11. Example “C” code implementation of GMRP	Provision of this standard, as modified by Clause 11
12. Generic Attribute Registration Protocol (GARP)	Provision of this standard
13. Example “C” code implementation of GARP	Provision of this standard
14. Bridge management	Replaced by Clause 12
15. Management protocol	Not applicable
16. Bridge performance	Provision of this standard
17. Rapid Spanning Tree Algorithm and Protocol	Provision of this standard, as modified by Clauses 13 and 14
Annex A (normative) PICS proforma	Replaced by Annex A
Annex B (informative) Calculating Spanning Tree parameters	Provision of this standard
Annex C (normative) Source-routing transparent bridge operation	Provision of this standard
Annex D (normative) PICS proforma for source routing transparent bridge operation	Provision of this standard
Annex E (normative) Allocation of Object Identifier values	Not applicable
Annex F (informative) Target topology, migration, and interoperability	Provision of this standard
Annex G (informative) Preserving the integrity of FCS fields in MAC Bridges	Provision of this standard
Annex H (informative) Design considerations for Traffic Class Expediting and Dynamic Multicast Filtering	Provision of this standard

2. References

The following standards contain provisions which, through reference in this text, constitute provisions of this standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ANSI X3.159-1989, American National Standards for Information Systems—Programming Language—C.[2]

IEEE Std 802™-1990, IEEE Standards for Local and Metropolitan Area Networks: Overview and Architecture.[3]

IEEE Std 802.1B™, 1995 [ISO/IEC 15802-2: 1995], IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Common specifications—Part 2: LAN/MAN Management.

IEEE Std 802.1D™-1993 [ISO/IEC 10038: 1993], IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local area networks—Media Access Control (MAC) bridges.

IEEE Std 802.1D, 1998 Edition [ISO/IEC 15802-3: 1998], IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Common specifications—Part 3: Media Access Control (MAC) Bridges.

IEEE Std 802.1F™-1993, IEEE Standards for Local and Metropolitan Area Networks: Common Definitions and Procedures for IEEE 802 Management Information.

IEEE Std 802.1H™-1997 [ISO/IEC 11802-5: 1997], IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and Metropolitan Area Networks—Technical Reports and Guidelines—Part 5: Media Access Control Bridging of Ethernet V2.0 in IEEE 802 Local Area Networks.

IEEE Std 802.1t™-2001, IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Common specifications—Part 3: Media Access Control (MAC) Bridges—Amendment 1.

IEEE Std 802.1w™-2001, IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Common specifications—Part 3: Media Access Control (MAC) Bridges—Amendment 2: Rapid Reconfiguration. [Amendment to IEEE Std 802.1D, 1998 Edition (ISO/IEC 15802-3:1998) and IEEE Std 802.1t-2001]

IEEE Std 802.1X™-2001, IEEE Standards for Local and Metropolitan Area Networks—Port Based Network Access Control.

IEEE Std 802.2™, 1998 Edition [ISO/IEC 8802-2: 1998], IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 2: Logical link control.

[2] ANSI publications are available from the Sales Department, American National Standards Institute, 11 West 42nd Street, 13th Floor, New York, NY 10036, USA.

[3] IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, USA (http://standards.ieee.org/).

IEEE Std 802.3™, 1998 Edition, Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications.

IEEE Std 802.3ac™-1998, Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Supplement to Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications: Frame Extensions for Virtual Bridged Local Area Network (VLAN) tagging on 802.3 Networks.

IEEE Std 802.4™-1990 [ISO/IEC 8802-4: 1990], IEEE Standard for Information processing systems—Local area networks—Part 4: Token-passing bus access method and physical layer specifications.[4]

IEEE Std 802.5™, 1998 Edition [ISO/IEC 8802-5: 1998], IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 5: Token ring access method and physical layer specifications.

IEEE Std 802.6™, 1994 Edition [ISO/IEC 8802-6: 1994], IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 6: Distributed Queue Dual Bus (DQDB) access method and physical layer specifications.[4]

IEEE Std 802.9™, 1996 Edition [ISO/IEC 8802-9: 1996], IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 9: Integrated Services (IS) LAN Interface at the Medium Access Control (MAC) and Physical (PHY) Layers.[4]

IEEE Std 802.9a™-1995, IEEE Standards for Local and Metropolitan Area Networks: Supplement to Integrated Services (IS) LAN Interface at the Medium Access Control (MAC) and Physical (PHY) Layers: Specification of ISLAN 16-T.

IEEE Std 802.12™, 1998 Edition [ISO/IEC 8802-12: 1998], IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 12: Demand-Priority access method, physical layer and repeater specifications.[4]

IETF RFC 1042, Postel & Reynolds, A Standard for the Transmission of IP Datagrams over IEEE 802 Networks, February 1988.[5]

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

IETF RFC 2104, HMAC: Keyed-Hashing for Message Authentication, Krawczyk, H., Bellare, M., and R. Canetti, February 1997.[6]

ISO 6937-2: 1994, Information technology—Coded graphic character set for text communication—Latin alphabet.[7]

[4] This IEEE Standard has been withdrawn; future revisions of IEEE Std 802.1Q will address its presence in the text of IEEE Std 802.1Q, 2003 Edition.

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

[6] IETF documents are available at http://www.ietf.org.

[7] ISO and ISO/IEC documents are available from the ISO Central Secretariat, 1 rue de Varembé, Case Postale 56, CH-1211, Genève 20, Switzerland/Suisse; and from the Sales Department, American National Standards Institute, 11 West 42nd Street, 13th Floor, New York, NY 10036, USA.

ISO/IEC 7498-1: 1994, Information processing systems—Open Systems Interconnection—Basic Reference Model—Part 1: The Basic Model.

ISO/IEC 7498-4: 1989, Information processing systems—Open Systems Interconnection—Basic Reference Model—Part 4: Management framework.

ISO/IEC DIS 8802-11, Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications.

ISO/IEC 8824: 1990, Information technology—Open Systems Interconnection—Specification of Abstract Syntax Notation One (ASN.1) (Provisionally retained edition).

ISO/IEC 8825: 1990, Information technology—Open Systems Interconnection—Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1) (Provisionally retained edition).

ISO 9314-2: 1989, Information processing systems—Fibre Distributed Data Interface—Part 2: FDDI Token Ring Media Access Control (MAC).

ISO/IEC 9595: 1998, Information technology—Open Systems Interconnection—Common management information service.

ISO/IEC 9596: 1998, Information technology—Open Systems Interconnection—Common management information protocol—Part 1: Specification.

ISO/IEC 15802-1: 1995, Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Common specifications—Part 1: Medium Access Control (MAC) service definition.

3. Definitions

The following terms are specific to this standard:

3.1 Boundary Port: A Bridge Port attaching an MST Bridge to a LAN that is not in the same region.

3.2 Common and Internal Spanning Tree (CIST): The single Spanning Tree calculated by STP and RSTP together with the logical continuation of that connectivity through MST Bridges and regions, calculated by MSTP to ensure that all LANs in the Bridged Local Area Network are simply and fully connected.

3.3 Common Spanning Tree (CST): The single Spanning Tree calculated by STP and RSTP, and by MSTP to connect MST Regions.

3.4 detagged frame: The detagged frame of an untagged frame is the frame itself. The detagged frame of a tagged frame or a priority-tagged frame is the frame that results from untagging the frame by the appropriate procedure.

NOTE—The procedure for untagging a frame is discussed in 9.1.

3.5 Ethernet Type-encoding: The use of the Type interpretation of an IEEE 802.3 Length/Type field value in a frame as a protocol identifier associated with the MAC Service user data carried in the frame.[8]

NOTES

1—The term *frame* is defined in 3.6.

2—Ethernet Type-encoding can be used with MAC Service user data carried on non-IEEE 802.3 MACs by means of the SNAP-based encapsulation techniques specified in IEEE Std 802.1H, IETF RFC 1042, and IETF RFC 1390.

3.6 Frame: A unit of data transmission on an IEEE 802 LAN MAC that conveys a protocol data unit (PDU) between MAC Service users. There are three types of frame; *untagged*, *VLAN-tagged,* and *priority-tagged*.

NOTE—The term *IEEE 802 LAN* is defined in IEEE Std 802.1D, 1998 Edition. *Untagged frame* is defined in 3.36, *VLAN-tagged frame* is defined in 3.40, and *priority-tagged frame* is defined in 3.20.

3.7 Frame relay: The function of the Forwarding Process that forwards frames between the Ports of a Bridge.

NOTE—The operation of the Forwarding Process is defined in 8.7.

3.8 Independent Virtual Local Area Network (VLAN) Learning (IVL): Configuration and operation of the Learning Process and the Filtering Database such that, for a given set of VLANs, if a given individual MAC Address is learned in one VLAN, that learned information is not used in forwarding decisions taken for that address relative to any other VLAN in the given set.

NOTE—In a Bridge that supports only IVL operation, the “given set of VLANs” is the set of all VLANs.

3.9 Independent Virtual Local Area Network (VLAN) Learning (IVL) Bridge: A type of Bridge that supports only Independent VLAN Learning.

3.10 Internal Spanning Tree (IST): The connectivity provided by the CIST within a given MST Region.

[8] The use of Ethernet Type values as a means of protocol identification was defined in the specification of Ethernet V2.0 (The Ethernet, AA-K759B-TK, Digital Equipment, Intel, and Xerox Corps., Nov. 1982).

3.11 Legacy region: A set of LAN segments interconnected such that there is physical connectivity between any pair of segments using only IEEE Std 802.1D, 1998 Edition-conformant, VLAN-unaware MAC Bridges.

NOTE—In other words, if, in a Bridged LAN containing both IEEE Std 802.1D, 1998 Edition and IEEE 802.1Q Bridges, all the IEEE 802.1Q Bridges were to be removed, the result would be a set of one or more Bridged LANs, each with its own distinct Spanning Tree. Each of those Bridged LANs is a legacy region. The term *VLAN-unaware* is defined in 3.41.

3.12 Logical Link Control (LLC) encoding: The use of LLC addressing information in a frame as a protocol identifier associated with the MAC Service user data carried in the frame.

3.13 MST Bridge: A Bridge capable of supporting the CST, and one or more MSTIs, and of selectively mapping frames classified in any given VLAN to the CST or a given MSTI.

3.14 MST Configuration Table: A configurable table that allocates each and every possible VLAN to the Common Spanning Tree or a specific Multiple Spanning Tree Instance.

3.15 MST Region: A set of LANs and MST Bridges physically connected via Ports on those MST Bridges, where each LAN's CIST Designated Bridge is an MST Bridge, and each Port is either the Designated Port on one of the LANs, or else a non-Designated Port of an MST Bridge that is connected to one of the LANs, whose MCID matches exactly the MCID of the Designated Bridge of that LAN.

NOTE—It follows from this definition that the MCID is the same for all LANs and Ports in the Region, and that the set of MST Bridges in the region are interconnected by the LANs.

3.16 Multiple Spanning Tree Algorithm and Protocol (MSTP): The Multiple Spanning Tree Algorithm and Protocol described in Clause 13 of this standard.

3.17 Multiple Spanning Tree Bridge Protocol Data Unit (MST BPDU): The MST BPDU specified in Clause 14 of this standard.

3.18 Multiple Spanning Tree (MST) Configuration Identifier: A name for, revision level, and a summary of a given allocation of VLANs to Spanning Trees.

NOTE—Each MST Bridge uses a single MST Configuration Table and Configuration Identifier.

3.19 Multiple Spanning Tree Instance (MSTI): One of a number of Spanning Trees calculated by MSTP within an MST Region, to provide a simply and fully connected active topology for frames classified as belonging to a VLAN that is mapped to the MSTI by the MST Configuration Table used by the MST Bridges of that MST Region.

3.20 Priority-tagged frame: A tagged frame whose tag header carries priority information, but carries no VLAN identification information.

NOTE—The term *tagged frame* is defined in 3.33; *tag header* is defined in 3.34. See also *VLAN-tagged frame* (3.40) and *untagged frame* (3.36).

3.21 protocol group database: Specifies a group of protocols by assigning a unique protocol group identifier to all protocols of the same group.

3.22 protocol group identifier: Designates a group of protocols that are associated together when assigning a VID to a frame.

3.23 protocol template: A tuple of values that specify a data-link encapsulation format and an identification of the protocol layer above the data-link layer.

3.24 Rapid Spanning Tree Algorithm and Protocol (RSTP): The Rapid Spanning Tree Algorithm and Protocol described in Clause 17 of IEEE Std 802.1w-2001.

3.25 Rapid Spanning Tree Bridge Protocol Data Unit (RST BPDU): The RST BPDU specified in Clause 9 of IEEE Std 802.1w-2001.

3.26 Shared Virtual Local Area Network (VLAN) Learning (SVL): Configuration and operation of the Learning Process and the Filtering Database such that, for a given set of VLANs, if an individual MAC Address is learned in one VLAN, that learned information is used in forwarding decisions taken for that address relative to all other VLANs in the given set.

NOTE—In a Bridge that supports only SVL operation, the “given set of VLANs” is the set of all VLANs.

3.27 Shared Virtual Local Area Network (VLAN) Learning (SVL) Bridge: A type of Bridge that supports only Shared VLAN Learning.

3.28 Shared Virtual Local Area Network (VLAN) Learning (SVL)/Independent Virtual Local Area Network (VLAN) Learning (IVL) Bridge: An SVL/IVL Bridge is a type of Bridge that simultaneously supports both Shared VLAN Learning and Independent VLAN Learning.

3.29 Single Spanning Tree (SST) Bridge: A Bridge capable of supporting only a single spanning tree, the CST. The single spanning tree may be supported by the Spanning Tree Algorithm and Protocol (STP) defined in IEEE Std 802.1D, 1998 Edition, or by the Rapid Spanning Tree Algorithm and Protocol (RSTP), defined in IEEE Std 802.1w-2001.

3.30 Spanning Tree: A simply and fully connected active topology formed from the arbitrary physical topology of connected Bridged Local Area Network components by relaying frames through selected bridge ports and not through others. The protocol parameters and states used and exchanged to facilitate the calculation of that active topology and to control the bridge relay function.

3.31 Spanning Tree Algorithm and Protocol (STP): The Spanning Tree Algorithm and Protocol described in Clause 8 of IEEE Std 802.1D, 1998 Edition.

3.32 Spanning Tree Bridge Protocol Data Unit (ST BPDU): A Bridge Protocol Data Unit specified for use by the Spanning Tree Algorithm and Protocol, i.e. a Configuration or Topology Change Notification BPDU as described in Clause 9 of IEEE Std 802.1D, 1998 Edition.

3.33 Tagged frame: A *tagged frame* is a frame that contains a tag header immediately following the Source MAC Address field of the frame or, if the frame contained a Routing Information field, immediately following the Routing Information field. There are two types of tagged frames: VLAN-tagged frames and priority-tagged frames.

NOTE—The term *tag header* is defined in 3.34, *priority-tagged frame* is defined in 3.20, and *VLAN-tagged frame* is defined in 3.40. See also *untagged frame* (3.36).

3.34 Tag header: A tag header allows user priority information, and optionally, VLAN identification information, to be associated with a frame.

NOTE—The structure of the tag header is defined in 9.3.

3.35 Terms used in IEEE Std 802.1D, 1998 Edition: The following terms used in this standard are used in IEEE Std 802.1D, 1998 Edition:
Active topology
Bridge Port
Bridged Local Area Network (also Bridged LAN)
GARP Participant
GARP Application
GIP Context
Group
IEEE 802 Local Area Network (also IEEE 802 LAN, or LAN)
Port

3.36 Untagged frame: An *untagged frame* is a frame that does not contain a tag header immediately following the Source MAC Address field of the frame or, if the frame contained a Routing Information field, immediately following the Routing Information field.

NOTE—The term *tag header* is defined in 3.34. See also *tagged frame* (3.33).

3.37 Virtual Bridged Local Area Network (LAN): A Bridged LAN in which the existence of one or more VLAN-aware Bridges allows the definition, creation, and maintenance of VLANs.

NOTE—The term *VLAN-aware* is defined in 3.39.

3.38 Virtual Local Area Network (VLAN): A subset of the active topology of a Bridged Local Area Network. Associated with each VLAN is a VLAN Identifier (VID).

3.39 VLAN-aware: A property of Bridges or end stations that recognize and support VLAN-tagged frames.

3.40 VLAN-tagged frame: A tagged frame whose tag header carries both VLAN identification and priority information.

3.41 VLAN-unaware: A property of Bridges or end stations that do not recognize VLAN-tagged frames.

4. Abbreviations

The following abbreviations are used in this standard:

BPDU	Bridge Protocol Data Unit (IEEE Std 802.1D, 1998 Edition)
CFI	Canonical Format Indicator (Annex F)
CIST	Common and Internal Spanning Tree
CST	Common Spanning Tree
E-ISS	Enhanced Internal Sublayer Service (7.1)
FCS	Frame Check Sequence (7.1)
FID	Filtering Identifier (8.11.3, 8.11.7)
GARP	Generic Attribute Registration Protocol (IEEE Std 802.1D, 1998 Edition)
GID	GARP Information Declaration (IEEE Std 802.1D, 1998 Edition)
GIP	GARP Information Propagation (IEEE Std 802.1D, 1998 Edition)
GMRP	GARP Multicast Registration Protocol (Clause 10, IEEE Std 802.1D, 1998 Ed.)
GVRP	GARP VLAN Registration Protocol (Clause 11)
ISS	Internal Sublayer Service (Clause 7, IEEE Std 802.1D, 1998 Edition)
IST	Internal Spanning Tree
IVL	Independent VLAN Learning (3.8)
LAN	Local Area Network (IEEE Std 802)
LS	Least-significant
LLC	Logical Link Control (IEEE Std 802.2)
MAC	Medium Access Control (IEEE Std 802)
MCID	MST Configuration Identifier
MIB	Management Information Base (ISO/IEC 7498-4)
MS	Most-significant
MSDU	MAC Service Data Unit (ISO/IEC 15802-1)
MST	Multiple Spanning Tree
MST BPDU	Multiple Spanning Tree Bridge Protocol Data Unit
MSTI	Multiple Spanning Tree Instance
MSTP	Multiple Spanning Tree Protocol
NCFI	Non-Canonical Format Indicator (Annex F)
PDU	Protocol Data Unit
PICS	Protocol Implementation Conformance Statement (Annex A)
PID	Protocol IDentifier
PVID	Port VID (8.4.4)
RIF	Routing Information Field (IEEE Std 802.5)
RST BPDU	Rapid Spanning Tree Bridge Protocol Data Unit
RSTP	Rapid Spanning Tree Protocol
SST	Single Spanning Tree
ST BPDU	Spanning Tree Bridge Protocol Data Unit
STP	Spanning Tree Protocol
STPID	SNAP-encoded Tag Protocol Identifier (9.3)
SVL	Shared VLAN Learning (3.26)
TCI	Tag Control Information (9.3)
TPID	Tag Protocol Identifier (9.3)
VID	VLAN Identifier (7.1, 8.4.4, 9.3)
VLAN	Virtual LAN (3.38)

5. Conformance

5.1 Static conformance requirements

A MAC Bridge for which conformance to this standard is claimed shall

a) Conform to the requirements of IEEE Std 802.1D, 1998 Edition, and IEEE Std 802.1w-2001, as modified by the provisions of this standard;

b) Relay and filter frames as described in 8.1 and specified in 8.5, 8.6, 8.7, 8.8, and 8.9;

c) On each Port, support at least one of the permissible values for the Acceptable Frame Types parameter, as defined in 8.4.3;

d) Support the following on each Port that supports untagged and priority-tagged frames:

   1) A Port VLAN Identifier (PVID) value (8.4.4);

   2) The ability to configure at least one VLAN whose untagged set includes that Port (8.8 and 8.11.9);

   3) Configuration of the PVID value via management operations (12.10);

   4) Configuration of Static Filtering Entries via management operations (12.7).

e) Support the ability to insert tag headers into, modify tag headers in, and remove tag headers from relayed frames, as described and specified in 7.1 and Clause 9, as required by the value(s) of the Acceptable Frame Types parameter that are supported on each Port, and by the ability of each Port to be configured to transmit VLAN-tagged frames and/or untagged frames. These requirements are summarized in Table 5-1 for frames relayed between any pair of Ports;

Table 5-1—Support requirements for insertion, removal, and modification of tag headers

		Reception Port receives as (and does not discard):		
		VLAN-tagged	**Priority-tagged**	**Untagged**
Transmission Port transmits as:	**Untagged**	Shall support removal of tag headers.	Shall support removal of tag headers.	N/A.
	VLAN-tagged	Shall support conversion of the tagged frame format if the format required for the destination MAC differs from the received format.	Shall support the insertion of a non-null Virtual LAN Identifier (VID) in tag headers, plus conversion of the tagged frame format if the format required for the destination MAC differs from the received format.	Shall support the insertion of tag headers of a format appropriate to the destination MAC, carrying a non-null VID.

f) Support the ability to perform automatic configuration and management of VLAN topology information by means of Generic Attribute Registration Protocol (GARP) VLAN Registration Protocol (GVRP) (Clause 11) on all Ports;

g) Support the ability for the Filtering Database to contain static and dynamic configuration information for at least one VLAN, by means of Static and Dynamic VLAN Registration Entries (8.11);

h) Support at least one Filtering Identifier (FID) (6.4, 8.11.3, 8.11.7, and 8.11.8);

i) Support the ability to allocate at least one VID to each FID that is supported (6.4, 8.11.3, 8.11.7, and 8.11.8).

NOTE 1—Under some circumstances, the ability for VLAN Bridges to successfully interoperate depends upon the number of FIDs supported, and the number of VIDs that can be allocated to each FID. These circumstances are discussed in Annex B, along with the implications with respect to interoperability.

j) A MAC bridge for which conformance to the provisions of this standard for an MST Bridge is claimed (5.2, 13) shall:
    1) Support the Multiple Spanning Tree Algorithm and Protocol, (MSTP) as specified in Clause 13 and the attendant conformance requirements stated in Clause 5 of IEEE Std 802.1w-2001;
    2) Support the CIST plus a stated maximum number of MSTIs, where that number is at least 2 (8.10.7) and at most 64 (13.14);

NOTE 2—In other words, a conformant MST Bridge must support a minimum of three spanning tree instances—the CIST and at least two additional MSTIs.

3) Support a stated maximum number of FIDs that is at least as large as the number of MSTIs supported (8.10.7);
4) Support the ability to associate each FID to a spanning tree (8.11.3);
5) Support the transmission and reception of MST Configuration Identifier information (8.11.2).
6) For each Port, be capable of supporting a set of port-state information for each spanning tree instance supported (8.4.1, 8.4.2, 13.34);
7) For each Port, be capable of supporting an instance of the Spanning Tree Protocol and its associated parameters for each spanning tree instance supported (8.12, 13);
8) Use the Bridge Group Address as specified in 8.14.3;
9) Support the default values for Bridge Forward Delay and Bridge Priority parameters specified in 13.23;
10) Support the operation of GVRP in each supported spanning tree context (11.2.3.3, 11.2.3.4);
11) Support the VLAN bridge management functions for the bridge protocol entity for each supported spanning tree, independently (12.8)
12) Support, in particular, management of the bridge priority parameters, and of the port priority and path cost parameters for every port, independently for each supported spanning tree (12.8.1.1, 12.8.1.3, 13.24);
13) Support the VLAN management functions for each supported spanning tree (12.10.1 and 12.11.1);
14) Support management of the MSTI configuration (12.12).

5.2 Options

A MAC Bridge for which conformance to this standard is claimed may

a) Support operation in Extended Filtering Mode (IEEE Std 802.1D, 1998 Edition, 6.6.5) and the operation of GARP Multicast Registration Protocol (GMRP) (IEEE Std 802.1D, 1998 Edition, Clause 10) as modified by Clause 10;
b) Support the ability for the Filtering Database to contain static and dynamic configuration information for more than one VLAN, by means of Static and Dynamic VLAN Registration Entries (8.11), up to a maximum of 4094 VLANs;

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

c) On each Port, support both of the permissible values for the Acceptable Frame Types parameter, as defined in 8.4.3. If both values are supported, then the implementation shall support configuration of the parameter value via management;
d) Support the ability to enable and disable Ingress Filtering (8.4.5);

e) Support the ability to configure more than one VLAN whose untagged set includes that Port (8.8 and 8.11.9);

f) Support the management functionality defined in Clause 12;

g) Support more than one FID (6.4, 8.11.3, 8.11.7, and 8.11.8);

h) Support the ability to allocate more than one VID to each FID that is supported (6.4, 8.11.3, 8.11.7, and 8.11.8);

i) Support the ability to configure VLAN Learning Constraints via management (8.11.7 and 12.10.3);

j) Support the ability to configure fixed VID to FID allocations via management (8.11.7.1 and 12.10.3);

k) Support any other optional capabilities defined in IEEE Std 802.1D, 1998 Edition, as modified by the provisions of this standard.

l) Support Port-and-Protocol-based VLAN classification, including multiple VID values per port, administrative control of the values of the multiple VIDs, and a Protocol Group Database. If this option is supported:

    1) One or more of the following Protocol Classifications and Protocol Template formats must be supported: Ethernet, RFC_1042, SNAP_8021H, SNAP_Other, or LLC_Other (8.6.1 and 8.6.2).

    2) The configuration of the contents of the Protocol Group Database may be supported.

m) Support the ability to configure the value of the Restricted_Group_Registration parameter (10.3.2 in IEEE Std 802.1t-2001) for each Port of the Bridge;

n) Support the ability to configure the value of the Restricted_VLAN_Registration parameter (11.2.3.2.3) for each Port of the Bridge.

o) Support MST operation (7, 8.4.1, 8.4.2, 8.6.2, 8.10.7, 8.11, 8.12, 8.14.3, 11.2.3.4, 11.3.1, 13, 14.);

p) A MAC bridge for which support of MSTP is claimed (13) may support a greater number of FIDs than spanning trees (8.10.7).

5.3 Protocol Implementation Conformance Statement (PICS)

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

5.4 MAC-specific bridging methods

MAC-specific bridging methods may exist. Use of a MAC-specific bridging method and the method specified in this standard on the same LAN shall

a) Not prevent communication between stations in a Bridged LAN.

b) Preserve the MAC Service.

c) Preserve the characteristics of each bridging method within its own domain.

d) Provide for the ability of both bridging techniques to coexist simultaneously on a LAN without adverse interaction.

IEEE Std 802.1D, 1998 Edition, Annex C, defines one such MAC-specific bridging method, source routing, and that method is also a provision of this standard. While this standard defines how source-routed frames can be transported in a VLAN environment, it does not attempt to specify VLAN aspects of the source routing bridging method itself.

6. Support of the MAC Service in VLANs

The provisions of Clause 6 of IEEE Std 802.1D apply to this standard, with the additions and modification defined in this clause.

6.1 Support of the MAC service

The MAC Service provided to end stations attached to a Virtual Bridged LAN is the (unconfirmed) connectionless mode MAC Service defined in ISO/IEC 15802-1. The MAC Service is defined as an abstraction of the features common to a number of specific MAC Services; it describes the transfer of user data between source and destination end stations, via MA-UNITDATA request primitives and corresponding MA-UNITDATA indication primitives issued at MAC Service access points. Each MA-UNITDATA request and indication primitive has four parameters: Destination Address, Source Address, MAC Service data unit (MSDU), and Priority.

The style of VLAN Bridge operation maximizes the availability of the MAC Service to end stations and assists in the maintenance of the Virtual Bridged LAN. It is therefore desirable that VLAN Bridges be capable of being configured in the Virtual Bridged LAN:

a) So as to provide redundant paths between end stations to enable the Virtual Bridged LAN to continue to provide the Service in the event of component failure (of VLAN Bridge or LAN).
b) So that the paths supported between end stations are predictable and configurable given the availability of Virtual Bridged LAN components.

The operation of VLAN Bridges supports the provision of the MAC Service only to devices that are authenticated and authorized for such use. Unauthorized devices may be denied access to the Virtual Bridged LAN, other than as necessary to support the protocol exchanges that are required by any authentication process that is supported.

NOTE—Authentication and authorization to access a LAN may be achieved by administrative or management mechanisms, or by means of an active authorization mechanism, such as is defined in IEEE Std 802.1X-2001.

6.2 Preservation of the MAC service

The MAC Service offered by a Virtual Bridged LAN consisting of LANs interconnected by VLAN Bridges is similar to that offered by a single LAN (see 6.3). In consequence,

a) A VLAN Bridge is not directly addressed by communicating end stations, except as an end station for management purposes: frames transmitted between end stations carry the MAC Address of the peer-end station in their Destination Address field, not the MAC Address, if any, of the VLAN Bridge.
b) All MAC Addresses must be unique within any set of VLANs supported by the Virtual Bridged LAN that share the same Filtering Identifier (FID).
c) The MAC Addresses of end stations are not restricted by the topology and configuration of the Virtual Bridged LAN.

6.3 Quality of service maintenance

6.3.1 Service availability

Service availability is measured as that fraction of some total time during which the MAC Service is provided. The operation of a VLAN Bridge can increase or lower the service availability.

The service availability can be increased by automatic reconfiguration of the Virtual Bridged LAN in order to avoid the use of a failed component (e.g., repeater, cable, or connector) in the data path. The service availability can be lowered by failure of a VLAN Bridge itself, through denial of service by the VLAN Bridge, or through frame filtering by the VLAN Bridge. Changes in topology, caused by component failures, the addition or removal of components, or by administrative changes, are detected and signalled by the following means:

a) Physical detection of component failure and signalling of that failure by the Enhanced Internal Sublayer Service (6.4 and 6.5);
b) Detection of component failure through the operation of a spanning tree algorithm and protocol;
c) Explicit signalling of reconfiguration events through the operation of a spanning tree algorithm and protocol.

Automatic reconfiguration can be achieved rapidly on the detection of a physical topology change (see Clause 17 of IEEE Std 802.1w-2001), thus minimizing any service denial that is caused by the reconfiguration.

A VLAN Bridge may deny service and discard frames (6.3.2) in order to preserve other aspects of the MAC Service (6.3.3 and 6.3.4) when automatic reconfiguration takes place. Service may be denied to end stations that do not benefit from the reconfiguration; hence, the service availability is lowered for those end stations. VLAN Bridges may filter frames in order to localize traffic in the Virtual Bridged LAN. Should an end station move, it may then be unable to receive frames from other end stations until the filtering information held by the VLAN Bridges is updated.

To minimize the effects of service denial caused by reconfiguration events, filtering information that has been dynamically learnt can be modified when automatic reconfiguration takes place, or in preparation for future reconfiguration events (Clause 17 and 17.10 of IEEE Std 802.1w-2001). However, filtering information that is statically configured cannot be modified in this way.

A VLAN Bridge may deny service and discard frames in order to prevent access to the network by devices that are not authorized for such access.

To maximize the service availability, no loss of service or delay in service provision should be caused by VLAN Bridges, except as a consequence of a failure, removal, or insertion of a Virtual Bridged LAN component, or as a consequence of the movement of an end station, or as a consequence of an attempt to perform unauthorized access. These are regarded as extraordinary events. The operation of any additional protocol necessary to maintain the quality of the MAC Service is thus limited to the configuration of the Virtual Bridged LAN, and is independent of individual instances of service provision.

NOTE 1—This is true only in circumstances where admission control mechanisms are not present, i.e., where the VLAN Bridges provide a "best effort" service. The specification and applicability of admission control mechanisms in VLAN Bridges is outside the scope of this standard.

NOTE 2—The operation of management upon the Bridge can result in the Bridge being reset, either as a result of a specific Bridge reset operation, or as a consequence of manipulating the Bridge's configuration. From the point of view of service availability, resetting the Bridge is an extraordinary event that has a similar effect to physical removal of the Bridge from the Virtual Bridged LAN, followed by reinsertion of the Bridge into the Virtual Bridged LAN.

6.3.2 Frame loss

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

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

a) Frame corruption during physical layer transmission or reception.
b) Frame discard by a VLAN Bridge because
   1) It is unable to transmit the frame within some maximum period of time and, hence, must discard the frame to prevent the maximum frame lifetime (6.3.6) from being exceeded.
   2) It is unable to continue to store the frame due to exhaustion of internal buffering capacity as frames continue to arrive at a rate in excess of that at which they can be transmitted.
   3) The size of the service data unit carried by the frame exceeds the maximum supported by the MAC procedures employed on the LAN to which the frame is to be relayed.
   4) Changes in the connected topology of the Virtual Bridged LAN necessitate frame discard for a limited period of time to maintain other aspects of Quality of Service (see 8.3.3 of IEEE Std 802.1D, 1998 Edition, and 17.9 of IEEE Std 802.1w-2001).
   5) The device attached to the Port is not authorized for access to the network.
   6) The configuration of Static Filtering Entries or Static VLAN Registration Entries in the Filtering Database (8.11.1, 8.11.2) disallows the forwarding of frames with particular destination addresses or VLAN classifications on specific Ports.

NOTE—As Static Filtering Entries and Static VLAN Registration Entries are associated with particular Ports or combinations of Ports, there is a possibility that mis-configuration of such entries will lead to unintended frame discard during or following automatic reconfiguration of the Virtual Bridged LAN.

6.3.3 Frame misordering

The MAC Service (9.2 of ISO/IEC 15802-1) permits a negligible rate of reordering of frames with a given user priority for a given combination of destination address and source address, transmitted on a given VLAN. MA_UNITDATA.indication service primitives corresponding to MA_UNITDATA.request primitives, with the same requested priority and for the same combination of VLAN classification, destination and source addresses, are received in the same order as the request primitives were processed.

NOTE 1—The operation of the Forwarding Process in VLAN Bridges (8.7) is such that the frame-ordering characteristics of the MAC Service are preserved.

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

NOTE 2—Where STP is in use (see Clause 8 of IEEE Std 802.1D, 1998 Edition), frame misordering cannot occur during normal operation. Where RSTP is in use (see Clause 17 of IEEE Std 802.1w-2001), there is an increased probability that frames that are in transit through the Virtual Bridged LAN will be misordered, due to the fact that a VLAN Bridge can buffer frames awaiting transmission through its Ports. The probability of misordering occurring as a result of such an event is dependent upon implementation choices, and is associated with Spanning Tree reconfiguration events. Some known LAN protocols, for example, LLC Type 2, are sensitive to frame duplication; in order to allow VLAN Bridges that support RSTP to be used in environments where sensitive protocols are in use, the forceVersion parameter (17.16.1 of IEEE Std 802.1w-2001) can be used to force a VLAN Bridge that supports RSTP to operate in an STP-compatible manner. A more detailed discussion of misordering in RSTP can be found in F.2.4 of IEEE Std 802.1w-2001.

6.3.4 Frame duplication

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

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

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

NOTE—Where RSTP is in use (see Clause 17 of IEEE Std 802.1w-2001), there is an increased probability that a Spanning Tree reconfiguration event can cause frames that are in transit through the Virtual Bridged LAN to be duplicated, due to the fact that a VLAN Bridge can buffer frames awaiting transmission through its Ports. As the probability of duplication occurring as a result of such an event is small, and the frequency of Spanning Tree reconfiguration events is also small, the degradation of the properties of the MAC service caused by this source of frame duplication is considered to be negligible. A more detailed discussion of frame duplication in RSTP can be found in F.2.4 of IEEE Std 802.1w-2001.

6.3.5 Transit delay

The MAC Service introduces a frame transit delay that is dependent on the particular media and MAC method employed. Frame transit delay is the elapsed time between an MA_UNITDATA.request primitive and the corresponding MA_UNITDATA.indication primitive. Elapsed time values are calculated only on Service Data Units that are successfully transferred.

Since the MAC Service is provided at an abstract interface within an end station, it is not possible to specify precisely the total frame transit delay. It is, however, possible to measure those components of delay associated with media access and with transmission and reception; and the transit delay introduced by an intermediate system, in this case a VLAN Bridge, can be measured.

The minimum additional transit delay introduced by a VLAN Bridge is the time taken to receive a frame plus that taken to access the media onto which the frame is to be relayed. Note that the frame is completely received before it is relayed as the Frame Check Sequence (FCS) is to be calculated and the frame discarded if in error.

6.3.6 Frame lifetime

The MAC Service ensures that there is an upper bound to the transit delay experienced for a particular instance of communication. This maximum frame lifetime is necessary to ensure the correct operation of higher layer protocols. The additional transit delay introduced by a VLAN Bridge is discussed in 6.3.5.

To enforce the maximum frame lifetime, a VLAN Bridge may be required to discard frames. Since the information provided by the MAC Sublayer to a VLAN Bridge does not include the transit delay already experienced by any particular frame, VLAN Bridges must discard frames to enforce a maximum delay in each VLAN Bridge.

The value of the maximum bridge transit delay is based on both the maximum delays imposed by all the VLAN Bridges in the Virtual Bridged LAN and the desired maximum frame lifetime. A recommended and an absolute maximum value are specified in Table 8-2 of IEEE Std 802.1D, 1998 Edition.

6.3.7 Undetected frame error rate

The MAC Service introduces a very low undetected frame error rate in transmitted frames. Undetected errors are protected against by the use of an FCS that is appended to the frame by the MAC Sublayer of the source station prior to transmission, and checked by the destination station on reception.

The FCS calculated for a given service data unit is dependent on the MAC method employed. It is therefore necessary to recalculate the FCS within a VLAN Bridge providing a relay function between IEEE 802 MACs of dissimilar types if differences in the method of calculation and/or the coverage of the FCS, or changes to the data that is within the coverage of the FCS, would lead to a different FCS being calculated for the service data unit by the two MAC methods. This introduces the possibility of additional undetected errors arising from the operation of a VLAN Bridge. For frames relayed between LANs of the same MAC type, the VLAN Bridge shall not introduce an undetected frame error rate greater than that which would have been achieved by preserving the FCS.

NOTE—Application of the techniques described in Annex G of IEEE Std 802.1D, 1998 Edition, allow an implementation to achieve an arbitrarily small increase in undetected frame error rate, even in cases where the data that is within the coverage of the FCS is changed. As a maintenance activity on this standard, revision of the wording of this requirement will be initiated, with a view to placing a quantitative limit on the increase in undetected frame error rate that is acceptable in a conformant implementation.

6.3.8 Maximum Service Data Unit Size

The Maximum Service Data Unit Size that can be supported by an IEEE 802 LAN varies with the MAC method and its associated parameters (speed, electrical characteristics, etc.). It may be constrained by the owner of the LAN. The Maximum Service Data Unit Size supported by a VLAN Bridge between two LANs is the smaller of that supported by the LANs. No attempt is made by a VLAN Bridge to relay a frame to a LAN that does not support the size of Service Data Unit conveyed by that frame.

6.3.9 Priority

The MAC Service includes user priority as a Quality of Service parameter. MA_UNITDATA.request primitives with a high priority may be given precedence over other request primitives made at the same station, or at other stations attached to the same LAN, and can give rise to earlier MA_UNITDATA.indication primitives.

The MAC Sublayer maps the requested user priorities onto the access priorities supported by the individual MAC method. The requested user priority may be conveyed to the destination station.

The transmission delay experienced by a frame in a VLAN Bridge can be managed by associating a user_priority with the frame.

The transmission delay comprises

a) A queuing delay until the frame becomes first in line for transmission on the Port, in accordance with the procedure for selecting frames for transmission described in 8.7.4;
b) The access delay for transmission of the frame.

Queueing delays can be managed using user_priority. Access delays can be managed using user_priority in MAC methods that support more than one access priority.

The user priority associated with a frame can be signaled by means of the priority signaling mechanisms inherent in some IEEE 802 LAN MAC types. Since not all IEEE 802 LAN MAC types are able to signal the user priority associated with a frame, VLAN Bridges regenerate user priority based upon a combination of signaled information and configuration information held in the VLAN Bridge.

The VLAN Bridge maps the user priority onto one or more traffic classes; VLAN Bridges that support more than one traffic class are able to support expedited classes of traffic. The Forwarding Process, 7.7 8.7, describes the use of user priority and traffic classes in VLAN Bridges. Given the constraints placed upon frame misordering in a VLAN Bridge, as expressed in 6.3.3, the mappings of priority and traffic class are static.

NOTE 1—The term Traffic Class, as used in this standard, is used only in the context of the operation of the priority handling and queueing functions of the Forwarding Process, as described in 8.7. Any other meanings attached to this term in other contexts do not apply to the use of the term in this standard.

The ability to signal user_priority in IEEE 802 LANs allows user_priority to be carried with end-to-end significance across a Virtual Bridged LAN. This, coupled with a consistent approach to the mapping of user_priority to traffic classes, and of user_priority to access_priority, allows consistent use of user_priority information to be made, according to the capabilities of the VLAN Bridges and MAC methods that are involved in the transmission path.

NOTE 2—This standard defines a frame format and associated procedures that can be used to carry user priority information across LAN MAC types that are not able to signal user priority. Use of the 802.1Q frame format allows the end-to-end significance of user_priority to be maintained regardless of the ability of individual LAN MAC types to signal priority.

Under normal circumstances, user_priority is not modified in transit through the relay function of a VLAN Bridge; however, there may be some circumstances where it is desirable for management purposes to control how user_priority is propagated. The User Priority Regeneration Table (Table ~~7-1~~ 8-1) provides the ability to map incoming user_priority values on a per-Port basis, under management control. In its default state, this table provides an identity mapping from user_priority values to Regenerated user_priority values; i.e., by default, the Regenerated user_priority is identical to the incoming user_priority.

6.3.10 Throughput

The total throughput provided by a Virtual Bridged LAN can be significantly greater than that provided by an equivalent single LAN. VLAN Bridges may localize traffic within the Virtual Bridged LAN by filtering frames. Filtering services available in Virtual Bridged LANs are described in 6.6 of IEEE Std 802.1D, 1998 Edition.

The throughput between end stations on individual LANs, communicating through a VLAN Bridge, can be lowered by frame discard in the VLAN Bridge due to the inability to transmit at the required rate on the LAN forming the path to the destination for an extended period.

6.4 Enhanced Internal Sublayer Service provided within VLAN Bridges

The Enhanced Internal Sublayer Service (E-ISS) is derived from the Internal Sublayer Service (ISS, defined in 6.4 of IEEE Std 802.1D, 1998 Edition) by augmenting that specification with elements necessary to the operation of the tagging and untagging functions of the VLAN Bridge. Within the attached end station, these elements can be considered to be either below the MAC Service boundary, and pertinent only to the operation of the service provider; or local matters not forming part of the peer-to-peer nature of the MAC Service.

Bridges that support these functions are known as *VLAN-aware Bridges* (3.17). The E-ISS defines the MAC Service provided to the relay function in VLAN-aware Bridges.

The relationships between the MAC Entity, the ISS, the E-ISS, and the MAC Relay Entity in a VLAN-aware Bridge are illustrated in Figure 6-1 and Figure 8-3.

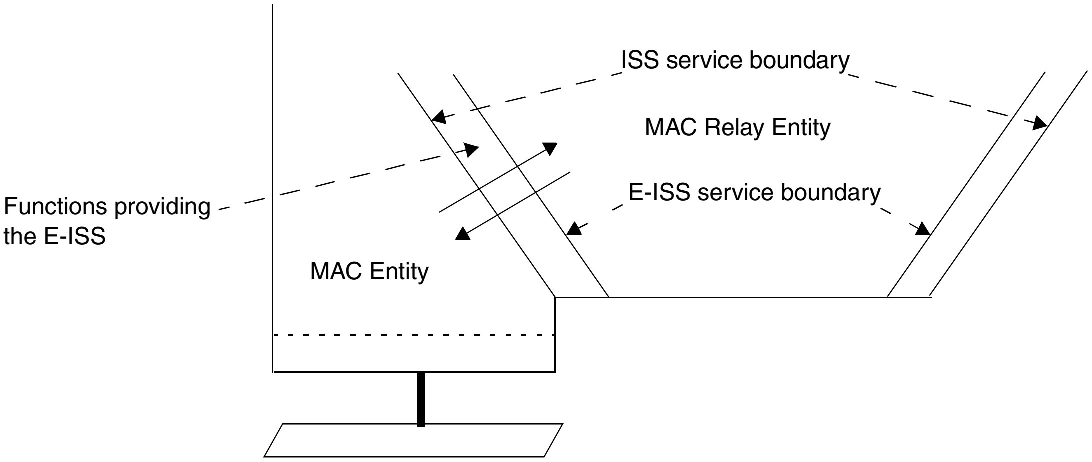


Figure 6-1—Relationships between MAC Entity, ISS, E-ISS, and MAC Relay Entity

6.4.1 E-ISS service definition

The unit-data primitives that define this service are

```
EM_UNITDATA.indication     (
                           frame_type,
                           mac_action,
                           destination_address,
                           source_address,
                           mac_service_data_unit,
                           user_ priority,
                           frame_check_sequence,
                           canonical_format_indicator,
                           vlan_identifier,
                           rif_information (optional)
                           )
```

Each data indication primitive corresponds to the receipt of a M_UNITDATA.indication primitive from the Internal Sublayer Service, as defined in 6.4 of IEEE Std 802.1D, 1998 Edition.

The **frame_type, mac_action, destination_address, source_address, mac_service_data_unit, user_ priority,** and **frame_check_sequence** parameters are as defined for the M_UNITDATA.indication primitive of the Internal Sublayer service.

The **canonical_format_indicator** parameter indicates whether embedded MAC Addresses carried in the mac_service_data_unit parameter are in Canonical format or Non-canonical format. The value False indicates Non-canonical format. The value True indicates Canonical format.

NOTE—The meanings of the terms Canonical format and Non-canonical format are discussed in Annex F.

The **vlan_identifier** parameter carries the VLAN identifier associated with the indication.

The **rif_information** parameter is present if a tag header was present in the indication, and if that tag header contained a Routing Information Field (RIF). Its value is equal to the value of the RIF.

```
EM_UNITDATA.request          (
                             frame_type,
                             mac_ action,
                             destination_address,
                             source_address,
                             mac_service_data_unit,
                             user_priority,
                             access_priority,
                             frame_check_sequence,
                             canonical_format_indicator,
                             vlan_classification,
                             rif_information (optional),
                             include_tag
                             )
```

A data request primitive is invoked in order to generate a M_UNITDATA.request primitive, as defined in the Internal Sublayer Service, 6.4 of IEEE Std 802.1D, 1998 Edition.

The **frame_type, mac_action, destination_address, source_address, mac_service_data_unit, user_priority, access_priority,** and **frame_check_sequence** parameters are as defined for the M_UNITDATA.request primitive of the Internal Sublayer service.

The definition of the **canonical_format_indicator** parameter is as defined for the EM_UNITDATA.indication.

The vlan_classification parameter carries the VLAN classification assigned to the frame by the forwarding process (8.6.1).

The **rif_information parameter**, if present, carries the value of any RIF information to be associated with the request.

The **include_tag** parameter carries a Boolean value. True indicates to the service provider that the mac_service_data_unit parameter of the data request shall include a tag header (9.3). False indicates that a tag header shall not be included.

6.4.2 Support of the E-ISS in VLAN-aware Bridges

6.4.2.1 Data indication primitives

On receipt of a data indication from the Internal Sublayer Service, an EM_UNITDATA.indication primitive is invoked, with parameter values as follows:

The **frame_type, mac_action, destination_address, source_address,** and **frame_check_sequence** parameters carry values equal to the corresponding parameters in the received data indication.

NOTE 1—The **mac_action** parameter only ever takes the value request_with_no_response for frames relayed by the Bridge. The **frame_check_sequence** parameter of the data indication carries the FCS value contained in the received frame. The original FCS associated with a frame is invalidated if there are changes to any fields of the frame, if fields are added or removed, or if bit ordering or other aspects of the frame encoding have changed. An invalid FCS is signalled in the E-ISS by an unspecified value in the frame_check_sequence parameter of the data request primitive. This signals the need for the FCS to be regenerated according to the normal procedures for the transmitting MAC. The options for regenerating the FCS under these circumstances are discussed in Annex G of IEEE Std 802.1D, 1998 Edition.

The value of the **mac_service_data_unit** parameter is determined as follows:

a) If the received mac_service_data_unit parameter contained a tag header (9.3), then the value used is equal to the value of the received mac_service_data_unit following removal of the tag header. Otherwise;
b) The value used is equal to the value of the received mac_service_data_unit.

The value of the **user_priority** parameter is determined as follows:

c) If the received mac_service_data_unit parameter contained a tag header (9.3), then the value contained in the user_priority field of the tag header is used. Otherwise;
d) The value of the received user_ priority parameter, regenerated as defined in 8.5.1 and 6.4 of IEEE Std 802.1D, 1998 Edition, is used.

The value of the **canonical_format_indicator** parameter is determined as follows:

e) If the received mac_service_data_unit parameter contained a tag header (9.3), then the value(s) contained in the Canonical Format Indicator (CFI) (and Non-Canonical Format Indicator [NCFI], if present) field(s) of the tag header are used to determine this parameter value, in accordance with the definition of the CFI and NCFI field(s) in Clause 9. Otherwise;
f) If the MAC entity that received the data indication was an ISO/IEC 8802-5 Token Ring MAC, then the parameter carries the value False. Otherwise;
g) The parameter carries the value True.

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

h) If the initial octets of the received mac_service_data_unit parameter contained a tag header (9.3), then the value contained in the VID field of the tag header is used. Otherwise;
i) A value equal to the null VLAN ID (as defined in Table 9-2) is used.

The value of the **rif_information** parameter is determined as follows:

j) If the initial octets of the received mac_service_data_unit parameter contained a tag header (9.3), and that tag header contained a RIF field in which one or more route descriptors were present, then the value contained in the RIF field is used. Otherwise;
k) The parameter is not present.

NOTE 2—This field can be present only in tag headers received using the 802.3/Ethernet or transparent FDDI MAC methods. The presence of one or more route descriptors indicates that there is source-routing information present in the received frame.

6.4.2.2 Data request primitives

On invocation of a data request primitive by a user of the E-ISS, an M-UNITDATA.request primitive is invoked, with parameter values as follows:

The **frame_type, mac_action, destination_address, source_address, user_priority,** and **access_priority** parameters carry values equal to the corresponding parameters in the received data request.

If the value of the **include_tag** parameter is False, the value of the mac_service_data_unit parameter is determined as follows:

a) If the destination MAC method is the same as the MAC method on which the corresponding data indication was received, then the value used is equal to the value of the mac_service_data_unit parameter received in the data request. Otherwise;
b) The value used is equal to the value of the mac_service_data_unit parameter received in the data request, modified, if necessary, in accordance with the procedures described in ISO/IEC 11802-5, IETF RFC 1042, and IETF RFC 1390.
c) If the canonical_format_indicator parameter indicates that the mac_service_data_unit may contain embedded MAC Addresses in a format inappropriate to the destination MAC method, then the Bridge shall either
    1) Convert any embedded MAC Addresses in the mac_service_data_unit to the format appropriate to the destination MAC method; or
    2) Discard the EISS data request without issuing a corresponding ISS data request.

If the value of the **include_tag parameter** is True, then a tag header, formatted as necessary for the destination MAC method, is inserted as the first N octets of the mac_service_data_unit parameter. The values of the user_priority, canonical_format_indicator, vlan_classification, and rif_information (if present) parameters are used to determine the contents of the tag header, in accordance with the structure defined in 9.2 and 9.3. The value inserted after the tag header is determined as follows:

d) If the destination MAC method is the same as the MAC method on which the corresponding data indication was received, then the value used is equal to the value of the mac_service_data_unit parameter received in the data request. Otherwise;
e) The value used is equal to the value of the mac_service_data_unit parameter received in the data request, modified, if necessary, in accordance with the procedures described in ISO/IEC 11802-5, IETF RFC 1042, and IETF RFC 1390.

The value of the **frame_check_sequence** parameter is determined as follows:

f) If the frame_check_sequence parameter received in the data request is either unspecified or still carries a valid value, then that value is used. Otherwise;
g) The value used is either derived from the received FCS information by modification to take account of the conditions that have caused it to become invalid, or the unspecified value is used.

NOTE—The original FCS associated with a frame is invalidated if there are changes to any fields of the frame, if fields are added or removed, or if bit ordering or other aspects of the frame encoding have changed. An invalid FCS is signalled in the E-ISS by an unspecified value in the frame_check_sequence parameter of the data request primitive. This signals the need for the FCS to be regenerated according to the normal procedures for the transmitting MAC. The options for regenerating the FCS under these circumstances are discussed in Annex G of IEEE Std 802.1D, 1998 Edition.

6.5 Support of the Internal Sublayer Service by IEEE Std 802.3 (CSMA/CD)

In addition to the provisions of 6.5.1 of IEEE Std 802.1D, 1998 Edition, on receipt of an M_UNITDATA.request primitive that represents a tagged frame, the implementation is permitted to adopt either of the following approaches with regard to the operation of Transmit Data Encapsulation for frames whose length would, using the procedure as described, be less than 68 octets:

a) Use the procedure as described in 6.5.1 of IEEE Std 802.1D, 1998 Edition. This can result in tagged frames of less than 68 octets (but at least 64 octets) being transmitted; or
b) Include additional octets before the FCS field in order for the transmitted frame length for such frames to be 68 octets. This results in a minimum tagged frame length of 68 octets.

When a tagged frame of less than 68 octets in length is received on a CSMA/CD LAN segment, and is forwarded as an untagged frame, the provisions of 6.5.1 of IEEE Std 802.1D, 1998 Edition, result in additional octets being included before the FCS field on transmission in order that the transmitted frame length meets the minimum frame size requirements of 3.2.7 in IEEE Std 802.3, 1998 Edition.

7. Principles of network operation

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

a) the operation of individual Bridges (Clause 8);
b) their participation in the Spanning Tree (Clause 8 of IEEE Std 802.1D, 1998 Edition), Rapid Spanning Tree (Clause 17 of IEEE Std 802.1w-2001), or Multiple Spanning Tree Protocol (Clause 13);
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

The operation of a Virtual Bridged Local Area Network, the Bridges, and the LANs, that compose that network comprises:

a) A physical topology comprising LANs, Bridges, and Bridge Ports. Each Bridge Port attaches a LAN to a Bridge and is capable of providing bidirectional connectivity for MAC user data frames. Each LAN is connected to every other LAN by a Bridge and zero or more other LANs and Bridges.
b) Calculation of one or more active topologies, each a loop-free subset of the physical topology.
c) Rules for the classification of MAC user data frames that allow each Bridge to allocate, directly or indirectly, each frame to one and only one active topology.
d) Management control of the connectivity provided for differently classified data frames by the selected active topology.
e) Implicit or explicit configuration of end station location information, identifying LANs with attached end stations that need to receive user data frames with a given destination address.
f) Communication of end station location information to allow Bridges to restrict user data frames to LANs in the path provided to their destination(s) by the chosen active topology.

These elements and their interrelationships are illustrated in Figure 7-1.

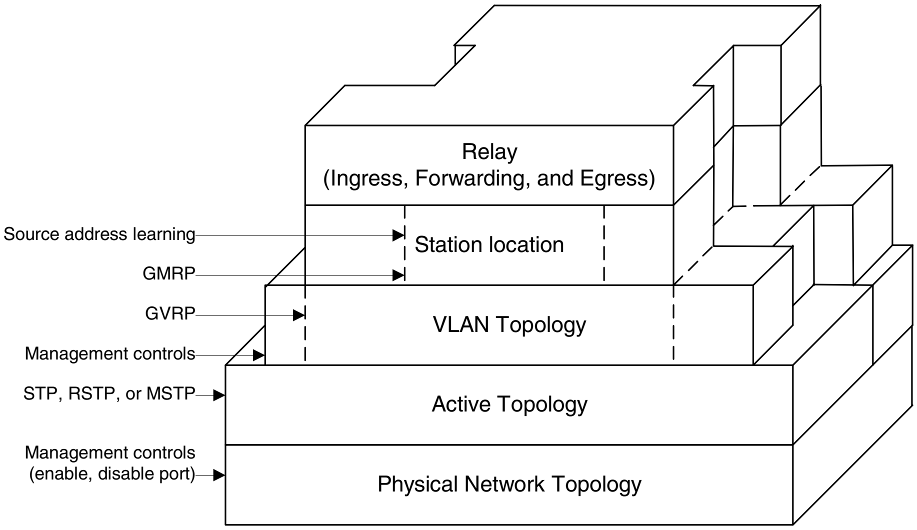


Figure 7-1—VLAN Bridging overview

NOTE 1—The physical and active topologies can be represented as bi-partite graphs. Bridges and LANs are nodes in these graphs (including LANs that are point-to-point links) and the Bridge Ports are edges.

NOTE 2—This standard applies the notion of a physical topology to media access control methods, like wireless, where there is no tangible physical connection between a Bridge and an attached LAN. A Bridge Port models this association just as for wired connectivity.

NOTE 3—Each of the active topologies [(b) above] does not necessarily span the entire network, but does span all those Bridges and LANS between which data frame connectivity is desired for frames allocated to that active topology.

NOTE 4—The destination addressing information associated with a MAC user data frame includes the VLAN classification of the frame [see (d) above].

NOTE 5—Implicit configuration [(d) above], or recognition, of end station location includes observation of the source address of the user data frame transmitted by that station. The user data frame itself communicates that location information [(e) above] along the portion of the chosen active topology leading to the frame's destination(s).

7.2 Use of VLANs

Virtual Local Area Networks (VLANs) and their VLAN Identifiers (VIDs) provide a convenient and consistent network wide reference for Bridges to:

a) identify rules for the classification of user data frames into VLANs;
b) effectively extend the source and destination MAC addresses, by treating frames and addressing information for different VLANs independently.
c) identify and select from different active topologies;
d) identify the configuration parameters that partition or restrict access from one part of the network to another.

Taken together these capabilities allow VLAN aware Bridges to emulate a number of separately manageable, or virtual, Bridged Local Area Networks. A LAN segment that has been selected by network management to receive frames assigned to a given VLAN is said to form part of, belong to, or be a member of the VLAN. Similarly, end stations that are attached to those LAN segments and that can receive frames assigned to the VLAN are said to be attached to that VLAN.

NOTE—Separate control over the transmission and over the reception of frames to and from LAN segments and end stations is possible. To avoid ambiguity VLAN membership is defined by reception.

Inclusion of the VID in VLAN-tagged frames guards against connectivity loops arising from differing classifications by different Bridges, and permits enhanced classification rules to be used by some Bridges while others simply forward the previously classified frames.

The VLAN tag allows frames to carry priority information, even if the frame has not been classified as belonging to a particular VLAN.

7.3 VLAN Topology

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

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

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

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

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

7.4 Locating end stations

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

NOTE 1—Individually Bridges do not determine the precise location of end stations, but merely determine which of their Bridge Ports need to forward frames towards the destination(s). For the system of Bridges this is sufficient to restrict frames to the paths necessary to reach the destination segments.

The multicast registration protocol (GMRP), specified in Clause 10 of IEEE Std 802.1D, 1998 Edition, allows end stations to advertise their presence and their desire to join (or leave) a multicast group in the context of a VLAN. The protocol communicates this information to other Bridges, using the VLAN and its active topology.

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

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

The Filtering Database architecture defined in this standard recognizes that

a) For some configurations, it is necessary to allow address information learned in one VLAN to be shared among a number of VLAN's. This is known as *Shared VLAN Learning* (3.26);
b) For some configurations, it is desirable to ensure that address information learned in one VLAN is not shared with other VLANs. This is known as *Independent VLAN Learning* (3.8);
c) For some configurations, it is immaterial as to whether learned information is shared between VLANs or not.

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

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

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

Within a given VLAN-Bridged LAN, there may be a combination of configuration requirements, so that individual VLAN Bridges may be called upon to share learned information, or not share it, according to the requirements of particular VLANs or groups of VLANs. The Filtering Database structure that is defined in this standard allows both Shared and Independent VLAN Learning to be implemented within the same VLAN Bridge; i.e., allows learned information to be shared between those VLANs for which Shared VLAN Learning is necessary, while also allowing learned information not to be shared between those VLANs for which Independent VLAN Learning is necessary. The precise requirements for each VLAN with respect to sharing or independence of learned information (if any) are made known to VLAN Bridges by means of a set of *VLAN Learning Constraints* (8.10.7.2) and fixed allocations of VLANs to filtering databases (8.10.7.1), which may be configured into the Bridges by means of management operations. By analyzing the set of learning constraints and fixed allocations for the VLANs that are currently active, the Bridge can determine

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

The manner in which this mapping of VLANs onto Filtering Databases is achieved is defined in 8.10.7; the result is that each VLAN is associated with exactly one Filtering Database.

The most general application of the Filtering Database specification in this standard is a Bridge that can support M independent Filtering Databases, and can map N VLANs onto each Filtering Database. Such a Bridge is known as an SVL/IVL Bridge (3.28).

The conformance requirements in this standard (5.1, 5.2) recognize that VLAN Bridges will be implemented with differing capabilities in order to meet a wide range of application needs, and that the full generality of the SVL/IVL approach is not always either necessary or desirable, as observed in the discussion in Annex B. In a given conformant implementation, there may be restrictions placed upon the number of Filtering Databases that can be supported, and/or the number of VLANs that can be mapped onto each Filtering Database. The full spectrum of conformant Filtering Database implementations is therefore as follows:

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

7.5 Ingress, Forwarding, and Egress Rules

The relay function provided by the Forwarding Process (8.6) of each individual Bridge controls:

a) Classification of each received frame as belonging to one and only one VLAN, and discard or acceptance of the frame for further processing on the basis of that classification and the received frame format, which can be one of three possible types:

   1) Untagged, and not explicitly identifying the frame as belonging to a particular VLAN;

   2) Priority-tagged, i.e., including a tag header conveying explicit user priority information but not identifying the frames as belonging to a specific VLAN;

   3) VLAN-tagged, i.e., explicitly identifying the frames as belonging to a particular VLAN.

   This aspect of relay implements the *ingress* rules.

b) Implementation of the decisions governing where each frame is to be forwarded as determined by the current extent of the VLAN topology (7.3), station location information (7.4), and the additional management controls specified in Clause 8. This aspect of relay implements the *forwarding* rules.

c) Determination of the appropriate frame format type, VLAN-tagged or untagged, for transmission through the selected Bridge Ports. This aspect of relay implements the *egress* rules.

The structuring of the relay functionality into the implementation of ingress, forwarding, and egress rules constitutes a generic approach to the provision of VLAN functionality. All VLAN-aware Bridges can correctly forward received frames that are already VLAN-tagged. These are classified as belonging to the VLAN identified by the VID in the tag header. All VLAN-aware Bridges can also classify untagged and priority-tagged frames received on any given port as belonging to a specified VLAN. In addition to this default Port-based ingress classification, this standard specifies an optional Port-and-Protocol-based classification.

The classification of untagged and priority-tagged frames, and the addition or removal of tag headers, is part of the relay functionality of a VLAN-aware Bridge and is only performed on frames that can be forwarded through other Bridge Ports. To facilitate coexistence with Bridges conforming to IEEE Std 802.1D, 1998 Edition, frames that carry control information to determine the active topology and current extent of each VLAN, i.e., spanning tree and GVRP BPDUs, are not forwarded. Permanently configured static entries in the filtering database (8.2, 8.3, and 8.13) ensure that such frames are discarded by the Forwarding Process (8.6).

NOTE 1—GARP PDUs destined for any GARP application are forwarded or filtered depending upon whether the application concerned is supported by the bridge, as specified in 8.13.

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

NOTE 1—Additional ingress classification methods may be the subject of future standardization.

NOTE 2—This classification of untagged and priority-tagged frames is part of the functionality of the MAC relay entity (Figure 8-3, Figure 8-4), and is therefore only of significance for received frames that are potentially to be forwarded through other Ports of the Bridge (see Spanning Tree).

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

d) All untagged; or
e) All VLAN tagged with the same VID.

NOTE 4—In other words, a Bridge can transmit untagged frames for some VLANs and VLAN-tagged frames for other VLANs on a given link, but cannot transmit using both formats for the same VLAN. The single format rule only applies to the frame transmission behavior of individual VLAN-aware stations; i.e., it does not express a requirement for a Bridge to police the behavior of the other stations attached to a segment and enforce a single format.

8. Principles of bridge operation

This clause establishes the principles of operation of a VLAN-aware Bridge, by reference to a model of that operation, as follows:

a) Explains the principal elements of Bridge operation and lists the functions that support these.
b) Establishes an architectural model for a Bridge that governs the provision of these functions.
c) Provides a model of the operation of a Bridge in terms of the processes and entities that support the functions.
d) Details the addressing requirements in a Bridged LAN and specifies the addressing of entities in a Bridge.

The provisions of this clause replace the provisions of Clause 7 of IEEE Std 802.1D, 1998 Edition in a VLAN-aware Bridge.

8.1 Bridge operation

The principal elements of Bridge operation are

a) Relay and filtering of frames.
b) Maintenance of the information required to make frame filtering and relaying decisions.
c) Management of the above.

8.1.1 Relay

A MAC Bridge relays individual MAC user data frames between the separate MACs of the Bridged LANs connected to its Ports. The order of frames shall be preserved as defined in 8.6.5.

The functions that support the relaying of frames and maintain the Quality of Service supported by the Bridge are

a) Frame reception.
b) Discard on received frame in error (6.3.2).
c) Frame discard if the frame_type is not user_data_frame, or if its mac_action parameter is not request_with_no_response (8.5, 6.4).
d) Regeneration of user priority, if required (6.4).
e) Application of VLAN ingress rules to classify each received frame to a particular VLAN (8.6.1).
f) Frame discard if the frame is not classified or the receiving port does not accept frame from that VLAN.
g) Frame discard following the application of filtering information.
h) Frame discard on transmittable service data unit size exceeded (6.3.8).
i) Forwarding of received frames to other Bridge Ports.
j) Frame discard following the application of VLAN egress rules if the service data unit cannot be mapped correctly or the receiving port does not transmit frames to the particular VLAN to which the frame belongs.
k) Selection of traffic class, following the application of filtering information.
l) Queuing of frames by traffic class.
m) Frame discard to ensure that a maximum bridge transit delay is not exceeded (6.3.6).
n) Selection of queued frames for transmission.
o) Selection of outbound access priority (6.3.9).
p) Mapping of service data units and recalculation of Frame Check Sequence, if required (8.6.9, 6.3.7).
q) Frame transmission.

8.1.2 Filtering and relaying information

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

a) Prevention of frame duplication: to ensure that frame duplication does not occur, by maintaining a loop-free logical topology;
b) Traffic segregation: to separate communication by different sets of users or groups of users of the Bridged LAN;
c) Traffic reduction: to confine frames to those parts of the Bridged LAN that lie on the path(s) between their source and destination(s);
d) Traffic expediting: to classify frames in order to expedite time critical traffic;
e) Conversion of frame formats: to present frames in a format (tagged or untagged) that is appropriate for the destination LAN and its attached stations.

8.1.2.1 Prevention of frame duplication

A Bridge filters frames, i.e., does not relay frames received by a Bridge Port to other Ports on that Bridge, in order to prevent the duplication of frames (6.3.4). The functions that support the use and maintenance of information for this purpose are

a) Configuration and calculation of Bridged LAN topology.
b) In MST Bridges, explicit configuration of the relationship between VIDs and spanning trees (8.11).

8.1.2.2 Traffic segregation

A Bridge can filter frames to confine them to the LAN segments that belong to the VLAN to which they are assigned, and thus define the VLAN's maximum extent (7.3). The functions that support the use and maintenance of information for this purpose are

a) Explicit configuration of the Ingress Rules associated with each Port (8.4, 8.6.1 8.9):
    1) Configuration of <commentRef>PVID for each Port, to associate a VID with untagged and priority-tagged frames;
    2) In Bridges implementing Port-and-Protocol based classification, the configuration of a VID for untagged and priority-tagged frames for each protocol;
    3) Enabling or disabling the application of Static VLAN Registration Entries to received frames through configuration of the Enable Ingress Filtering parameter;
b) Explicit configuration of the Egress Rules associated with each Port (8.6.4):
    1) Configuration of Static VLAN Registration Entries.

A Bridge can filter frames to partially partition a Virtual Bridged Local Area Network. Frames assigned to any given VLAN and addressed to specific end stations or groups of end stations can be excluded from relay to certain Bridge Ports. The functions that support the use and maintenance of information for this purpose are:

c) Permanent configuration of Reserved Addresses (Table 8-10);
d) Configuration of Static Filtering Entries (8.10.1) and Group Registration Entries (8.10.4).

NOTE—The use of VLANs is generally less error prone and is preferred to filtering using destination addresses if a Bridged Local Area Network is to be partitioned for reasons of scale, efficiency, management, or security. Destination address filtering is the only mechanism available to Bridges that are not VLAN aware.

8.1.2.3 Traffic reduction

A Bridge can filter frames to confine them to LAN segments that either have end stations attached to their assigned VLAN or that connect those LAN segments, and thus define the current practical extent of the

VLAN (7.4). LANs not attaching to or forming part of the path between the source and destination(s) of any given communication do not have to support the transmission of related frames, potentially improving the quality of the MAC service for other communications. The functions that support the use and maintenance of information for this purpose are:

a) Automatic learning of dynamic filtering information for unicast destination addresses through observation of source addresses of frames, together with the ageing out or flushing of that information to support the movement of end stations and changes in active topology;
b) Ageing out or flushing of dynamic filtering information that has been learned;
c) Automatic inclusion and removal of Bridge Ports in the VLAN, through configuration of Dynamic VLAN Registration Entries by means of GVRP (8.10.5 and 11.2);
d) Explicit configuration of management controls associated with the operation of GVRP by means of Static VLAN Registration Entries (8.10.2 and 11.2);
e) Automatic configuration of Group Registration Entries by means of GMRP exchanges;
f) Explicit configuration of the management controls associated with the operation of GMRP by means of Group Registration Entries.

8.1.2.4 Traffic expediting

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

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

8.1.2.5 Conversion of frame formats

A Bridge adds and removes tag headers (9.3) from frames, and performs the associated frame translations that may be required, in accordance with the egress rules (8.6.4). The function that supports the use and maintenance of information for this purpose is

a) Explicit configuration of tagging requirements on egress for each Port (8.10.2 and 8.10.9).

8.1.3 Bridge management

The functions that support Bridge Management control and monitor the provision of the above functions. They are specified in Clause 12.

8.2 Bridge architecture

8.2.1 Architectural model of a Bridge

Figure 8-1 gives an example of the physical topology of a Bridged LAN. The component LANs are interconnected by means of MAC Bridges; each Port of a MAC Bridge connects to a single LAN. Figure 7-1 illustrates a Bridge with two Ports, and Figure 6-1 illustrates the architecture of such a Bridge.

A Bridge is modeled as consisting of

a) A MAC Relay Entity that interconnects the Bridge's Ports;
b) At least two Ports;
c) Higher layer entities, including at least a Spanning Tree Protocol Entity[9].

[9] The Spanning Tree Protocol Entity was formerly known as the Bridge Protocol Entity.

8.2.2 MAC Relay Entity

The MAC Relay Entity handles the MAC independent functions of relaying frames between Bridge Ports, filtering frames, and learning filtering information. It uses the Enhanced Internal Sublayer Service provided by the separate MAC Entities for each Port. (The Enhanced Internal Sublayer Service and its support are described in 6.4 and 6.5.) Frames are relayed between Ports attached to different LANs.

8.2.3 Ports

Each Bridge Port transmits and receives frames to and from the LAN to which it is attached. An individual MAC Entity permanently associated with the Port provides the Enhanced Internal Sublayer Service used for frame transmission and reception. The MAC Entity handles all the MAC method dependent functions (MAC protocol and procedures) as specified in the relevant standard for that IEEE 802 LAN MAC technology.

8.2.4 Higher Layer Entities

The Spanning Tree Protocol Entity handles calculation and configuration of Bridged LAN topology.

The Spanning Tree Protocol Entity and other higher layer protocol users, such as Bridge Management (8.1.3) and GARP application entities including GARP Participants (Clause 12 of IEEE Std 802.1D, 1998 Edition), make use of Logical Link Control procedures. These procedures are provided separately for each Port, and use the MAC Service provided by the individual MAC Entities.

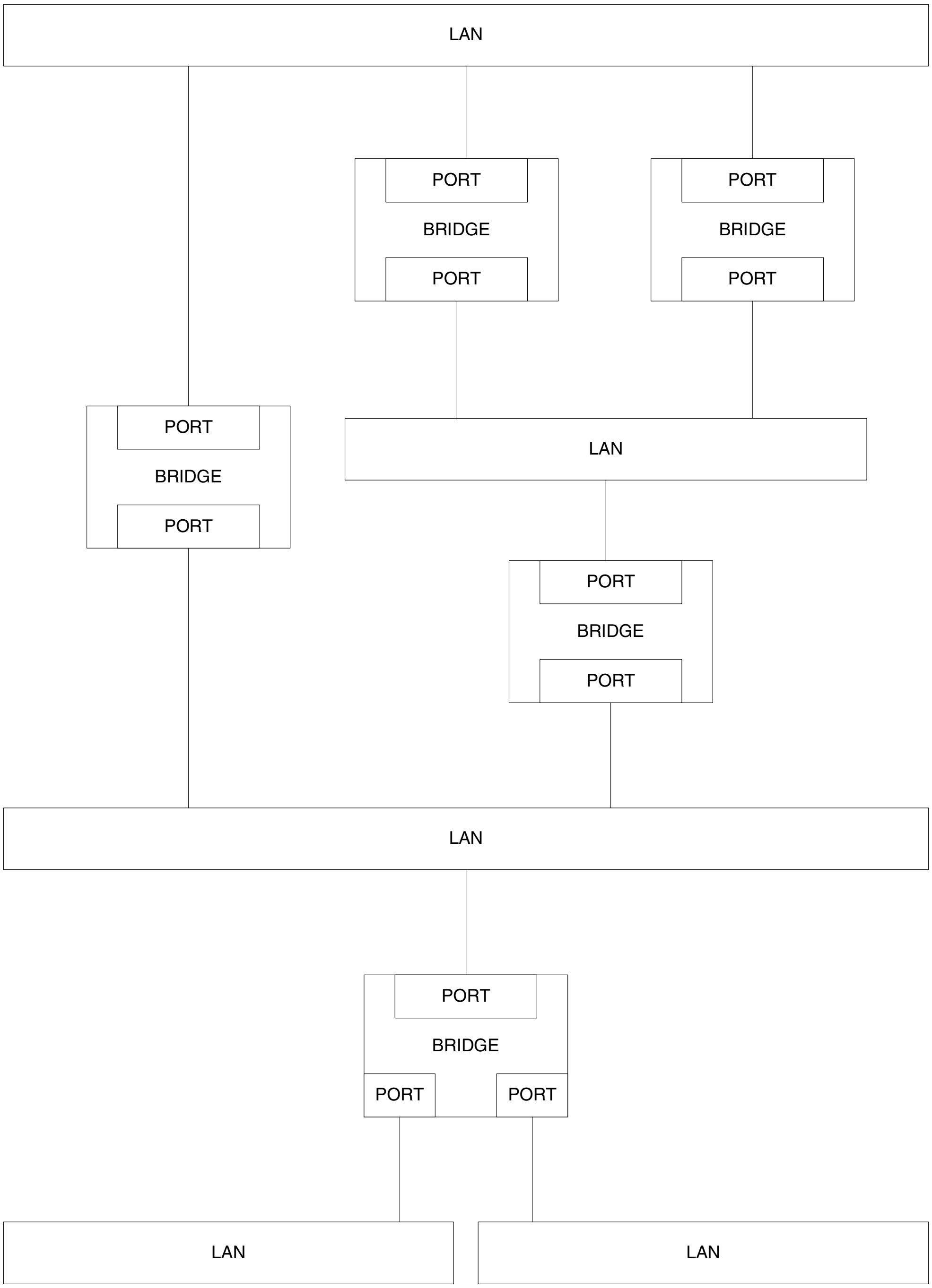


Figure 8-1—Example of a Bridged LAN

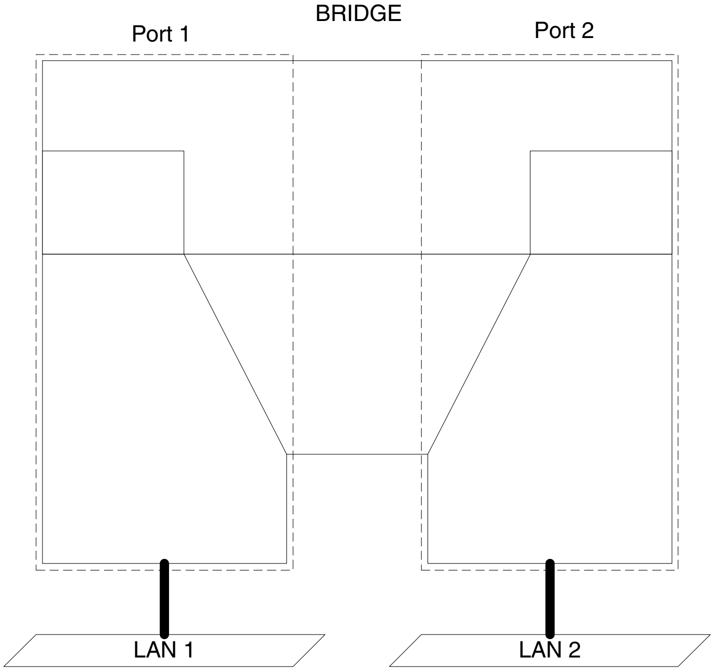


Figure 8-2—Bridge ports

8.3 Model of operation

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

Subclauses 8.5 and 8.7 specify the MAC Relay Entity's use of the Enhanced Internal Sublayer Service. State information associated with each Port governs the Port's participation in the Bridged LAN. (Port States are specified in detail in 17.5 of IEEE Std 802.1D, 1998 Edition.).

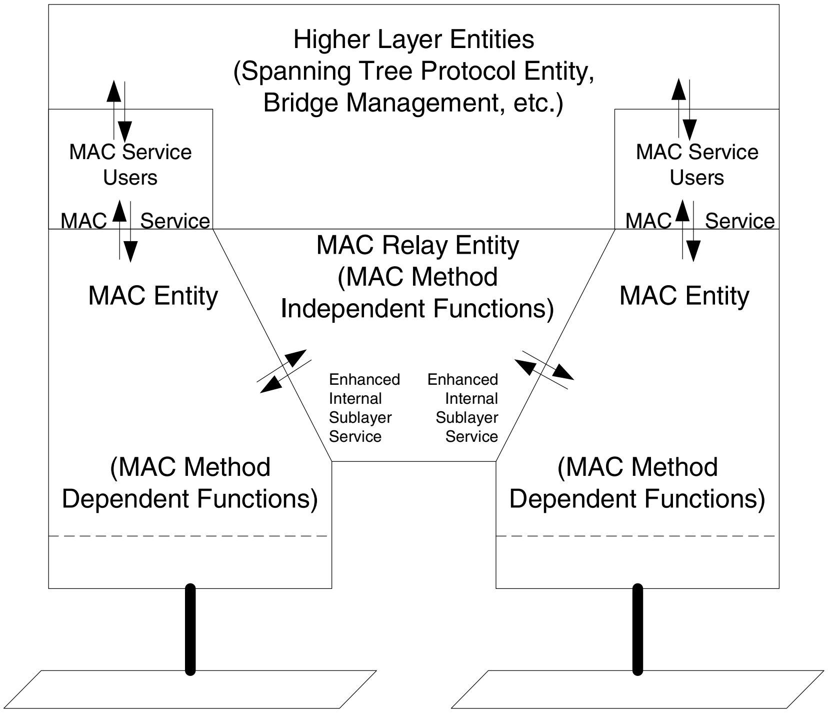


Figure 8-3—VLAN Bridge architecture

Frames are accepted for transmission and delivered on reception to and from processes and entities that model the operation of the MAC Relay Entity in a Bridge. These are

a) The Forwarding Process (8.6), that:
   1) Classifies received frames according to their VLAN membership (8.6.1), can filter frames based on the absence of a VID in the received frame (8.4.3), and can filter frames based on the frame’s VLAN identifier (8.4.5), on the basis of the ingress rules (8.6.1) for the receiving Port;
   2) Forwards received frames that are to be relayed to other Bridge Ports, filtering frames on the basis of information contained in the Filtering Database (8.10), the state of the Bridge Ports (8.4), and, in MST Bridges, the mapping of VIDs to spanning trees supported by the Bridge (8.6.2, 8.11);
   3) Determines, for a given VLAN, through which Ports frames may be transmitted, and in what format, on the basis of the egress rules (8.6.4) for the transmitting Port.
b) The Learning Process (8.8), which, by observing the source addresses and VIDs of frames classified by the Forwarding Process according to the ingress rules (8.6.1), updates the Filtering Database (8.10), conditionally on the Port state (8.4);
c) The Filtering Database (8.10), which holds filtering information and supports queries by the Forwarding Process as to whether frames with given values of the destination MAC Address field and VID should be forwarded to a given Port.

Each Bridge Port also functions as an end station providing the MAC Service to LLC, which in turn supports operation of the Spanning Tree Protocol Entity (8.12) and of other possible users of LLC, such as protocols providing Bridge Management (8.13).

Each Bridge Port shall support the operation of LLC Type 1 procedures in order to support the operation of the Spanning Tree Protocol Entity. Bridge Ports may support other types of LLC procedures, which may be used by other protocols.

Figure 8-4 illustrates a single instance of frame relay between the Ports of a Bridge with two Ports.

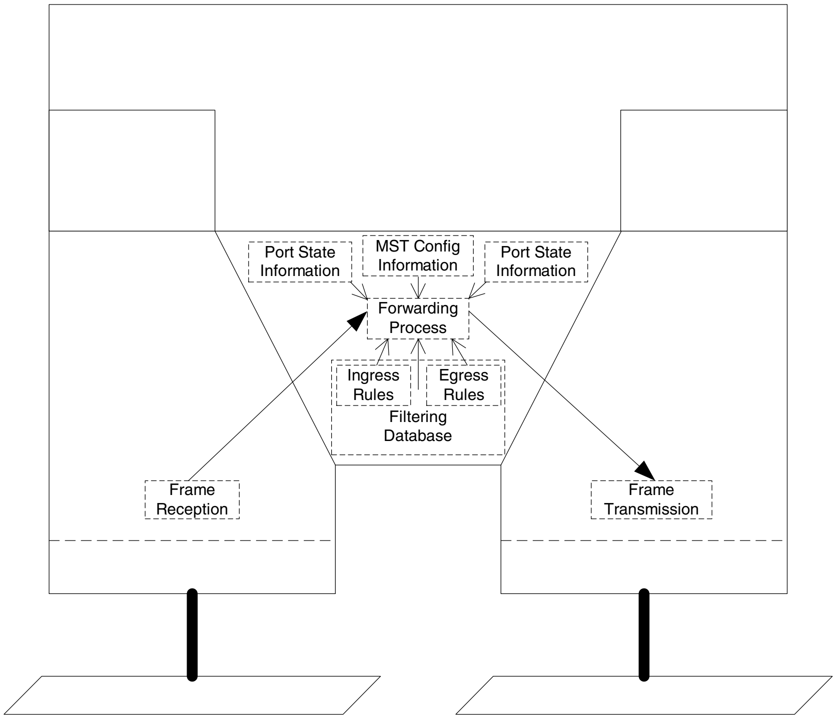


Figure 8-4—Relaying MAC frames

Figure 8-5 illustrates the inclusion of information carried by a single frame, received on one of the Ports of a Bridge with two Ports, in the Filtering Database.

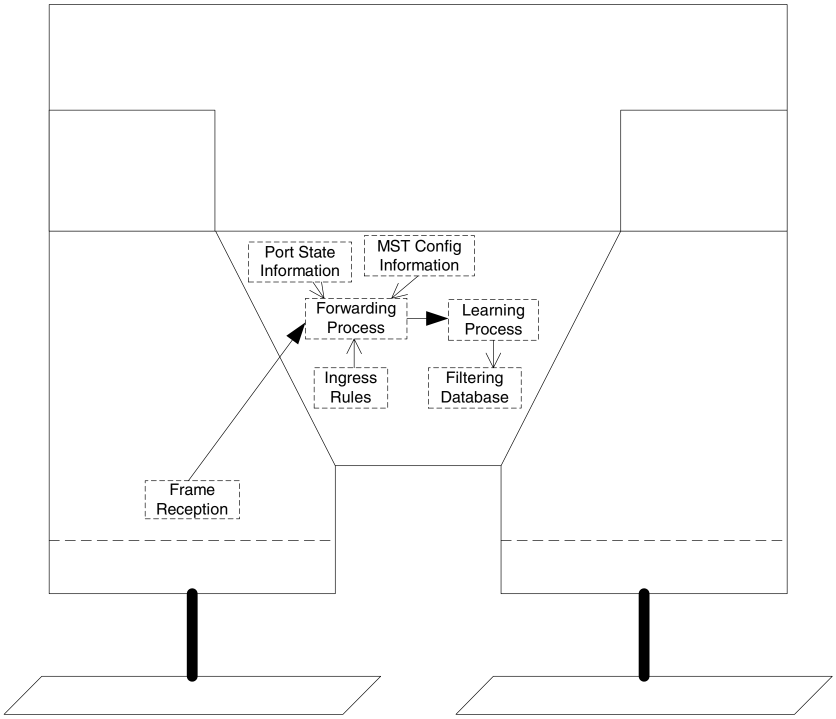


Figure 8-5—Observation of network traffic

Figure 8-6 illustrates the reception and transmission of Bridge Protocol Data Units by the Spanning Tree Protocol Entity.

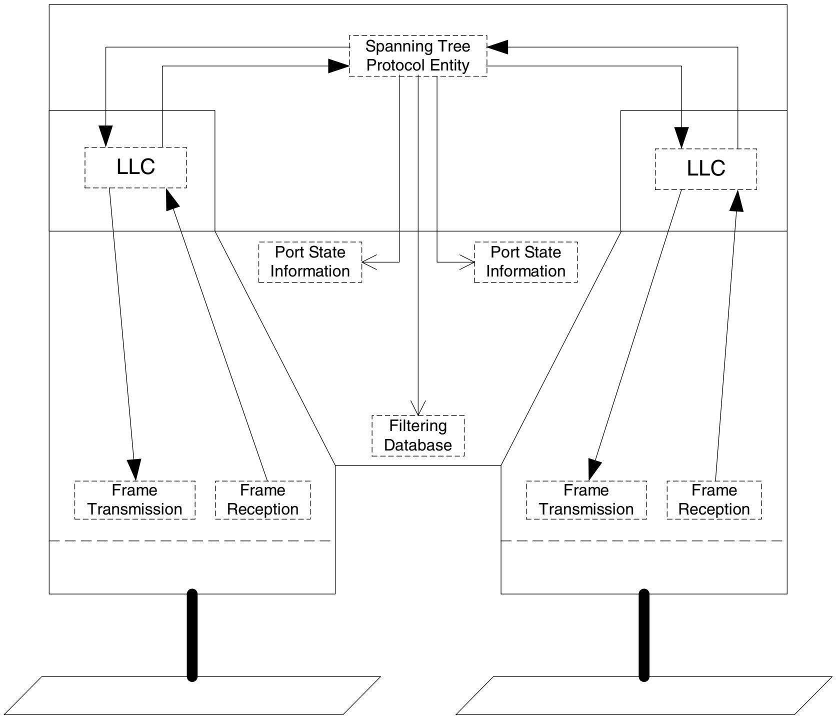


Figure 8-6—Operation of Spanning Tree protocol

Figure 8-7 illustrates the reception and transmission of GARP Protocol Data Units by a GARP Protocol Entity (8.12).

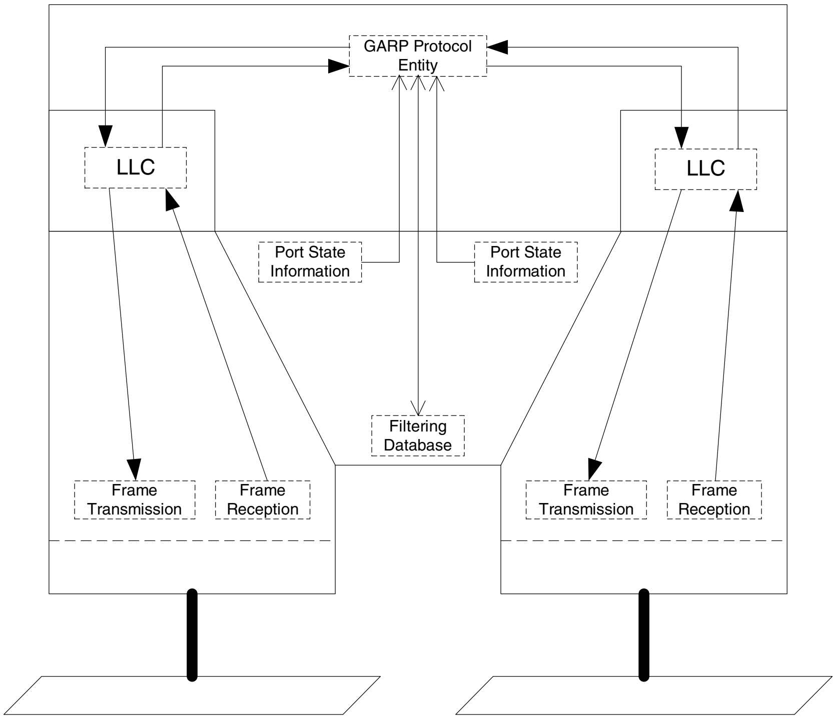


Figure 8-7—Operation of the GARP protocol

8.4 Port states and parameters

8.4.1 Forwarding states

The active topology of a Bridged LAN at any time is the set of communication paths formed by interconnecting the LANs and Bridges by the forwarding Ports. The function of the distributed Spanning Tree (STP, Clause 8 of IEEE Std 802.1D, 1998 Edition), and Rapid Spanning Tree (RSTP, Clause 17 of IEEE Std 802.1D, 1998 Edition) protocols used by SST Bridges, is to construct an active topology that is simply connected relative to communication between any given pair of MAC Addresses used to address end stations on the LANs, irrespective of the VLAN classification of frames used in that communication. The Multiple Spanning Tree Protocol (MSTP, Clause 13) used by MST Bridges constructs one or more active topologies. Each active topology is simply and fully connected relative to communication between any given pair of end stations using frames consistently classified by Bridges as belonging to a given VLAN or VLANs.

STP and RSTP construct a single spanning tree, the Common Spanning Tree (CST). MSTP constructs multiple spanning trees, the Common and Internal Spanning Tree (CIST) and additional Multiple Spanning Tree Instances (MSTIs). An MST Bridge allocates all frames classified as belonging to a given VLAN to the CIST or to one of the MSTIs.

State information associated with each Bridge Port for each spanning tree governs whether or not the Port participates in relaying MAC frames allocated to that tree. A Port can be disabled by management, in which case it plays no part in the operation of the Bridged LAN; a Port that is not disabled can be dynamically excluded from participation in frame relaying by operation of the Spanning Tree algorithm. If neither of these applies to a Port, it is described as forwarding for that spanning tree.

Figure 8-6 illustrates the operation of the Spanning Tree Protocol Entity, which operates the Spanning Tree algorithm and its related protocols, and its modification of Port state information as part of determining the active topology of the Bridged LAN. The Port states associated with the determination of the active topology are specified in detail in 17.5 of IEEE Std 802.1D, 1998 Edition.

Figure 8-4 illustrates the Forwarding Process's use of Port state information: first, for a Port receiving a frame, in order to determine whether the received frame is to be relayed through any other Ports; and second, for another Port in order to determine whether the relayed frame is to be forwarded through that particular Port.

8.4.2 Learning states

The incorporation of end station location information in the Filtering Database by the Learning Process also depends on the active topology. If information associated with frames received on a Port and allocated to a given spanning tree is to be incorporated in the Filtering Database by the Learning Process, then the Port is described as being in a learning state for that spanning tree; otherwise, it is in a non-learning state. Figure 8-5 illustrates the use of the Port state information for a Port receiving a frame, by the Learning Process, in order to determine whether the station location information is to be incorporated in the Filtering Database.

8.4.3 Acceptable Frame Types

Associated with each Port of a VLAN Bridge is an Acceptable Frame Types parameter that controls the reception of VLAN-tagged and non VLAN-tagged frames on that Port. Valid values for this parameter are

a) *Admit Only VLAN-tagged frames*;
b) *Admit All Frames.*

If this parameter is set to Admit Only VLAN-tagged frames, any frames received on that Port that carry no VID (i.e., untagged frames or priority-tagged frames) are discarded by the ingress rule checking function of the Forwarding Process (8.6.1).

Frames that are not discarded as a result of this parameter value are classified and processed according to the ingress rules (8.6.1) that apply to that Port.

Each Port of the Bridge shall support at least one of these values, and may support both. Where both values are supported,

c) The implementation shall support the ability to configure the value of the parameter by means of the management operations defined in Clause 12; and

d) The default value of the parameter shall be *Admit All Frames*.

8.4.4 Port VLAN identifier and VID Set

A VLAN Bridge supports Port-based VLAN classification, and may, in addition, support Port-and-Protocol-based VLAN classification.

In Port-based VLAN classification within a Bridge, the VID associated with an untagged or priority-tagged frame (i.e., a frame with no tag header, or a frame with a tag header that carries the null VLAN ID) is determined, based on the Port of arrival of the frame into the Bridge, as described in 8.6.1 and 8.9. This classification mechanism requires the association of a specific Port VLAN Identifier, or PVID, with each of the Bridge's Ports. In this case, the PVID for a given Port provides the VID for untagged and priority-tagged frames received through that Port.

In addition to the PVID, for bridges that implement Port-and-Protocol-based VLAN classification, the VID associated with an Untagged or Priority-tagged Frame is determined based on the Port of arrival of the frame into the bridge and on the protocol identifier of the frame, as described in 8.6.1 and 8.9. This classification mechanism requires the association of multiple VIDs with each of the Ports of the Bridge: this is known as the "VID Set" for that port. Each VID of a Port of a Bridge that supports Port-and-Protocol-based VLAN classification is also associated with a Protocol Group Identifier. A Protocol Group Identifier is not relevant in a Bridge that supports only Port-based VLAN classification. The contents of the VID Set for each port may be configured by management. The VID Set is in addition to the PVID value described above.

The PVID and VID Set for each Port shall contain valid VID values (Table 9-2).

NOTE 1—This rule ensures that the process of ingress classification of frames always associates a valid VID with each received frame. As a consequence, a VLAN-aware Bridge can never transmit priority-tagged frames; all frames transmitted are either untagged or carry a valid VID in their tag header.

The PVID and VID Set values may be configured by management, if management operations are supported by the implementation. If no PVID value has been explicitly configured for a Port, the PVID shall assume the value of the default PVID defined in Table 9-2 and the VID Set shall be empty.

NOTE 2—If a Bridge is configured so that, for any Port, all the members of the VID Set of that port assume the same VID, then it is impossible to tell from the frame relay behavior of the bridge whether the Bridge supports Port-based or Port-and-Protocol-based VLAN classification. In particular, the default frame relay behavior (the frame relay behavior before any administrative actions on the Bridge) of a Bridge that supports Port-and-Protocol-based VLAN classification is the same as the default frame relay behavior of a Bridge that supports only Port-based VLAN classification.

8.4.5 Enable Ingress Filtering

An Enable Ingress Filtering parameter is associated with each Port. If the Enable Ingress Filtering parameter for a given Port is set, the ingress rule checking function of the Forwarding Process (8.6.1) shall discard any frame received on that Port whose VLAN classification does not include that Port in its Member set (8.10.9). If the parameter is reset for that Port, the ingress rules shall not discard frames received on that Port on the basis of their VLAN classification.

The default value for this parameter is reset, i.e., Disable Ingress Filtering, for all Ports. The value of this parameter may be configured by means of the management operations defined in Clause 12, if management operations are supported by the implementation. If the implementation supports the ability to enable Ingress Filtering on any Port, then it shall also support the ability to disable Ingress Filtering on those Ports.

8.5 Frame reception

The individual MAC Entity associated with each Bridge Port examines all frames received on the LAN to which it is attached.

All error-free received frames give rise to EM_UNITDATA indication primitives, which shall be handled as follows.

NOTE 1—A frame that is in error, as defined by the relevant MAC specification, is discarded by the MAC Entity without giving rise to any EM_UNITDATA indication: see 7.2 and 6.4.

Frames with EM_UNITDATA.indication primitive frame_type and mac_action parameter values of user_data_frame and request_with_no_response, respectively (7.2 and 6.4), shall be submitted to the Forwarding Process (8.6).

Frames with other values of frame_type and mac_action parameters, (e.g., request_with_response and response frames), shall not be submitted to the Forwarding Process (8.6).

Frames with a frame_type of user_data_frame and addressed to the Bridge Port as an end station shall be submitted to the MAC Service user. Such frames carry either the individual MAC Address of the Port or a group address associated with the Port (8.14) in the destination address field. Frames submitted to the MAC Service user can also be submitted to the Forwarding Process (8.6), as specified above.

Frames addressed to a Bridge Port as an end station, and relayed to that Bridge Port from other Bridge Ports in the same Bridge by the Forwarding Process, shall also be submitted to the MAC Service user.

NOTE 2—The consequence of the above is that frames “relayed to that Bridge Port” are both submitted to that Port’s MAC Service user and transmitted on the LAN to which that Port is attached (see 8.14.7).

No other frames shall be submitted to the MAC Service user.

8.5.1 Regenerating user priority

The user_priority of received frames is regenerated using priority information contained in the frame and the User Priority Regeneration Table for the reception Port. For each reception Port, the User Priority Regeneration Table has eight entries, corresponding to the eight possible values of user_priority (0 through 7). Each entry specifies, for the given value of received user_priority, the corresponding Regenerated user_priority value.

NOTE 1—IEEE 802 LAN technologies signal a maximum of eight user_priority values. H.2 of IEEE Std 802.1D, 1998 Edition contains further explanation of the use of user_priority values and how they map to traffic classes.

NOTE 2—User priority is regenerated only for frames that do not carry tag headers; see 7.1.2.1.

Table 8-1 defines the default values of Regenerated user_priority for the eight possible values of the user_priority parameter received in a data indication; these values shall be used as the initial values of the corresponding entries of the User Priority Regeneration Table for each Port.

Optionally, the ability to modify the values in the User Priority Regeneration Table by management means may be supported, as described in Clause 12. If this capability is provided, the value of the table entries may be independently settable for each reception Port and for each value of received user_priority, and the Bridge may have the capability to use the full range of values in the parameter ranges specified in Table 8-1.

NOTE 3—It is important to ensure that the regeneration and mapping of user priority within the Bridge is consistent with the end-to-end significance attached to that user priority in the Bridged LAN. Within a given Bridge, the values chosen for the User Priority Regeneration Table for a given Port should be consistent with the priority to be associated with traffic received through that Port across the rest of the Bridged LAN, and should generate appropriate access priority values for each transmission MAC method. The user priority value regenerated via the User Priority Regeneration Table on reception is used:

- Via the traffic class table (8.6.5) to determine the traffic class for a given outbound Port, and
- Via fixed, MAC method-specific mappings (8.6.7) to determine the access priority that will be used for a given outbound MAC method.

Table 8-1 shows the default values for the regeneration of user priority. Table 8-2 shows the default values for the traffic class table, for all possible numbers of supported traffic classes. Table 8-3 shows the fixed mappings from user priority to access priority that are required for different outbound MAC methods.

Table 8-1—User priority regeneration

Received user priority	Default regenerated user priority	Range
0	0	0–7
1	1	0–7
2	2	0–7
3	3	0–7
4	4	0–7
5	5	0-7
6	6	0–7
7	7	0–7

8.6 The Forwarding Process

Frames submitted to the Forwarding Process after being received at any given Bridge Port (8.5) shall be forwarded through the other Bridge Ports subject to the constituent functions of the Forwarding Process. These functions classify frames according to their VLAN membership (8.6.1), enforce topology restrictions (8.6.2), use Filtering Database information to filter frames (8.6.3), use the Egress Rules to discard frames (8.6.4), queue frames (8.6.5), select queued frames for transmission (8.6.6), map priorities (8.6.7), determine the format (tagged or untagged) of the transmitted frame (8.6.8), and recalculate FCS if required (8.6.9).

The Forwarding Process functions are described in 8.6.1–8.6.9 in terms of the action taken for a given frame received on a given Port (termed "the reception Port"). The frame can be forwarded for transmission on some Ports (termed "transmission Ports"), and is discarded without being transmitted at the other Ports.

NOTE—The model of operation of the Forwarding Process described in this standard is limited to the operation of the relay function of the MAC Bridge, and does not take into consideration what may occur in real implementations once frames are passed to the MAC for transmission. In some MAC implementations, and under some traffic conditions, a degree of indeterminacy may be introduced between the modeled description of the process of passing selected frames to the MAC for transmission and the actual sequence of frames as visible on the LAN medium itself. Examples can be found in the handling of access_priority in Token-Passing Bus MACs, or in the effect of different values for Token Holding Time in FDDI LANs. Such indeterminacy could result in apparent violation of the queuing/de-queueing and prioritizing rules described for the Forwarding Process, when observing traffic on the medium. As a consequence, in some implementations of this standard, it may prove to be impossible to test conformance to the standard simply by relating observed LAN traffic to the described model of the forwarding process; conformance tests would have to allow for the (permissible) behavior of the MAC implementations as well.

Figure 8-4 illustrates the operation of the Forwarding Process in a single instance of frame relay between the Ports of a Bridge with two Ports. Figure 8-8 illustrates the detailed operation of the Forwarding Process.

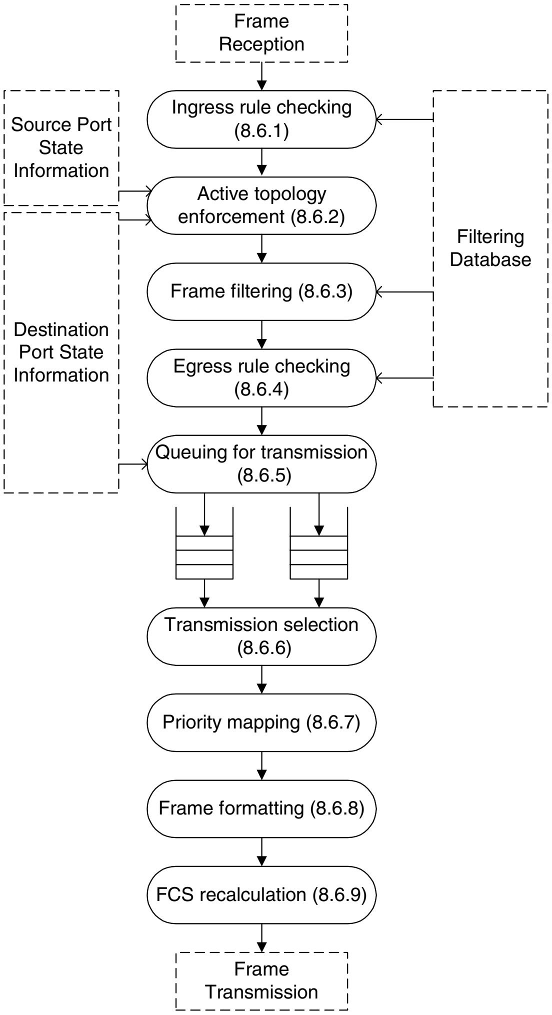

Frame Reception
Source Port State Information
Ingress rule checking (8.6.1)
Active topology enforcement (8.6.2)
Filtering Database
Destination Port State Information
Frame filtering (8.6.3)
Egress rule checking (8.6.4)
Queuing for transmission (8.6.5)
Transmission selection (8.6.6)
Priority mapping (8.6.7)
Frame formatting (8.6.8)
FCS recalculation (8.6.9)
Frame Transmission

Figure 8-8—Illustration of the detailed operation of the Forwarding Process

8.6.1 Ingress rule checking

Each received frame is assigned to a VLAN, and the corresponding VID is associated with the frame, as specified by the VLAN classification rules (8.9).

The frame shall be discarded and not forwarded to any other Port or submitted to the Learning Process if:

a) The VID is the value FFF, reserved in Table 9-2 for implementation use; or
b) The Acceptable Frame Types parameter (8.4.3) for the receiving Port is set to the value Admit Only VLAN-tagged frames, and the VID is the null VLAN ID; or
c) The Enable Ingress Filtering parameter (8.4.5) is set for the receiving Port, and that Port is not in the Member set (11.9) associated with the VID.

Otherwise the frame is permitted ingress to the VLAN, and shall be forwarded through the other Bridge Ports subject to the additional constituent functions of the Forwarding Process (8.6.2 through 8.6.9), and submitted to the Learning Process (8.8).

8.6.2 Active topology enforcement

Each received frame is allocated to a spanning tree by the Forwarding Process, using the VID. The Forwarding Process selects each Port as a potential transmission Port if, and only if

a) The Port on which the frame was received is in a forwarding state for that spanning tree (8.4 of IEEE Std 802.1D, 1998 Edition), and
b) The Port considered for transmission is in a forwarding state for that spanning tree, and
c) The Port considered for transmission is not the same as the Port on which the frame was received, and
d) The size of the mac_service_data_unit conveyed by the frame does not exceed the maximum size of mac_service_data_unit supported by the LAN to which the Port considered for transmission is attached.

For each Port not selected as a potential transmission Port the frame shall be discarded.

An SST Bridge allocates all frames to a single spanning tree, the Common Spanning Tree (CST).

An MST Bridge allocates all frames with a given VID to the CIST or to a Multiple Spanning Tree Instance (MSTI). The allocation can be controlled by configuration of the MST Configuration Table (8.11.1) maintained by the Forwarding Process, subject to constraints (if any) imposed by the allocation of VIDs to FIDs (8.10.7). VIDs allocated to different spanning trees shall also be allocated to different FIDs. VIDs allocated to a given spanning tree may share the same FID.

8.6.3 Frame filtering

Filtering decisions are taken by the Forwarding Process on the basis of

a) The destination MAC Address carried in a received frame;
b) The VID associated with the received frame;
c) The information contained in the Filtering Database for that MAC Address and VID;
d) The default Group filtering behavior for the potential transmission Port (8.10.6).

For each potential transmission Port selected as in 8.6.2, the frame shall be forwarded, or discarded (i.e., filtered), on the basis of this information, in accordance with the definition of the Filtering Database entry types (8.10.1, 8.10.3, and 8.10.4). The required forwarding and filtering behavior is summarized in 8.10.6, 8.10.8, Table 8-5, Table 8-6, and Table 8-7.

8.6.4 Egress rule checking

Frames shall be filtered, i.e., discarded, if

a) For the frame's VID, as determined by applying the ingress rules (8.6.1), the transmission Port is not present in the Member set (8.10.9); or
b) The value of the include_tag parameter, determined as shown below, is False, and the Bridge does not support the ability to translate embedded MAC Address information from the format indicated by the canonical_format_indicator parameter to the format appropriate to the MAC method on which the data request will be carried.

NOTE 1—The meanings of the terms Canonical format and Non-canonical format are discussed in Annex F.

The value of the include_tag parameter in the EM_UNITDATA.request primitive is determined as follows:

c) If, for the frame's VID, as determined by ingress rule checking (8.6.1), the transmission Port is present in the untagged set (8.10.9), then the value False is used. Otherwise;
d) The value True is used.

NOTE 2—As all incoming frames, including priority-tagged frames, are classified as belonging to a VLAN by the ingress rule checking (8.6.1), the transmitting Port only transmits VLAN-tagged frames or untagged frames, and can never transmit priority-tagged frames. Hence, a station sending a priority-tagged frame via a VLAN Bridge will receive a response that is either VLAN-tagged or untagged, depending upon the state of the untagged set for the VLAN concerned.

The value of the canonical_format_indicator parameter of the data request primitive is equal to the value of that parameter as received in the corresponding data indication.

8.6.5 Queuing for transmission

The Forwarding Process provides storage for queued frames, awaiting an opportunity to submit these for transmission to the individual MAC Entities associated with each Bridge Port. The order of frames received on the same Bridge Port shall be preserved for

a) Unicast frames with a given user_priority (regenerated as defined in 8.5.1) for a given combination of destination_address and source_address;
b) Group-addressed frames with a given user_priority (regenerated as defined in 8.5.1) for a given destination_address.

The Forwarding Process may provide more than one transmission queue for a given Bridge Port. Frames are assigned to storage queue(s) on the basis of their user_priority using a traffic class table that is part of the state information associated with each Port. The table indicates, for each possible value of user_priority, the corresponding value of traffic class that shall be assigned. Values of user_priority range from 0 through 7. Queues correspond one-to-one with traffic classes.

NOTE 1—Annex H.2 of IEEE Std 802.1D, 1998 Edition contains further explanation of the use of user_priority values and how they map to traffic classes.

For management purposes, up to eight traffic classes are supported by the traffic class tables in order to allow for separate queues for each level of user_priority. Traffic classes are numbered 0 through N-1, where N is the number of traffic classes associated with a given outbound Port. Management of traffic class information is optional. Traffic class 0 corresponds to non-expedited traffic; non-zero traffic classes are expedited classes of traffic.

NOTE 2—In a given Bridge, it is permissible to implement different numbers of traffic classes for each Port. Ports associated with MAC methods that support a single transmission priority, such as CSMA/CD, can support more than one traffic class.

Where the Forwarding Process does not support expedited classes of traffic for a given Port, in other words, where there is a single traffic class associated with the Port, all values of user_priority map to traffic class 0. In bridges which support expedited traffic, the recommended mapping of user_priority to traffic class, for the number of traffic classes implemented, is as shown in Table 8-2. Each entry in the body of the table is the traffic class assigned to traffic with a given user_priority, for a given number of available traffic classes.

Table 8-2—Recommended user priority to traffic class mappings

		Number of Available Traffic Classes							
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
User Priority	**0 (Default)**	0	0	0	1	1	1	1	2
	1	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	1
	3	0	0	0	1	1	2	2	3
	4	0	1	1	2	2	3	3	4
	5	0	1	1	2	3	4	4	5
	6	0	1	2	3	4	5	5	6
	7	0	1	2	3	4	5	6	7

NOTE—The rationale behind the choice of values shown in this table is discussed in Annex H.2 of IEEE Std 802.1D, 1998 Edition. A consequence of the mapping shown is that frames carrying the default user priority are given preferential treatment relative to user priority 1 and 2 in Bridges that implement four or more Traffic Classes.

A frame queued by the Forwarding Process for transmission on a Port shall be removed from that queue on submission to the individual MAC Entity for that Port. No further attempt shall be made to transmit the frame on that Port even if the transmission is known to have failed.

A frame queued by the Forwarding Process for transmission on a Port can be removed from that queue, and not subsequently transmitted, if the time for which buffering is guaranteed has been exceeded for that frame.

A frame queued for transmission on a Port shall be removed from that queue if that is necessary to ensure that the maximum bridge transit delay (6.3.6) will not be exceeded at the time at which the frame would subsequently be transmitted.

A frame queued for transmission on a Port shall be removed from that queue if the associated Port leaves the forwarding state.

Removal of a frame from a queue for any particular Port does not of itself imply that it shall be removed from a queue for transmission on any other Port.

NOTE 3—The assumption underlying the model of operation of the Forwarding Process is that it is fed by frame reception with the parameters present in data indication primitives, and feeds these parameters to frame transmission in the form of data request primitives. The act of queueing a frame therefore amounts to placing the parameters of a data request primitive on an outbound queue.

8.6.6 Transmission selection

The following algorithm shall be supported by all Bridges as the default algorithm for selecting frames for transmission:

a) For each Port, frames are selected for transmission on the basis of the traffic classes that the Port supports. For a given supported value of traffic class, frames are selected from the corresponding queue for transmission only if all queues corresponding to numerically higher values of traffic class supported by the Port are empty at the time of selection;
b) For a given queue, the order in which frames are selected for transmission shall maintain the ordering requirement specified in 8.6.5.

Additional algorithms, selectable by management means, may be supported as an implementation option so long as the requirements of 8.6.5 are met.

8.6.7 Priority mapping

The user_priority parameter in an EM_UNITDATA.request primitive (7.1) shall be equal to the user_priority parameter in the corresponding data indication.

The mapping of user_priority to outbound access_priority is achieved via fixed, MAC method-specific mappings. The access_priority parameter in an EM_UNITDATA.request primitive (7.1) shall be determined from the user_priority in accordance with the values shown in Table 8-3 for the MAC methods that will carry the data request. The values shown in Table 8-3 are not modifiable by management or other means.

The table shows two columns for the 8802-5 MAC method. The mapping in the column marked “8802-5 (alternate)” is included in order to permit backwards compatibility with equipment manufactured in accordance with ISO/IEC 10038: 1993; however, the use of this mapping reduces the number of available access priority values to three. For this reason, it is recommended that the column marked “8802-5 (default)” is supported as the default mapping where backward compatibility is not an issue.

8.6.8 Frame formatting

The format of the frame transmitted on an outbound Port is determined by the value of the include_tag and the canonical_format_indicator parameters in the EM_UNITDATA.request primitive, and the MAC method supported by the Port. Where the include_tag parameter is TRUE, the frame shall be formatted in accordance with the tagged frame format defined in Clause 9.

8.6.9 FCS recalculation

Where a frame is being forwarded between two individual MAC Entities of the same IEEE 802 LAN type, and relaying the frame involves no changes to the data that is within the FCS coverage, the FCS received in the EM_UNITDATA.indication primitive may be supplied in the corresponding EM_UNITDATA.request primitive and not recalculated (7.1, 7.2, 6.3.7).

Table 8-3—Outbound access priorities

user_priority	Outbound Access Priority per MAC method								
	802.3	8802-4	8802-5 (default)	8802-5 (alternate)	8802-6	802.9a*	8802.11	8802-12	FDDI
0	0	0	0	4	0	0	0	0	0
1	0	1	1	4	1	0	0	0	1
2	0	2	2	4	2	0	0	0	2
3	0	3	3	4	3	0	0	0	3
4	0	4	4	4	4	0	0	4	4
5	0	5	5	5	5	0	0	4	5
6	0	6	6	6	6	0	0	4	6
7	0	7	6	6	7	0	0	4	6

*In the absence of a definition, in 6.5 of IEEE Std 802.1D, 1998 Edition, of support by IEEE Std 802.9a-1995, it is assumed that for this MAC method, access priority 0 will map to “low.”

Where a frame is being forwarded between two individual MAC Entities of different types, recalculation of the FCS is necessary if the differences between the LAN MAC methods is such that an FCS calculated according to the MAC procedures for the destination MAC method would differ from the FCS carried by the received frame, or if relaying the frame involves changes to the data that is within the FCS coverage. Where necessary, the FCS is recalculated according to the specific MAC procedures of the transmitting MAC entity.

NOTE—There are two possibilities for recreating a valid FCS. The first is to generate a new FCS by algorithmically modifying the received FCS, based on knowledge of the FCS algorithm and the transformations that the frame has undergone between reception and transmission. The second is to rely on the normal MAC procedures to recalculate the FCS for the outgoing frame. The former approach may be preferable in terms of its ability to protect against increased levels of undetected frame errors. Annex G of IEEE Std 802.1D, 1998 Edition discusses these possibilities in more detail. The frame_check_sequence parameter of the Enhanced Internal Sublayer Service (7.1) is able to signal the validity, or otherwise, of the FCS; an unspecified value in this parameter in a data request indicates to the transmitting MAC that the received FCS is no longer valid, and the FCS must therefore be recalculated.

FCS recalculation is necessary if any of the following conditions are true:

a) The algorithm used to determine the FCS differs between the MAC methods used by the two MAC entities;

b) The FCS coverage differs between the MAC methods used by the two MAC entities;

c) Relaying the frame between the two MAC entities involves changes to the data that is within the coverage of the FCS (e.g., the frame was tagged on one link, but not on the other).

8.7 Frame transmission

The individual MAC Entity associated with each Bridge Port transmits frames submitted to it by the MAC Relay Entity.

Relayed frames are submitted for transmission by the Forwarding Process. The EM_UNITDATA.request primitive associated with such frames conveys the values of the source and destination address fields received in the corresponding EM_UNITDATA.indication primitive.

LLC Protocol Data Units are submitted by LLC as a user of the MAC Service provided by the Bridge Port. Frames transmitted to convey such Protocol Data Units carry the individual MAC Address of the Port in the source address field.

Each frame is transmitted subject to the MAC procedures to be observed for that specific IEEE 802 LAN technology. The values of the frame_type and mac_action parameters of the corresponding EM_UNITDATA.request primitive shall be user_data_frame and request_with_no_response respectively (7.2; 6.5 of IEEE Std 802.1D, 1998 Edition).

Frames transmitted following a request by the LLC user of the MAC Service provided by the Bridge Port shall also be submitted to the MAC Relay Entity.

8.8 The Learning Process

The Learning Process observes the source MAC Addresses of frames received on each Port and updates the Filtering Database conditionally on the state of the receiving Port. The VID associated with the frame is used to ensure that the address information is learned relative to the frame's VLAN.

Frames are submitted to the Learning Process following VLAN classification, as specified in 8.6.1.

The Learning Process can deduce the Port through which particular end stations in the Bridged LAN can be reached by inspection of the source MAC Address field and VID of received frames. It records such information in the Filtering Database (8.10). It shall create or update a Dynamic Filtering Entry (8.10.3) associated with the frame's VID (8.10.9), associating the reception Port with the source MAC Address, if and only if

a) The Port on which the frame was received is in a state that allows learning (17.5 of IEEE Std 802.1D, 1998 Edition), and
b) The source address field of the frame denotes a specific end station, i.e., is not a group MAC Address, and
c) The resulting number of entries would not exceed the capacity of the Filtering Database, and
d) The Member set (8.10.9) for the frame's VID includes at least one Port.

NOTE—If the Member set for a given VID is the empty set, then that VLAN is not currently active, and the Bridge will therefore filter all frames destined for that VLAN, regardless of their destination address. There is therefore no reason to include MAC Address filtering information in the Filtering Database for that VLAN until such a time as it becomes active.

If the Filtering Database is already filled up to its capacity, but a new entry would otherwise be made, then an existing entry may be removed to make room for the new entry.

Figure 8-5 illustrates the operation of the Learning Process in the inclusion of station location information carried by a single frame, received on one of the Ports of a Bridge, in the Filtering Database.

8.9 VLAN classification

Each frame received by a VLAN Bridge shall be classified as belonging to exactly one VLAN by associating a VID value with the received frame. The classification is achieved as follows:

a) If the vlan_identifier parameter carried in a received data indication is the null VLAN ID (Table 9-2), and the Bridge supports only Port-based VLAN classification, then the VID for the Frame is the unique PVID associated with the Port through which the frame was received (8.4.4). Otherwise:

b) If the vlan_identifier parameter carried in a received data indication is the null VLAN ID and the Bridge supports Port-and-Protocol-based VLAN classification, then the VID for the Frame is selected from the VID Set of the port through which the Frame was received. The VID selected is the member of the VID Set (8.4.4) for which the associated Protocol Group Identifier (8.9.3) is equal to the Protocol Group Identifier of the Frame. If no matches are found then the VID for the frame is the PVID associated with the Port. Otherwise:

c) The VID for the Frame is the vlan_identifier parameter value.

NOTE 1—As defined in 6.4.2.1, the vlan_identifier parameter carries the null VLAN ID if the frame was not VLAN-tagged. There are two cases; either the frame was untagged, or the frame was tagged and the tag header carried a VID value equal to the null VLAN ID (i.e., a priority-tagged frame).

NOTE 2—VIDs of value FFF cannot be configured in any Filtering Database entry (see Table 9-2). Consequently, any incoming frame whose VLAN classification is FFF will be discarded by the Forwarding Process.

The VID value thus identified, known as the *VLAN classification* of the frame, is used as the value of the vlan_classification parameter of any corresponding data request primitives.

8.9.1 Protocol Classification

For frames received from the E-ISS on bridge ports which implement Port-and-Protocol-based VLAN classification, the following procedures are followed in order to classify the frame's format and protocol. These procedures are described as if they assigned values to parameters that are used as input to ingress rule checking (8.6.1).

The **detagged_frame_type** parameter indicates the frame format. The value is determined as follows:

a) If the frame is Untagged or Priority Tagged, this parameter is present and indicates the link-layer encapsulation format of the *Detagged Frame*. The Detagged Frame of an Untagged Frame is the Frame itself. The Detagged Frame of a Tagged Frame or Priority Tagged Frame is the Frame which results from untagging the Frame by the procedure described in 9.1. The value of detagged_frame_type is as follows:

    1) Ethernet, if the Detagged Frame uses Type-encapsulated 802.3 format

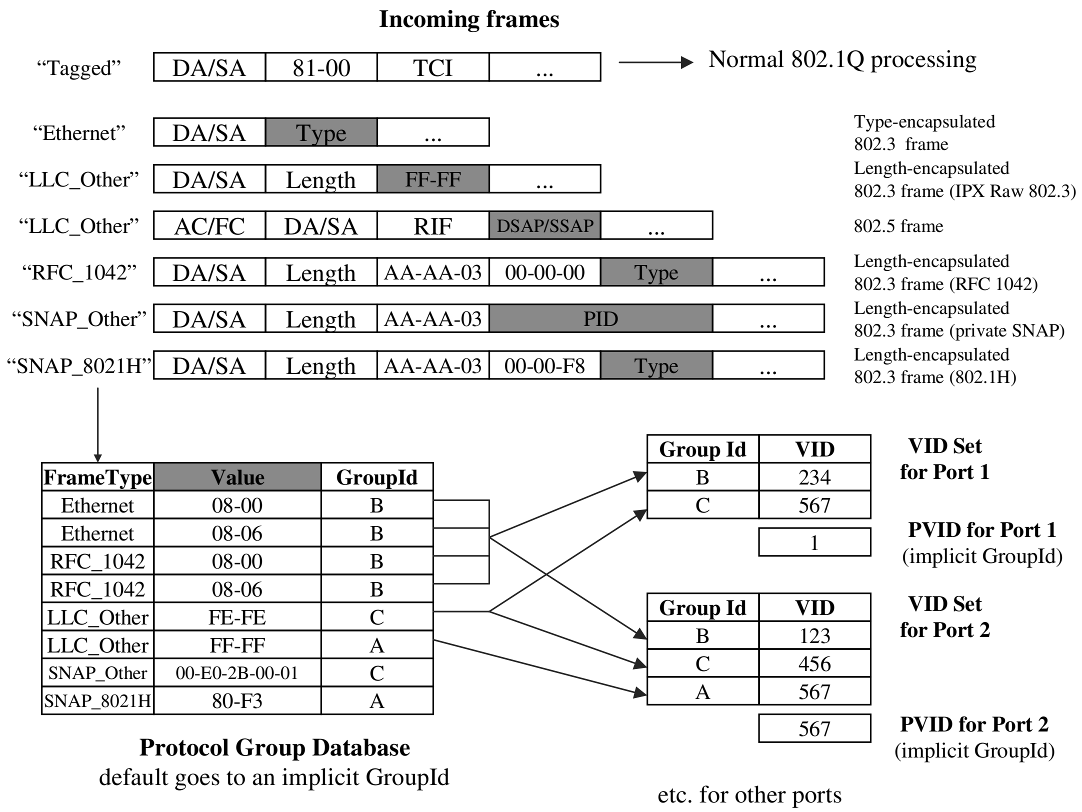
    2) RFC_1042, if the Detagged Frame is of the format specified by 10.5 in IEEE Std 802-2001 for the encoding of an IEEE 802.3 Type Field in an 802.2/SNAP header (this supersedes the original definition, which appeared in RFC 1042)

    3) SNAP_8021H, if the Detagged Frame is of the format specified by IEEE Std 802.1H, 1997 Edition, for the encoding of an IEEE 802.3 Type Field in an 802.2/SNAP header

    4) SNAP_Other, if the Detagged Frame contains an LLC UI PDU with DSAP and SSAP fields equal to the LLC address reserved for SNAP and the 5-octet SNAP Protocol Identifier (PID) value is not in either of the ranges used for RFC_1042 or SNAP_8021H above

    5) LLC_Other, if the Detagged Frame contains both a DSAP and an SSAP address field in the positions specified by IEEE 802.2 Logical Link Control, but is not any of the formats described for LLC frames above

b) Else the parameter is not present.


Incoming frames
"Tagged" DA/SA 81-00 TCI ...
Normal 802.1Q processing
"Ethernet" DA/SA Type ...
Type-encapsulated 802.3 frame
"LLC_Other" DA/SA Length FF-FF ...
Length-encapsulated 802.3 frame (IPX Raw 802.3)
"LLC_Other" AC/FC DA/SA RIF DSAP/SSAP ...
802.5 frame
"RFC_1042" DA/SA Length AA-AA-03 00-00-00 Type ...
Length-encapsulated 802.3 frame (RFC 1042)
"SNAP_Other" DA/SA Length AA-AA-03 PID ...
Length-encapsulated 802.3 frame (private SNAP)
"SNAP_8021H" DA/SA Length AA-AA-03 00-00-F8 Type ...
Length-encapsulated 802.3 frame (802.1H)
VID Set for Port 1
PVID for Port 1 (implicit GroupId)
1
VID Set for Port 2
PVID for Port 2 (implicit GroupId)
567
Protocol Group Database
default goes to an implicit GroupId
etc. for other ports

FrameType	Value	GroupId
Ethernet	08-00	B
Ethernet	08-06	B
RFC_1042	08-00	B
RFC_1042	08-06	B
LLC_Other	FE-FE	C
LLC_Other	FF-FF	A
SNAP_Other	00-E0-2B-00-01	C
SNAP_8021H	80-F3	A

Group Id	VID
B	234
C	567

Group Id	VID
B	123
C	456
A	567

NOTE—The PID shown in this figure is a Protocol Identifier, as defined in 5.3 of IEEE Std 802. It is a 5-octet value consisting of a 3-octet OUI value followed by a 2-octet locally administered identifier.

Figure 8-9—Example of operation of port-and-protocol based classification

The **ethertype** parameter is present if the detagged_frame_type parameter is present and has the value Ethernet, RFC_1042, or SNAP_8021H. Its value is the IEEE 802.3 Type Field present in the Detagged Frame.The value is determined as follows:

c) If the detagged_frame_type parameter is present and has the value Ethernet, RFC_1042, or SNAP_8021H, then this parameter is present and has the value of the IEEE 802.3 Type Field present in the Detagged Frame.

d) Else the parameter is not present.

The **llc_saps** parameter is present if the detagged_frame_type parameter is present and has the value LLC_Other. Its value is determined as follows:

e) If the detagged_frame_type parameter is present and has the value LLC_Other then this parameter is present and its value is the pair of LLC 802.2 DSAP and SSAP address field values.

f) Else the parameter is not present.

NOTE 1—A frame that is encapsulated using values of hex FF/FF in the position where an LLC header is to be expected (as defined by IEEE Std 802.2, 1998 Edition) is known as a "Novell IPX Raw" encapsulation. Such frames do not conform to IEEE Std 802.2, 1998 Edition, in that they do not include some of the other required LLC fields. For the purposes of this standard, they are treated as LLC_Other, regardless of whether they are legal LLC frames or not.

NOTE 2—Bridges are not required, for the purposes of this standard, to completely verify the format of frames as meeting IEEE Std 802.2 or not: they are only required to recognize the DSAP and SSAP fields of such frames.

The **snap_pid** parameter is present if the detagged_frame_type parameter is present and has the value SNAP_Other. Its value is determined as follows:

g) If the detagged_frame_type parameter is present and has the value SNAP_Other then the parameter is present and its value is the contents of the 5 octets following the LLC header, i.e., the PID field.
h) Else the parameter is not present.

8.9.2 Protocol Templates

In a Bridge that supports Port-and-Protocol-based VLAN classification, a Protocol Template is a tuple that specifies a protocol to be identified in received frames. A Protocol Template has one of the following formats:

a) A value “Ethernet” and a 16-bit IEEE 802.3 Type Field value
b) A value “RFC_1042” and a 16-bit IEEE 802.3 Type Field value
c) A value “SNAP_8021H” and a 16-bit IEEE 802.3 Type Field value
d) A value “SNAP_Other” and a 40-bit PID value
e) A value “LLC_Other” and a pair of IEEE 802.2 LSAP values: DSAP and SSAP

A Protocol Template *matches* a Frame if

f) The Frame’s detagged_frame_type is Ethernet, the Protocol Template is of type Ethernet, and the frame’s IEEE 802.3 Type Field is equal to the value of the IEEE 802.3 Type Field of the Protocol Template, or
g) The Frame’s detagged_frame_type is RFC_1042, the Protocol Template is of type RFC_1042 and the frame’s IEEE 802.3 Type Field is equal to the IEEE 802.3 Type Field of the Protocol Template, or
h) The Frame’s detagged_frame_type is SNAP_8021H, the Protocol Template is of type SNAP_8021H, and the frame’s IEEE 802.3 Type Field is equal to the IEEE 802.3 Type Field of the Protocol Template, or
i) The Frame’s detagged_frame_type is SNAP_Other, the Protocol Template is of type SNAP_Other, and the frame’s snap_pid is equal to the PID of the Protocol Template, or
j) The Frame’s detagged_frame_type is LLC_Other, the Protocol Template is of type LLC_Other, and the frame’s llc_saps matches the value of the DSAP and SSAP of the Protocol Template.

NOTE—If a port does not support Protocol Templates of the Frame’s detagged_frame_type then no match will occur.

8.9.3 Protocol Group Identifiers

A Bridge that supports Port-and-Protocol-based VLAN classification shall support Protocol Group Identifiers.

A Protocol Group Identifier, shown as “Group Id” in Figure 8-9, designates a group of protocols that will be associated with one member of the VID Set of a Port. The association of protocols into groups is established by the contents of the Protocol Group Database, as described in 8.9.4. The identifier has scope only within a single bridge.

There is an implicit Protocol Group Identifier that is assigned to frames that match none of the entries in the Protocol Group Database. Therefore, every incoming Frame can be assigned to a Protocol Group Identifier.

8.9.4 Protocol Group Database

A Bridge that supports Port-and-Protocol-based VLAN classification~~,~~ shall support a single Protocol Group Database. The Protocol Group Database groups together a set of one or more Protocols by assigning them the same Protocol Group Identifier (8.9.3). Each entry of the Protocol Group Database comprises the following:

a) A Protocol Template
b) A Protocol Group Identifier

The Protocol Group Database specifies a mapping from Protocol Templates to Protocol Group Identifiers: if two entries of the Protocol Group Database contain different Protocol Group Identifiers then their Protocol Templates must also be different.

The entries of the Protocol Group Database may be configured by management. A Bridge that supports Port-and-Protocol-based VLAN classification shall support at least one of the formats of Protocol Template.

An implicit Protocol Group Database entry exists that matches all frames: this entry is invoked for frames that do not match the template of any of the other entries. It references an implicit Protocol Group Identifier that selects the PVID on each port. In this way, it is ensured that all incoming Frames are matched by a Protocol Group Identifier and, hence, are assigned to a VID.

NOTE—If there are no entries in the Protocol Group Database, then the frame relay behavior of this Bridge is identical to the frame relay behavior of a Bridge having the same number of Ports that supports only Port-based VLAN classification.

8.10 The Filtering Database

The Filtering Database supports queries by the Forwarding Process as to whether frames received by the Forwarding Process, with given values of destination MAC Address parameter and VID, are to be forwarded through a given potential transmission Port (8.6.2 and 8.6.3). It contains filtering information in the form of filtering entries that are either

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

Two entry types are used to represent static filtering information. The Static Filtering Entry represents static information in the Filtering Database for individual and for group MAC Addresses. It allows administrative control of

c) Forwarding of frames with particular destination addresses; and
d) The inclusion in the Filtering Database of dynamic filtering information associated with Extended Filtering Services, and use of this information.

The Filtering Database shall contain entries of the Static Filtering Entry type.

The Static VLAN Registration Entry represents all static information in the Filtering Database for VLANs. It allows administrative control of

e) Forwarding of frames with particular VIDs;
f) The inclusion/removal of tag headers in forwarded frames; and
g) The inclusion in the Filtering Database of dynamic VLAN membership information, and use of this information.

The Filtering Database may contain entries of the Static VLAN Registration Entry type.

Static filtering information is added to, modified, and removed from the Filtering Database only under explicit management control. It shall not be automatically removed by any ageing mechanism. Management of static filtering information may be carried out by use of the remote management capability provided by Bridge Management (8.13) using the operations specified in Clause 12.

Three entry types are used to represent dynamic filtering information:

h) Dynamic Filtering Entries are used to specify the Ports on which individual MAC Addresses have been learned. They are created and updated by the Learning Process (8.8), and are subject to ageing and removal by the Filtering Database.

i) Group Registration Entries support the registration of group MAC Addresses. They are created, updated, and removed by the GMRP protocol in support of Extended Filtering Services (8.10.4; 6.6.5 of IEEE Std 802.1D, 1998 Edition; Clause 10 of IEEE Std 802.1D, 1998 Edition), subject to the state of the Restricted_Group_Registration management control (10.3.2.3 of IEEE Std 802.1D, 1998 Edition). If the value of this control is TRUE, then the creation of a Group Registration Entry is not permitted unless a Static Filtering Entry exists that permits dynamic registration for the Group concerned.

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

Static Filtering Entries and Group Registration Entries comprise

k) A MAC Address specification;

l) A VLAN Identifier (VID);

m) A Port Map, with a control element for each outbound Port to specify filtering for that MAC Address specification and VID.

Dynamic Filtering Entries comprise

n) A MAC Address specification;

o) A locally significant Filtering Identifier (FID; see 8.10.7);

p) A Port Map, with a control element for each outbound Port to specify filtering for that MAC Address specification in the VLAN(s) allocated to that FID.

Static and Dynamic VLAN Registration Entries comprise

q) A VLAN Identifier;

r) A Port Map, with a control element for each outbound Port to specify filtering for the VLAN.

Dynamic filtering information may be read by use of the remote management capability provided by Bridge Management (8.13) using the operations specified in Clause 12.

The Filtering Services supported by a Bridge (Basic and Extended Filtering Services) determine the default behavior of the Bridge with respect to the forwarding of frames destined for group MAC Addresses. In Bridges that support Extended Filtering Services, the default forwarding behavior for group MAC Addresses, for each Port, and for each VID, can be configured both statically and dynamically by means of Static Filtering Entries and/or Group Registration Entries that can carry the following MAC Address specifications:

s) All Group Addresses, for which no more specific Static Filtering Entry exists;
t) All Unregistered Group Addresses (i.e., all group MAC Addresses for which no Group Registration Entry exists), for which no more specific Static Filtering Entry exists.

NOTE 1—The All Group Addresses specification s) above, when used in a Static Filtering Entry with an appropriate control specification, provides the ability to configure a Bridge that supports Extended Filtering Services to behave as a Bridge that supports only Basic Filtering Services on some or all of its Ports. This might be done for the following reasons:

— The Ports concerned serve "legacy" devices that wish to receive multicast traffic, but are unable to register Group membership;
— The Ports concerned serve devices that need to receive all multicast traffic, such as routers or diagnostic devices.

The Filtering Database shall support the creation, updating and removal of Dynamic Filtering Entries by the Learning Process (8.8). In Bridges that support Extended Filtering Services, the Filtering Database shall support the creation, updating, and removal of Group Registration Entries by GMRP (Clause 10 of IEEE Std 802.1D, 1998 Edition).

Figure 8-4 illustrates use of the Filtering Database by the Forwarding Process in a single instance of frame relay between the Ports of a Bridge with two Ports.

Figure 8-5 illustrates the creation or update of a dynamic entry in the Filtering Database by the Learning Process. The entries in the Filtering Database allow MAC Address information to be learned independently for each VLAN or set of VLANs, by relating a MAC Address to the VLAN or set of VLANs on which that address was learned. This has the effect of creating independent Filtering Databases for each VLAN or set of VLANs that is present in the Bridged LAN.

NOTE 2—This standard specifies a single Filtering Database that contains all Filtering Database entries; however, the inclusion of VIDs and FIDs in the filtering entries effectively provides distinct IEEE Std 802.1D-style Filtering Databases per VLAN or set of VLANs.

NOTE 3—The ability to create VLAN-dependent Filtering Database entries allows a VLAN Bridge to support

— Multiple end stations with the same individual MAC Address residing on different VLANs;
— End stations with multiple interfaces, each using the same individual MAC Address,
as long as not more than one end station or interface that uses a given MAC Address resides in a given VLAN.

Figure 8-6 illustrates the operation of the Spanning Tree Protocol Entity (8.12), which operates the Spanning Tree Algorithm and Protocol, and its notification of the Filtering Database of changes in active topology signaled by that protocol.

There are no standardized constraints on the size of the Filtering Database in an implementation for which conformance to this standard is claimed. The PICS Proforma in Annex A requires the following to be specified for a given implementation:

u) The total number of entries (Static Filtering Entries, Dynamic Filtering Entries, Group Registration Entries, Static VLAN Registration Entries, and Dynamic VLAN Registration Entries) that the implementation of the Filtering Database can support, and
v) Of that total number, the total number of VLAN Registration Entries (static and dynamic) that the Filtering Database can support.

8.10.1 Static Filtering Entries

A Static Filtering Entry specifies

a) A MAC Address specification, comprising
   1) An Individual MAC Address; or
   2) A group MAC Address; or
   3) All Group Addresses, for which no more specific Static Filtering Entry exists; or
   4) All Unregistered Group Addresses, for which no more specific Static Filtering Entry exists.
b) The VID of the VLAN to which the static filtering information applies;
c) A Port Map, containing a control element for each outbound Port, specifying that a frame with a destination MAC Address and VID that meets this specification is to be
   1) Forwarded, independently of any dynamic filtering information held by the Filtering Database; or
   2) Filtered, independently of any dynamic filtering information; or
   3) Forwarded or filtered on the basis of dynamic filtering information, or on the basis of the default Group filtering behavior for the outbound Port (8.10.6) if no dynamic filtering information is present specifically for the MAC Address.

All Bridges shall have the capability to support the first two values for the MAC Address specification, and all three values for each control element for all Static Filtering Entries (i.e., shall have the capability to support a1, a2, c1, c2, and c3 above).

A Bridge that supports Extended Filtering Services shall have the capability to support all four values for the MAC Address specification and all three control element values for all Static Filtering Entries.

For a given MAC Address specification, a separate Static Filtering Entry with a distinct Port Map may be created for each VLAN from which frames are received by the Forwarding Process.

In addition to controlling the forwarding of frames, Static Filtering Entries for group MAC Addresses provide the Registrar Administrative Control values for the GMRP protocol (Clauses 10, 12, and 12.9.1 of IEEE Std 802.1D, 1998 Edition). Static configuration of forwarding of specific group addressed frames to an outbound port indicates Registration Fixed on that port: a desire to receive frames addressed to that Group even in the absence of dynamic information. Static configuration of filtering of frames that might otherwise be sent to an outbound port indicates Registration Forbidden. The absence of a Static Filtering Entry for the group address, or the configuration of forwarding or filtering on the basis of dynamic filtering information, indicates Normal Registration.

8.10.2 Static VLAN Registration Entries

A Static VLAN Registration Entry specifies

a) The VID of the VLAN to which the static filtering information applies;
b) A Port Map, consisting of a control element for each outbound Port, specifying
   1) The Registrar Administrative Control values for the GVRP protocol (Clause 11) for the VLAN specified. In addition to providing control over the operation of GVRP, these values can also directly affect the forwarding behavior of the Bridge, as described in 8.10.9. The values that can be represented are
      i) Registration Fixed; or
      ii) Registration Forbidden; or
      iii) Normal Registration.
   2) Whether frames destined for the VLAN specified are to be VLAN-tagged or untagged when forwarded through this Port.

All Bridges shall be capable of supporting all values for each control element for all Static VLAN Registration Entries; however, the ability to support more than one untagged VLAN on egress on any given Port is optional (see 5.1 and 5.2).

NOTE—In other words, it shall be possible to configure any VLAN as untagged on egress, but it is an implementation option as to whether only a single untagged VLAN per Port on egress is supported, or whether multiple untagged VLANs per Port on egress are supported.

A separate Static VLAN Registration Entry with a distinct Port Map may be created for each VLAN from which frames are received by the Forwarding Process.

8.10.3 Dynamic Filtering Entries

A Dynamic Filtering Entry specifies

a) An individual MAC Address;
b) The FID, an identifier assigned by the MAC Bridge (8.10.7) to identify a set of VIDs for which no more than one Dynamic Filtering Entry can exist for any individual MAC Address;

NOTE 1—An FID identifies a set of VLANs among which *Shared VLAN Learning* (3.26) takes place. Any pair of FIDs identifies two sets of VLANs between which *Independent VLAN Learning* (3.8) takes place. The allocation of FIDs by a Bridge is described in 8.10.7.

c) A Port Map that specifies forwarding of frames destined for that MAC Address and FID to a single Port.

NOTE 2—This is equivalent to specifying a single port number; hence, this specification is directly equivalent to the specification of dynamic entries in ISO/IEC 10038: 1993.

Dynamic Filtering Entries are created and updated by the Learning Process (8.8). They shall be automatically removed after a specified time, the Ageing Time, has elapsed since the entry was created or last updated. No more than one Dynamic Filtering Entry shall be created in the Filtering Database for a given combination of MAC Address and FID.

Dynamic Filtering Entries cannot be created or updated by management.

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

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

The Ageing Time may be set by management (Clause 12). A range of applicable values and a recommended default is specified in Table 8-4; this is suggested to remove the need for explicit configuration in most cases. If the value of Ageing Time can be set by management, the Bridge shall have the capability to use values in the range specified, with a granularity of 1 s.

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

Table 8-4—Ageing time parameter value

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

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

8.10.4 Group Registration Entries

A Group Registration Entry specifies

a) A MAC Address specification, comprising
   1) A group MAC Address; or
   2) All Group Addresses, for which no more specific Static Filtering Entry exists; or
   3) All Unregistered Group Addresses, for which no more specific Static Filtering Entry exists.
b) The VID of the VLAN in which the dynamic filtering information was registered;
c) A Port Map, consisting of a control element for each outbound Port, which specifies forwarding (Registered) or filtering (Not registered) of frames destined to the MAC Address and VID.

Group Registration Entries are created, modified and deleted by the operation of GMRP (Clause 10 of IEEE Std 802.1D, 1998 Edition, as modified by Clause 10 of this standard). No more than one Group Registration Entry shall be created in the Filtering Database for a given combination of MAC Address specification and VID.

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

The creation of Group Registration Entries is subject to the Restricted_Group_Registration management control (10.3.2.3 of IEEE Std 802.1D, 1998 Edition). If the value of this control is TRUE, a dynamic entry for a given Group may only be created if a Static Filtering Entry already exists for that Group, in which the Registrar Administrative Control value is Normal Registration.

8.10.5 Dynamic VLAN Registration Entries

A Dynamic VLAN Registration Entry specifies

a) The VID of the VLAN to which the dynamic filtering information applies;
b) A Port Map with a control element for each outbound Port specifying whether the VLAN is registered on that Port.

A separate Dynamic VLAN Registration Entry with a distinct Port Map may be created for each VLAN from which frames are received by the Forwarding Process.

The creation of Dynamic VLAN Registration Entries is subject to the Restricted_VLAN_Registration management control (11.2.3.2.3). If the value of this control is TRUE, a dynamic entry for a given VLAN may only be created if a Static VLAN Registration Entry already exists for that VLAN, in which the Registrar Administrative Control value is Normal Registration.

8.10.6 Default Group filtering behavior

Forwarding and filtering of group-addressed frames may be managed by specifying defaults for each VLAN and outbound Port. The behavior of each of these defaults, as modified by the control elements of more explicit Filtering Database entries applicable to a given frame's MAC Address, VLAN classification, and outbound Port is as follows:

NOTE 1—As stated in 8.10.1, for a given MAC Address there may be separate Static Filtering Entries with a distinct Port Map for each VLAN.

a) *Forward All Groups.* The frame is forwarded, unless an explicit Static Filtering Entry specifies filtering independent of any dynamic filtering information.
b) *Forward Unregistered Groups.* The frame is forwarded, unless
   1) An explicit Static Filtering Entry specifies filtering independent of any dynamic filtering information; or
   2) An explicit Static Filtering Entry specifies forwarding or filtering on the basis of dynamic filtering information, and an applicable explicit Group Registration Entry exists specifying filtering; or
   3) An applicable explicit Static Filtering Entry does not exist, but an applicable Group Registration entry specifies filtering.
c) *Filter Unregistered Groups.* The frame is filtered unless
   1) An explicit Static Filtering Entry specifies forwarding independent of any dynamic filtering information; or
   2) An explicit Static Filtering Entry specifies forwarding or filtering on the basis of dynamic filtering information, and an applicable explicit Group Registration Entry exists specifying forwarding; or
   3) An applicable explicit Static Filtering Entry does not exist, but an applicable Group Registration entry specifies forwarding.

In Bridges that support only Basic Filtering Services, the default Group filtering behavior is Forward All Groups for all Ports of the Bridge, for all VLANs.

NOTE 2—Forward All Groups corresponds directly to the behavior specified in ISO/IEC 10038: 1993 when forwarding group MAC Addressed frames for which no static filtering information exists in the Filtering Database. Forward All Groups makes use of information contained in Static Filtering Entries for specific group MAC Addresses, but overrides any information contained in Group Registration Entries. Forward Unregistered Groups is analogous to the forwarding behavior of a Bridge with respect to individual MAC Addresses. If there is no static or dynamic information for a specific group MAC Address, then the frame is forwarded; otherwise, the frame is forwarded in accordance with the statically configured or dynamically learned information.

In Bridges that support Extended Filtering Services, the default Group filtering behavior for each outbound Port for each VLAN is determined by the following information contained in the Filtering Database:

d) Any Static Filtering Entries applicable to that VLAN with a MAC Address specification of All Group Addresses or All Unregistered Group Addresses;
e) Any Group Registration Entries applicable to that VLAN with a MAC Address specification of All Group Addresses or All Unregistered Group Addresses.

The means whereby this information determines the default Group filtering behavior is specified in 8.10.8, Table 8-6, and Table 8-7.

NOTE 3—The result is that the default Group filtering behavior for each VLAN can be configured for each Port of the Bridge via Static Filtering Entries, determined dynamically via Group Registration Entries created/updated by GMRP (Clause 10), or both. For example, in the absence of any static or dynamic information in the Filtering Database for All Group Addresses or All Unregistered Group Addresses, the default Group filtering behavior will be Filter Unregistered Groups on all Ports, for all VLANs. Subsequently, the creation of a Dynamic Group Registration Entry for All Unregistered Group Addresses indicating “Registered” for a given VLAN on a given Port would cause that Port to exhibit Forward Unregistered Groups behavior for that VLAN. Similarly, creating a Static Filtering Entry for All Group Addresses indicating “Registration Fixed” on a given Port for that VLAN would cause that Port to exhibit Forward All Groups behavior.

Hence, by using appropriate combinations of “Registration Fixed,” “Registration Forbidden,” and “Normal Registration” in the Port Maps of Static Filtering Entries for the All Group Addresses and All Unregistered Group Addresses address specifications, it is possible, for a given Port and VLAN, to

— Fix the default Group filtering behavior to be just one of the three behaviors described above; or
— Restrict the choice of behaviors to a subset of the three, and allow GMRP registrations (or their absence) to determine the final choice; or
— Allow any one of the three behaviors to be adopted, in accordance with any registrations received via GMRP.

8.10.7 Allocation of VIDs to FIDs

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

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

a) The set of *VLAN Learning Constraints* that have been configured into the Bridge (by means of the management operations defined in Clause 12);
b) Any fixed mappings of VIDs to FIDs that may have been configured into the Bridge (by means of the management operations defined in Clause 12);
c) The *set of active VLANs* (i.e., those VLANs on whose behalf the Bridge may be called upon to forward frames). A VLAN is active if either of the following is true:
    1) The VLAN’s Member set (8.10.9) contains one Port that is in a forwarding state, and at least one other Port of the Bridge is both in a forwarding state and has Ingress Filtering (8.4.5) disabled;
    2) The VLAN’s Member set contains two or more Ports that are in a forwarding state.
d) The capabilities of the Bridge with respect to the number of FIDs that it can support, and the number of VIDs that can be allocated to each FID.

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

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

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

NOTE—In an SVL/IVL Bridge (3.28), a number of FIDs are supported, and one or more VID can be mapped to each FID. In an SVL Bridge (3.27), a single FID is supported, and all VIDs are mapped to that FID. In an IVL Bridge (3.9), a number of FIDs are supported, and only one VID can be mapped to each FID.

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

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

An MST Bridge shall ensure that the maximum supported numbers of FIDs and VLANs can be associated unambiguously. This requires either 1) a number of fixed VID to FID allocations at least equal to the maximum number of VLANs supported; or 2) one I Constraint entry per FID supported and one S Constraint entry per MSTI supported, or both. (8.10.7.1).

8.10.7.1 Fixed and dynamic VID to FID allocations

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

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

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

8.10.7.2 VLAN Learning Constraints

There are two types of VLAN Learning Constraint:

a) A Shared Learning Constraint (or S Constraint) asserts that Shared VLAN Learning shall occur between a pair of identified VLANs. S Constraints are of the form {A S B}, where A and B are VIDs. An S constraint is interpreted as meaning that Shared VLAN Learning shall occur between the VLANs identified by the pair of VIDs;
b) An Independent Learning Constraint (or I Constraint) asserts that a given VLAN is a member of a set of VLANs amongst which Independent VLAN Learning shall occur. I Constraints are of the form {A I N}, where A is a VID and N is an Independent Set Identifier. An I Constraint is interpreted as meaning that Independent VLAN Learning shall occur among the set of VLANs comprising VLAN A and all other VLANs identified in I Constraints that carry the same Independent Set Identifier, N.

A given VID may appear in any number (including zero) of S Constraints and/or I Constraints.

NOTE 1—S Constraints are

— *Symmetric*: e.g., {A S B} and {B S A} both express an identical constraint;
— *Transitive*: e.g., {A S B}, {B S C} implies the existence of a third constraint, {A S C};
— *Reflexive*: e.g., {A S A} is a valid S Constraint.

I Constraints are not

— *Symmetric*: e.g., {A I 1} and {1 I A} express different constraints;
— *Transitive*: e.g., ({A I 1}, {B I 1}, {B I 2}, {C I 2}) does not imply either {A I 2} or {C I 1}.

The allocation of VIDs to FIDs shall be such that, for all members of the set of active VLANs (8.10.7),

c) A given VID shall be allocated to exactly one FID;
d) If a given VID appears in an I Constraint, then it shall not be allocated to the same FID as any other VID that appears in an I Constraint with the same Independent Set Identifier;

e) If a given VID appears in an S Constraint (either explicit, or implied by the transitive nature of the specification), then it shall be allocated to the same FID as the other VID identified in the same S Constraint;

f) If a VID does not appear in any S or I Constraints, then the Bridge may allocate that VID to any FID of its choice.

NOTE 2—The intent is that the set of Learning Constraints is defined on a global basis; i.e., that all VLAN-aware Bridges are configured with the same set of constraints (although individual constraints may well be defined and distributed by different managers/administrators). Any Bridge therefore sees the complete picture in terms of the Learning Constraints that apply to all VLANs present in the Bridged LAN, regardless of whether they all apply to VLANs that are present in that particular Bridge. This standard provides the definition, in Clause 12, of managed objects and operations that model how individual constraints can be configured in a Bridge; however, the issue of how a distributed management system might ensure the consistent setting of constraints in all Bridges in a Bridged LAN is not addressed by this standard.

8.10.7.3 VLAN Learning Constraint inconsistencies and violations

The application of the rules specified in 8.10.7.2, coupled with any fixed allocations of VIDs to FIDs that may have been configured, can result in the Bridge detecting Learning Constraint inconsistencies and/or violations (i.e., can result in situations where there are inherent contradictions in the combined specification of the VLAN Learning Constraints and the fixed allocations, or the Bridge’s own limitations mean that it cannot meet the set of VLAN Learning Constraints that have been imposed upon it).

A Bridge detects a Learning Constraint inconsistency if

a) The VLAN Learning Constraints, coupled with any fixed VID to FID allocations, are such that, if any given pair of VLANs became members of the set of active VLANs (8.10.7), the result would be a simultaneous requirement for Independent VLAN Learning and for Shared VLAN Learning for those two VLANs. Such an inconsistency would require the Bridge to allocate that pair of VIDs both to the same FID and to different FIDs.

Learning Constraint inconsistencies are detected when a management operation (12.10.3) attempts to set a new Learning Constraint value, or to modify the fixed VID to FID allocations. If the new constraint or allocation that is the subject of the operation is inconsistent with those already configured in the Bridge, then the management operation shall not be performed and an error response shall be returned.

A Bridge detects a Learning Constraint violation if

b) The Bridge does not support the ability to map more than one VID to any given FID, and the VLAN Learning Constraints indicate that two or more members of the active set of VLANs require to be mapped to the same FID; or

c) The number of FIDs required in order to correctly configure the Bridge to meet the VLAN Learning Constraints and fixed VID to FID allocations for all members of the active set of VLANs exceeds the number of FIDs supported by the Bridge.

Learning Constraint violations are detected

d) When a VLAN that was hitherto not a member of the set of active VLANs (8.10.7) becomes active, either as a result of management action or as a result of the operation of GVRP, resulting in the Bridge no longer being able to support the defined set of constraints and/or fixed allocations for the set of active VLANs; or

e) When other management reconfiguration actions, such as defining a new Learning Constraint or fixed VID to FID allocation, results in the Bridge no longer being able to support the defined set of constraints and/or fixed allocations for the set of active VLANs.

On detection of a violation, the Bridge issues the Notify Learning Constraint Violation management notification (12.10.3.10), in order to alert any management stations to the existence of the violation. There is the potential for a single change in configuration to result in more than one VLAN whose constraints cannot be met; in such cases, multiple notifications are generated.

8.10.8 Querying the Filtering Database

If a frame is classified into a VLAN containing a given outbound Port in its member set (8.10.9), forwarding or filtering through that Port is determined by the control elements of filtering entries for the frame's destination MAC Address and for VLANs with the same VID or Filtering Identifier (FID, 8.10.7) as the frame's VLAN.

Each entry in the Filtering Database for a MAC Address comprises

a) A MAC Address specification;
b) A VID or, in the case of Dynamic Filtering Entries, an FID;
c) A Port Map, with a control element for each outbound Port.

For Dynamic Filtering Entries, the FID that corresponds to a given VID is determined as specified in 8.10.7.

For a given VID, a given individual MAC Address specification can be included in the Filtering Database in a Static Filtering Entry, a Dynamic Filtering Entry, both or neither. Table 8-5 combines Static Filtering Entry and Dynamic Filtering Entry information for an individual MAC Address to specify forwarding, or filtering, of a frame with that destination MAC Address and VID through an outbound Port.

NOTE 1—The use of FID in this table for Static Filtering Entries, and the text in parentheses in the headings, reflects the fact that, where more than one VID maps to a given FID, there may be more than one Static Filtering Entry that affects the forwarding decision for a given individual MAC Address. The effect of all Static Filtering Entries for that address, and for VIDs that correspond to that FID, is combined, such that, for a given outbound Port:
— IF <any static entry for any VIDs that map to that FID specifies Forwarding> THEN <result = Forwarding>
— ELSE IF <any static entry for any VIDs that map to that FID specifies Filtering> THEN <result = Filtering>
— ELSE <result = Use Dynamic Filtering Information>

Table 8-5—Combining Static and Dynamic Filtering Entries for an individual MAC Address

Filtering Information	**Control Elements in any Static Filtering Entry or Entries for this individual MAC Address, FID, and outbound Port specify:**				
	Forward (Any Static Filtering Entry for this Address/FID/Port specifies Forward)	**Filter (No Static Filtering Entry for this Address/FID/Port specifies Forward)**	**Use Dynamic Filtering Information (No Static Filtering Entry for this Address/FID/Port specifies Forward or Filter), or no Static Filtering Entry present. Dynamic Filtering Entry Control Element for this individual MAC Address, FID and outbound Port specifies:**		
			Forward	**Filter**	**No Dynamic Filtering Entry present**
Result	Forward	Filter	Forward	Filter	Forward

Table 8-6 specifies the result, Registered or Not Registered, of combining a Static Filtering Entry and a Group Registration Entry for the "All Group Addresses" address specification, and for the "All Unregistered Group Addresses" address specification for an outbound Port.

Table 8-6—Combining Static Filtering Entry and Group Registration Entry for “All Group Addresses” and “All Unregistered Group Addresses”

Filtering Information	Static Filtering Entry Control Element for this group MAC Address, VID, and outbound Port specifies:				
	Registration Fixed (Forward)	Registration Forbidden (Filter)	Use Group Registration Information, or no Static Filtering Entry present. Group Registration Entry Control Element for this group MAC Address, VID and outbound Port specifies:		
			Registered (Forward)	Not Registered (Filter)	No Group Registration Entry present
Result	Registered	Not Registered	Registered	Not Registered	Not Registered

Table 8-7 combines Static Filtering Entry and Group Registration Entry information for a specific group MAC Address with the Table 8-6 results for All Group Addresses and All Unregistered Group Addresses to specify forwarding, or filtering, of a frame with that destination group MAC Address through an outbound Port.

Table 8-7—Forwarding or Filtering for specific group MAC Addresses

				Static Filtering Entry Control Element for this group MAC Address, VID and outbound Port specifies:				
				Registration Fixed (Forward)	Registration Forbidden (Filter)	Use Group Registration Information, or no Static Filtering Entry present. Group Registration Entry Control Element for this group MAC Address, VID and outbound Port specifies:		
						Registered (Forward)	Not Registered (Filter)	No Group Registration Entry present
All Group Addresses control elements for this VID and Port specify (Table 8-6):	Not Registered	All Unregistered Group Addresses control elements for this VID and Port specify (Table 8-6):	Not Registered	Forward	Filter	Forward	Filter	Filter (Filter Unregistered Groups)
			Registered	Forward	Filter	Forward	Filter	Forward (Forward Unregistered Groups)
	Registered			Forward	Filter	Forward (Forward All Groups)	Forward (Forward All Groups)	Forward (Forward All Groups)

Where a given VID is allocated to the same FID as one or more other VIDs, it is an implementation option as to whether

d) The results shown in Table 8-7 directly determine the forwarding/filtering decision for a given VID and group MAC Address (i.e., the operation of the Bridge with respect to group MAC Addresses ignores the allocation of VIDs to FIDs); or

e) The results for a given MAC Address and VID are combined with the corresponding results for that MAC Address for each other VID that is allocated to the same FID, so that if the Table 8-7 result is Forward in any one VLAN that shares that FID, then frames for that group MAC Address will be forwarded for all VLANs that share that FID (i.e., the operation of the Bridge with respect to group MAC Addresses takes account of the allocation of VIDs to FIDs).

NOTE 2—In case d), the implementation effectively operates a single FDB per VLAN for group MAC Addresses. In case e), the implementation combines static and registered information for group MAC Addresses in accordance with the VID to FID allocations currently in force, in much the same manner as for individual MAC Addresses.

8.10.9 Determination of the member set and untagged set for a VLAN

The VLAN configuration information contained in the Filtering Database for a given VLAN may include a Static VLAN Registration Entry (8.10.2) and/or a Dynamic VLAN Registration Entry (8.10.5). This information defines, for that VLAN:

a) The *member set,* consisting of the set of Ports through which members of the VLAN can currently be reached;

b) The *untagged set,* consisting of the set of Ports through which, if frames destined for the VLAN are to be transmitted, they shall be transmitted without tag headers. For all other Ports (i.e., all Ports that are not members of the untagged set), if frames destined for the VLAN are to be transmitted, they shall be transmitted with tag headers.

NOTE 1—As the ingress rule checking function of the Forwarding Process (8.6.1) always associates a non-null VLAN ID with an incoming frame, all frames (including received frames that were priority-tagged and carried the null VLAN ID in their tag header) will be transmitted with or without a tag header in accordance with the membership of the untagged set for their VID.

For a given VID, the Filtering Database can contain a Static VLAN Registration Entry, a Dynamic VLAN Registration Entry, both or neither. Table 8-8 combines Static VLAN Registration Entry and Dynamic VLAN Registration Entry information for a VLAN and Port to give a result *member*, or *not member*, for the Port. The member set for a given VLAN consists of all Ports for which the result is member.

Table 8-8—Determination of whether a Port is in a VLAN's member set

Filtering Information	Static VLAN Registration Entry Control Element for this VID and Port specifies:				
	Registration Fixed	Registration Forbidden	Normal Registration, or no Static VLAN Registration Entry present. Dynamic VLAN Registration Entry Control Element for this VID and Port specifies:		
			Registered	Not Registered	No Dynamic VLAN Registration Entry present
Result	Member	Not member	Member	Not member	Not member

Membership of the untagged set for a given VLAN is derived from Static VLAN Registration Entry information contained in the Filtering Database as follows:

c) If there is no Static VLAN Registration Entry for the VLAN, then the untagged set is the empty set; otherwise,
d) The untagged set is equal to the set of Ports for which the Port Map in the Static VLAN Registration Entry indicates that frames are to be transmitted untagged.

The untagged set and the member set for a given VLAN are used by the Forwarding Process in applying the ingress rules (8.6.1) and the egress rules (8.6.4) for that VLAN.

The initial state of the Permanent Database contains a Static VLAN Registration Entry for the VLAN corresponding to the Default PVID (Table 9-2). The Port Map in this entry specifies Registration Fixed and forwarding untagged for all Ports of the Bridge. This entry may be modified or removed from the Permanent Database by means of the management operations defined in Clause 12 if the implementation supports these operations.

NOTE 2—This causes the default tagging state for the PVID to be untagged, and for all other VIDs to be tagged, unless otherwise configured; however, the management configuration mechanisms allow any VID (including the PVID) to be specified as VLAN-tagged or untagged on any Port. Under normal circumstances, the appropriate configuration for the PVID would be untagged on an access Port or a hybrid Port, and VLAN-tagged on a trunk Port (Annex D discusses the terms *access Port*, *hybrid Port*, and *trunk Port*).

8.10.10 Permanent Database

The Permanent Database provides fixed storage for a number of Static Filtering Entries and Static VLAN Registration Entries. The Filtering Database shall be initialized with the Filtering Database Entries contained in this fixed data store.

Entries may be added to and removed from the Permanent Database under explicit management control, using the management functionality defined in Clause 12. Changes to the contents of Static Filtering Entries or Static VLAN Registration Entries in the Permanent Database do not affect forwarding and filtering decisions taken by the Forwarding Process or the egress rules until such a time as the Filtering Database is re-initialized.

NOTE 1—This aspect of the Permanent Database can be viewed as providing a “boot image” for the Filtering Database, defining the contents of all initial entries, before any dynamic filtering information is added.

NOTE 2—Subclause 10.3.2.3 of IEEE Std 802.1D, 1998 Edition defines an initial state for the contents of the Permanent Database, required for the purposes of GMRP operation.

8.11 MST configuration information

In order to support multiple spanning trees, an MST Bridge has to be configured with an unambiguous assignment of VIDs to spanning trees. This is achieved by:

a) Ensuring that the allocation of VIDs to FIDs (8.10.7) is unambiguous; and
b) Ensuring that each FID supported by the Bridge is allocated to exactly one Spanning Tree.

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

The second requirement is met by means of the FID to MSTI Allocation Table (8.11.3).

The combination of the VID to FID allocations and the FID to MSTI allocations defines a mapping of VIDs to spanning trees, represented by the MST Configuration Table (8.11.1).

8.11.1 MST Configuration Table

The MST Configuration Table defines, for each VID, the MSTID of the spanning tree instance to which the VID is allocated.

The MST Configuration Table cannot be configured directly; configuration of the table occurs as a consequence of configuring the relationships between VIDs and FIDs (8.10.7), and between FIDs and spanning trees (8.11.3).

8.11.2 MST Configuration Identification

In order for two or more MST Bridges to be members of the same MST Region (3.29), it is necessary for those Bridges to be directly connected together (i.e., interconnected only by means of LAN segments, without intervening Bridges that are not members of the region), and for them to support the same MST Region configuration. Two MST Region configurations are considered to be the same if the correspondence between VIDs and spanning trees is identical in both configurations.

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

In order to ensure that adjacent MST Bridges are able to determine whether they are part of the same MST Region, the MST BPDU supports the communication of an MST Configuration Identifier (13.7).

NOTE 2—As the MST Configuration Identifier is smaller than the mapping information that it summarizes, there is a small but finite possibility that two MST Bridges will assume that they have the same MST Region Configuration when this is not actually the case. However, given the size of the identifier, this standard assumes that this possibility is sufficiently small that it can safely be ignored. Appropriate use of the Configuration Name and Revision Level portions of the identifier can remove the possibility of an accidental match between MST Configuration Identifiers that are derived from different configurations within a single administrative domain (see 13.7).

8.11.3 FID to MSTI Allocation Table

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

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

8.12 Spanning Tree Protocol Entity and GARP Protocol Entities

The Spanning Tree Protocol Entity operates the Spanning Tree Algorithm and associated protocol (STP, RSTP, or MSTP).

The Spanning Tree Protocol Entities of Bridges attached to a given individual LAN in a Bridged LAN communicate by exchanging Bridge Protocol Data Units (BPDUs).

Figure 8-6 illustrates the operation of the Spanning Tree Protocol Entity including the reception and transmission of frames containing BPDUs, the modification of the state information associated with individual Bridge Ports, and notification of the Filtering Database of changes in active topology.

A given MST bridge is not required to support all of the spanning trees that exist within the MST bridged LAN. That is, the number of spanning trees operated by the Spanning Tree Protocol Entity in a given bridge may be different from the number operated by that in another bridge. However, as a direct consequence of the conditions stated in 8.11.2, the number of instances of the Spanning Tree Protocol operated by a given MST Bridge is the same for all Bridges that are members of a given MST Region.

The GARP Protocol Entities operate the Algorithms and Protocols associated with the GARP Applications supported by the Bridge, and consist of the set of GARP Participants for those GARP Applications (Clause 10 and 12.3 of IEEE Std 802.1D, 1998 Edition).

The GARP Protocol Entities of Bridges attached to a given individual LAN in a Bridged LAN communicate by exchanging GARP Protocol Data Units (GARP PDUs).

Figure 8-7 illustrates the operation of a GARP Protocol Entity including the reception and transmission of frames containing GARP PDUs, the use of control information contained in the Filtering Database, and notification of the Filtering Database of changes in filtering information.

8.13 Bridge Management

Remote management facilities may be provided by the Bridge. Bridge Management is modeled as being performed by means of the Bridge Management Entity. The facilities provided by Bridge Management, and the operations that support these facilities, are specified in Clause 12.

Bridge Management protocols use the MAC Service provided by the Bridged LAN.

8.14 Addressing

All MAC Entities communicating across a Bridged LAN shall use 48-bit addresses. These may be Universally Administered Addresses, Locally Administered Addresses, or a combination of both.

8.14.1 End stations

Frames transmitted between end stations using the MAC Service provided by a Bridged LAN carry the MAC Address of the source and destination end stations in the source and destination address fields of the frames, respectively. The address, or other means of identification, of a Bridge is not carried in frames transmitted between end stations for the purpose of frame relay in the Bridged LAN.

The broadcast address and other group MAC Addresses apply to the use of the MAC Service provided by a Bridged LAN as a whole. In the absence of explicit filters configured via management as Static Filtering Entries, or via GMRP as Group Registration Entries (8.10 and Clause 12 of this standard, and Clause 10 of IEEE Std 802.1D, 1998 Edition), frames with such destination addresses are relayed throughout the Bridged LAN.

8.14.2 Bridge Ports

The individual MAC Entity associated with each Bridge Port shall have a separate individual MAC Address. This address is used for any MAC procedures required by the particular MAC method employed.

Frames that are received from the LAN to which a Port is attached and that carry a MAC Address for the Port in the destination address field are submitted to the MAC Service User (LLC) exactly as for an end station.

8.14.3 Bridge Protocol Entities and GARP Protocol Entities

Bridge Protocol Entities only receive and transmit BPDUs. These are only received and transmitted from other Bridge Protocol Entities (or where two Bridge Ports are connected to the same LAN, to and from themselves).

GARP Protocol Entities only receive and transmit GARP PDUs (12.11 of IEEE Std 802.1D, 1998 Edition) that are formatted according to the requirements of the GARP Applications they support. These are only received and transmitted from other GARP Protocol Entities.

A Spanning Tree Protocol Entity or a GARP Protocol Entity uses the DL_UNITDATA.request primitive (see ISO/IEC 8802-2) provided by the individual LLC Entities associated with each active Bridge Port to transmit BPDUs or GARP PDUs. Each PDU is transmitted on one selected Bridge Port. PDUs are received through corresponding DL_UNITDATA.indication primitives. The source_address and destination_address parameters of the DL_UNITDATA.request primitive shall both denote the standard LLC address assigned to the Bridge Spanning Tree Protocol. This identifies the Spanning Tree Protocol Entity and the GARP Protocol Entity among other users of LLC.

Each DL_UNITDATA.request primitive gives rise to the transmission of an LLC UI command PDU, which conveys the BPDU or GARP PDU in its information field. The source and destination LLC address fields are set to the values supplied in the request primitive.

The value assigned to the Bridge Spanning Tree Protocol LLC address is given in Table 8-9.[10]

Table 8-9—Standard LLC address assignment

Assignment	Value
Bridge spanning tree protocol	01000010

Code Representation: The least significant bit of the value shown is the right-most. The bits increase in significance from right to left. It should be noted that the code representation used here has been chosen in order to maintain consistency with the representation used elsewhere in this standard; however, it differs from the representation used in ISO/IEC 11802-1: 1997.

IEEE Std 802.1D, 1998 Edition defines a Protocol Identifier field, present in all BPDUs (Clause 9 of IEEE Std 802.1D, 1998 Edition) and GARP PDUs (12.11 of IEEE Std 802.1D, 1998 Edition), which serves to identify different protocols supported by Bridge Protocol Entities and GARP Protocol Entities, within the scope of the LLC address assignment. This standard specifies a single value of the Protocol Identifier, defined in Clause 9 of IEEE Std 802.1D, 1998 Edition, for use in BPDUs. This value serves to identify BPDUs exchanged between Bridge Protocol Entities operating the Spanning Tree Algorithm and Protocol specified in Clause 8 of IEEE Std 802.1D, 1998 Edition. A second value of this protocol identifier for use in GARP PDUs is defined in 12.11 of IEEE Std 802.1D, 1998 Edition. This value serves to identify GARP PDUs exchanged between GARP Participants operating the GARP protocol specified in Clause 12 of IEEE Std 802.1D, 1998 Edition. Further values of this field are reserved for future standardization.

A Spanning Tree Protocol Entity or GARP Protocol Entity that receives a BPDU or a GARP PDU with an unknown Protocol Identifier shall discard that PDU.

[10] ISO/IEC TR 11802-1: 1997, Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Technical reports and guidelines—Part 1: The structure and coding of Logical Link Control addresses in Local Area Networks, contains the full list of standard LLC address assignments, and documents the criteria for assignment.

A Spanning Tree Protocol Entity that operates the Spanning Tree Algorithm and Protocol specified in Clause 8 of IEEE Std 802.1D, 1998 Edition always transmits BPDUs addressed to all other Bridge Protocol Entities attached to the LAN on which the frame containing the BPDU is transmitted. A group address shall be used in the destination address field to address this group of Entities. This group address shall be configured in the Permanent Database (8.14.6) in order to confine BPDUs to the individual LAN on which they are transmitted.

A 48-bit universally administered address, known as the Bridge Group Address, has been assigned for this purpose. Its value is specified inTable 8-10. SST Bridges that use 48-bit universally administered addresses shall use this address as the destination address of all MAC frames conveying BPDUs in an SST environment. MST Bridges shall use this address as the destination address of all MAC frames conveying BPDUs for the Multiple Spanning Tree Protocol (MSTP, clause 13) and the Internal Sub-Tree Protocol (ISTP, 14.3).

Table 8-10—Reserved addresses

Assignment	Value
Bridge Group Address	01-80-C2-00-00-00
IEEE Std 802.3, 1998 Edition, Full Duplex PAUSE operation	01-80-C2-00-00-01
IEEE Std 802.3ad Slow_Protocols_Multicast address	01-80-C2-00-00-02
IEEE Std 802.1X PAE address	01-80-C2-00-00-03
Reserved for future standardization	01-80-C2-00-00-04
Reserved for future standardization	01-80-C2-00-00-05
Reserved for future standardization	01-80-C2-00-00-06
Reserved for future standardization	01-80-C2-00-00-07
Reserved for future standardization	01-80-C2-00-00-08
Reserved for future standardization	01-80-C2-00-00-09
Reserved for future standardization	01-80-C2-00-00-0A
Reserved for future standardization	01-80-C2-00-00-0B
Reserved for future standardization	01-80-C2-00-00-0C
Reserved for future standardization	01-80-C2-00-00-0D
Reserved for future standardization	01-80-C2-00-00-0E
Reserved for future standardization	01-80-C2-00-00-0F

A GARP Protocol Entity that

a) Operates the GARP protocol specified in Clause 12 of IEEE Std 802.1D, 1998 Edition; and
b) Supports a given GARP Application,

always transmits GARP PDUs addressed to all other GARP Protocol Entities that

c) Implement the same GARP Application; and
d) Are attached to the LAN on which the frame containing the GARP PDU is transmitted.

A group MAC Address, specific to the GARP Application concerned, shall be used as the destination MAC Address field to address this group of GARP Protocol Entities. A set of 48-bit Universal Addresses, known as GARP Application addresses, have been assigned for that purpose. The values of the GARP Application addresses are defined in Table 12-1 of IEEE Std 802.1D, 1998 Edition. These group MAC Addresses are reserved for assignment to standard protocols, according to the criteria for such assignments (Clause 5.5 of ISO/IEC TR 11802-2).

NOTE—Table 11-1 allocates a group MAC Address for use by the GVRP application; however, the value allocated in that table is one of the GARP Application addresses reserved by Table 12-1 of IEEE Std 802.1D, 1998 Edition.

In Bridges that do not support any GARP applications, the set of GARP Application addresses should not be configured in the Filtering Database (8.10) or the Permanent Database (8.10.10). In Bridges that support one or more GARP applications, the set of GARP Application addresses should be configured as Static Filtering Entries in the Filtering Database (8.10.1) and Permanent Database (8.10.10) as follows:

e) GARP Application addresses assigned to GARP Applications that are supported by the Bridge should be configured in order to confine GARP PDUs for that GARP Application to the individual LAN on which they are transmitted;

f) GARP Application addresses assigned to GARP Applications that are not supported by the Bridge should not be configured in the Filtering Database or Permanent Database.

The source address field of MAC frames conveying BPDUs or GARP PDUs contains the individual MAC Address for the Bridge Port through which the PDU is transmitted (8.14.2).

8.14.4 Bridge Management Entities

Bridge Management Entities transmit and receive protocol data units using the Service provided by the individual LLC Entities associated with each Bridge Port. Each of these in turn uses the MAC Service, which is provided by the individual MAC Entities associated with that Port and supported by the Bridged LAN as a whole.

As a user of the MAC Service provided by a Bridged LAN, the Bridge Management Entity may be attached to any point in the Bridged LAN. Frames addressed to the Bridge Management Entity will be relayed by Bridges if necessary to reach the LAN to which it is attached.

In order to ensure that received frames are not duplicated, the basic requirement in a single LAN or a Bridged LAN that a unique address be associated with each point of attachment shall be met.

A Bridge Management Entity for a specific Bridge is addressed by one or more individual MAC Addresses in conjunction with the higher layer protocol identifier and addressing information. It may share one or more points of attachment to the Bridged LAN with the Ports of the Bridge with which it is associated. It is recommended that it make use of the MAC Service provided by all the MAC Entities associated with each Bridge Port, i.e., that it be reachable through each Bridge Port using frames carrying the individual MAC Address of that Port in the destination address field.

This standard specifies a standard group MAC Address for public use which serves to convey management requests to the Bridge Management Entities associated with all Bridge Ports attached to a Bridged LAN. A management request that is conveyed in a MAC frame carrying this address value in the destination address field will generally elicit multiple responses from a single Bridge. This address is known as the All LANs Bridge Management Group Address and takes the value specified in Table 8-11.

Table 8-11—Addressing bridge management

Assignment	Value
All LANs Bridge Management Group Address	01-80-C2-00-00-10

8.14.5 Unique identification of a Bridge

A unique 48-bit Universally Administered MAC Address, termed the Bridge Address, shall be assigned to each Bridge. The Bridge Address may be the individual MAC Address of a Bridge Port, in which case use of the address of the lowest numbered Bridge Port (Port 1) is recommended.

NOTE—The Spanning Tree Protocol (Clause 8 of IEEE Std 802.1D, 1998 Edition) requires that a single unique identifier be associated with each Bridge. That identifier is derived from the Bridge Address as specified in 8.5.1.3, 8.5.3.7, and 9.2.5 of IEEE Std 802.1D, 1998 Edition.

8.14.6 Reserved addresses

Frames containing any of the group MAC Addresses specified in Table 8-1 in their destination address field shall not be relayed by the Bridge. They shall be configured in the Permanent Database. Management shall not provide the capability to modify or remove these entries from the Permanent or the Filtering Databases. These group MAC Addresses are reserved for assignment to standard protocols, according to the criteria for such assignments (Clause 5.5 of ISO/IEC TR 11802-2).

8.14.7 Points of attachment and connectivity for Higher Layer Entities

Higher Layer Entities such as the Spanning Tree Protocol Entity and GARP Protocol Entity (8.12), and Bridge Management (8.13) are modeled as being connected directly to the Bridged LAN via one or more points of attachment. From the point of view of their attachment to the Bridged LAN, Higher Layer Entities associated with a Bridge can be regarded as if they are distinct end stations, directly connected to one or more of the LAN segments served by the Bridge Ports, in the same way as any other end station is connected to the Bridged LAN. In practice, the Higher Layer Entities will, in many cases, share the same physical points of attachment used by the relay function of the Bridge, as stated in 8.14; however, from the point of view of the transmission and reception of frames by these functions, the behavior is the same as if they were contained in logically separate end stations with points of attachment “outside” the Port(s) with which they are associated. Figure 8-10 is functionally equivalent to Figure 6-1, but illustrates this logical separation between the points of attachment used by the Higher Layer Entities and points of attachment used by the MAC Relay Entity.

Higher Layer Entities fall into two distinct categories:

a) Those entities, such as the Bridge Management Entity, that require only a single point of attachment to the Bridged LAN;
b) Those entities, such as Bridge Protocol Entities and GARP Participants, that require a point of attachment per Port of the Bridge.

The fundamental distinction between these two categories is that for the latter, it is essential for the operation of the entity concerned that it is able to associate received frames with the LAN segment on which those frames were originally seen by the Bridge, and that it is able to transmit frames to peer entities that are connected directly to that LAN segment. It is therefore essential that

c) It does not receive frames via a point of attachment associated with one Port that have been relayed by the Bridge from other Ports; and
d) Frames that it transmits via one point of attachment are not relayed by the Bridge to any other Ports.

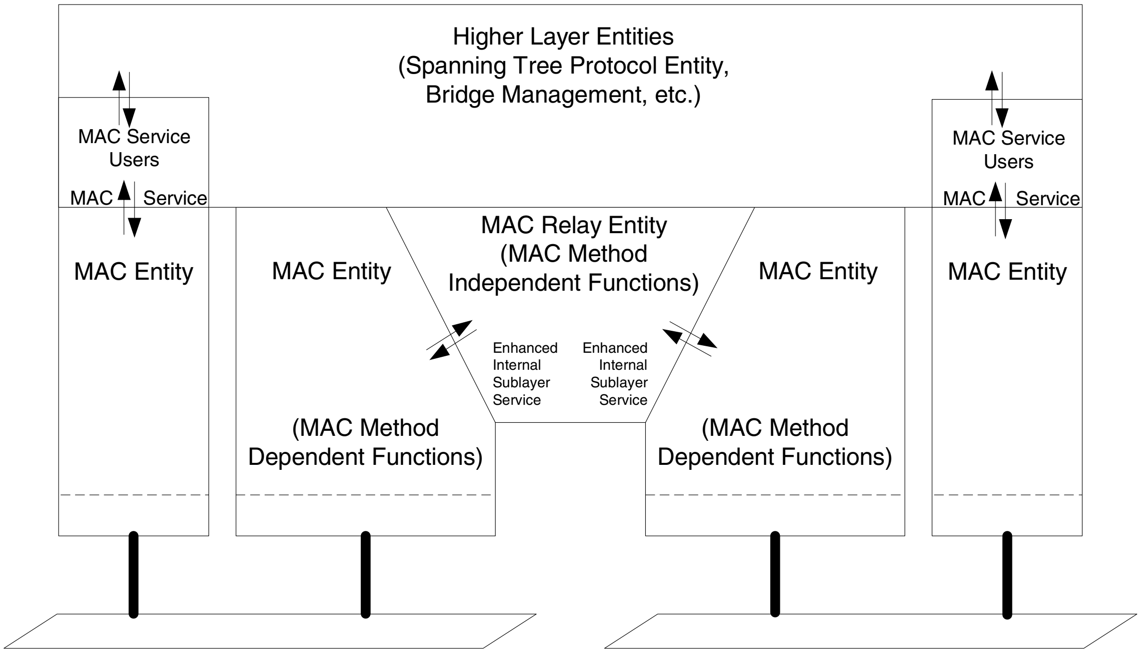


Figure 8-10—Logical separation of points of attachment used by Higher Layer Entities and the MAC Relay Entity

For this reason, the MAC Addresses used to reach entities of this type are permanently configured in the Filtering Database in order to prevent the Bridge from relaying such frames received via any Port to any other Port of the Bridge, as defined in 8.14.3 and 8.14.6.

NOTE 1—The MAC Addresses used to address such entities are generally group MAC Addresses.

The MAC Relay Entity forwards a frame received on one Port through the other Port(s) of the Bridge, subject to the following control information permitting such forwarding to take place:

e) The Port state information (8.4) associated with the Port on which the frame was received;

f) The information held in the Filtering Database (8.10);

g) The Port state information (8.4) associated with the Port(s) on which the frame is potentially to be transmitted.

This is illustrated in Figure 8-11, where the control information represented by the Port state and Filtering Database information is represented as a series of switches (shown in the open, disconnected state) inserted in the forwarding path provided by the MAC Relay Entity. For the Bridge to forward a given frame between two Ports, all three switches must be in the closed state. This figure also illustrates that the controls placed in the forwarding path have no effect upon the ability of a Higher Layer Entity to transmit and receive frames directly onto a given LAN segment via the point of attachment to that segment (e.g., from entity A to segment A); they only affect the path taken by any indirect transmission/reception (e.g., from entity A to segment B).

Figure 8-12 illustrates the state of the forwarding path with respect to frames destined for Higher Layer Entities that require per-Port points of attachment. The fact that the Filtering Databases in all Bridges are permanently configured to prevent relay of frames addressed to these entities means that they can receive frames only via their direct points of attachment (i.e., from segment A to entity A, and from segment B to entity B), regardless of the Port states.

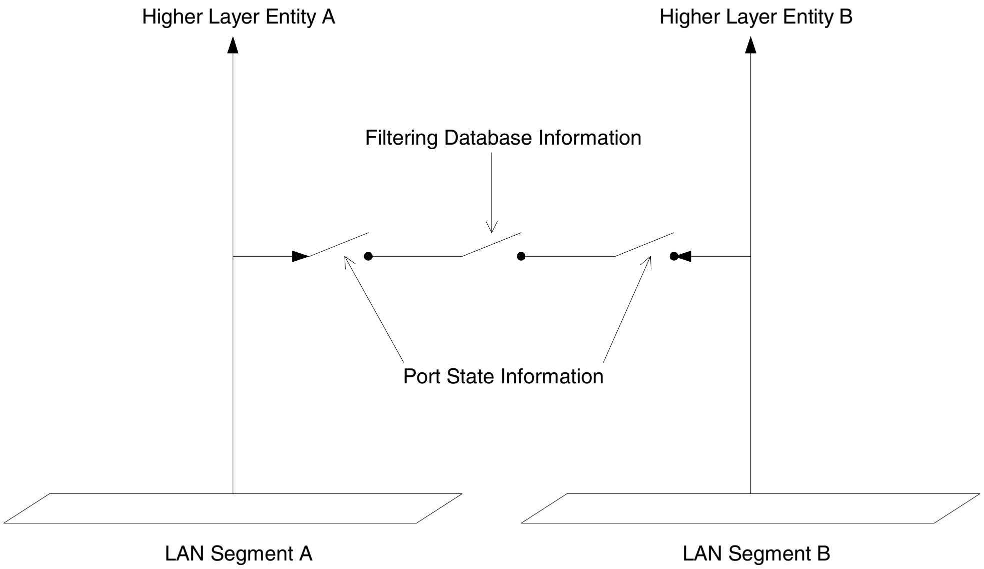


Figure 8-11—Effect of control information on the forwarding path

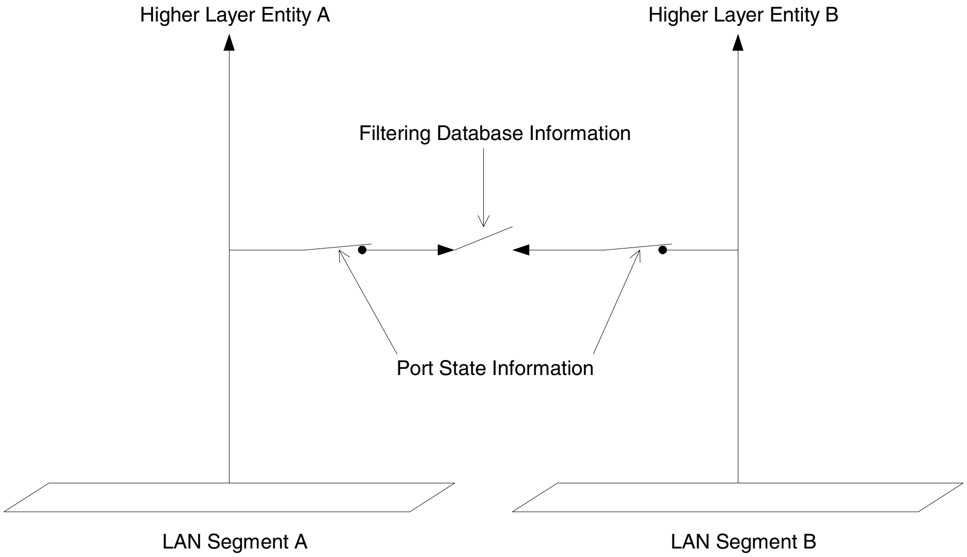


Figure 8-12—Per-Port points of attachment

Figure 8-13 illustrates the state of the forwarding path with respect to frames destined for a Higher Layer Entity that requires only a single point of attachment, for the case where the Port states and Filtering Database states permit relay of frames. Frames destined for the Higher Layer Entity that originate on LAN segment B are relayed by the Bridge, and are both received by the entity and transmitted on LAN segment A.

Figure 8-14 illustrates the state of the forwarding path with respect to frames destined for a Higher Layer Entity that requires only a single point of attachment, for the case where one of the Port states does not permit relay. Frames destined for the Higher Layer Entity that originate on LAN segment A are received by the entity; however, frames that originate on LAN segment B are not relayed by the Bridge, and can therefore only be received by the entity if there is some other forwarding path provided by other components of the Bridged LAN between segments A and B.

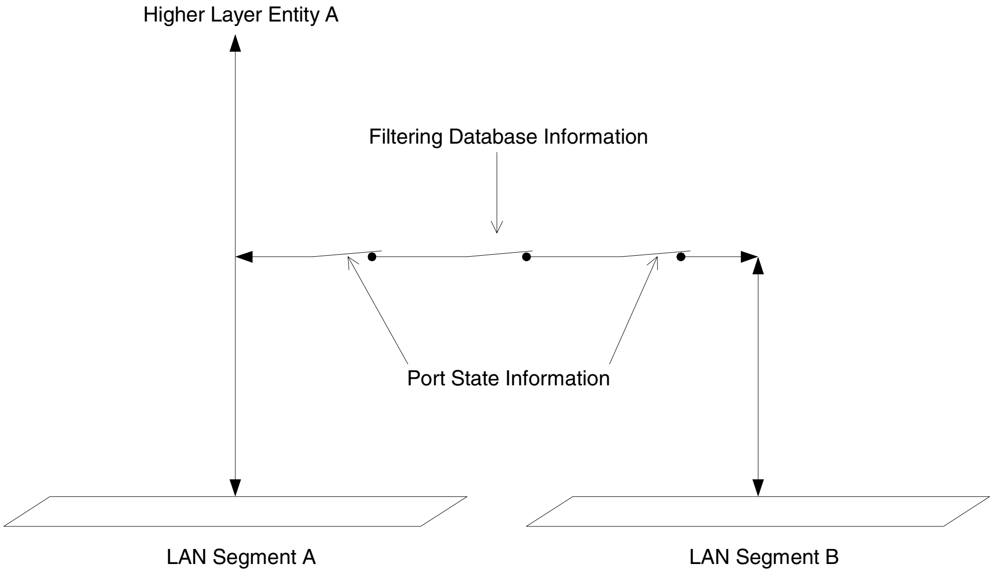


Figure 8-13—Single point of attachment—relay permitted

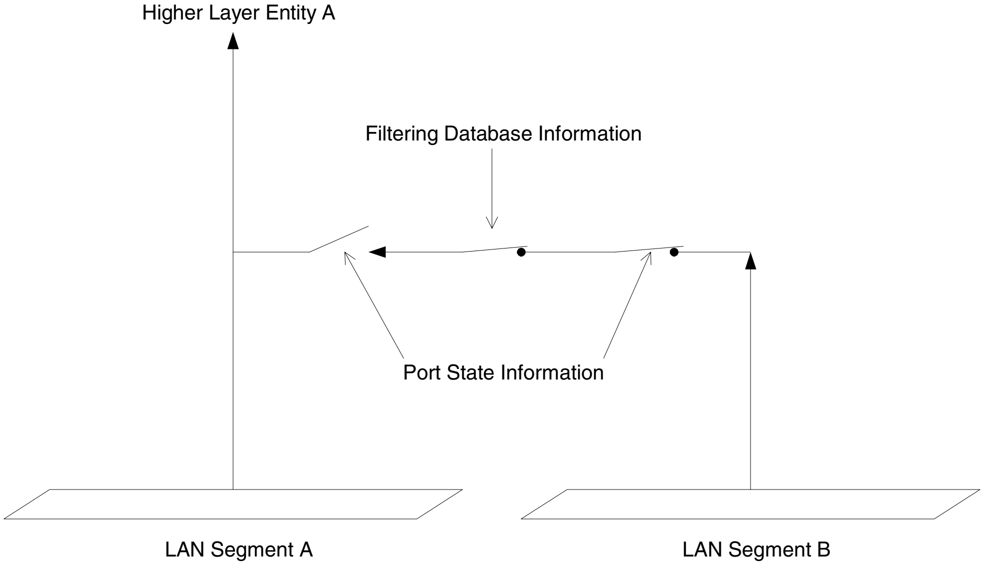


Figure 8-14—Single point of attachment—relay not permitted

NOTE 2—If the Port state shown in Figure 8-14 occurs as a result of the normal operation of the Spanning Tree (as opposed to being a result of equipment failure, or administrative control of Port state information), then such a path will exist, either via another Port of this Bridge (not shown in the diagram) connected to segment A, or via one or more Bridges providing a path between segments A and B. If there is no active Spanning Tree path from segment B to segment A, then the Bridged LAN has partitioned into two separate Bridged LANs, one on either side of this Port, and the Higher Layer Entity shown is only reachable via segment A.

In VLAN-aware Bridges, two more switches appear in the forwarding path, corresponding to the actions taken by the Forwarding Process (8.6.1 and 8.6.4) in applying the ingress and egress rules (8.6.1 and 8.6.4), as illustrated in Figure 8-15.

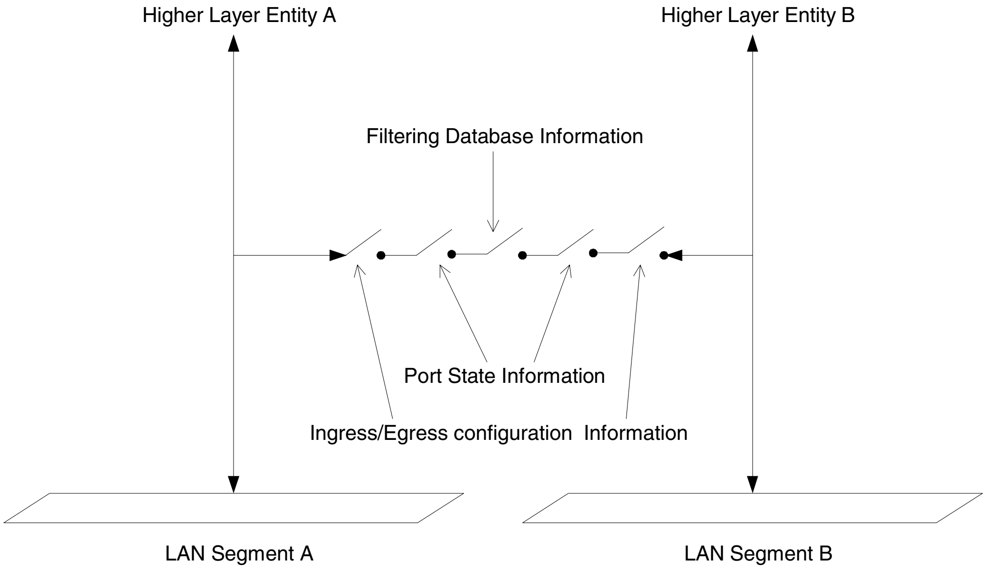


Figure 8-15—Ingress/egress control information in the forwarding path

As with Port state information, the configuration of the ingress and egress rules does not affect the reception of frames received on the same LAN segment as a Higher Layer Entity's point of attachment. For example, the reception of a frame by Higher Layer Entity A that was transmitted on LAN Segment A is unaffected by the ingress or egress configuration of either Port. However, for Higher Layer Entities that require only a single point of attachment, the ingress and egress configuration affects the forwarding path. For example, frames destined for Higher Layer Entity A that are transmitted on LAN Segment B would be subjected to the ingress rules that apply to Port B and the egress rules that apply to Port A.

The decision as to whether frames transmitted by Higher Layer Entities are VLAN-tagged or untagged depends upon the Higher Layer Entity concerned, and the connectivity that it requires

h) Spanning Tree BPDUs transmitted by the Spanning Tree Protocol Entity are not forwarded by Bridges, and must be visible to all other Spanning Tree Protocol Entities attached to the same LAN segment. Such frames shall be transmitted untagged;

NOTE 3—Any BPDUs or GVRP PDUs that carry a tag header are not recognized as well-formed BPDUs or GVRP PDUs and are not forwarded by the Bridge.

i) The definition of the GVRP application (11.2.3) calls for all GVRP frames to be transmitted untagged for similar reasons;

j) The definition of the GMRP application (Clause 10) calls for all GMRP frames originating from VLAN-aware devices to be transmitted VLAN-tagged, in order for the VID in the tag to be used to identify the VLAN context in which the registration applies;

k) It may be necessary for PDUs transmitted for Bridge Management (8.13) to be VLAN-tagged in order to achieve the necessary connectivity for management in a VLAN Bridged LAN. In order to access a Bridge Management entity located in a region of the network that is served only by a given set of VLANs, it may be necessary to communicate with that entity using frames VLAN-tagged with one of the VIDs concerned, unless one of those VIDs also happens to be the PVID (or a member of the VID Set that matches the appropriate Protocol) for the Port serving the management station.

9. Tagged frame format

Tagging of frames is performed for the following purposes:

a) To allow user_priority information to be added to frames carried on IEEE 802 LAN MAC types that have no inherent ability to signal priority information at the MAC protocol level;
b) To allow a frame to carry a VID;
c) To allow the frame to indicate the format of MAC Address information carried in MAC user data;
d) To allow VLANs to be supported across different MAC types.

This clause describes the tag format used for tagging frames, as follows:

e) Subclause 9.1 gives an overview of tagging;
f) Subclause 9.2 defines the data representations that are used in the descriptions of the tag field formats;
g) Subclause 9.3 describes the structure of the tag header.

Further analysis of the frame formats, the format translations that can occur when frames are tagged or untagged when relayed between different MAC methods, and a description of the tagging/untagging procedure can be found in Annex C.

The description of the tagged frame structure, both here and in Annex C, is based on two generic frame formats:

h) The frame format used in IEEE Std 802.3 MACs, and which is used with minor variations in other MACs where the native Link Layer protocol identification mechanism is based on a choice between the Type interpretation and Length interpretation of the Length/Type field. The Type interpretation is used where a Type value provides the protocol identification; the Length interpretation is used where LLC addressing provides the protocol identification. LAN MAC methods that make use of this frame format are referred to in this standard as 802.3/Ethernet MAC methods;
i) The frame format used in IEEE Std 802.5 and FDDI MACs, and used with minor variations in other MACs where the native Link Layer protocol identification mechanism is based on LLC addressing, and where the frame may also be able to carry source-routing information. LAN MAC methods that make use of this frame format are referred to in this standard as Token Ring/FDDI MAC methods.

For MACs other than 802.3, 8802-5, and FDDI, the approach used is to apply these frame formats, with appropriate modification to the overall frame structure, as appropriate to the MAC concerned. For example:

j) MACs such as 8802-4 and 8802-6 that use LLC as the native Link Layer protocol identification would adopt format i). If they do not provide native support for source routing, the variant of this format that is used in transparent FDDI LANs would be used;
k) MACs such as 8802-12 that can support compatibility with 802.3 and 8802-5 MACs would adopt either format h) or format i), depending upon which compatibility mode was in operation.

9.1 Overview

Tagging a frame requires

a) The addition of a tag header to the frame. This header is inserted immediately following the destination MAC Address and source MAC Address (and routing, if present) fields of the frame to be transmitted;

b) If the source and destination MAC methods differ, tagging the frame may involve translation or encapsulation of the remainder of the frame, as specified in IEEE Std 802.1H, IETF RFC 1042, and IETF RFC 1390;

c) Recomputation of the Frame Check Sequence (FCS).

When relaying a tagged frame between 802.3/Ethernet MACs, a Bridge may adjust the PAD field such that the minimum size of a transmitted tagged frame is 68 octets (7.2).

The tag header carries the following information:

d) The Tag Protocol Identifier (TPID) appropriate to the MAC method concerned, as described in 9.3.1. This protocol identifier identifies the frame as a tagged frame, conforming to the tagging format described in this standard.

e) Tag Control Information (TCI) as described in 9.3.2. The TCI consists of the following elements:

   1) User_priority, as described in 9.3.2.1. This field allows the tagged frame to carry user_priority information across Bridged LANs in which individual LAN segments may be unable to signal priority information (e.g., IEEE 802.3/Ethernet segments).

   2) Canonical Format Indicator (CFI), as described in 9.3.2.2. This field is used

      i) In Token Ring/source-routed FDDI MAC methods, to signal the bit order of address information carried in the encapsulated frame; and

      ii) In IEEE 802.3/Ethernet and transparent FDDI MAC methods, to signal the presence or absence of an Embedded Source-Routing Information Field (E-RIF), and, in combination with the Non-canonical Format Indicator (NCFI) carried in the RIF, to signal the bit order of address information carried in the encapsulated frame.

   NOTE 1—The meaning of Canonical format as applied to MAC Addresses, and the implications of the format of addresses on the requirements for frame translation, are discussed in Annex F.

   3) VLAN Identifier (VID), as described in 9.3.2.3. This field uniquely identifies the VLAN to which the frame belongs.

f) In IEEE 802.3/Ethernet and FDDI MAC methods, an E-RIF is included, if required by the state of the CFI flag in the TCI. If present, in addition to providing the ability to carry source-routing information, this field includes a further flag, the NCFI, that signals the bit order of address information carried in the encapsulated frame. The structure of this field, as used in this context, is described in 9.3.3.

NOTE 2—The ability of the tag header to carry embedded source-routing information using 802.3/Ethernet and FDDI MAC methods does not imply a requirement on the part of a pure 802.3/Ethernet Bridge or a transparent FDDI Bridge to support source routing. This capability is provided simply to allow traffic that originates in, and is destined for, a source-routed environment to transit as VLAN-tagged traffic across a non-source-routed environment. “Tunnelling” of source-routed frames across transparent media in this manner is still required to follow the rules for source routing as defined in IEEE Std 802.1D, 1998 Edition (see 1.3, 5.4); in particular, Bridges that support only transparent operation are not permitted to forward any frames received that have the RII bit set in the source MAC Address field. Any use of the capability of using the E-RIF to carry real source-routing information across transparent LANs can therefore only be made by Bridges and/or end stations that support source routing. Once RIF information has been encapsulated in this way, transparent Bridges can treat the frames as transparent frames, and forward/filter them accordingly. This standard does not specify any forwarding decisions based on the E-RIF.

The structure of the tagged frame allows the following types of information to be identified and carried in tagged frames across all MAC methods:

g) Ethernet Type-encoded (E) and LLC-encoded (L) information (see 3.1, 3.2);

NOTE 3—The distinction between E and L is not represented in the tag header itself, but is identifiable by examination of the data carried in the tagged frame; in 802.3/Ethernet MAC methods, by examining the value in the Length/Type field, and in Token Ring/FDDI MAC methods, by the presence/absence of the SNAP-based protocol identifiers used in the encapsulation formats described in IEEE Std 802.1H, IETF RFC 1042, and IETF RFC 1390.

h) Frames in which any MAC Addresses embedded in the MAC data are carried in Canonical (C) or Non-canonical (N) format;
i) Source-routed (R) and transparent (T) frames.

NOTE 4—These abbreviations are used here and in Annex C to refer to different types of data frame. For example, a frame carrying LLC-encoded information, Non-canonical embedded addresses and source-routing information, is abbreviated to L-N-R; a frame carrying Ethernet Type-encoded information with Canonical addresses and no source-routing information would be E-C-T.

Relaying a tagged frame requires

j) If the frame formats used on the source and destination MAC methods differ, translation of the tag header to the format appropriate for the destination MAC method;
k) If the source and destination MAC methods differ, relaying the frame may require translation of the remainder of the frame, as defined in IEEE Std 802.1H, IETF RFC 1042, and IETF RFC 1390;
l) Inclusion/adjustment of the PAD field, if necessary, where the destination MAC method is 802.3/ Ethernet (7.2);
m) Re-computation of the Frame Check Sequence (FCS) if necessary.

Untagging a tagged frame requires

n) The removal of the tag header, retaining the RIF in the appropriate position in the final untagged frame, or restoring an embedded RIF to the appropriate position in the final untagged frame, if necessary;
o) If the frame formats used on the source and destination MAC methods differ, untagging the frame may require translation of the remainder of the frame, as defined in IEEE Std 802.1H, IETF RFC 1042, and IETF RFC 1390;
p) Adjustment of the PAD field, if necessary, where the destination MAC method is 802.3/Ethernet (7.2);
q) Re-computation of the Frame Check Sequence (FCS).

The frame translations defined in IEEE Std 802.1H, IETF RFC 1042, and IETF RFC 1390 are applied, as necessary, to all frames carrying Ethernet Type-encoded information relayed by VLAN-aware Bridges. Use of the CFI flag in the tagged frame allows

r) The format of embedded MAC Address information to be signalled end-to-end across a VLAN without the need for MAC Address format translation by VLAN-aware Bridges while the frame is in tagged format, regardless of the MAC methods involved in carrying the tagged frame from source to destination;

NOTE 5—In other words, from the point of view of IEEE Std 802.1H, IETF RFC 1042, and IETF RFC 1390 translation, the representation of Ethernet Type-encoded information is always as would be expected for the underlying MAC method, and in tagged frames, the format of embedded address information is always as indicated by the CFI/NCFI.

s) Source-routing information to be carried end-to-end across a VLAN, regardless of the MAC methods involved in carrying the tagged frame from source to destination.

The primary purpose in allowing the distinction between Canonical and Non-canonical information to be represented in the tagged frame is to permit such information to be carried across a VLAN without the need for VLAN-aware Bridges to translate the format of embedded MAC Addresses en route; however, Bridges that untag frames may still need to take the address format into consideration, and perform the appropriate translation if necessary, and if the Bridge supports such translation. For example, in order to successfully untag an L-N-T frame for transmission onto an 802.3/Ethernet segment, it would be necessary to convert any embedded MAC Addresses to Canonical format in order for that frame to be meaningful to the end stations on that segment.

The ability to support translation of embedded MAC Addresses between Canonical and Non-canonical formats (and vice versa) when transmitting an untagged frame is not required by this standard. In Bridges that do not support such translation capability on a given outbound Port, frames that may require such translation before being forwarded as untagged frames on that Port shall be discarded.

NOTE 6—In particular, this means that a Bridge that supports only one MAC type on all Ports is not required to support MAC Address translation when untagging a frame that originated on a different MAC type.

Tagging of frames occurs when an untagged frame is relayed by a Bridge onto a LAN segment for which that frame is required by the egress rules (8.8) to be transmitted in tagged format.

Untagging of frames occurs when a tagged frame is relayed by a Bridge onto a LAN segment for which that frame is required by the egress rules (8.8) to be transmitted in untagged format.

9.2 Transmission and representation of octets

In this clause, octets in a PDU (or a field of a PDU) are numbered starting from 1 and increasing in the order in which they are put into a MAC Service Data Unit (MSDU).

The bits in an octet are numbered from 1 to 8, where 1 is the least significant bit.

Where consecutive octets are used to represent a binary number, the lower octet number carries the most significant value.

Where the value of a field is represented in hexadecimal notation, as a sequence of two-digit hexadecimal values separated by hyphens (e.g., A1-5B-03), the leftmost hexadecimal value (A1 in this example) appears in the lowest numbered octet of the field and the rightmost hexadecimal value (03 in this example) appears in the highest numbered octet of the field.

When the terms *set* and *reset* are used in the text to indicate the values of single-bit fields, *set* is encoded as a binary 1 and *reset* as a binary 0 (zero).

When the encoding of a PDU (or a field within a PDU) is represented using a diagram, the following representations are used:

a) Octets are shown with the lowest numbered octet nearest the top of the page, the octet numbering increasing from the top to bottom; or
b) Octets are shown with the lowest numbered octet nearest the left of the page, the octet numbering increasing from left to right;
c) Within an octet, bits are shown with bit 8 to the left and bit 1 to the right.

9.3 Structure of the tag header

The tag header consists of the following components:

a) The Tag Protocol Identifier (TPID) as described in 9.3.1;
b) The Tag Control Information (TCI) as described in 9.3.2;

c) In 802.3/Ethernet and non-Source-Rrouted FDDI frames (i.e., FDDI frames in which the RII bit is reset), the E-RIF, if required by the state of the CFI.

There are three forms of the tag header, depending upon the type of encoding used for the TPID and the underlying MAC type. The overall structure of the three forms of header is illustrated in Figure 9-1.

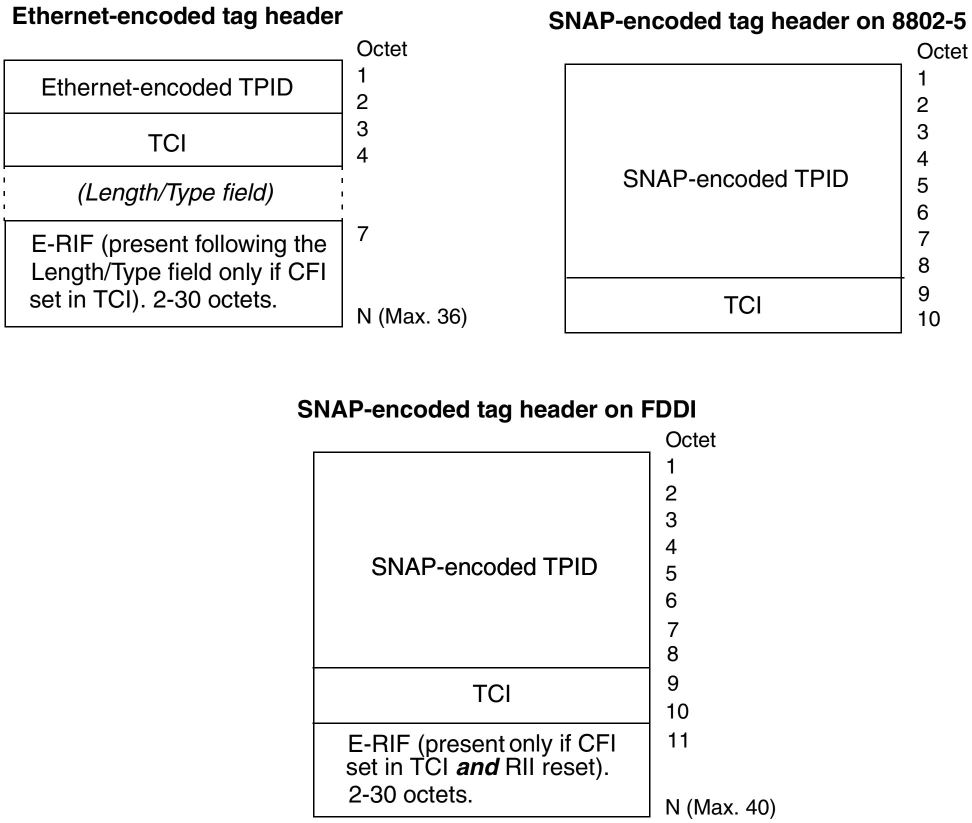

Ethernet-encoded tag header
Octet
Ethernet-encoded TPID
1
2
TCI
3
4
(Length/Type field)
E-RIF (present following the Length/Type field only if CFI set in TCI). 2-30 octets.
7
N (Max. 36)
SNAP-encoded tag header on 8802-5
Octet
SNAP-encoded TPID
1
2
3
4
5
6
7
8
TCI
9
10
SNAP-encoded tag header on FDDI
Octet
SNAP-encoded TPID
1
2
3
4
5
6
7
8
TCI
9
10
E-RIF (present only if CFI set in TCI *and* RII reset). 2-30 octets.
11
N (Max. 40)

Figure 9-1—Tag header formats

The Ethernet-encoded form of the tag header is used where the tagged frame is to be transmitted using 802.3/Ethernet MAC methods.

The two SNAP-encoded forms of the tag header are used where the tagged frame is to be transmitted on Token Ring and FDDI MAC methods:

d) On 8802-5 MACs, any RIF information, if present, appears in the normal position in the frame, i.e., directly following the destination MAC Address field, and is not part on the tag header.

e) On FDDI MACs, RIF information, if present, may be carried either in the normal position in the frame (i.e., directly following the destination MAC Address field), or within the E-RIF field in the tag header.

NOTE—The SNAP-encoded frame format on FDDI allows FDDI LANs to be operated in transparent mode and/or in source routing mode.

9.3.1 Tag Protocol Identifier (TPID) format

The structure of the TPID field takes two forms, depending upon whether the field is Ethernet encoded (9.3.1.1) or SNAP encoded (9.3.1.2). The TPID carries an Ethernet Type value (802.1QTagType), which identifies the frame as a tagged frame. The value of 802.1QTagType is defined in Table 9-1.

Table 9-1—802.1Q Ethernet Type allocations

Name	Value
802.1Q Tag Protocol Type (802.1QTagType)	81-00

9.3.1.1 Ethernet-encoded TPID

The Ethernet-encoded TPID (ETPID) field is two octets in length. The ETPID carries the value of the 802.1QTagType, as defined in Table 9-1.

Figure 9-2 illustrates the structure of the ETPID.

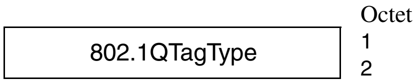


Figure 9-2—Ethernet-encoded TPID format

9.3.1.2 SNAP-encoded TPID

The SNAP-encoded TPID (STPID) is eight octets in length, encoded in SNAP format, as follows:

a) Octets numbered 1 through 3 carry the standard SNAP header, consisting of the hexadecimal value AA-AA-03;
b) Octets numbered 4 through 6 carry the SNAP PID, consisting of the hexadecimal value 00-00-00;
c) Octets 7 and 8 carry the 802.1QTagType, as defined in Table 9-1.

Figure 9-3 illustrates the structure of the STPID.

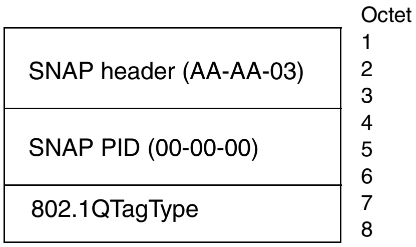


Figure 9-3—SNAP-encoded TPID format

9.3.2 Tag Control Information (TCI) format

The TCI field is two octets in length, and contains user_priority, CFI and VID (VLAN Identifier) fields. Figure 9-4 illustrates the structure of the TCI field.

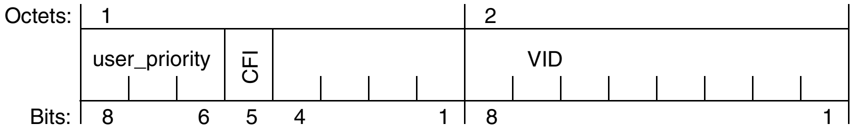


Figure 9-4—Tag Control Information (TCI) format

9.3.2.1 user_priority

The user_priority field is three bits in length, interpreted as a binary number. The user_priority is therefore capable of representing eight priority levels, 0 through 7. The use and interpretation of this field is defined in IEEE Std 802.1D, 1998 Edition.

9.3.2.2 CFI format

The Canonical Format Indicator (CFI) is a single bit flag value. CFI reset indicates that all MAC Address information that may be present in the MAC data carried by the frame is in Canonical format.

The meaning of the CFI when set depends upon the variant of the tag header in which it appears.

a) In a SNAP-encoded tag header transmitted using 8802-5 MAC methods, CFI has the following meanings:
   1) When set, indicates that all MAC Address information that may be present in the MAC data carried by the frame is in Non-canonical format (N);
   2) When reset, indicates that all MAC Address information that may be present in the MAC data carried by the frame is in Canonical format (C).
b) In an Ethernet-encoded tag header, transmitted using 802.3/Ethernet MAC methods, CFI has the following meanings:
   1) When set, indicates that the E-RIF field is present in the tag header, and that the NCFI bit in the RIF determines whether MAC Address information that may be present in the MAC data carried by the frame is in Canonical (C) or Non-canonical (N) format;
   2) When reset, indicates that the E-RIF field is not present in the tag header, and that all MAC Address information that may be present in the MAC data carried by the frame is in Canonical format (C).
c) In a SNAP-encoded tag header transmitted using FDDI MAC methods, CFI has the following meanings:
   1) When the frame takes the source-routed form (i.e., the RII bit is set in the frame's source MAC Address field and a RIF follows the source MAC Address), the interpretation of the CFI bit is as defined in a) for SNAP-encoded tag headers transmitted using 8802-5 MAC methods. The E-RIF field is not present in this form;
   2) When the frame takes the transparent form (i.e., the RII bit is reset in the frame's source MAC Address field and there is no RIF following the source MAC Address), the interpretation of the CFI bit and the presence or absence of the E-RIF is as defined in b) for Ethernet-encoded tag headers transmitted using 802.3/Ethernet MAC methods.

NOTE 1—The decision as to whether the source-routed form or the transparent form is used on FDDI is a local matter, and depends upon local knowledge in a Bridge or end station as to whether the FDDI LAN is capable of supporting source-routed traffic. The transparent form allows source-routing information to be transparently "tunneled" across LANs that do not support source routing; i.e., LANs where there may be intermediate transparent Bridges in the transmission path that would discard source-routed frames.

NOTE 2—In order to correctly relay frames between differing media types, the MAC Relay function of the Bridge needs to know the MAC type associated with each port. The means by which this information is provided to the MAC Relay function is a local matter.

9.3.2.3 VID format

The twelve-bit VLAN Identifier (VID) field uniquely identify the VLAN to which the frame belongs. The VID is encoded as an unsigned binary number. Table 9-2 identifies values of the VID field that have specific meanings or uses; the remaining values of VID are available for general use as VLAN identifiers.

A priority-tagged frame is a tagged frame whose tag header contains a VID value equal to the null VLAN ID.

NOTE 1—The specification of the ingress and egress rules for VLAN-Aware Bridges (8.6, 8.8) is such that a Bridge does not propagate priority-tagged frames; a received priority-tagged frame will acquire an appropriate VLAN classification on ingress as determined by the Ingress rules (8.6) and will, therefore, either be forwarded as an untagged frame~~,~~ or as a tagged frame tagged with that VLAN classification, depending upon the egress configuration for that VLAN. P~~p~~riority-tagged frames are therefore only ever generated by end stations.

A VLAN-tagged frame is a tagged frame whose tag header contains a VID value other than the null VLAN ID.

Table 9-2—Reserved VID values

VID value (hexadecimal)	Meaning/Use
0	The null VLAN ID. Indicates that the tag header contains only user_priority information; no VLAN identifier is present in the frame. This VID value shall not be configured as a PVID or a member of a VID Set, or configured in any Filtering Database entry, or used in any Management operation..
1	The default PVID value used for classifying frames on ingress through a Bridge Port. The PVID value of a Port can be changed by management.
FFF	Reserved for implementation use. This VID value shall not be configured as a PVID or a member of a VID Set, configured in any Filtering Database entry, used in any Management operation, or transmitted in a tag header.

A Bridge may implement the ability to support less than the full range of VID values; i.e., for a given implementation, an upper limit, N, is defined for the VID values supported, where N is less than or equal to 4094. All implementations shall support the use of all VID values in the range 0 through their defined maximum VID, N.

NOTE 2—There is a distinction made here between the range of VID *values* (0 through N) that an implementation can support as identifiers for its active VLANs, and the maximum number of active VLANs (V) that it is able to support at any one time. An implementation that supports a maximum of, say, only 16 active VLANs (V=16) can support VIDs for those VLANs that are chosen from anywhere in the full VID number space (i.e., support N=4094), or from a subset of that number space (i.e., support N<4094). Therefore N is always greater than or equal to V. As the support of a limited range of VIDs (i.e., N < 4094) may result in difficulties in an environment where different values of N are supported by different Bridges in the same Bridged LAN, it is recommended that new implementations of this standard support the use of the full range of VIDs (i.e., N=4094), even if the number of active VLANs (V) that the implementation supports is less than 4094.

9.3.3 Embedded RIF format

The E-RIF that can appear in Ethernet-encoded tag headers, and in the transparent form of SNAP-encoded tag headers on FDDI, is a modification of the RIF as defined in IEEE Std 802.1D, 1998 Edition, C.3.3.2. When present, it immediately follows the Length/Type field in the 802.3/Ethernet tagged frame, or immediately follows the TCI field in an FDDI frame. It consists of two components:

a) A two-octet Route Control Field (RC);
b) Zero or more octets of Route Descriptors (up to a maximum of 28 octets), as defined by RC.

The structure and semantics associated with the Route Descriptors are as defined in IEEE Std 802.1D, 1998 Edition, C.3.3.2.

Figure 9-5 illustrates the format of the RC component, as used in the E-RIF. The fields of the RC, and their usage, are defined in the following subclauses.

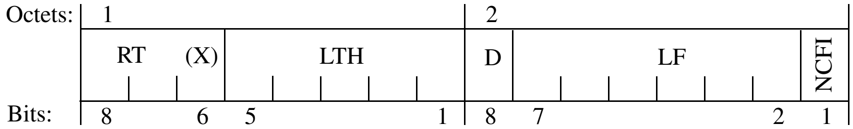


Figure 9-5—E-RIF Route Control (RC) field

Note that the definition of the E-RIF and its use within tag headers does not affect the definition of the RIF used in untagged frames in a source routing environment.

NOTE—The use of E-RIF fields in 802.3/Ethernet and FDDI frames is further discussed in 9.3.3.6.

9.3.3.1 Definition of the Routing Type (RT) field in the E-RIF

The definition of this field is as defined in IEEE Std 802.1D, 1998 Edition, C.3.3.2, with the addition that an RT value of 01X indicates a transparent frame. The value of the rightmost bit, X, is ignored; i.e., it is the binary value 01 in bits 8 and 7 that is used to signal a transparent frame.

NOTE 1—In effect, the RT bits in the E-RIF encode the state of the RII bit that would appear in an equivalent frame in a source-routed environment. RT values 01X encode RII reset; RT values 00X or 1XX encode RII set.

The transparent frame value indicates that, with the exception of the NCFI (9.3.3.5), the remainder of the E-RIF shall be discarded if the frame is forwarded using 8802-5 MAC methods. An E-RIF containing an RT value indicating a transparent frame contains no route descriptors, and is therefore exactly 2 octets in length (however, the value of the LTH field is set to zero in this case; see 9.3.3.2).

The least significant bit of the RT field, marked (X) is reserved, as defined in IEEE Std 802.1D, 1998 Edition, C.3.3.2.

The following rules ensure that the interpretation and use of the RT field by VLAN-aware devices is unambiguous, and does not conflict with use by non-VLAN-aware devices:

a) Where an untagged, source-routed frame is received from a Token Ring/FDDI LAN and is relayed as a tagged frame either on 802.3/Ethernet or on Token Ring/FDDI, if the received value of the RT field was 0XX, then the value of the RT field in the E-RIF or RIF in the tagged frame shall be transmitted as 000 (i.e., any 01X is converted to 000);
b) Where an untagged, transparent frame is received from a Token Ring LAN and is relayed as a tagged frame either on 802.3/Ethernet or on FDDI, then the tag header will carry an E-RIF in which the value of the RT field shall be 010;
c) Where a VLAN-aware end station on Token Ring/FDDI generates source-routed, tagged frames in the source-routed form (i.e., where the RIF appears in the normal position for a source-routed frame and there is no E-RIF in the tag header), then it shall not transmit RT values of 010 or 011 in the RIF;
d) Where a VLAN-aware end station on 802.3/Ethernet or FDDI generates source-routed, tagged frames in the transparent form (i.e., where there is source-routing information present that is carried in the E-RIF in the tag header, but there is no RIF in the normal position for a source-routed frame) then it shall not use RT values of 010 or 011 in the E-RIF;

NOTE 2—In other words, when source-routed frames are tunnelled across a transparent environment, the state of the RT bits signal that the E-RIF carries real source-routing information. An equivalent frame generated in a source-routed environment would have RII set, and the source-routing information would appear as a RIF in the normal position.

e) "X" in these rules is taken to mean "Ignored upon receipt, transmitted as zero."

NOTE 3—The use of an RT value of 01X to indicate a transparent frame applies only to RT values carried in the E-RIF; values of 01X appearing in the RIF of a normal source-routed frame (whether tagged or untagged) are never interpreted in this way.

NOTE 4—In addition to its use as a means of tunnelling source-routed frames across transparent LANs, the above rules for the use of RT values in the E-RIF also provide the possibility of stations attached to transparent LANs using source routing to communicate with stations attached to source-routed LANs. In particular, bullet d) states the rule that makes such communication possible.

9.3.3.2 Definition of the Length (LTH) field in the E-RIF

This field is as defined in IEEE Std 802.1D, 1998 Edition, C.3.3.2, with the exception that if the value of the RT field in the E-RIF is 01X, indicating a transparent frame, then the LTH field shall carry a value of 0.

NOTE—The use of a zero length in conjunction with the transparent RT indicator ensures that there is no possibility of such frames being misinterpreted as valid Specifically Routed frames by devices that support source routing.

9.3.3.3 Definition of the Direction Bit (D) field in the E-RIF

This field is as defined in IEEE Std 802.1D, 1998 Edition, C.3.3.2.

9.3.3.4 Definition of the Largest Frame (LF) field in the E-RIF

This field is as defined in IEEE Std 802.1D, 1998 Edition, C.3.3.2, and is used accordingly for all tagged frames transmitted using 802.3/Ethernet or FDDI MAC methods that carry an E-RIF, whether source-routed or transparent. For frames transmitted using 802.3/Ethernet MAC methods, the value of this field shall indicate a largest frame size of 1470 octets or less.

9.3.3.5 Definition of the NCFI field in the E-RIF

The Non-canonical Format Indicator field of the E-RIF has the following meanings:

a) When reset, indicates that all MAC Address information that may be present in the MAC data carried by the frame is in Non-canonical format (N);
b) When set, indicates that all MAC Address information that may be present in the MAC data carried by the frame is in Canonical format (C).

In source-routed frames on Token Ring/FDDI MAC methods, this bit in the RIF is reserved, and its value is preserved across Bridges; its value is normally reset.

Where a source-routed frame is received from a Token Ring/FDDI LAN and relayed as a tagged frame containing an E-RIF on 802.3/Ethernet or FDDI (i.e., where RIF information carried in the normal position for a source-routed frame is embedded in an E-RIF), the received value of this field is replaced by the appropriate N or C value.

Where a tagged frame containing an E-RIF is relayed from an 802.3/Ethernet or FDDI LAN onto a Token Ring or FDDI LAN as a source-routed frame (i.e., when the RIF information carried in the E-RIF is restored to its normal position in the frame), this bit in the RIF is reset in the frame transmitted onto the destination LAN.

9.3.3.6 E-RIF usage in tagged frames on 802.3/Ethernet and FDDI

There are three cases where a tagged frame will carry an E-RIF:

a) It is a transparent frame that carries E-N or L-N information;
b) It is a source-routed frame that carries E-N or L-N information;
c) It is a source-routed frame that carries L-C information.

Case a) occurs if

d) An untagged frame containing E-N or L-N data, and with no RIF (RII reset), is received from an 8802-5 LAN and is transmitted as a VLAN-tagged frame on 802.3/Ethernet or FDDI; or
e) A tagged frame with CFI set, and with no RIF (RII reset), is received from an 8802-5 LAN and is transmitted as a VLAN-tagged frame on 802.3/Ethernet or FDDI.

Case b) occurs if

f) An untagged frame containing E-N or L-N data, and with a RIF (RII set), is received from a Token Ring/FDDI LAN and is transmitted as a VLAN-tagged frame on 802.3/Ethernet, or on an FDDI LAN that does not support source routing; or
g) A tagged frame with CFI set, and with a RIF (RII set), is received from a Token Ring/FDDI LAN and is transmitted as a VLAN-tagged frame on 802.3/Ethernet, or on an FDDI LAN that does not support source routing.

Case c) occurs if

h) An untagged frame containing L-C data, and with a RIF (RII set), is received from a Token Ring/FDDI LAN and is transmitted as a VLAN-tagged frame on 802.3/Ethernet, or on an FDDI LAN that does not support source routing; or
i) A tagged frame with CFI reset, and with a RIF (RII set), is received from a Token Ring/FDDI LAN and is transmitted as a VLAN-tagged frame on 802.3/Ethernet, or on an FDDI LAN that does not support source routing.

The other possible frame representations on Token Ring/FDDI, namely, transparent frames carrying either L-C or E-C information, are represented on 802.3/Ethernet media as VLAN-tagged frames with CFI reset (i.e., the RIF is not present).

In case a), the RIF is created from scratch as part of the frame translation. The RIF in this case consists of only 2 octets, with field values as follows:

j) RT is set to the binary value 010, to indicate a transparent frame;
k) LTH is set to 0;
l) D and LF fields are set to 0;
m) NCFI is reset to indicate that the format is Non-canonical.

In cases b) and c), the RIF contained in the frame to be translated is used unmodified, with the two exceptions that

n) Received RT values of 0XX are translated to 000 in the RT field of the E-RIF;
o) The NCFI field is set appropriately to indicate the format (Canonical or Non-canonical) of the data carried in the frame.

10. Use of GMRP in VLANs

The GARP Multicast Registration Protocol, GMRP, defined in Clause 10 of IEEE Std 802.1D, 1998 Edition, allows the declaration and dissemination of Group membership information, in order to permit GMRP-aware Bridges to filter frames destined for group MAC Addresses on Ports through which potential recipients of such frames cannot be reached. The specification in IEEE Std 802.1D, 1998 Edition calls for the propagation of GMRP registrations only in the GARP Information Propagation (GIP) Context known as the *Base Spanning Tree Context* (IEEE Std 802.1D, 1998 Edition, Clauses 10 and 12.3.4); i.e., propagation of GMRP information occurs among the set of Ports of a Bridge that are part of the *active topology* (IEEE Std 802.1D, 1998 Edition, 7.4) of the Spanning Tree resulting from operation of the Spanning Tree Algorithm and Protocol defined in IEEE Std 802.1D, 1998 Edition, Clause 8. This GIP Context is identified by a GIP Context Identifier of 0.

In Bridged LAN environments that support the definition and management of VLANs in accordance with this standard, the operation of GMRP as specified in IEEE Std 802.1D, 1998 Edition is extended to permit GMRP to operate in multiple GIP contexts, defined by the set of VLANs that are active in the Bridged LAN; these are known as *VLAN Contexts.*

The use of GMRP in a VLAN Context allows GMRP registrations to be made that are specific to that VLAN; i.e., it allows the Group filtering behavior for one VLAN to be independent of the Group filtering behavior for other VLANs. The following subclauses define the extensions to the definition of GMRP that permit its use in VLAN contexts.

With the exception of the extensions defined in this standard, the operation of GMRP and the conformance requirements associated with GMRP are as defined in IEEE Std 802.1D, 1998 Edition.

10.1 Definition of a VLAN Context

The GIP Context Identifier used to identify a VLAN Context shall be equal to the VID used to identify the corresponding VLAN.

The set of Ports of a Bridge defined to be part of the active topology for a given VLAN Context shall be equal to the set of Ports of a Bridge for which the following are true:

a) The Port is a member of the *Member set* (8.11.9) for that VLAN; and
b) The Port is one of the Ports of the Bridge that are part of the active topology for the spanning tree that supports that VLAN.

NOTE—The above definition applies equally to SST and MST environments. It ensures that GMRP operates in either environment, without the GMRP implementations needing to be aware whether the VLAN contexts that apply are all supported by the same spanning tree, as in the SST environment, or are potentially distributed across two or more spanning trees, as in the MST environment.

10.2 GMRP Participants and GIP Contexts

For each Port of the Bridge, a distinct instance of the GMRP Participant can exist for each VLAN Context supported by the Bridge. Each GMRP Participant maintains its own set of GARP Applicant and Registrar state machines, and its own Leave All state machine. There is no GMRP Participant associated with the Base Spanning Tree Context.

A given GARP Participant, operating in a given GIP Context, manipulates only the Port Filtering Mode and Group Registration Entry information for the context concerned. In the case of Group Registration Entries, the GIP Context Identifier value corresponds to the value of the VID field of the entry.

10.3 Context identification in GMRP PDUs

Implementations of GMRP conformant to the specification of GMRP in IEEE Std 802.1D, 1998 Edition exchange PDUs in the Base Spanning Tree Context; such PDUs are transmitted and received by GMRP Participants as untagged frames.

Implementations of GMRP in VLAN Bridges apply the same ingress rules (8.6) to received GMRP PDUs that are defined for the reception Port. Therefore

a) GMRP frames with no VLAN classification (i.e., untagged or priority-tagged GMRP frames) are discarded if the Acceptable Frame Types parameter (8.4.3) for the Port is set to *Admit Only VLAN-tagged frames.* Otherwise, they are classified according to the PVID for the Port;
b) VLAN-tagged GMRP frames are classified according to the VID carried in the tag header;
c) If Ingress Filtering (8.4.5) is enabled, and if the Port is not in the Member set (8.11.9) for the GMRP frame's VLAN classification, then the frame is discarded.
d) If the VLAN classification of the frame is outside the range of VIDs supported by the implementation (9.3.2.3), then the frame is discarded.

The VLAN classification thus associated with received GMRP PDUs establishes the VLAN Context for the received PDU, and identifies the GARP Participant instance to which the PDU is directed.

GMRP PDUs transmitted by GMRP Participants are VLAN classified according to the VLAN Context associated with that Participant. GMRP Participants in VLAN Bridges apply the same egress rules that are defined for the transmission Port (8.8). Therefore

e) GMRP PDUs are transmitted through a given Port only if the value of the Member Set for the Port for the VLAN concerned indicates that the VLAN is registered on that Port;
f) GMRP PDUs are transmitted as VLAN-tagged frames or as untagged frames in accordance with the state of the Untagged Set (8.11.9) for that Port for the VLAN concerned. Where VLAN-tagged frames are transmitted, the VID field of the tag header carries the VLAN Context Identifier value.

10.4 Default Group filtering behavior and GMRP propagation

The propagation of GMRP registrations within a VLAN context has implications with respect to the choice of default Group filtering behavior within a Bridged LAN. As GMRP frames are transmitted only on outbound Ports that are in the Member set (8.11.9) for the VLAN concerned, propagation of Group registrations by a given Bridge occurs only towards regions of the Bridged LAN where that VLAN has been (statically or dynamically) registered. This is illustrated in Figure 10-1; dotted lines in the diagram show those regions of the LAN where propagation of registrations for Group M in VLAN V does not occur. Consequently, the Filtering Databases of the lower two Bridges will not contain any Dynamic Group Registration Entry for Group M in VLAN V.

The action of these two Bridges on receipt of frames, on either of their lower Ports, destined for Group M and VLAN V, will depend upon the Default Group Filtering Behavior adopted by their upper Ports, which are the Ports that are in the Member set for VLAN V. If the Default Group Filtering Behavior is either Forward All Groups or Forward Unregistered Groups, then these Bridges will forward the frames. If the Default Group Filtering Behavior is Filter Unregistered Groups, then these Bridges will filter the frames. In the scenario shown, the choice of Default Group Filtering Behavior is therefore crucial with respect to whether or not end station S, or any other station that is "outside" the VLAN, is able to send frames to members of the Group. The choice between Filter Unregistered Groups and the other default behaviors therefore has the effect of defining VLANs that are closed to external unregistered traffic (Filter Unregistered Groups) or open to external unregistered traffic (Either of the other default behaviors).

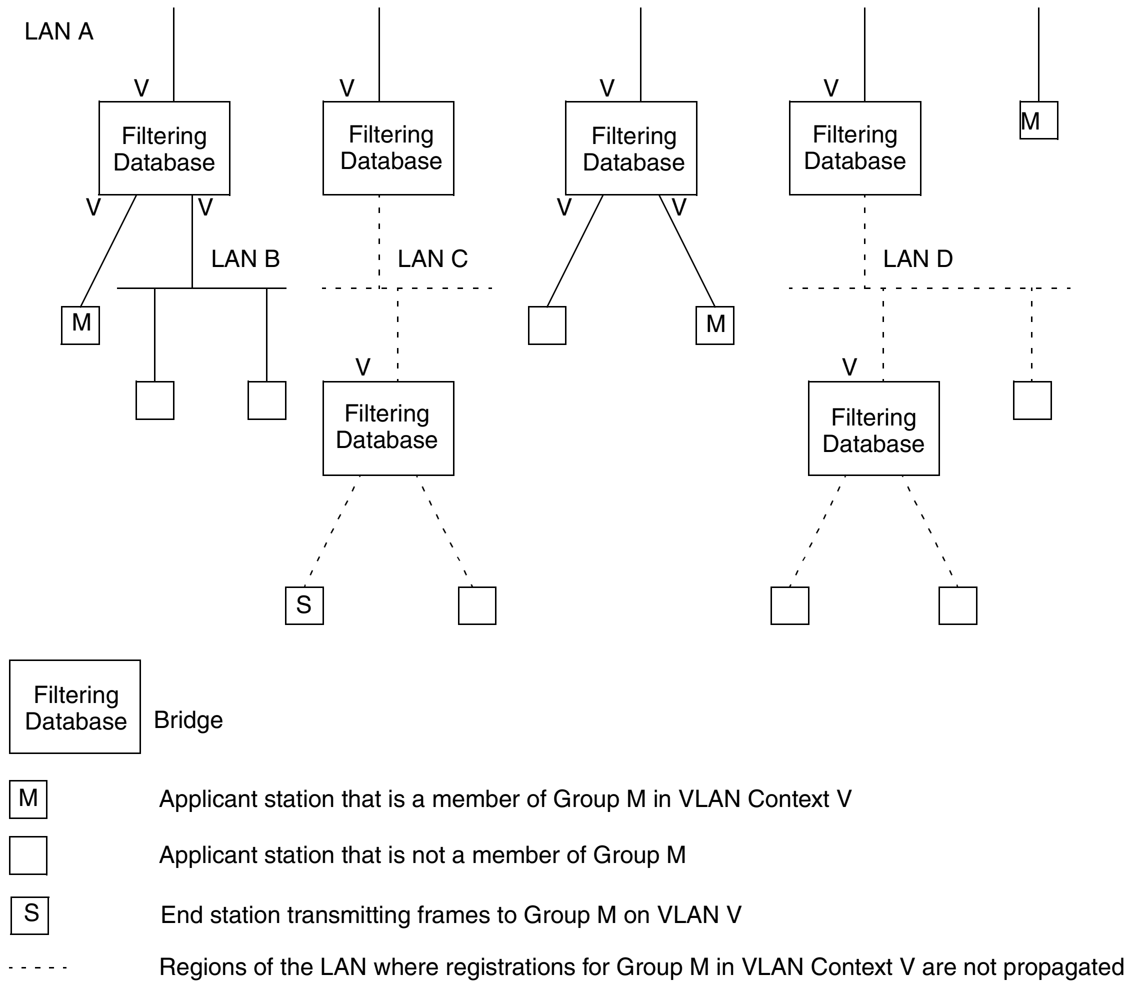

LAN A
V
Filtering Database
V
V
LAN B
M
V
Filtering Database
LAN C
V
Filtering Database
S
V
Filtering Database
V
V
M
V
Filtering Database
LAN D
V
Filtering Database
M
Filtering Database
Bridge
M
Applicant station that is a member of Group M in VLAN Context V
Applicant station that is not a member of Group M
S
End station transmitting frames to Group M on VLAN V
Regions of the LAN where registrations for Group M in VLAN Context V are not propagated

Figure 10-1—Example of GMRP propagation in a VLAN context

11. VLAN topology management

The egress rules (8.8) defined for the Forwarding Process in VLAN Bridges rely on the existence of configuration information for each VLAN that defines the set of Ports of the Bridge through which one or more members are reachable. This set of Ports is known as the Member Set (8.11.9), and its membership is determined by the presence or absence of configuration information in the Filtering Database, in the form of Static and Dynamic VLAN Registration Entries (8.11.2, 8.11.5).

Reliable operation of the VLAN infrastructure requires VLAN membership information held in the Filtering Database to be maintained in a consistent manner across all VLAN-aware Bridges in the Bridged LAN, in order to ensure that frames destined for end station(s) on a given VLAN can be correctly delivered, regardless of where in the Bridged LAN the frame is generated. Maintenance of this information by end stations that are sources of VLAN-tagged frames can allow such stations to suppress transmission of such frames if no members exist for the VLAN concerned.

This standard defines the following mechanisms that allow VLAN membership information to be configured:

a) Dynamic configuration and distribution of VLAN membership information by means of the GARP VLAN Registration Protocol (GVRP), as described in 11.2;
b) Static configuration of VLAN membership information via Management mechanisms, as described in Clause 12, which allow configuration of Static VLAN Registration Entries.

These mechanisms provide for the configuration of VLAN membership information as a result of

c) Dynamic registration actions taken by end stations or Bridges in the bridged LAN;
d) Administrative actions.

11.1 Static and dynamic VLAN configuration

The combined functionality provided by the ability to configure Static VLAN Registration Entries in the Filtering Database, coupled with the use of the Restricted_VLAN_Registration control (11.2.3.2.3) and the ability of GVRP to dynamically create and update Dynamic VLAN Registration Entries, offers the following possibilities with respect to how VLANs are configured on a given Port:

a) *Static configuration only.* The management facilities described in Clause 12 are used to establish precisely which VLANs have this Port in their Member set, and the GVRP management controls are used to disable the operation of the GVRP protocol on that Port. Hence, any use of GVRP by devices reachable via that Port is ignored, and the Member set for all VLANs can therefore only be determined by means of static entries in the Filtering Database.
b) *Dynamic configuration only.* The operation of GVRP is relied upon to establish Dynamic VLAN Registration Entries that will dynamically reflect which VLANs are registered on the Port, their contents changing as the configuration of the network changes. The GVRP management controls are set to enable the operation of the GVRP protocol on that Port.
c) *Combined static and dynamic configuration.* The static configuration mechanisms are used in order to configure some VLAN membership information; for other VLANs, GVRP is relied upon to establish the configuration. The GVRP management controls are set to enable the operation of the GVRP protocol on that Port.

All of the above approaches are supported by the mechanisms defined in this standard, and each approach is applicable in different circumstances. For example:

d) Use of static configuration may be appropriate on Ports where the configuration of the attached devices is fixed, or where the network administrator wishes to establish an administrative boundary outside of which any GVRP registration information is to be ignored. For example, it might be desirable for all Ports serving end user devices to be statically configured in order to ensure that particular end users have access only to particular VLANs.

e) Use of dynamic configuration may be appropriate on Ports where the VLAN configuration is inherently dynamic; where users of particular VLANs can connect to the network via different Ports on an ad hoc basis, or where it is desirable to allow dynamic reconfiguration in the face of Spanning Tree topology changes. In particular, if the "core" of the Virtual Bridged LAN contains redundant paths that are pruned by the operation of Spanning Tree, then it is desirable for Bridge Ports that form the core network to be dynamically configured.

f) Use of both static and dynamic configuration may be appropriate on Ports where it is desirable to place restrictions on the configuration of some VLANs, while maintaining the flexibility of dynamic registration for others. For example, on Ports serving mobile end user devices, this would maintain the benefits of dynamic VLAN registration from the point of view of traffic reduction, while still allowing administrative control over access to some VLANs via that Port.

11.2 GARP VLAN Registration Protocol

The GARP VLAN Registration Protocol (GVRP) defines a *GARP Application* that provides the VLAN registration service defined in 11.2.2. GVRP makes use of GARP Information Declaration (GID) and GARP Information Propagation (GIP), which provide the common state machine descriptions and the common information propagation mechanisms defined for use in GARP-based applications. The GARP architecture, GID, and GIP are defined in IEEE Std 802.1D, 1998 Edition, Clause 12.

GVRP provides a mechanism for dynamic maintenance of the contents of Dynamic VLAN Registration Entries for each VLAN, and for propagating the information they contain to other Bridges. This information allows GVRP-aware devices to dynamically establish and update their knowledge of the set of VLANs that currently have active members, and through which Ports those members can be reached.

11.2.1 GVRP overview

The operation of GVRP is closely similar to the operation of GMRP (IEEE Std 802.1D, 1998 Edition, Clause 10), which is used for registering Group membership information. The primary differences are as follows:

a) The attribute values carried by the protocol are 12-bit VID values, rather than 48-bit MAC Addresses and Group service requirement information;

b) The act of registering/deregistering a VID affects the contents of Dynamic VLAN Registration Entries (8.11.5), rather than the contents of Group Registration Entries (8.11.4).

c) In an SST environment, there is a single GVRP Participant per port, rather than one GVRP Participant per VLAN per port, and the GVRP Participants all operate in a single GIP context;

d) In an MST environment, there is again a single GVRP Participant per port, but each GVRP Participant operates in multiple GIP contexts.

GVRP allows both end stations and Bridges in a Bridged LAN to issue and revoke declarations relating to membership of VLANs. The effect of issuing such a declaration is that each GVRP Participant that receives the declaration will create or update a Dynamic VLAN Registration Entry in the Filtering Database to indicate that VLAN is registered on the reception Port. Subsequently, if all Participants on a segment that had an interest in a given VID revoke their declarations, the Port attached to that segment is set to Unregistered in the Dynamic VLAN Registration Entry for that VLAN by each GVRP Participant attached to that segment.

Figure 11-1 illustrates the architecture of GVRP in the case of a two-Port Bridge and an end station.

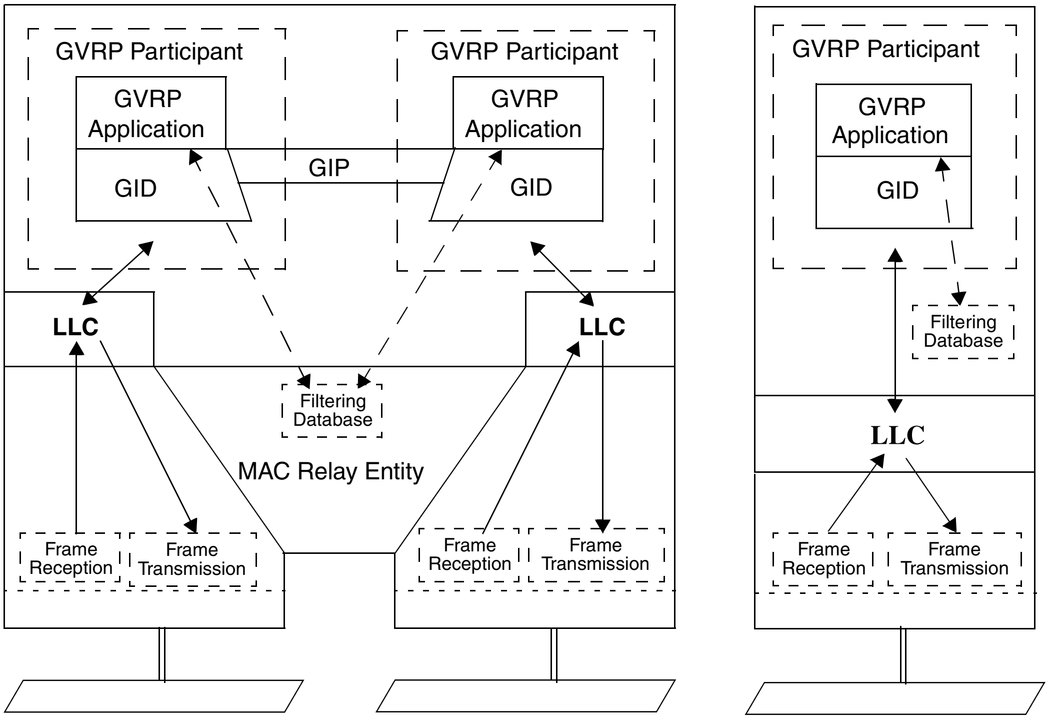


Figure 11-1—Operation of GVRP

As shown in the diagram, the GVRP Participant consists of the following components:

e) The GVRP Application, described in 11.2.3;

f) GARP Information Propagation (GIP), described in IEEE Std 802.1D, 1998 Edition, Clause 12;

g) GARP Information Declaration, described in IEEE Std 802.1D, 1998 Edition, Clause 12.

11.2.1.1 Behavior of end stations

VLAN-aware end stations participate in GVRP protocol activity, as appropriate for the set of VLANs of which they are currently members. GVRP provides a way for such an end station to ensure that the VLAN(s) of which it is a member are registered for each Port on any LAN segment to which the end station is attached. GVRP also provides for that VID information to be propagated across the Spanning Tree to other VLAN-aware devices, as described in 11.2.1.2.

Incoming VLAN membership information (from all other devices on the same LAN segment) allows such end stations to “source prune” (i.e., discard at source; see IEEE Std 802.1D, 1998 Edition, 10.2.2) any traffic destined for VLANs that currently have no other members in the Bridged LAN, thus avoiding the generation of unnecessary traffic on their local LAN segments. This is illustrated in Figure 11-1 by a Filtering Database shown as being present in the end station.

NOTE—Non-VLAN-aware end stations have no need to register VLAN membership via GVRP; indeed, this would be impossible for them to achieve if truly VLAN-unaware, as they would have no knowledge of the set of VLANs in which they participate. Their VLAN registration requirements are taken care of by means of the configuration of PVIDs (and possibly other VLAN classification mechanisms) and the propagation of registered VLAN IDs by the Bridges.

11.2.1.2 Behavior of Bridges

VLAN-aware Bridges register and propagate VLAN memberships on all Bridge Ports that are part of the active topology of the underlying Spanning Tree(s). Incoming VID registration and de-registration information is used to update the Dynamic VLAN Registration Entries associated with each VLAN. Any changes in the state of registration of a given VID on a given Port are propagated on Ports that are part of the active topology of the underlying Spanning Tree, in order to ensure that other GVRP-aware devices in the Bridged LAN update their Filtering Databases appropriately. In Bridges that support multiple Spanning Tree instances, the MST Configuration Table (12.12, 13.7) is used to determine which spanning tree instance is to be used to propagate registration information for each supported VLAN.

The Filtering Databases in all GVRP-aware devices are thus automatically configured such that the Port Map in the Dynamic VLAN Registration Entry for a given VID indicates that a given Port is registered if one or more members of the corresponding VLAN are reachable through the Port.

NOTE—The information that determines whether frames destined for each VLAN are transmitted VLAN-tagged or untagged is carried in Static VLAN Registration Entries (8.11.2); if no such entry exists for a VLAN, then it is assumed that frames for that VLAN are transmitted VLAN-tagged on all Ports. Therefore, if the configuration information held in the Filtering Database for a given VLAN consists only of information configured by the operation of GVRP (i.e., only a Dynamic VLAN Registration Entry), then all traffic for that VLAN will be VLAN-tagged on transmission.

11.2.1.3 Use of the PVID and VID Set

The initial state of the Permanent Database contains a Static VLAN Registration Entry for the Default PVID, in which the Port Map indicates Registration Fixed on all Ports. This ensures that in the default state, where the value of every PVID of each Port is the Default PVID and where the VID Set of each Port is empty, membership of the Default PVID is propagated across the Bridged LAN to all other GVRP-aware devices. Subsequent management action may change both the Permanent Database and the Filtering Database in order to modify or remove this initial setting, and may change the PVID and/or VID Set value(s) on any Port of the Bridge.

NOTE—In the absence of any modification of these initial settings, this ensures that connectivity is established across the Bridged LAN for the VLAN corresponding to the Default PVID.

11.2.2 VLAN registration service definition

The VLAN registration service allows MAC Service users to indicate to the MAC Service provider the set of VLANs in which they wish to participate; i.e., that the MAC Service user wishes to receive traffic destined for members of that set of VLANs. The service primitives allow the service user to:

a) Register membership of a VLAN;
b) De-register membership of a VLAN.

Provision of these services is achieved by means of GVRP and its associated procedures, as described in 11.2.3.

ES_REGISTER_VLAN_MEMBER (VID)

Indicates to the MAC Service provider that the MAC Service user wishes to receive frames destined for the VLAN identified by the VID parameter.

ES_DEREGISTER_VLAN_MEMBER (VID)

Indicates to the MAC Service provider that the MAC Service user no longer wishes to receive frames destined for the VLAN identified by the VID parameter.

The use of these services can result in the propagation of VID information across the Spanning Tree, affecting the contents of Dynamic VLAN Registration Entries (8.11.5) in Bridges and end stations in the Bridged LAN, and thereby affecting the frame forwarding behavior of those Bridges and end stations.

11.2.3 Definition of the GVRP Application

11.2.3.1 Definition of GARP protocol elements

11.2.3.1.1 GVRP Application address

The group MAC Address used as the destination address for GARP PDUs destined for GVRP Participants shall be the GVRP address identified in Table 11-1. Received PDUs that are constructed in accordance with the PDU format defined in IEEE Std 802.1D, 1998 Edition, 12.11, and which carry a destination MAC Address equal to the GVRP address are processed as follows:

a) In Bridges and end stations that support the operation of GVRP, all such PDUs shall be submitted to the GVRP Participant associated with the receiving Port for further processing;
b) In Bridges that do not support the operation of GVRP, all such PDUs shall be submitted to the Forwarding Process.

Table 11-1—GVRP Application address

Assignment	Value
GVRP address	01-80-C2-00-00-21

NOTE—The GVRP Application Address has been allocated from the set of GARP Application addresses defined in IEEE Std 802.1D, 1998 Edition, Table 12-1, using the MAC Address contained in the second entry of that table.

11.2.3.1.2 Encoding of GVRP Attribute Types

The operation of GVRP defines a single Attribute Type (IEEE Std 802.1D, 1998 Edition, 12.11.2.2) that are carried in GARP protocol exchanges; the VID Attribute Type. The VID Attribute Type is used to identify values of VLAN Identifiers (VIDs). The value of the Group Attribute Type carried in GVRP PDUs shall be 1.

11.2.3.1.3 Encoding of GVRP Attribute Values

Values of instances of the VID Attribute Type shall be encoded as Attribute Values in GARP PDUs (IEEE Std 802.1D, 1998 Edition, 12.11.2.6) as two octets, taken to represent an unsigned binary number, and equal to the hexadecimal value of the VLAN identifier that is to be encoded.

11.2.3.2 Provision and support of the VLAN registration service

11.2.3.2.1 End system VLAN membership declaration

The GVRP Application element of a GVRP Participant provides the dynamic registration and de-registration services defined in 11.2.2, as follows:

On receipt of an ES_REGISTER_VLAN_MEMBER service primitive, the GVRP Participant issues a GID_Join.request service primitive (IEEE Std 802.1D, 1998 Edition, 12.3.2.1). The attribute_type parameter of the request carries the value of the VID Attribute Type (11.2.3.1.2) and the attribute_value parameter carries the value of the VID parameter carried in the ES_REGISTER_VLAN_MEMBER primitive.

On receipt of an ES_DEREGISTER_VLAN_MEMBER service primitive, the GVRP Participant issues a GID_Leave.request service primitive (IEEE Std 802.1D, 1998 Edition, 12.3.2.1). The attribute_type parameter of the request carries the value of the VID Attribute Type (11.2.3.1.2) and the attribute_value parameter carries the value of the VID parameter carried in the ES_REGISTER_VLAN_MEMBER primitive.

11.2.3.2.2 VLAN membership registration

The GVRP Application element of a GVRP Participant responds to registration and de-registration events signalled by GID as follows:

On receipt of a GID_Join.indication (ISO/IEC 15802-3, 12.3.2.2) whose attribute_type is equal to the value of the VID Attribute Type (11.2.3.1.2), the GVRP Application element indicates the reception Port as Registered in the Port Map of the Dynamic VLAN Registration Entry for the VID indicated by the attribute_value parameter. If no such entry exists, there is sufficient room in the Filtering Database, and the VID is within the range of values supported by the implementation (see 9.3.2.3), then an entry is created. If not, then the indication is not propagated and the registration fails.

The creation of new Dynamic VLAN Registration Entries can be restricted by use of the Restricted_VLAN_Registration control (11.2.3.2.3). If the value of this control is TRUE, then creation of a new dynamic entry is permitted only if there is a Static VLAN Registration Entry for the VLAN concerned, in which the Registrar Administrative Control value is Normal Registration.

On receipt of a GID_Join.indication (IEEE Std 802.1D, 1998 Edition, 12.3.2.2) whose attribute_type is equal to the value of the VID Attribute Type (11.2.3.1.2), the GVRP Application element indicates the reception Port as Registered in the Port Map of the Dynamic VLAN Registration Entry for the VID indicated by the attribute_value parameter. If no such entry exists, there is sufficient room in the Filtering Database, and the VID is within the range of values supported by the implementation (see 9.3.2.3), then an entry is created. If not, then the indication is not propagated and the registration fails.

11.2.3.2.3 Administrative controls

The provision of static control over the declaration or registration state of the state machines associated with the GVRP Application is achieved by means of the Registrar Administrative Control parameters provided by GARP (IEEE Std 802.1D, 1998 Edition, 12.9.1). These parameters are represented as Static VLAN Registration Entries in the Filtering Database (8.11.2). Where management capability is implemented, these parameters can be manipulated by means of the management functionality defined in 12.7.

The provision of static control over the ability of Applicant state machines to participate in protocol exchanges is achieved by means of the Applicant Administrative Control parameters associated with the operation of GARP (IEEE Std 802.1D, 1998 Edition, 12.9.2). Where management capability is implemented, the Applicant Administrative Control parameters can be applied and modified by means of the management functionality defined in 12.9.

Further administrative control over dynamic VLAN registration can be achieved, if supported, by means of a per-Port Restricted_VLAN_Registration control parameter. If the value of this control is TRUE for a given Port, the creation or modification of Dynamic VLAN Registration Entries as a result of GVRP exchanges on that Port shall be restricted only to those VLANs for which Static VLAN Registration Entries exist in which the Registrar Administrative Control value is Normal Registration. If the value of the Restricted_VLAN_Registration control is FALSE, dynamic VLAN registration is not so restricted. The recommended default value of this parameter is FALSE. Where management capability is implemented, the value of the Restricted_VLAN_Registration control can be manipulated by means of the management functionality defined in 12.10. If management of this parameter is not supported, the value of this parameter shall be FALSE for all Ports.

11.2.3.3 GIP context for GVRP

In an SST environment, GVRP operates in the Base Spanning Tree Context (12.3.4 of IEEE Std 802.1D, 1998 Edition); i.e. GVRP operates only on the CIST defined by IEEE Std 802.1D, 1998 Edition. Consequently, all GVRP PDUs sent and received by GVRP Participants in SST bridges are transmitted as untagged frames.

11.2.3.4 GIP contexts for GVRP in MST environments

In an MST environment, GVRP operates in multiple spanning-tree contexts, one for each of the spanning trees. Each spanning-tree context consists of the ports that are in the forwarding state for the corresponding spanning tree. The GIP context for the GVRP Participants associated with a given spanning tree is the same as the spanning-tree context. MST bridges can identify the GIP contexts using the mappings of VID values to MSTID values (see 8.11.2). All GVRP PDUs sent and received by GVRP Participants in MST bridges are transmitted as untagged frames.

11.3 Conformance to GVRP

This subclause defines the conformance requirements for implementations claiming conformance to GVRP. Two cases are covered; implementation of GVRP in MAC Bridges and implementation of GVRP in end stations. Although this standard is principally concerned with defining the requirements for MAC Bridges, the conformance requirements for end station implementations of GVRP are included in order to give useful guidance to such implementations. The PICS proforma defined in Annex A is concerned only with conformance claims with respect to MAC Bridges.

11.3.1 Conformance to GVRP in MAC Bridges

A MAC Bridge for which conformance to GVRP is claimed shall

a) Conform to the operation of the GARP Applicant and Registrar state machines, and the LeaveAll generation mechanism, as defined in IEEE Std 802.1D, 1998 Edition, 12.8.1, 12.8.2, and 12.8.3;
b) Exchange GARP PDUs as required by those state machines, formatted in accordance with the generic PDU format described in IEEE Std 802.1D, 1998 Edition, 12.11, and able to carry application-specific information as defined in 11.2.3, using the GVRP Application address as defined in Table 11-1;
c) propagate registration information
    1) in an SST bridge, in accordance with the operation of GIP for the Base Spanning Tree Context, as specified in 12.3.3 and 12.3.4 of IEEE Std 802.1D, 1998 Edition; or
    2) in an MST bridge, in accordance with the operation of GIP for multiple Spanning Tree contexts as specified in 11.2.3.4 of this standard.
d) Implement the GVRP Application component as defined in 11.2;
e) Forward, filter or discard MAC frames carrying any GARP Application address as the destination MAC Address in accordance with the requirements of 8.14.3.

11.3.2 Conformance to GVRP in end stations

An end station for which conformance to GVRP is claimed shall

a) Conform to the operation of one of
    1) The Applicant state machine, as defined in IEEE Std 802.1D, 1998 Edition, 12.8.1; or
    2) The Applicant Only state machine, as defined in IEEE Std 802.1D, 1998 Edition, 12.8.5; or
    3) The Simple Applicant state machine, as defined in IEEE Std 802.1D, 1998 Edition, 12.8.6;

b) Exchange GARP PDUs as required by the GARP state machine(s) implemented, formatted in accordance with the generic PDU format described in IEEE Std 802.1D, 1998 Edition, 12.11, and able to carry application-specific information as defined in 11.2.3, using the GVRP Application address as defined in Table 11-1;
c) Support the provision of end system registration and de-registration as defined in 11.2;
d) Discard MAC frames carrying any GARP Application address as the destination MAC Address in accordance with the requirements of 8.14.3.

An end station for which conformance to GVRP is claimed may optionally

e) Conform to the operation of the GARP Registrar state machine and the LeaveAll generation mechanism, as defined in IEEE Std 802.1D, 1998 Edition, 12.8.2 and 12.8.3; and
f) Support the provision of VLAN registration and de-registration as defined in 11.2; and
g) Filter outgoing frames destined for group MAC Addresses in accordance with registered VLAN membership information, in a manner consistent with the operation of the filtering function of the forwarding process described in 8.7.2 and the operation of the egress rules defined in 8.8.

It is recommended that only those end stations that require the ability to perform Source Pruning (11.2.1.1) conform to the operation of the Applicant state machine (IEEE Std 802.1D, 1998 Edition, 12.8.1).

For the reasons stated in IEEE Std 802.1D, 1998 Edition, 12.7.9, it is recommended that end stations that do not require the ability to perform Source Pruning implement the Applicant Only state machine (IEEE Std 802.1D, 1998 Edition, 12.8.5), in preference to the Simple Applicant state machine (IEEE Std 802.1D, 1998 Edition, 12.8.6).

NOTE—End stations that implement only a) 2) and b) through d) are equivalent to the description of the Applicant Only Participant (IEEE Std 802.1D, 1998 Edition, 12.7.7); those that implement a) 3) and b) through d) are equivalent to the description of the Simple Applicant Participant (IEEE Std 802.1D, 1998 Edition, 12.7.8). Such end stations require only the ability to register membership of one or more VLANs, and revoke that membership at some later point in time; for this reason, there is no requirement to support the operation of the Registrar or Leave All state machines.
End stations that implement a) 1) and b) through g) are able to perform “source pruning” as described in 11.2.1.1; i.e., to suppress the transmission of frames destined for VLANs that currently have no membership. Consequently, such end stations need to support the full Applicant state machine, in combination with the Registrar and Leave All state machines.

11.4 Procedural model

11.4.1 Purpose

This section contains an example implementation of the GVRP application defined in this Clause. This “C” code description is included in order to demonstrate the structure of the GVRP application, and to show that a reasonably low overhead implementation can be constructed. The implementation has been designed with the intent of maximizing clarity and generality, not for compactness.

The example implementation is shown in two sections:

a) Header files for the GVRP application (11.4.2);
b) The GVRP application code (11.4.3).

The example implementation also references the GARP application independent “C” code contained in IEEE Std 802.1D, 1998 Edition:

c) Header files for the GARP application independent code (IEEE Std 802.1D, 1998 Edition, 13.2);
d) The GARP application independent code (IEEE Std 802.1D, 1998 Edition, 13.3).

The separation shown in the documentation of the application dependent (GVRP) and application independent (GARP) aspects of the implementation gives a clear illustration of what is involved in implementing additional applications using the same basic GARP state machines. The code is intended to be largely self-documenting, by means of in-line comments.

11.4.2 GVRP application-specific header files

11.4.2.1 gvr.h

```
/* gvr.h */
#ifndef gvr_h__
#define gvr_h__

#include “garp.h”
#include “gid.h”
#include “gip.h”

/******************************************************************************
 * GVR : GARP VLAN REGISTRATION APPLICATION : GARP ATTRIBUTES
 ******************************************************************************
 */

typedef unsigned Vlan_id;

typedef enum {All_attributes, Vlan_attribute}
             Attribute_type;

/******************************************************************************
 * GVR : GARP VLAN REGISTRATION APPLICATION : CREATION, DESTRUCTION
 ******************************************************************************
 */

extern Boolean gvr_create_gvr(int process_id, void **gvr);
   /*
    * Creates a new instance of GVR, allocating and initializing a control
    * block, returning True and a pointer to this instance if creation succeeds.
    * Also creates instances of GVD (the GARP VLAN database) and of GIP
    * (which controls information propagation).
    *
    * Ports are created by the system and added to GVR separately (see
    * gvr_added_port() and gvr_removed_port() below).
    *
    * The operating system supplied process_id is for use in subsequent calls
    * to operating system services. The system itself ensures the temporal
    * scope of process_id, guarding against timers yet to expire for destroyed
    * processes, etc. Although process_id will be implicitly supplied by many
    * if not most systems, it is made explicit in this implementation for
    * clarity.
    *
    */

extern void gvr_destroy_gvr(void *gvr);
   /*
    * Destroys an instance of GVR, destroying and deallocating the associated
    * instances of GVD and GIP, and any instances of GID remaining.
    */

extern void gvr_added_port(void *my_gvr, int port_no);
   /*
    * The system has created a new port for this application and added it to
    * the ring of GID ports. This function ensures that Static VLAN Entries
    * from the Permanent Database are represented in the GVD database (which
    * provides VLAN ID to GID index mapping) and have GID machines in the newly
    * added port (with the correct management control state). This can result
    * in the creation of new GID machines or modification of the state of
    * existing machines.
    *
    * Newly created ports are “connected” for the purpose of GARP information
    * propagation using the separate function gip_connect_port(). This should
    * be called after this function, gvr_added_port. It may cause GVRP/GIP
    * to propagate information from the static management controls through
    * other ports.
    *
    * It is assumed that new ports will be “connected” correctly before the
    * application continues as determined by the active topology of the network,
    * i.e., if stp_forwarding(port_no) gvr_connect_port(port_no);.
    *
    * As the system continues to run it should invoke gip_disconnect_port()
    * and gip_connect_port() as required to maintain the required connectivity.
    */

extern void gvr_removed_port(void *my_gvr, int port_no);
   /*
    * The system has removed and destroyed the GID port. This function should
    * provide any application-specific cleanup required.
    */

/******************************************************************************
 * GVR : GARP VLAN REGISTRATION APPLICATION : JOIN, LEAVE INDICATIONS
 ******************************************************************************
 */

extern void gvr_join_indication(void *my_gvr, void *my_port,
                              unsigned joining_gid_index);
   /*
    *
    */

extern void gvr_join_leave_propagated(void *my_gvr, void *my_port,
                                    unsigned gid_index);
   /*
    *
    */

extern void gvr_leave_indication(void *my_gvr, void *my_port,
                               unsigned leaving_gid_index);
   /*
    *
    */


/******************************************************************************
 * GVR : GARP VLAN REGISTRATION APPLICATION : PROTOCOL AND MANAGEMENT EVENTS
 ******************************************************************************
 */

extern void gvr_rcv(void *my_gvr, void *my_port, void *pdu);
   /*
    * Process an entire received pdu for this instance of GVR.
    */

extern void gvr_tx(void *my_gvr, void *my_port);
   /*
    * Transmit a pdu for this instance of GVR.
    */

#endif /* gvr_h__ */
```

11.4.2.2 gvd.h/* gvd.h */

```
#ifndef gvd_h__
#define gvd_h__

#include “sys.h”

/******************************************************************************
 * GVD : GARP VLAN DATABASE
 ******************************************************************************
 *
 * The GARP VLAN Database maps VLAN IDs into compact GID indexes and
 * vice versa. It contains VLAN ID to index mappings for all the VLAN IDs
 * dynamically registered (except when there is a database overflow, which
 * should be an event that rarely occurs through appropriate
 * sizing) and for all those for which static controls exist.
 *
 * Taken together with the GID machines for each port (which are identified
 * by the GID indexes provided by GVD), GVD logically provides the Static VLAN
 * Registration Entries and the Dynamic VLAN Registration Entries of the abstract
 * Filtering Database (see 8.11).
 *
 * Static VLAN Registration Entries are included in this database (and have GID
 * machines defined) on an as-needed basis as ports are added to the GVR
 * Application. This example implementation assumes that the necessary information
 * is taken from Static Filtering Entries kept in a Permanent Database outside
 * the example implementation. Static VLAN Entries can also be added, changed, or
 * removed as the running system is managed.
 *
 * VLAN Registration Entries are added and removed by GVRP. Note that a
 * single VLAN ID will only give rise to one entry in this database, and one
 * GID machine per port. That machine provides the functionality for both
 * the Static VLAN Entry and the VLAN Registration Entry.
 */

extern Boolean gvd_create_gvd(int max_vlans, void **gvd);
   /*
    * Creates a new instance of gvd, allocating space for up to max_vlans
    * VLAN IDs.
    *
    * Returns True if the creation succeeded together with a pointer to the
    * gvd information.
    */

extern void gvd_destroy_gvd(void *gvd);
   /*
    * Destroys the instance of gvd, releasing previously allocated database and
    * control space.
    */

extern Boolean gvd_find_entry(  void *my_gvd, Vlan_id key,
                              unsigned *found_at_index);

extern Boolean gvd_create_entry(void *my_gvd, Vlan_id key,
                              unsigned *created_at_index);

extern Boolean gvd_delete_entry(void *my_gvd,
                              unsigned  delete_at_index);

extern Boolean gvd_get_key(     void *my_gvd, unsigned index, Vlan_id *key);

#endif /* gvd_h__ */
```

11.4.2.3 gvf.h

```
/* gvf.h */
#ifndef gvf_h__
#define gvf_h__

#include “sys.h”
#include “prw.h”
#include “gvr.h”

/*******************************************************************************
 * GVF : GARP VLAN REGISTRATION APPLICATION PDU FORMATTING
 *******************************************************************************
 */

typedef struct
{  /*
    * This data structure saves the temporary state required to parse GVR
    * PDUs in particular. Gpdu provides a common basis for GARP Application
    * formatters, additional state can be added here as required by GVF.
    */

    Gpdu gpdu;

} Gvf;

typedef struct /* Gvf_msg_data */
{
    Attribute_type attribute;

    Gid_event   event;

    Vlan_id        key1;

} Gvf_msg;

extern void    gvf_rdmsg_init(Pdu *pdu, Gvf **gvf);

extern void    gvf_wrmsg_init(Gvf *gvf, Pdu  *pdu, int vlan_id);

extern Boolean gvf_rdmsg(     Gvf *gvf, Gvf_msg *msg);

extern Boolean gvf_wrmsg(     Gvf *gvf, Gvf_msg *msg);

#endif /* gvf_h__ */
```

11.4.2.4 vfdb.h

```
/* vfdb.h */
#ifndef vfdb_h__
#define vfdb_h__

#include “sys.h”

/*******************************************************************************
 * VFDB : VLAN ACTIVE FILTERING DATABASE INTERFACE
 *******************************************************************************
 */

extern void vfdb_filter( int port_no, unsigned vlan_id);
extern void vfdb_forward(int port_no, unsigned vlan_id);

#endif /* vfdb_h__ */
```

11.4.3 GVRP application code

11.4.3.1 gvr.c

```
/* gvr.c */

#include “gvr.h”
#include “gid.h”
#include “gip.h”
#include “garp.h”
#include “gvd.h”
#include “gvf.h”
#include “vfdb.h”


/******************************************************************************
 * GVR : GARP VLAN REGISTRATION APPLICATION : IMPLEMENTATION SIZING
 ******************************************************************************
 */

enum {Max_vlans = 100};

enum {Number_of_gid_machines = Max_vlans};

enum {Unused_index = Number_of_gid_machines};

/******************************************************************************
 * GVR : GARP VLAN REGISTRATION APPLICATION : CREATION, DESTRUCTION
 ******************************************************************************
 */

typedef struct /* gvr */
{
    Garp      g;

    unsigned  vlan_id;

    void     *gvd; /* VLAN Registration Entry Database */

    unsigned  number_of_gvd_entries;

    unsigned  last_gvd_used_plus1;

} Gvr;


Boolean gvr_create_gvr(int process_id, void **gvr)
{   /*
    */
    Gvr *my_gvr;

    if (!sysmalloc(sizeof(Gvr), &my_gvr))
        goto gvr_creation_failure;

    my_gvr->g.process_id   = process_id;
    my_gvr->g.gid          = NULL;

    if (!gip_create_gip(Number_of_gid_machines, &my_gvr->g.gip))
```

```
        goto gip_creation_failure;

    my_gvr->g.max_gid_index = Number_of_gid_machines - 1;
    my_gvr->g.last_gid_used = Zero;

    my_gvr->g.join_indication_fn  = gvr_join_indication;
    my_gvr->g.leave_indication_fn = gvr_leave_indication;
    my_gvr->g.join_propagated_fn  = gvr_join_leave_propagated;
    my_gvr->g.leave_propagated_fn = gvr_join_leave_propagated;
    my_gvr->g.transmit_fn         = gvr_tx;
    my_gvr->g.added_port_fn       = gvr_added_port;
    my_gvr->g.removed_port_fn     = gvr_removed_port;

    if (!gvd_create_gvd(Max_vlans, &my_gvr->gvd))
        goto gvd_creation_failure;

    my_gvr->number_of_gvd_entries = Max_vlans;
    my_gvr->last_gvd_used_plus1   = 0;

    *gvr = my_gvr;    return(True);
gvd_creation_failure: gip_destroy_gip(my_gvr->g.gip);
gip_creation_failure: sysfree(my_gvr);
gvr_creation_failure: return(False);
}


void gvr_destroy_gvr(Gvr *my_gvr)
{
    Gid *my_port;

    gvd_destroy_gvd(my_gvr->gvd);
    gip_destroy_gip(my_gvr->g.gip);

    while ((my_port = my_gvr->g.gid) != NULL)
        gid_destroy_port(&my_gvr->g, my_port->port_no);
}


void gvr_added_port(Gvr *my_gvr, int port_no)
{  /*
     * Query the Permanent Database for Static VLAN Entries with “Registration
     * Forbidden” or “Registration Fixed” for this Port. Repeat the following
     * steps until there are no more entries to be found.
     *
     * Check that the VLAN ID is represented in VLD. If not, create it, then
     * create GID machines for all the other Ports with control state “Normal
     * Registration” and create the GID machine for this Port. Change the
     * control state for this Port’s GID machine to forbidden or fixed as
     * required.
     *
    */
}


void gvr_removed_port(Gvr *my_gvr, int port_no)
{  /*
     * Provide any GVR specific cleanup or management alert functions for the
     * removed Port.
    */
}


/******************************************************************************
 * GVR : GARP VLAN REGISTRATION APPLICATION : JOIN, LEAVE INDICATIONS
 ******************************************************************************
 */

void gvr_join_indication(Gvr *my_gvr, Gid *my_port, unsigned gid_index)
{  /*
    *
    */
    Vlan_id      key;

    gvd_get_key(my_gvr->gvd, gid_index, &key);
    vfdb_forward(my_port->port_no, key);
}


void gvr_join_leave_propagated(Gvr *my_gvr, Gid *my_port, unsigned gid_index)
{  /*
    * Nothing to be done since, unlike GMR with its Forward All Unregistered
    * Port mode, a join indication on one Port does not cause filtering to be
    * instantiated on another.
    */
}


void gvr_leave_indication(Gvr *my_gvr, Gid *my_port, unsigned leaving_gid_index)
{  /*
    *
    */
    Vlan_id  key;

    gvd_get_key(my_gvr->gvd, leaving_gid_index, &key);
    vfdb_filter(my_port->port_no, key);
}

/******************************************************************************
 * GVR : GARP VLAN REGISTRATION APPLICATION : RECEIVE MESSAGE PROCESSING
 ******************************************************************************
 */

static void gvr_db_full(Gvr *my_gvr, Gid *my_port)
{  /*
    * Placeholder for management alert functions indicating registrations
    * for more VLANs have been received than can be accepted.
    */
}

static void gvr_rcv_msg(Gvr *my_gvr, Gid *my_port, Gvf_msg *msg)
{  /*
    * Process one received message.
    *
    * A LeaveAll message never causes an indication (join or leave directly),
    * even for the point-to-point link protocol enhancements (where an
    * ordinary Leave does). No further work is needed here.
    *
    * A LeaveAllRange message is currently treated exactly as a LeaveAll
    * (i.e., the range is ignored).
    *
    * All the remaining messages refer to a single attribute (i.e., a single
    * registered VLAN). Try to find a matching entry in the gvd database.
    * If one is found, dispatch the message to a routine that will
    * handle both the local GID effects and the GIP propagation to other Ports.
    *
    * If no entry is found, Leave and Empty messages can be discarded, but
    * JoinIn and JoinEmpty messages demand further treatment. First, an attempt
    * is made to create a new entry using free space (in the database, which
    * corresponds to a free GID machine set). If this fails, an attempt may be
    * made to recover space from a machine set that is in an unused or less
    * significant state. Finally, the database is considered full and the received
    * message is discarded.
    *
    * Once (if) an entry is found, Leave, Empty, JoinIn, and JoinEmpty are
    * all submitted to GID (gid_rcv_msg()), which will generate and propagate
    * Join or Leave indications as necessary.
    *
    * JoinIn and JoinEmpty may cause Join indications, which are then propagated
    * by GIP.
    *
    * On a shared medium, Leave and Empty will not give rise to indications
    * immediately. However, this routine does test for and propagate
    * Leave indications so that it can be used unchanged with a point-to-point
    * protocol enhancement.
    *
    */

    unsigned gid_index  = Unused_index;

    if (   (msg->event == Gid_rcv_leaveall)
        || (msg->event == Gid_rcv_leaveall_range))
    {
        gid_rcv_leaveall(my_port);
    }
    else
    {
        if (!gvd_find_entry(my_gvr->gvd, msg->key1, &gid_index))
        {
            if (   (msg->event == Gid_rcv_joinin)
                || (msg->event == Gid_rcv_joinempty))
            {
                if (!gvd_create_entry(my_gvr->gvd, msg->key1, &gid_index))
                {
                    if (gid_find_unused(&my_gvr->g, Zero, &gid_index))
                    {
                        gvd_delete_entry(my_gvr->gvd, gid_index);
                        (void) gvd_create_entry(my_gvr->gvd, msg->key1,
                                                &gid_index);
                    }
                    else
                        gvr_db_full(my_gvr, my_port);
        }   }   }

        if (gid_index != Unused_index)
            gid_rcv_msg(my_port, gid_index, msg->event);
}   }


void gvr_rcv(Gvr *my_gvr, Gid *my_port, Pdu *pdu)
{  /*
    * Process an entire received pdu for this instance of GVR: initialize
    * the GVF pdu parsing routine, and, while messages last, read and process
    * them one at a time.
    */
    Gvf      gvf;
    Gvf_msg  msg;

    gvf_rdmsg_init(&gvf, pdu);

    while (gvf_rdmsg(&gvf, &msg))
        gvr_rcv_msg(my_gvr, my_port, &msg);
}


/******************************************************************************
```

```
 * GVR : GARP VLAN REGISTRATION APPLICATION : TRANSMIT PROCESSING
 ******************************************************************************
 */

static void gvr_tx_msg(Gvr *my_gvr, unsigned gid_index, Gvf_msg *msg)
{  /*
    * Fill in msg fields for transmission.
    */

    if (msg->event == Gid_tx_leaveall)
    {
        msg->attribute = All_attributes;
    }
    else
    {
        msg->attribute = Vlan_attribute;

        gvd_get_key(my_gvr->gvd, gid_index, &msg->key1);
}   }

void gvr_tx(Gvr *my_gvr, Gid *my_port){  /*
    * Get and prepare a pdu for the transmission, if one is not available,
    * simply return; if there is more to transmit, GID will reschedule a call
    * to this function.
    *
    * Get messages to transmit from GID and pack them into the pdu using GVF
    * (GARP VLAN pdu Formatter).
    */
    Pdu         *pdu;
    Gvf          gvf;
    Gvf_msg      msg;
    Gid_event    tx_event;
    unsigned     gid_index;

    if ((tx_event = gid_next_tx(my_port, &gid_index)) != Gid_null)
    {
        if (syspdu_alloc(&pdu))
        {
            gvf_wrmsg_init(&gvf, pdu, my_gvr->vlan_id);

            do
            {
                msg.event = tx_event;

                gvr_tx_msg(my_gvr, gid_index, &msg);

                if (!gvf_wrmsg(&gvf, &msg))
                {
                    gid_untx(my_port);
                    break;
                }
            } while ((tx_event  = gid_next_tx(my_port, &gid_index))
                           != Gid_null);

            syspdu_tx(pdu, my_port->port_no);
}   }   }
```

12. VLAN Bridge Management

This clause defines the set of managed objects, and their functionality, that allow administrative configuration of VLANs.

This clause

a) Introduces the functions of management to assist in the identification of the requirements placed on Bridges for the support of management facilities.
b) Establishes the correspondence between the Processes used to model the operation of the Bridge (8.3) and the managed objects of the Bridge.
c) Specifies the management operations supported by each managed object.

12.1 Management functions

Management functions relate to the users' needs for facilities that support the planning, organization, supervision, control, protection, and security of communications resources, and account for their use. These facilities may be categorized as supporting the functional areas of Configuration, Fault, Performance, Security, and Accounting Management. Each of these is summarized in 12.1.1 through 12.1.5, together with the facilities commonly required for the management of communication resources, and the particular facilities provided in that functional area by Bridge Management.

12.1.1 Configuration Management

Configuration Management provides for the identification of communications resources, initialization, reset and close-down, the supply of operational parameters, and the establishment and discovery of the relationship between resources. The facilities provided by Bridge Management in this functional area are

a) The identification of all Bridges that together make up the Bridged LAN and their respective locations and, as a consequence of that identification, the location of specific end stations to particular individual LANs.
b) The ability to remotely reset, i.e., reinitialize, specified Bridges.
c) The ability to control the priority with which a Bridge Port transmits frames.
d) The ability to force a specific configuration of a spanning tree.
e) The ability to control the propagation of frames with specific group MAC Addresses to certain parts of the configured Bridged LAN.
f) The ability to identify the VLANs in use, and through which Ports of the Bridge and for which Protocols frames destined for a given VLAN may be received and/or forwarded.

12.1.2 Fault Management

Fault Management provides for fault prevention, detection, diagnosis, and correction. The facilities provided by Bridge Management in this functional area are

a) The ability to identify and correct Bridge malfunctions, including error logging and reporting.

12.1.3 Performance Management

Performance Management provides for evaluation of the behavior of communications resources and of the effectiveness of communication activities. The facilities provided by Bridge Management in this functional area are

a) The ability to gather statistics relating to performance and traffic analysis. Specific metrics include network utilization, frame forward, and frame discard counts for individual Ports within a Bridge.

12.1.4 Security Management

Security Management provides for the protection of resources. Bridge Management does not provide any specific facilities in this functional area.

12.1.5 Accounting Management

Accounting Management provides for the identification and distribution of costs and the setting of charges. Bridge Management does not provide any specific facilities in this functional area.

12.2 Managed objects

Managed objects model the semantics of management operations. Operations upon an object supply information concerning, or facilitate control over, the Process or Entity associated with that object.

The managed resources of a MAC Bridge are those of the Processes and Entities established in 8.3 of this standard and 12.2 of IEEE Std 802.1D, 1998 Edition. Specifically,

a) The Bridge Management Entity (12.4 and 8.13).
b) The individual MAC Entities associated with each Bridge Port (12.5, 8.2, 8.5, and 8.9).
c) The Forwarding Process of the MAC Relay Entity (12.6, 8.2, and 8.7).
d) The Filtering Database of the MAC Relay Entity (12.7 and 8.11).
e) The Bridge Protocol Entity (12.8 and 8.12 of this standard; Clauses 8 and 17 of IEEE Std 802.1D, 1998 Edition).
f) GARP Participants (Clause 12 of IEEE Std 802.1D, 1998 Edition);
g) GVRP participants (12.10, Clause 11);
h) GMRP participants (12.11, Clause 10 of IEEE Std 802.1D, 1998 Edition);
i) The MST Configuration Table (12.12).

The management of each of these resources is described in terms of managed objects and operations below.

NOTE—The values specified in this clause, as inputs and outputs of management operations, are abstract information elements. Questions of formats or encodings are a matter for particular protocols that convey or otherwise represent this information.

12.3 Data types

This subclause specifies the semantics of operations independent of their encoding in management protocol. The data types of the parameters of operations are defined only as required for that specification.

The following data types are used:

a) Boolean.
b) Enumerated, for a collection of named values.
c) Unsigned, for all parameters specified as “the number of” some quantity, and for Spanning Tree priority values that are numerically compared. When comparing Spanning Tree priority values, the lower number represents the higher priority value.
d) MAC Address.
e) Latin1 String, as defined by ANSI X3.159, for all text strings.

f) Time Interval, an Unsigned value representing a positive integral number of seconds, for all Spanning Tree protocol timeout parameters;
g) Counter, for all parameters specified as a “count” of some quantity. A counter increments and wraps with a modulus of 2 to the power of 64.
h) GARP Time Interval, an Unsigned value representing a positive integral number of centiseconds, for all GARP protocol time-out parameters.
i) Port Number, an Unsigned value assigned to a Port as part of a Port Identifier. Valid Port Numbers are in the range 1 through 4095;
j) Port Priority, an Unsigned value used to represent the priority component of a Port Identifier. Valid Port Priorities are in the range 0 through 240, in steps of 16;
k) Bridge Priority, an Unsigned value used to represent the priority component of a Bridge Identifier. Valid Bridge Priorities are in the range 0 through 61440, in steps of 4096.

12.4 Bridge Management Entity

The Bridge Management Entity is described in 8.13.

The objects which comprise this managed resource are

a) The Bridge Configuration (12.4.1).
b) The Port Configuration for each Port (12.4.2).

12.4.1 Bridge Configuration

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

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

a) Discover Bridge (12.4.1.1);
b) Read Bridge (12.4.1.2);
c) Set Bridge Name (12.4.1.3);
d) Reset Bridge (12.4.1.4).

12.4.1.1 Discover Bridge

12.4.1.1.1 Purpose

To solicit configuration information regarding the Bridge(s) in the Bridged LAN.

12.4.1.1.2 Inputs

a) Inclusion Range, a set of ordered pairs of specific MAC Addresses. Each pair specifies a range of MAC Addresses. A Bridge shall respond if and only if
    1) For one of the pairs, the numerical comparison of its Bridge Address with each MAC Address of the pair shows it to be greater than or equal to the first, and
    2) Less than or equal to the second, and
    3) Its Bridge Address does not appear in the Exclusion List parameter below.

The numerical comparison of one MAC Address with another, for the purpose of this operation, is achieved by deriving a number from the MAC Address according to the following procedure. The consecutive octets of the MAC Address are taken to represent a binary number; the first octet that would be transmitted on a LAN medium when the MAC Address is used in the source or destination fields of a MAC frame has the most significant value, the next octet the next most significant value. Within each octet the first bit of each octet is the least significant bit.

b) Exclusion List, a list of specific MAC Addresses.

12.4.1.1.3 Outputs

a) Bridge Address—the MAC Address for the Bridge from which the Bridge Identifiers used by the Spanning Tree Algorithm and Protocol, the Rapid Spanning Tree Protocol, and the Multiple Spanning Tree Protocol are derived (8.14.5, 13.23 of this standard; 8.5.1.3 and 17.17.2 of IEEE Std 802.1D, 1998 Edition).
b) Bridge Name—a text string of up to 32 characters, of locally determined significance.
c) Number of Ports—the number of Bridge Ports (MAC Entities).
d) Port Addresses—a list specifying the following for each Port:
   1) Port Number—the number of the Bridge Port (13.24 of this standard and 8.5.5.1 of IEEE Std 802.1D, 1998 Edition).
   2) Port Address—the specific MAC Address of the individual MAC Entity associated with the Port (8.14.2).
e) Uptime—count in seconds of the time elapsed since the Bridge was last reset or initialized (13.23.1 of this standard and 8.8.1 of IEEE Std 802.1D, 1998 Edition).

NOTE—Events that are considered to reset or initialize the Bridge include changing the MST Configuration Identifier.

12.4.1.2 Read Bridge

12.4.1.2.1 Purpose

To obtain general information regarding the Bridge.

12.4.1.2.2 Inputs

None.

12.4.1.2.3 Outputs

a) Bridge Address—the MAC Address for the Bridge from which the Bridge Identifiers used by the Spanning Tree Algorithm and Protocol and the Multiple Spanning Tree Protocol are derived (8.14.5, 13.23 of this standard, and 8.5.1.3 of IEEE Std 802.1D, 1998 Edition).
b) Bridge Name—a text string of up to 32 characters, of locally determined significance.
c) Number of Ports—the number of Bridge Ports (MAC Entities).
d) Port Addresses—a list specifying the following for each Port:
   1) Port Number (13.24 of this standard and 8.5.5.1 of IEEE Std 802.1D, 1998 Edition).
   2) Port Address—the specific MAC Address of the individual MAC Entity associated with the Port (8.14.2).
e) Uptime—count in seconds of the time elapsed since the Bridge was last reset or initialized (13.23.1 of this standard and 8.8.1 of IEEE Std 802.1D, 1998 Edition).

12.4.1.3 Set Bridge Name

12.4.1.3.1 Purpose

To associate a text string, readable by the Read Bridge operation, with a Bridge.

12.4.1.3.2 Inputs

a) Bridge Name—a text string of up to 32 characters.

12.4.1.3.3 Outputs

None.

12.4.1.4 Reset Bridge

12.4.1.4.1 Purpose

To reset the specified Bridge. The Filtering Database is cleared and initialized with the entries specified in the Permanent Database, and the Bridge Protocol Entity is initialized (13.23.1 of this standard and 8.8.1 of IEEE Std 802.1D, 1998 Edition).

12.4.1.4.2 Inputs

None.

12.4.1.4.3 Outputs

None.

12.4.2 Port configuration

The Port Configuration object models the operations that modify, or inquire about, the configuration of the Ports of a Bridge. There are a fixed set of Bridge Ports per Bridge (one for each MAC interface), and each is identified by a permanently allocated Port Number.

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

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

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

a) Read Port (12.4.2.1);
b) Set Port Name (12.4.2.2).

12.4.2.1 Read Port

12.4.2.1.1 Purpose

To obtain general information regarding a specific Bridge Port.

12.4.2.1.2 Inputs

a) Port Number—the number of the Bridge Port (13.24 of this standard, 8.5.5.1 of IEEE Std 802.1D, 1998 Edition).

12.4.2.1.3 Outputs

a) Port Name—a text string of up to 32 characters, of locally determined significance.
b) Port Type—the MAC Entity type of the Port (IEEE Std 802.3; ISO/IEC 8802-4; ISO/IEC 8802-5; ISO/IEC 8802-6; ISO/IEC 8802-9; IEEE Std 802.9a-1995; ISO/IEC 8802-11; ISO/IEC 8802-12 (IEEE Std 802.3 format); ISO/IEC 8802-12 (ISO/IEC 8802-5 format); ISO 9314; other).

12.4.2.2 Set Port Name

12.4.2.2.1 Purpose

To associate a text string, readable by the Read Port operation, with a Bridge Port.

12.4.2.2.2 Inputs

a) Port Number (13.24 of this standard, 8.5.5.1 of IEEE Std 802.1D, 1998 Edition).
b) Port Name—a text string of up to 32 characters.

12.4.2.2.3 Outputs

None.

12.5 MAC entities

The Management Operations and Facilities provided by the MAC Entities are those specified in the Layer Management standards of the individual MACs. A MAC Entity is associated with each Bridge Port.

12.6 Forwarding process

The Forwarding Process contains information relating to the forwarding of frames. Counters are maintained that provide information on the number of frames forwarded, filtered, and dropped due to error. Configuration data, defining how frame priority is handled, is maintained by the Forwarding Process.

The objects that comprise this managed resource are

a) The Port Counters (12.6.1).
b) The Priority Handling objects for each Port (12.6.2).
c) The Traffic Class Table for each Port (12.6.3).

12.6.1 The Port Counters

The Port Counters object models the operations that can be performed on the Port counters of the Forwarding Process resource. There are multiple instances (one for each VLAN for each MAC Entity) of the Port Counters object per Bridge.

The management operation that can be performed on the Port Counters is Read Forwarding Port Counters (12.6.1.1).

12.6.1.1 Read forwarding port counters

12.6.1.1.1 Purpose

To read the forwarding counters associated with a specific Bridge Port.

12.6.1.1.2 Inputs

a) Port Number (13.24 of this standard, 8.5.5.1 of IEEE Std 802.1D, 1998 Edition);
b) Optionally, VLAN Identifier (9.3.2.3).

If the VLAN Identifier parameter is supported, then the forwarding Port counters are maintained per VLAN per Port. If the parameter is not supported, then the forwarding Port counters are maintained per Port only.

12.6.1.1.3 Outputs

a) Frames Received—count of all valid frames received (including BPDUs, frames addressed to the Bridge as an end station and frames that were submitted to the Forwarding Process, 8.5).
b) Optionally, Octets Received—count of the total number of octets in all valid frames received (including BPDUs, frames addressed to the Bridge as an end station, and frames that were submitted to the Forwarding Process).
c) Discard Inbound—count of valid frames received that were discarded by the Forwarding Process (8.7).
d) Forward Outbound—count of frames forwarded to the associated MAC Entity (8.9).
e) Discard Lack of Buffers—count of frames that were to be transmitted through the associated Port but were discarded due to lack of buffers (8.7.3).
f) Discard Transit Delay Exceeded—count of frames that were to be transmitted but were discarded due to the maximum bridge transit delay being exceeded (buffering may have been available, 8.7.3).
g) Discard on Error—count of frames that were to be forwarded on the associated MAC but could not be transmitted (e.g., frame would be too large, IEEE Std 802.1D, 6.3.8).
h) If Ingress Filtering is supported (8.4.5), Discard on Ingress Filtering—count of frames that were discarded as a result of Ingress Filtering being enabled.
i) Optionally, Discard on Error Details—a list of 16 elements, each containing the source address of a frame and the reason why the frame was discarded (frame too large). The list is maintained as a circular buffer. The reasons for discard on error, at present, are
    1) Transmissible service data unit size exceeded; or
    2) Discard due to Ingress Filtering. The VID associated with the last discarded frame is recorded.

12.6.2 Priority handling

The Priority Handling object models the operations that can be performed upon, or inquire about, the Default User Priority parameter, the User Priority Regeneration Table parameter, and the Outbound Access Priority Table parameter for each Port. The operations that can be performed on this object are

a) Read Port Default User Priority (12.6.2.1);
b) Set Port Default User Priority (12.6.2.2);
c) Read Port User Priority Regeneration Table (12.6.2.3);
d) Set Port User Priority Regeneration Table (12.6.2.4);
e) Read Outbound Access Priority Table (12.6.2.5).

12.6.2.1 Read Port Default User Priority

12.6.2.1.1 Purpose

To read the current state of the Default User Priority parameter (6.4 of IEEE Std 802.1D, 1998 Edition) for a specific Bridge Port.

12.6.2.1.2 Inputs

a) Port number.

12.6.2.1.3 Outputs

a) Default User Priority value—Integer in range 0–7.

12.6.2.2 Set Port Default User Priority

12.6.2.2.1 Purpose

To set the current state of the Default User Priority parameter (6.4 of IEEE Std 802.1D, 1998 Edition) for a specific Bridge Port.

12.6.2.2.2 Inputs

a) Port number;
b) Default User Priority value—Integer in range 0–7.

12.6.2.2.3 Outputs

None.

12.6.2.3 Read Port User Priority Regeneration Table

12.6.2.3.1 Purpose

To read the current state of the User Priority Regeneration Table parameter (8.5.1) for a specific Bridge Port.

12.6.2.3.2 Inputs

a) Port number.

12.6.2.3.3 Outputs

a) Regenerated User Priority value for Received User Priority 0—Integer in range 0–7.
b) Regenerated User Priority value for Received User Priority 1—Integer in range 0–7.
c) Regenerated User Priority value for Received User Priority 2—Integer in range 0–7.
d) Regenerated User Priority value for Received User Priority 3—Integer in range 0–7.
e) Regenerated User Priority value for Received User Priority 4—Integer in range 0–7.
f) Regenerated User Priority value for Received User Priority 5—Integer in range 0–7.
g) Regenerated User Priority value for Received User Priority 6—Integer in range 0–7.
h) Regenerated User Priority value for Received User Priority 7—Integer in range 0–7.

12.6.2.4 Set Port User Priority Regeneration Table

12.6.2.4.1 Purpose

To set the current state of the User Priority Regeneration Table parameter (8.5.1) for a specific Bridge Port.

12.6.2.4.2 Inputs

a) Port number;
b) Regenerated User Priority value for Received User Priority 0—Integer in range 0–7.
c) Regenerated User Priority value for Received User Priority 1—Integer in range 0–7.
d) Regenerated User Priority value for Received User Priority 2—Integer in range 0–7.
e) Regenerated User Priority value for Received User Priority 3—Integer in range 0–7.
f) Regenerated User Priority value for Received User Priority 4—Integer in range 0–7.
g) Regenerated User Priority value for Received User Priority 5—Integer in range 0–7.
h) Regenerated User Priority value for Received User Priority 6—Integer in range 0–7.
i) Regenerated User Priority value for Received User Priority 7—Integer in range 0–7.

12.6.2.4.3 Outputs

None.

12.6.2.5 Read Outbound Access Priority Table

12.6.2.5.1 Purpose

To read the state of the Outbound Access Priority Table parameter (Table 8-3) for a specific Bridge Port.

12.6.2.5.2 Inputs

a) Port number.

12.6.2.5.3 Outputs

a) Access Priority value for User Priority 0—Integer in range 0–7.
b) Access Priority value for User Priority 1—Integer in range 0–7.
c) Access Priority value for User Priority 2—Integer in range 0–7.
d) Access Priority value for User Priority 3—Integer in range 0–7.
e) Access Priority value for User Priority 4—Integer in range 0–7.
f) Access Priority value for User Priority 5—Integer in range 0–7.
g) Access Priority value for User Priority 6—Integer in range 0–7.
h) Access Priority value for User Priority 7—Integer in range 0–7.

12.6.3 Traffic Class Table

The Traffic Class Table object models the operations that can be performed upon, or inquire about, the current contents of the Traffic Class Table (8.7.3) for a given Port. The operations that can be performed on this object are Read Port Traffic Class Table and Set Port Traffic Class Table.

12.6.3.1 Read Port Traffic Class Table

12.6.3.1.1 Purpose

To read the contents of the Traffic Class Table (8.7.3) for a given Port.

12.6.3.1.2 Inputs

a) Port Number.

12.6.3.1.3 Outputs

a) The number of Traffic Classes, in the range 1 through 8, supported on the Port;
b) For each value of Traffic Class supported on the Port, the value of the Traffic Class in the range 0 through 7, and the set of user_priority values assigned to that Traffic Class.

12.6.3.2 Set Port Traffic Class Table

12.6.3.2.1 Purpose

To set the contents of the Traffic Class Table (8.7.3) for a given Port.

12.6.3.2.2 Inputs

a) Port number;
b) For each value of Traffic Class supported on the Port, the value of the Traffic Class in the range 0 through 7, and the set of user_priority values assigned to that Traffic Class.

NOTE—If a Traffic Class value greater than the largest Traffic Class available on the Port is specified, then the value applied to the Traffic Class Table is the largest available Traffic Class.

12.6.3.2.3 Outputs

None.

12.7 Filtering Database

The Filtering Database is described in 8.11. It contains filtering information used by the Forwarding Process (8.7) in deciding through which Ports of the Bridge frames should be forwarded.

The objects that comprise this managed resource are

a) The Filtering Database (12.7.1);
b) The Static Filtering Entries (12.7.2);
c) The Dynamic Filtering Entries (12.7.3);
d) The Group Registration Entries (12.7.4);
e) The Static VLAN Registration Entries (12.7.5);
f) The Dynamic VLAN Registration Entries (12.7.5);
g) The Permanent Database (12.7.6).

12.7.1 The Filtering Database

The Filtering Database object models the operations that can be performed on, or affect, the Filtering Database as a whole. There is a single Filtering Database object per Bridge.

The management operations that can be performed on the Database are:

a) Read Filtering Database (12.7.1.1);
b) Set Filtering Database Ageing Time (12.7.1.2);
c) Read Permanent Database (12.7.6.1);

d) Create Filtering Entry (12.7.7.1);
e) Delete Filtering Entry (12.7.7.2);
f) Read Filtering Entry (12.7.7.3);
g) Read Filtering Entry Range (12.7.7.4).

12.7.1.1 Read Filtering Database

12.7.1.1.1 Purpose

To obtain general information regarding the Bridge's Filtering Database.

12.7.1.1.2 Inputs

None.

12.7.1.1.3 Outputs

a) Filtering Database Size—the maximum number of entries that can be held in the Filtering Database.
b) Number of Static Filtering Entries—the number of Static Filtering Entries (8.11.1) currently in the Filtering Database;
c) Number of Dynamic Filtering Entries—the number of Dynamic Filtering Entries (8.11.3) currently in the Filtering Database;
d) Number of Static VLAN Registration Entries—the number of Static VLAN Registration Entries (8.11.2) currently in the Filtering Database;
e) Number of Dynamic VLAN Registration Entries—the number of Dynamic VLAN Registration Entries (8.11.5) currently in the Filtering Database.
f) Ageing Time—for ageing out Dynamic Filtering Entries when the Port associated with the entry is in the Forwarding state (8.11.3).
g) If Extended Filtering Services are supported, Number of Group Registration Entries—the number of Group Registration Entries (8.11.4) currently in the Filtering Database;

12.7.1.2 Set Filtering Database Ageing Time

12.7.1.2.1 Purpose

To set the ageing time for Dynamic Filtering Entries (8.11.3).

12.7.1.2.2 Inputs

a) Ageing Time.

12.7.1.2.3 Outputs

None.

12.7.2 A Static Filtering Entry

A Static Filtering Entry object models the operations that can be performed on a single Static Filtering Entry in the Filtering Database. The set of Static Filtering Entry objects within the Filtering Database changes only under management control.

A Static Filtering Entry object supports the following operations:

a) Create Filtering Entry (12.7.7.1);

b) Delete Filtering Entry (12.7.7.2);
c) Read Filtering Entry (12.7.7.3);
d) Read Filtering Entry Range (12.7.7.4).

12.7.3 A Dynamic Filtering Entry

A Dynamic Filtering Entry object models the operations that can be performed on a single Dynamic Filtering Entry (i.e., one that is created by the Learning Process as a result of the observation of network traffic) in the Filtering Database.

A Dynamic Filtering Entry object supports the following operations:

a) Delete Filtering Entry (12.7.7.2);
b) Read Filtering Entry (12.7.7.3);
c) Read Filtering Entry Range (12.7.7.4).

12.7.4 A Group Registration Entry

A Group Registration Entry object models the operations that can be performed on a single Group Registration Entry in the Filtering Database. The set of Group Registration Entry objects within the Filtering Database changes only as a result of GARP protocol exchanges.

A Group Registration Entry object supports the following operations:

a) Read Filtering Entry (12.7.7.3);
b) Read Filtering Entry Range (12.7.7.4).

12.7.5 A VLAN Registration Entry

A VLAN Registration Entry object models the operations that can be performed on a single VLAN Registration Entry in the Filtering Database. The set of VLAN Registration Entry objects within the Filtering Database changes under management control and also as a result of GARP protocol exchanges.

12.7.5.1 Static VLAN Registration Entry object

A Static VLAN Registration Entry object supports the following operations:

a) Create Filtering Entry (12.7.7.1);
b) Delete Filtering Entry (12.7.7.2);
c) Read Filtering Entry (12.7.7.3);
d) Read Filtering Entry Range (12.7.7.4).

12.7.5.2 Dynamic VLAN Registration Entry object

A Dynamic VLAN Registration Entry object supports the following operations:

a) Read Filtering Entry (12.7.7.3);
b) Read Filtering Entry Range (12.7.7.4).

12.7.6 Permanent Database

The Permanent Database object models the operations that can be performed on, or affect, the Permanent Database. There is a single Permanent Database per Filtering Database.

The management operations that can be performed on the Permanent Database are

a) Read Permanent Database (12.7.6.1);
b) Create Filtering Entry (12.7.7.1);
c) Delete Filtering Entry (12.7.7.2);
d) Read Filtering Entry (12.7.7.3);
e) Read Filtering Entry Range (12.7.7.4).

12.7.6.1 Read Permanent Database

12.7.6.1.1 Purpose

To obtain general information regarding the Permanent Database (8.11.10).

12.7.6.1.2 Inputs

None.

12.7.6.1.3 Outputs

a) Permanent Database Size—maximum number of entries that can be held in the Permanent Database.
b) Number of Static Filtering Entries—number of Static Filtering Entries (8.11.1) currently in the Permanent Database;
c) Number of Static VLAN Registration Entries—number of Static VLAN Registration Entries (8.11.2) currently in the Permanent Database.

12.7.7 General Filtering Database operations

In these operations on the Filtering Database, the operation parameters make use of VID values, even when operating upon a Dynamic Filtering Entry (8.11.3) whose structure carries an FID rather than a VID. In this case, the value used in the VID parameter can be any VID that has been allocated to the FID concerned (8.11.7).

12.7.7.1 Create Filtering Entry

12.7.7.1.1 Purpose

To create or update a Static Filtering Entry (8.11.1) or Static VLAN Registration Entry (8.11.2) in the Filtering Database or Permanent Database. Only static entries may be created in the Filtering Database or Permanent Database.

12.7.7.1.2 Inputs

a) Identifier—Filtering Database or Permanent Database.
b) Address—MAC Address of the entry (not present in VLAN Registration Entries).
c) VID—VLAN Identifier of the entry.
d) Port Map—a set of control indicators, one for each Port, as specified in 8.11.1 and 8.11.2.

12.7.7.1.3 Outputs

None.

12.7.7.2 Delete Filtering Entry

12.7.7.2.1 Purpose

To delete a Filtering Entry or VLAN Registration Entry from the Filtering Database or Permanent Database.

12.7.7.2.2 Inputs

a) Identifier—Filtering Database or Permanent Database.
b) Address—MAC Address of the desired entry (not present in VLAN Registration Entries).
c) VID—VLAN Identifier of the entry.

12.7.7.2.3 Outputs

None.

12.7.7.3 Read Filtering Entry

12.7.7.3.1 Purpose

To read a Filtering Entry, Group Registration Entry, or VLAN Registration Entry from the Filtering or Permanent Databases.

12.7.7.3.2 Inputs

a) Identifier—Filtering Database or Permanent Database.
b) Address—MAC Address of the desired entry (not present in VLAN Registration Entries).
c) VID—VLAN Identifier of the entry.
d) Type—Static or Dynamic entry.

12.7.7.3.3 Outputs

a) Address—MAC Address of the desired entry (not present in VLAN Registration Entries).
b) VID—VLAN Identifier of the entry.
c) Type—Static or Dynamic entry.
d) Port Map—a set of control indicators as appropriate for the entry, as specified in 8.11.1 through 8.11.5.

12.7.7.4 Read Filtering Entry range

12.7.7.4.1 Purpose

To read a range of Filtering Database entries (of any type) from the Filtering or Permanent Databases.

Since the number of values to be returned in the requested range may have exceeded the capacity of the service data unit conveying the management response, the returned entry range is identified. The indices that define the range take on values from zero up to Filtering Database Size minus one.

12.7.7.4.2 Inputs

a) Identifier—Filtering Database or Permanent Database.
b) Start Index—inclusive starting index of the desired entry range.
c) Stop Index—inclusive ending index of the desired range.

12.7.7.4.3 Outputs

a) Start Index—inclusive starting index of the returned entry range.
b) Stop Index—inclusive ending index of the returned entry range.
c) For each index returned:
   1) Address—MAC Address of the desired entry (not present in VLAN Registration Entries).
   2) VID—VLAN Identifier of the entry.
   3) Type—Static or Dynamic entry.
   4) Port Map—a set of control indicators as appropriate for the entry, as specified in 8.11.1 through 8.11.5.

12.8 Bridge Protocol Entity

The Bridge Protocol Entity is described in 8.12 and Clause 13 of this standard, and Clauses 8 and 17 of IEEE Std 802.1D, 1998 Edition.

The objects that comprise this managed resource are

a) The Protocol Entity itself.
b) The Ports under its control.

12.8.1 The Protocol Entity

The Protocol Entity object models the operations that can be performed upon, or inquire about, the operation of the Spanning Tree Algorithm and Protocol. There is a single Protocol Entity per Bridge; it can, therefore, be identified as a single fixed component of the Protocol Entity resource.

The management operations that can be performed on the Protocol Entity are

a) Read CIST Bridge Protocol Parameters (12.8.1.1);
b) Read MSTI Bridge Protocol Parameters (12.8.1.2);
c) Set CIST Bridge Protocol Parameters (12.8.1.3).
d) Set MSTI Bridge Protocol Parameters (12.8.1.4).

12.8.1.1 Read CIST Bridge Protocol Parameters

12.8.1.1.1 Purpose

To obtain information regarding the Bridge's Bridge Protocol Entity for the CIST.

12.8.1.1.2 Inputs

None.

12.8.1.1.3 Outputs

a) Bridge Identifier—as defined in 8.5.3 of IEEE Std 802.1D, 1998 Edition. The Bridge Identifier for the CIST.
b) Time Since Topology Change—in an STP Bridge, the count in seconds of the time elapsed since the Topology Change flag parameter for the Bridge (8.5.3.12 of IEEE Std 802.1D, 1998 Edition) was last True, or in an RSTP or MSTP Bridge, the count in seconds since tcWhile timer (13.21 of this standard or 17.15.7 of IEEE Std 802.1D, 1998 Edition) for any Port was non-zero.
c) Topology Change Count—in an STP Bridge, the count of the times the Topology Change flag parameter for the Bridge has been set (i.e., transitioned from False to True) since the Bridge was

powered on or initialized, or in an RSTP or MSTP Bridge, the count of times that there has been at least one non-zero tcWhile timer (13.21 of this standard or 17.15.7 of IEEE Std 802.1D, 1998 Edition).

d) Topology Change—in an STP Bridge, the value of the Topolgy Change parameter (8.5.3.12 of IEEE Std 802.1D, 1998 Edition), or in an RSTP or MSTP Bridge, asserted if the tcWhile timer for any Port for the CIST (13.21 of this standard or 17.15.7 of IEEE Std 802.1D, 1998 Edition) is non-zero.
e) Designated Root (13.23.3 of this standard, 8.5.3.1 and 17.18.7 of IEEE Std 802.1D, 1998 Edition).
f) Root Path Cost (13.23.3 of this standard, 8.5.3.1 and 17.18.7 of IEEE Std 802.1D, 1998 Edition).
g) Root Port (13.23.5 of this standard, 8.5.3.3 and 17.17.5 of IEEE Std 802.1D, 1998 Edition).
h) Max Age (13.23.7 of this standard, 8.5.3.4 and 17.18.18 of IEEE Std 802.1D, 1998 Edition).
i) Forward Delay (13.23.7 of this standard, 8.5.3.6 and 17.16.2 of IEEE Std 802.1D, 1998 Edition).
j) Bridge Max Age (13.23.4 of this standard, 8.5.3.7 and 17.17.4 of IEEE Std 802.1D, 1998 Edition).
k) Bridge Hello Time (13.23.4 of this standard, 8.5.3.9 and 17.17.4 of IEEE Std 802.1D, 1998 Edition). This parameter is present only if the Bridge supports STP or RSTP.
l) Bridge Forward Delay (13.23.4 of this standard, 8.5.3.10 and 17.17.4 of IEEE Std 802.1D, 1998 Edition).
m) Hold Time (8.5.3.14 of IEEE Std 802.1D, 1998 Edition) or Transmission Limit (TxHoldCount in 13.22 of this standard and 17.16.6 of IEEE Std 802.1D, 1998 Edition).

The following parameter is present only if the Bridge supports RSTP or MSTP:

n) forceVersion—the value of the Force Protocol Version parameter for the Bridge (13.6.2 of this standard and 17.16.1 of IEEE Std 802.1D, 1998 Edition)

The following additional parameters are present only if the Bridge supports MSTP:

o) CIST Regional Root Identifier (13.16.4). The Bridge Identifier of the current CIST Regional Root.
p) CIST Path Cost. The CIST path cost from the transmitting Bridge to the CIST Regional Root.

12.8.1.2 Read MSTI Bridge Protocol Parameters

12.8.1.2.1 Purpose

In an MST Bridge, to obtain information regarding the Bridge's Bridge Protocol Entity for the specified Spanning Tree instance.

12.8.1.2.2 Inputs

a) MSTID—Identifies the set of parameters that will be returned. For Bridges that support MSTP, this parameter is the identifier of the spanning tree for which the operation is being performed. This parameter takes a value in the range 1 through 4094.

12.8.1.2.3 Outputs

a) MSTID—Identifies the set of parameters that are being returned. This parameter is the identifier of the MST Instance for which the operation is being performed.
b) Bridge Identifier—as defined in 13.23.2. The Bridge Identifier for the spanning tree instance identified by the MSTID.
c) Time Since Topology Change—count in seconds of the time elapsed since tcWhile (13.21) was last non-zero for any Port for the given MSTI.
d) Topology Change Count—count of the times tcWhile (13.21) has been non-zero for any Port for the given MSTI since the Bridge was powered on or initialized.
e) Topology Change (tcWhile, 13.21). True if tcWhile is non-zero for any Port for the given MST.

f) Designated Root (13.23.9). The Bridge Identifier of the Root Bridge for the spanning tree instance identified by the MSTID.
g) Root Path Cost (13.23.9). The path cost from the transmitting Bridge to the Root Bridge for the spanning tree instance identified by the MSTID.
h) Root Port (13.23.11). The Root Port for the spanning tree instance identified by the MSTID.

12.8.1.3 Set CIST Bridge Protocol Parameters

12.8.1.3.1 Purpose

To modify parameters in the Bridge's Bridge Protocol Entity for the CIST, in order to force a configuration of the spanning tree and/or tune the reconfiguration time to suit a specific topology. In RSTP and MSTP implementations, this operation causes these values to be set for all Ports of the Bridge.

12.8.1.3.2 Inputs

a) Bridge Max Age—the new value (13.23.4 of this standard, 8.5.3.8 and 17.17.4 of IEEE Std 802.1D, 1998 Edition).
b) Bridge Hello Time—the new value (13.23.4 of this standard, 8.5.3.9 and 17.17.4 of IEEE Std 802.1D, 1998 Edition) This parameter is present only if the Bridge supports STP or RSTP.
c) Bridge Forward Delay—the new value (13.23.4 of this standard, 8.5.3.10 and 17.17.4 of IEEE Std 802.1D, 1998 Edition).
d) Bridge Priority—the new value of the priority part of the Bridge Identifier (13.23.2 of this standard, 8.5.3.7 of IEEE Std 802.1D, 1998 Edition) for the CIST.

The following parameter is present only if the Bridge supports RSTP or MSTP:

e) forceVersion—the new value of the Force Protocol Version parameter for the Bridge (13.6.2 of this standard and 17.16.1 of IEEE Std 802.1D, 1998 Edition).

12.8.1.3.3 Outputs

a) Operation status. This takes one of the following values:
   1) Operation rejected due to invalid Bridge Priority value (12.3); or
   2) Operation rejected due to the specified Max Age, Hello Time, or Forward Delay values being outside the range specified by IEEE Std 802.1D (see 12.8.1.3.4); or
   3) Operation rejected due to the specified Max Age, Hello Time, or Forward Delay values not being in compliance with the requirements of IEEE Std 802.1D (see 12.8.1.3.4); or
   4) Operation accepted.

12.8.1.3.4 Procedure

In the following description, the references to Bridge Hello Time apply only to Bridges that support STP or RSTP.

The input parameter values are checked for compliance with 8.10.2 of IEEE Std 802.1D, 1998 Edition (STP Bridges), 17.28 of IEEE Std 802.1D, 1998 Edition (RSTP Bridges), or their definitions in Clause 13. If they do not comply, or the value of Bridge Max Age or Bridge Forward Delay is less than the lower limit of the range specified in Table 8-3 of IEEE Std 802.1D, 1998 Edition (STP Bridges), or Table 17-5 of IEEE Std 802.1D, 1998 Edition (RSTP and MSTP Bridges), then no action shall be taken for any of the supplied parameters. If the value of any of Bridge Max Age, Bridge Forward Delay, or Bridge Hello Time is outside the range specified in Table 8-3 of IEEE Std 802.1D, 1998 Edition (STP Bridges), or Table 17-5 of IEEE Std 802.1D, 1998 Edition (RSTP and MSTP Bridges), then the Bridge need not take action.

Otherwise:

a) The Bridge's Bridge Max Age, Bridge Hello Time, and Bridge Forward Delay parameters are set to the supplied values.
b) In STP Bridges, the Set Bridge Priority procedure (8.8.4 of IEEE Std 802.1D, 1998 Edition) is used to set the priority part of the Bridge Identifier to the supplied value.
c) In RSTP and MSTP Bridges, the priority component of the Bridge Identifier (13.23.4 of this standard, 17.17.3 of IEEE Std 802.1D, 1998 Edition) is updated using the supplied value. For all Ports of the Bridge, the reselect for the CIST parameter (13.24 of this standard, 17.18.29 of IEEE Std 802.1D, 1998 Edition) is set TRUE, and the selected parameter for the CIST (13.24 of this standard, 17.18.31 of IEEE Std 802.1D, 1998 Edition) is set FALSE.

12.8.1.4 Set MSTI Bridge Protocol Parameters

12.8.1.4.1 Purpose

To modify parameters in the Bridge's Bridge Protocol Entity for the specified Spanning Tree instance, in order to force a configuration of the spanning tree and/or tune the reconfiguration time to suit a specific topology.

12.8.1.4.2 Inputs

a) MSTID—Identifies the set of parameters upon which the operation will be performed.
b) Bridge Priority—the new value of the priority part of the Bridge Identifier (13.23.2) for the Spanning Tree instance identified by the MSTID.

12.8.1.4.3 Outputs

a) Operation status. This takes one of the following values:
   1) Operation rejected due to invalid Bridge Priority value (12.3); or
   2) Operation rejected due to invalid MSTID (i.e, there is currently no MST Instance with that value of MSTID supported by the Bridge); or
   3) Operation accepted.

12.8.1.4.4 Procedure

The Bridge Priority parameter value is checked for compliance with its definition in Clause 13. If it does not comply, then no action shall be taken.

Otherwise, the priority part of the Bridge Identifier is set to the supplied value for the specified Spanning Tree instance.

12.8.2 Bridge Port

A Bridge Port object models the operations related to an individual Bridge Port in relation to the operation of the Spanning Tree Algorithm and Protocol. There are a fixed set of Bridge Ports per Bridge; each can, therefore, be identified by a permanently allocated Port Number, as a fixed component of the Protocol Entity resource.

The management operations that can be performed on a Bridge Port are

a) Read CIST Port Parameters (12.8.2.1);
b) Read MSTI Port Parameters (12.8.2.2);
c) Force CIST Port State (12.8.2.3);

d) Force MSTI Port State (12.8.2.4);
e) Set CIST Port Parameters (12.8.2.5);
f) Set MSTI Port Parameters (12.8.2.6);
g) Force BPDU Migration Check (12.8.2.7).

12.8.2.1 Read CIST Port Parameters

12.8.2.1.1 Purpose

To obtain information regarding a specific Port within the Bridge's Bridge Protocol Entity, for the CIST.

12.8.2.1.2 Inputs

a) Port Number—the number of the Bridge Port.

12.8.2.1.3 Outputs

a) Uptime—count in seconds of the time elapsed since the Port was last reset or initialized (BEGIN, 13.23).
b) State—the current state of the Port (i.e., Disabled, Listening, Learning, Forwarding, or Blocking) (8.4, 13.34 of this standard, 8.4, 8.5.5.2, and 17.5 of IEEE Std 802.1D, 1998 Edition).
c) Port Identifier—the unique Port identifier comprising two parts, the Port Number and the Port Priority field (13.24.8 of this standard, 8.5.5.1 and 17.18.16 of IEEE Std 802.1D, 1998 Edition).
d) Path Cost (8.5.5.3 and 17.16.5 of IEEE Std 802.1D, 1998 Edition).
e) Designated Root (13.24.8 of this standard, 8.5.5.4 and 17.18.17 of IEEE Std 802.1D, 1998 Edition).
f) Designated Cost (13.24.8 of this standard, 8.5.5.5 and 17.18.17 of IEEE Std 802.1D, 1998 Edition).
g) Designated Bridge (13.24.8 of this standard, 8.5.5.6 and 17.18.17 of IEEE Std 802.1D, 1998 Edition).
h) Designated Port (13.24.8 of this standard, 8.5.5.7 and 17.18.17 of IEEE Std 802.1D, 1998 Edition).
i) Topology Change Acknowledge (8.5.5.8 and 17.18.37 of IEEE Std 802.1D, 1998 Edition).
j) Hello Time (13.24.9 of this standard, 8.5.3.5 and 17.18.18 of IEEE Std 802.1D, 1998 Edition).
k) adminEdgePort (18.3.3 of IEEE Std 802.1D, 1998 Edition). Present in implementations that support the identification of edge ports.
l) operEdgePort (18.3.4 of IEEE Std 802.1D, 1998 Edition). Present in implementations that support the identification of edge ports.
m) MAC Enabled - the current state of the MAC Enabled parameter (6.4.2 of IEEE Std 802.1D, 1998 Edition). Present if the implementation supports the MAC Enabled parameter.
n) MAC Operational - the current state of the MAC Operational parameter (6.4.2 of IEEE Std 802.1D, 1998 Edition). Present if the implementation supports the MAC Operational parameter.
o) adminPointToPointMAC - the current state of the adminPointToPointMAC parameter (6.4.3 of IEEE Std 802.1D, 1998 Edition). Present if the implementation supports the adminPointToPointMAC parameter.
p) operPointToPointMAC - the current state of the operPointToPointMAC parameter (6.4.3 of IEEE Std 802.1D, 1998 Edition). Present if the implementation supports the operPointToPointMAC parameter.

The following additional parameters are present only if the Bridge supports MSTP:

q) CIST Regional Root Identifier (13.16.4). The Bridge Identifier of the current CIST Regional Root.
r) CIST Path Cost. The CIST path cost from the transmitting Bridge to the CIST Regional Root.
s) Port Hello Time. The administrative value of Hello Time for the Port (13.22).

12.8.2.2 Read MSTI Port Parameters

12.8.2.2.1 Purpose

To obtain information regarding a specific Port within the Bridge's Bridge Protocol Entity, for a given MSTI.

12.8.2.2.2 Inputs

a) Port Number—the number of the Bridge Port.
b) MSTID—Identifies the set of parameters that will be returned, in the range 1 through 4094.

12.8.2.2.3 Outputs

a) MSTID—Identifies the set of parameters that are being returned. This parameter is the identifier of the spanning tree for which the operation is being performed.
b) Uptime—count in seconds of the time elapsed since the Port was last reset or initialized (BEGIN, 13.23).
c) State—the current state of the Port (i.e., Disabled, Listening, Learning, Forwarding, or Blocking) (8.4, 13.34).
d) Port Identifier—the unique Port identifier comprising two parts, the Port Number and the Port Priority field (13.24.17).
e) Path Cost (13.36.1).
f) Designated Root (13.24.17).
g) Designated Cost (13.24.17).
h) Designated Bridge (13.24.17).
i) Designated Port (13.24.17).

12.8.2.3 Force CIST port state

12.8.2.3.1 Purpose

To set the Administrative Bridge Port State (see 17.5 of IEEE Std 802.1D, 1998 Edition) for the specified Port to Disabled or Enabled, for the CIST.

12.8.2.3.2 Inputs

a) Port Number—the number of the Bridge Port.
b) State—either Disabled or Enabled.

12.8.2.3.3 Outputs

None.

12.8.2.3.4 Procedure

In Bridges that support STP, if the selected state is Disabled, the Disable Port procedure (8.8.3 of IEEE Std 802.1D, 1998 Edition) is used for the specified Port for the specified Spanning Tree instance. If the selected state is Enabled, the Enable Port procedure (8.8.2 of IEEE Std 802.1D, 1998 Edition) is used.

In Bridges that support RSTP or MSTP, the effect of changing this parameter is as defined in 17.5 of IEEE Std 802.1D, 1998 Edition.

12.8.2.4 Force MSTI port state

12.8.2.4.1 Purpose

To set the Administrative Bridge Port State (see 17.5 of IEEE Std 802.1D, 1998 Edition) for the specified Port to Disabled or Enabled, for the specified Spanning Tree instance.

12.8.2.4.2 Inputs

a) MSTID—Identifies the set of parameters upon which the operation will be performed. The identifier of the spanning tree for which the operation is being performed. The parameter can take a value in the range 1 through 4094.
b) Port Number—the number of the Bridge Port.
c) State—either Disabled or Enabled.

12.8.2.4.3 Outputs

None.

12.8.2.4.4 Procedure

The effect of changing this parameter is as defined in 17.5 of IEEE Std 802.1D, 1998 Edition.

12.8.2.5 Set CIST port parameters

12.8.2.5.1 Purpose

To modify parameters for a Port in the Bridge's Bridge Protocol Entity in order to force a configuration of the spanning tree for the CIST.

12.8.2.5.2 Inputs

a) Port Number—the number of the Bridge Port.
b) Path Cost—the new value (13.36.1 of this standard, 8.5.5.3 and 17.16.5 of IEEE Std 802.1D, 1998 Edition).
c) Port Priority—the new value of the priority field for the Port Identifier (13.24.8 of this standard, 8.5.5.1 and 17.18.7 of IEEE Std 802.1D, 1998 Edition).
d) adminEdgePort—the new value of the adminEdgePort parameter (18.3.3 of IEEE Std 802.1D, 1998 Edition). Present in implementations that support the identification of edge ports.
e) MAC Enabled - the new value of the MAC Enabled parameter (6.4.2 of IEEE Std 802.1D, 1998 Edition). May be present if the implementation supports the MAC Enabled parameter.
f) adminPointToPointMAC - the new value of the adminPointToPointMAC parameter (6.4.3 of IEEE Std 802.1D, 1998 Edition). May be present if the implementation supports the adminPointToPointMAC parameter.

12.8.2.5.3 Outputs

a) Operation status. This takes one of the following values:
   1) Operation rejected due to invalid Port Priority value (12.3); or
   2) Operation accepted.

12.8.2.5.4 Procedure

In STP Bridges, the Set Path Cost procedure (8.8.6 of IEEE Std 802.1D, 1998 Edition) is used to set the Path Cost parameter for the specified Port for the specified spanning tree instance. The Set Port Priority procedure (8.8.5 of IEEE Std 802.1D, 1998 Edition) is used to set the priority part of the Port Identifier (8.5.5.1 of IEEE Std 802.1D, 1998 Edition) for the CIST to the supplied value.

In RSTP and MSTP Bridges, the Path Cost (13.36.1 of this standard, 17.16.5 of IEEE Std 802.1D, 1998 Edition) and Port Priority (17.18.7 of IEEE Std 802.1D, 1998 Edition) parameters for the Port are updated using the supplied values. The reselect parameter value for the CIST for the Port (13.24 of this standard, 17.18.29 of IEEE Std 802.1D, 1998 Edition) is set TRUE, and the selected parameter for the CIST for the Port (13.24 of this standard, 17.18.31 of IEEE Std 802.1D, 1998 Edition) is set FALSE.

12.8.2.6 Set MSTI port parameters

12.8.2.6.1 Purpose

To modify parameters for a Port in the Bridge's Bridge Protocol Entity in order to force a configuration of the spanning tree for the specified Spanning Tree instance.

12.8.2.6.2 Inputs

a) MSTID—Identifies the set of parameters upon which the operation will be performed. This parameter is the identifier of the spanning tree for which the operation is being performed.
b) Port Number—the number of the Bridge Port.
c) Path Cost—the new value (13.36.1).
d) Port Priority—the new value of the priority field for the Port Identifier (13.24.17).

12.8.2.6.3 Outputs

a) Operation status. This takes one of the following values:
   1) Operation rejected due to invalid Port Priority value (12.3); or
   2) Operation rejected due to invalid MSTID (i.e, there is currently no spanning tree instance with that value of MSTID supported by the Bridge); or
   3) Operation accepted.

12.8.2.6.4 Procedure

The Path Cost (13.36.1 of this standard, 17.16.5 of IEEE Std 802.1D, 1998 Edition) and Port Priority (17.18.7 of IEEE Std 802.1D, 1998 Edition) parameters for the specified MSTI and Port are updated using the supplied values. The reselect parameter value for the MSTI for the Port (13.24) is set TRUE, and the selected parameter for the MSTI for the Port () is set FALSE.

12.8.2.7 Force BPDU Migration Check

This operation is available only in Bridges that support RSTP or MSTP, as specified in Clause 13 of this standard or Clause 17 of IEEE Std 802.1D, 1998 Edition.

12.8.2.7.1 Purpose

To force the specified Port to transmit RST or MST BPDUs (see 13.29 of this standard and 17.26 of IEEE Std 802.1D, 1998 Edition).

12.8.2.7.2 Inputs

a) Port Number—the number of the Bridge Port.

12.8.2.7.3 Outputs

None.

12.8.2.7.4 Procedure

The mcheck variable (17.18.10 of IEEE Std 802.1D, 1998 Edition) for the specified Port is set to the value TRUE if the value of the forceVersion variable (13.6.2 of this standard, 17.16.1 of IEEE Std 802.1D, 1998 Edition) is greater than or equal to 2.

12.9 GARP Entities

The operation of GARP is described in Clause 12 of IEEE Std 802.1D, 1998 Edition.

The objects that comprise this managed resource are

a) The GARP Timer objects (12.9.1);
b) The GARP Attribute Type objects (12.9.2);
c) The GARP State Machine objects (12.9.3).

12.9.1 The GARP Timer object

The GARP Timer object models the operations that can be performed upon, or inquire about, the current settings of the timers used by the GARP protocol on a given Port. The management operations that can be performed on the GARP Participant are

a) Read GARP Timers (12.9.1.1);
b) Set GARP Timers (12.9.1.2).

NOTE—The GARP timer values modeled by this object are the values used to initialize timer instances that are used within the GARP state machines, not the timer instances themselves. Hence, there is a single GARP Timer object per Port, regardless of whether the Bridge supports single or multiple spanning trees.

12.9.1.1 Read GARP Timers

12.9.1.1.1 Purpose

To read the current GARP Timers for a given Port.

12.9.1.1.2 Inputs

a) The Port identifier.

12.9.1.1.3 Outputs

a) Current value of JoinTime—Centiseconds (12.10.2.1 and 12.12.1 of IEEE Std 802.1D, 1998 Edition);
b) Current value of LeaveTime—Centiseconds (12.10.2.2 and 12.12.1 of IEEE Std 802.1D, 1998 Edition);

c) Current value of LeaveAllTime—Centiseconds (12.10.2.3 and 12.12.1 of IEEE Std 802.1D, 1998 Edition).

12.9.1.2 Set GARP Timers

12.9.1.2.1 Purpose

To set new values for the GARP Timers for a given Port.

12.9.1.2.2 Inputs

a) The Port identifier;
b) New value of JoinTime—Centiseconds (12.10.2.1 and 12.12.1 of IEEE Std 802.1D, 1998 Edition);
c) New value of LeaveTime—Centiseconds (12.10.2.2 and 12.12.1 of IEEE Std 802.1D, 1998 Edition);
d) New value of LeaveAllTime—Centiseconds (12.10.2.3 and 12.12.1 of IEEE Std 802.1D, 1998 Edition).

12.9.1.2.3 Outputs

None.

12.9.2 The GARP Attribute Type object

The GARP Attribute Type object models the operations that can be performed upon, or inquire about, the operation of GARP for a given Attribute Type (12.11.2.2 of IEEE Std 802.1D, 1998 Edition). The management operations that can be performed on a GARP Attribute Type are

a) Read GARP Applicant Controls (12.9.2.1);
b) Set GARP Applicant Controls (12.9.2.2).

12.9.2.1 Read GARP Applicant Controls

12.9.2.1.1 Purpose

To read the current values of the GARP Applicant Administrative control parameters (12.9.2 of IEEE Std 802.1D, 1998 Edition) associated with all GARP Participants for a given Port, GARP Application and Attribute Type.

12.9.2.1.2 Inputs

a) The Port identifier;
b) The GARP Application address (Table 12-1 of IEEE Std 802.1D, 1998 Edition);
c) The Attribute Type (12.11.2.5 of IEEE Std 802.1D, 1998 Edition).

12.9.2.1.3 Outputs

a) The current Applicant Administrative Control Value (12.9.2 of IEEE Std 802.1D, 1998 Edition);
b) Failed Registrations—Count of the number of times that this GARP Application has failed to register an attribute of this type due to lack of space in the Filtering Database (12.10.1.6).

12.9.2.2 Set GARP Applicant Controls

12.9.2.2.1 Purpose

To set new values for the GARP Applicant Administrative control parameters (12.9.2 of IEEE Std 802.1D, 1998 Edition) associated with all GARP Participants for a given Port, GARP Application and Attribute Type.

12.9.2.2.2 Inputs

a) The Port identifier;
b) The GARP Application address (Table 12-1 of IEEE Std 802.1D, 1998 Edition);
c) The Attribute Type (12.11.2.5 of IEEE Std 802.1D, 1998 Edition) associated with the state machine;
d) The desired Applicant Administrative Control Value (12.9.2 of IEEE Std 802.1D, 1998 Edition).

12.9.2.2.3 Outputs

None.

12.9.3 The GARP State Machine object

The GARP State Machine object models the operations that can be performed upon, or inquire about, the operation of GARP for a given State Machine.

The management operation that can be performed on a GARP State Machine is Read GARP State.

12.9.3.1 Read GARP State

12.9.3.1.1 Purpose

To read the current value of an instance of a GARP state machine.

12.9.3.1.2 Inputs

a) The Port identifier;
b) The GARP Application address (Table 12-1 of IEEE Std 802.1D, 1998 Edition);
c) The GIP Context (12.3.4 of IEEE Std 802.1D, 1998 Edition);
d) The Attribute Type (12.11.2.2 of IEEE Std 802.1D, 1998 Edition) associated with the state machine;
e) The Attribute Value (12.11.2.6 of IEEE Std 802.1D, 1998 Edition) associated with the state machine.

12.9.3.1.3 Outputs

a) The current value of the combined Applicant and Registrar state machine for the attribute (Table 12-6 of IEEE Std 802.1D, 1998 Edition);
b) Optionally, Originator address—the MAC Address of the originator of the most recent GARP PDU that was responsible for causing a state change in this state machine (12.9.1 of IEEE Std 802.1D, 1998 Edition).

12.10 Bridge VLAN managed objects

The following managed objects define the semantics of the management operations that can be performed upon the VLAN aspects of a Bridge:

a) The Bridge VLAN Configuration managed object (12.10.1);

b) The VLAN Configuration managed object (12.10.2);
c) The VLAN Learning Constraints managed object (12.10.3).

12.10.1 Bridge VLAN Configuration managed object

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

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

a) Read Bridge VLAN Configuration (12.10.1.1);
b) Configure PVID and VID Set values (12.10.1.2);
c) Configure Acceptable Frame Types parameters (12.10.1.3);
d) Configure Enable Ingress Filtering parameters (12.10.1.4);
e) Reset VLAN Bridge (12.10.1.5);
f) Notify VLAN registration failure (12.10.1.6);
g) Configure Restricted_VLAN_Registration parameters (12.10.1.3);
h) Configure Protocol Group Database (12.10.1.8);
i) Configure VLAN Learning Constraints (12.10.3).

12.10.1.1 Read Bridge VLAN Configuration

12.10.1.1.1 Purpose

To obtain general VLAN information from a Bridge.

12.10.1.1.2 Inputs

None.

12.10.1.1.3 Outputs

a) The 802.1Q VLAN Version number. Reported as “1” by VLAN Bridges that support only SST operation, and reported as “2” by VLAN Bridges that support MST operation;

NOTE—No 802.1Q VLAN version numbers other than 1 and 2 are currently specified.

b) The optional VLAN features supported by the implementation:
   1) The maximum number of VLANs supported;
   2) Whether the implementation supports the ability to override the default PVID setting, and its egress status (VLAN-tagged or untagged) on each Port;
   3) For a Bridge that supports Port-and-Protocol-based VLAN classification, which of the Protocol Template formats (8.6.2) are supported by the implementation.
   4) For MST Bridges, the maximum number of MSTIs supported within an MST Region (i.e., the number of Spanning Tree instances that can be supported in addition to the CIST). For SST Bridges, this parameter may either be omitted or reported as “0”.
c) For each Port:
   1) the Port number;
   2) the PVID value (8.4.4) currently assigned to that Port;
   3) For a Bridge that supports Port-and-Protocol-based VLAN classification, whether the implementation supports Port-and-Protocol-based VLAN classification on that Port;

4) For a Bridge that supports Port-and-Protocol-based VLAN classification on that Port, the maximum number of entries supported in the VID Set on that Port; the VID value and Protocol Group Identifier currently assigned to each entry in the VID Set (8.4.4) on that Port;
5) the state of the Acceptable Frame Types parameter (8.4.3). The permissible values for this parameter are:
   i) Admit only VLAN-tagged frames;
   ii) Admit all frames.
3) the state of the Enable Ingress Filtering parameter (8.4.5); Enabled or Disabled;
4) the state of the Restricted_VLAN_Registration parameter (11.2.3.2.3), TRUE or FALSE.

d) For a Bridge that supports Port-and-Protocol-based VLAN classification: the contents of the Protocol Group Database comprising a set of {Protocol Template, Protocol Group Identifier} bindings (8.6.1, 8.6.3, and 8.6.4); the maximum number of entries supported in the Protocol Group Database.

12.10.1.2 Configure PVID and VID Set values

12.10.1.2.1 Purpose

To configure the PVID and VID Set value(s) (8.4.4) associated with one or more Ports.

12.10.1.2.2 Inputs

a) For each Port to be configured, a Port number and the PVID value to be associated with that Port
b) In addition, for a Bridge that supports Port-and-Protocol-based VLAN classification: for each Port to be configured, a Port number, a Protocol Group Identifier, and a VID value for the member of the Port’s VID Set that is to be configured.

12.10.1.2.3 Outputs

a) Operation status for each Port to be configured. This takes one of the following values:
   1) Operation rejected due to there being no spare VID Set entries on this Port; or
   2) Operation rejected due to the PVID or VID being out of the supported range for this Port; or
   3) Operation accepted.

12.10.1.3 Configure Acceptable Frame Types parameters

12.10.1.3.1 Purpose

To configure the Acceptable Frame Types parameter (8.4.3) associated with one or more Ports.

12.10.1.3.2 Inputs

a) For each Port to be configured, a Port number and the value of the Acceptable Frame Types parameter to be associated with that Port. The permissible values of this parameter are (as defined in 8.4.3):
   1) Admit only VLAN-tagged frames;
   2) Admit all frames.

12.10.1.3.3 Outputs

None.

12.10.1.4 Configure Enable Ingress Filtering parameters

12.10.1.4.1 Purpose

To configure the Enable Ingress Filtering parameter(s) (8.4.5) associated with one or more Ports.

12.10.1.4.2 Inputs

a) For each Port to be configured, a Port number and the value of the Enable Ingress Filtering parameter to be associated with that Port. The permissible values for the parameter are
   1) Enabled;
   2) Disabled.

12.10.1.4.3 Outputs

None.

12.10.1.5 Reset VLAN Bridge

12.10.1.5.1 Purpose

To reset all statically configured VLAN-related information in the Bridge to its default state. This operation

a) Deletes all VLAN Configuration managed objects;
b) Resets the PVID associated with each Bridge Port to the Default PVID value (Table 9-2);
c) Removes all entries in the Protocol Group Database and removes all members of the VID Set on each port, for a Bridge that supports Port-and-Protocol-based VLAN classification;
d) Resets the Acceptable Frame Types parameter value associated with each Port to the default value (8.4.3).

12.10.1.5.2 Inputs

None.

12.10.1.5.3 Outputs

None.

12.10.1.6 Notify VLAN registration failure

12.10.1.6.1 Purpose

To notify a manager that GVRP (11.2.3) has failed to register a given VLAN owing to lack of resources in the Filtering Database for the creation of a Dynamic VLAN Registration Entry (8.11.5), or owing to the Restricted_VLAN_Registration parameter being set to TRUE.

12.10.1.6.2 Inputs

None.

12.10.1.6.3 Outputs

a) The VID of the VLAN that GVRP failed to register;
b) The Port number of the Port on which the registration request was received;

c) The reason for the failure:
   1) Lack of Resources; or
   2) Registration Restricted; or
   3) Unsupported VID value.

12.10.1.7 Configure Restricted_VLAN_Registration parameters

12.10.1.7.1 Purpose

To configure the Restricted_VLAN_Registration parameter (11.2.3.2.3) associated with one or more Ports.

12.10.1.7.2 Inputs

a) For each Port to be configured, a Port number and the value of the Restricted_VLAN_Registration parameter. The permissible values of this parameter are (as defined in 11.2.3.2.3) as follows:
   1) TRUE;
   2) FALSE.

12.10.1.7.3 Outputs

None.

12.10.1.8 Configure Protocol Group Database

To configure a Protocol Group Database (8.6.4) entry. This operation is not applicable to a Bridge that does not support Port-and-Protocol-based VLAN classification.

NOTE—Implementation of the Configure Protocol Group Database operation is not mandatory; conformant implementations may implement a fixed set of Protocol Group Database entries.

12.10.1.8.1 Inputs

a) A value representing the frame format to be matched: Ethernet, RFC_1042, SNAP_8021H, SNAP_Other or LLC_Other (8.6.1);
b) One of
   1) An IEEE 802.3 Type value, for matching frame formats of Ethernet, RFC_1042, or SNAP_8021H;
   2) A 40-bit Protocol ID (PID), for matching frame formats of SNAP_Other;
   3) A pair of IEEE 802.2 DSAP and SSAP address field values, for matching frame formats of LLC_Other;
c) A Protocol Group Identifier (8.6.3).

12.10.1.8.2 Outputs

a) Operation status. This takes one of the following values:
   1) Operation rejected due to there being no spare Protocol Group Database entries; or
   2) Operation rejected due to an unsupported frame format; or
   3) Operation rejected due to an unsupported value for an IEEE 802.3 Type value, PID, DSAP, or SSAP; or
   4) Operation accepted.

12.10.2 VLAN Configuration managed object

The VLAN Configuration object models operations that modify, or enquire about, the configuration of a particular VLAN within a Bridge. There are multiple VLAN Configuration objects per Bridge; only one such object can exist for a given VLAN ID.

The management operations that can be performed on the VLAN Configuration are:

a) Read VLAN Configuration (12.10.2.1);
b) Create VLAN Configuration (12.10.2.2);
c) Delete VLAN Configuration (12.10.2.3);

12.10.2.1 Read VLAN Configuration

12.10.2.1.1 Purpose

To obtain general information regarding a specific VLAN Configuration.

12.10.2.1.2 Inputs

a) VLAN Identifier: a 12-bit VID.

12.10.2.1.3 Outputs

a) VLAN Name: A text string of up to 32 characters of locally determined significance;
b) List of Untagged Ports: The set of Port numbers for which this VLAN ID is a member of the Untagged set (8.11.9) for that Port;
c) List of Egress Ports: The set of Port numbers for which this VLAN ID is a member of the Member set (8.11.9) for that Port.

NOTE—The values of the Member set and the Untagged set are determined by the values held in VLAN Registration Entries in the Filtering Database (8.11.2, 8.11.5, and 8.11.9).

12.10.2.2 Create VLAN Configuration

12.10.2.2.1 Purpose

To create or update a VLAN Configuration managed object.

12.10.2.2.2 Inputs

a) VLAN Identifier: a 12-bit VID;
b) VLAN Name: A text string of up to 32 characters of locally determined significance.

NOTE—Static configuration of the Member set and the Untagged set is achieved by means of the management operations for manipulation of VLAN Registration Entries (12.7.5).

12.10.2.2.3 Outputs

None.

12.10.2.3 Delete VLAN Configuration

12.10.2.3.1 Purpose

To delete a VLAN Configuration managed object.

12.10.2.3.2 Inputs

a) VLAN Identifier: a 12-bit VID;

12.10.2.3.3 Outputs

None.

12.10.3 The VLAN Learning Constraints managed object

The VLAN Learning Constraints managed object models operations that modify, or enquire about, the set of VLAN Learning Constraints (8.11.7.2) and VID to FID allocations (8.11.7.1) that apply to the operation of the Learning Process and the Filtering Database. There is a single VLAN Learning Constraints managed object per Bridge. The object is modeled as a pair of fixed-length tables, as follows:

a) A Learning Constraint table in which each table entry either defines a single Learning Constraint or is undefined. For some of the operations that can be performed upon the table, an *entry index* is used; this identifies the number of the entry in the table, where index number 1 is the first, and N is the last (where the table contains N entries).

NOTE—The number of Learning Constraint table entries supported is an implementation option. This standard does not provide any distribution mechanism to ensure that the same set of constraints is configured in all Bridges; individual Bridges can be configured by use of the management operations defined in this subclause (for example, via the use of SNMP operating upon a VLAN Bridge MIB), but there is no in-built consistency checking to ensure that all Bridges have been provided with the same constraint information. Hence, any such consistency checking is the responsibility of the network administrator and the management applications employed in the LAN.

b) A VID to FID allocation table (8.11.7.1) with an entry per VID supported by the implementation. Each table entry indicates, for that VID, that there is currently
   1) No allocation defined; or
   2) A fixed allocation to FID X; or
   3) A dynamic allocation to FID X.

The management operations that can be performed on the VLAN Learning Constraints managed object are

c) Read VLAN Learning Constraints (12.10.3.1);
d) Read VLAN Learning Constraints for VID (12.10.3.2);
e) Set VLAN Learning Constraint (12.10.3.3);
f) Delete VLAN Learning Constraint (12.10.3.4);
g) Read VID to FID allocations (12.10.3.5);
h) Read FID allocation for VID (12.10.3.6);
i) Read VIDs allocated to FID (12.10.3.7);
j) Set VID to FID allocation (12.10.3.8);
k) Delete VID to FID allocation (12.10.3.9);
l) Notify Learning Constraint Violation (12.10.3.10).

12.10.3.1 Read VLAN Learning Constraints

12.10.3.1.1 Purpose

To read the contents of a range of one or more entries in the VLAN Learning Constraints table.

12.10.3.1.2 Inputs

a) First Entry—Entry Index of first entry to be read;
b) Last Entry—Entry Index of last entry to be read.

12.10.3.1.3 Outputs

a) List of Entries—for each entry that was read:
    1) The Entry Index;
    2) The type of the Learning Constraint: Undefined, S or I;
    3) The value of the Learning Constraint, which is one of:
        i) Undefined, indicating an empty element in the table;
        ii) An S Constraint value, consisting of a pair of VIDs;
        iii) An I Constraint value, consisting of a VID and an Independent Set Identifier.

NOTE—Where this operation is implemented using a remote management protocol, PDU size constraints may restrict the number of entries that are actually read to fewer than was requested in the input parameters. In such cases, retrieving the remainder of the desired entry range can be achieved by repeating the operation with a modified entry range specification.

12.10.3.2 Read VLAN Learning Constraints for VID

12.10.3.2.1 Purpose

To read all the VLAN Learning Constraints for a given VID.

12.10.3.2.2 Inputs

a) VID—The VLAN Identifier to which the read operation applies.

12.10.3.2.3 Outputs

a) All learning constraint values that identify the VID requested. Each value returned is either
    1) An S Constraint value, consisting of a pair of VIDs; or
    2) An I Constraint value, consisting of a VID and an Independent Set Identifier.

12.10.3.3 Set VLAN Learning Constraint

12.10.3.3.1 Purpose

To modify the contents of one of the entries in the VLAN Learning Constraints table.

12.10.3.3.2 Inputs

a) Entry Index—Entry index of the entry to be set;
b) The type of the Learning Constraint: S or I;
c) The value of the Learning Constraint, which is either:
    1) An S Constraint value, consisting of a pair of VIDs; or
    2) An I Constraint value, consisting of a VID and an Independent Set Identifier.

12.10.3.3.3 Outputs

a) Operation status. This takes one of the following values:
   1) Operation rejected due to inconsistent learning constraint specification (8.11.7.3)—The Set operation requested setting a constraint that is inconsistent with another constraint already defined in the constraint table. The operation returns the value of the constraint concerned; or
   2) Operation rejected due to inconsistent fixed VID to FID allocation (8.11.7.3)—The Set operation requested setting a constraint that is inconsistent with a fixed VID to FID allocation already defined in the allocation table. The operation returns the value of the fixed allocation concerned; or
   3) Operation rejected due to entry index exceeding the maximum index supported by the constraint table; or
   4) Operation rejected due to conflict with FID to MSTID allocations (12.12.2)—The Set operation requested setting a constraint that cannot be reconciled with the current FID to MSTID allocations represented by the FID to MSTID Allocation Table; or

NOTE—It is not possible to specify a shared VLAN learning constraint for VLANs that do not share the same Spanning Tree instance.

   5) Operation accepted.

12.10.3.3.4 Procedure

In MST Bridges, the Configuration Digest element of the MST Configuration Identifier is re-calculated, in accordance with the definition in 13.7, following any change in the VLAN Learning Constraints that results in a change in the allocation of VIDs to spanning trees.

12.10.3.4 Delete VLAN Learning Constraint

12.10.3.4.1 Purpose

To remove one of the entries in the VLAN Learning Constraints table. This operation has the effect of setting the value of the specified table entry to “Undefined.”

12.10.3.4.2 Inputs

a) Entry Index—Entry index of the entry to be deleted.

12.10.3.4.3 Outputs

a) Operation status. This takes one of the following values:
   1) Operation rejected due to entry index exceeding the maximum index supported by the constraint table; or
   2) Operation accepted.

12.10.3.4.4 Procedure

In MST Bridges, the Configuration Digest element of the MST Configuration Identifier is re-calculated, in accordance with the definition in 13.7, following any change in the VLAN Learning Constraints that results in a change in the allocation of VIDs to spanning trees.

12.10.3.5 Read VID to FID allocations

12.10.3.5.1 Purpose

To read the contents of a range of one or more entries in the VID to FID allocation table.

12.10.3.5.2 Inputs

a) First Entry—VID of first entry to be read;
b) Last Entry—VID of last entry to be read.

12.10.3.5.3 Outputs

a) List of Entries—For each entry that was read:
   1) VID—The VLAN Identifier for this entry;
   2) Allocation Type—The type of the allocation: Undefined, Fixed or Dynamic;
   3) FID—The FID to which the VID is allocated (if not of type Undefined).

NOTE—Where this operation is implemented using a remote management protocol, PDU size constraints may restrict the number of entries that are actually read to fewer than was requested in the input parameters. In such cases, retrieving the remainder of the desired entry range can be achieved by repeating the operation with a modified entry range specification.

12.10.3.6 Read FID allocation for VID

12.10.3.6.1 Purpose

To read the FID to which a specified VID is currently allocated.

12.10.3.6.2 Inputs

a) VID—The VLAN Identifier to which the read operation applies.

12.10.3.6.3 Outputs

a) VID—the VLAN Identifier to which the read operation applies;
b) Allocation Type—the type of the allocation: Undefined, Fixed or Dynamic;
c) FID—the FID to which the VID is allocated (if not of type Undefined).

12.10.3.7 Read VIDs allocated to FID

12.10.3.7.1 Purpose

To read all the VIDs currently allocated to a given FID.

12.10.3.7.2 Inputs

a) FID—the Filtering Identifier to which the read operation applies.

12.10.3.7.3 Outputs

a) FID—the Filtering Identifier to which the read operation applies
b) Allocation List—a list of allocations for this FID. For each element in the list:
   1) Allocation Type—the type of the allocation: Fixed or Dynamic;
   2) VID—the VID that is allocated.

12.10.3.8 Set VID to FID allocation

12.10.3.8.1 Purpose

To establish a fixed allocation of a VID to an FID.

12.10.3.8.2 Inputs

a) VID—the VID of the entry to be set;
b) FID—the FID to which the VID is to be allocated.

12.10.3.8.3 Outputs

a) Operation status. This takes one of the following values:
   1) Operation rejected due to inconsistent learning constraint specification (8.11.7.3)—The Set operation requested setting a fixed allocation that is inconsistent with a VLAN Learning Constraint. The operation returns the value of the VLAN Learning Constraint concerned; or
   2) Operation rejected due to VID exceeding the maximum VID supported by the allocation table; or
   3) Operation rejected due to FID exceeding the maximum ID supported by the implementation; or
   4) Operation accepted.

12.10.3.8.4 Procedure

In MST Bridges, the Configuration Digest element of the MST Configuration Identifier is recalculated, in accordance with the definition in 13.7, following any change in the allocation of VIDs to FIDs that results in a change in the allocation of VIDs to spanning trees.

12.10.3.9 Delete VID to FID allocation

12.10.3.9.1 Purpose

To remove a fixed VID to FID allocation from the VID to FID allocation table. This operation has the effect of setting the value of the specified table entry to “Undefined.”

NOTE—If the VID concerned represents a currently active VLAN, then removal of a fixed allocation may result in the “Undefined” value in the table immediately being replaced by a dynamic allocation to an FID.

12.10.3.9.2 Inputs

a) VID—VID of the allocation to be deleted.

12.10.3.9.3 Outputs

a) Operation status. This takes one of the following values:
   1) Operation rejected due to VID exceeding the maximum value supported by the allocation table; or
   2) Operation accepted.

12.10.3.9.4 Procedure

In MST Bridges, the Configuration Digest element of the MST Configuration Identifier is re-calculated, in accordance with the definition in 13.7, and the MSTP state machine variables are reinitialized by asserting BEGIN, following any change in the allocation of VIDs to FIDs that results in a change in the allocation of VIDs to spanning trees.

12.10.3.10 Notify Learning Constraint Violation

12.10.3.10.1 Purpose

To alert the Manager to the existence of a Learning Constraint violation (8.11.7.3). This is an unsolicited notification from the management entity of the Bridge, issued upon detection of the constraint violation.

NOTE—As indicated in 8.11.7.3, a single change in configuration, such as the registration of a new VID by GVRP or the addition of a new learning constraint, can give rise to more than one violation being notified, depending upon the set of learning constraints currently configured in the Bridge.

12.10.3.10.2 Inputs

a) None.

12.10.3.10.3 Outputs

a) Violation Type/Argument—one of
    1) Shared VLAN Learning not supported. The argument returned indicates the VIDs of a pair of active VLANs for which an S constraint exists.
    2) Independent VLAN Learning not supported. The argument returned indicates the VIDs of a pair of active VLANs for which I constraints exist that contain the same independent set identifier.
    3) Required FID range not supported. The argument returned indicates
        i) The VID that the Bridge is unable to allocate to an FID;
        ii) The maximum number of FIDs supported by the Bridge.

The violation type *Required FID range not supported* is detected only by IVL or IVL/SVL Bridges that support fewer than 4094 FIDs.

12.11 GMRP entities

The following managed objects define the semantics of the management operations that can be performed upon the operation of GMRP in a Bridge:

a) The GMRP Configuration managed object (12.10.1).

12.11.1 GMRP Configuration managed object

The GMRP Configuration managed object models operations that modify, or enquire about, the overall configuration of the operation of GMRP. There is a single GMRP Configuration managed object per Bridge.

The management operations that can be performed on the GMRP Configuration managed object are as follows:

a) Read GMRP Configuration (12.10.1.1);
b) Notify Group registration failure (12.10.1.6);
c) Configure Restricted_Group_Registration parameters (12.11.1.3).

12.11.1.1 Read GMRP Configuration

12.11.1.1.1 Purpose

To obtain general GMRP configuration information from a Bridge.

12.11.1.1.2 Inputs

None.

12.11.1.1.3 Outputs

a) For each Port:
   1) The Port number;
   2) The state of the Restricted_Group_Registration parameter (10.3.2.3 in IEEE Std 802.1D, 1998 Edition), TRUE or FALSE.

12.11.1.2 Notify Group registration failure

12.11.1.2.1 Purpose

To notify a manager that GMRP has failed to register a given Group owing to lack of resources in the Filtering Database for the creation of a Group Registration Entry (8.11.4).

12.11.1.2.2 Inputs

None.

12.11.1.2.3 Outputs

a) The MAC address of the Group that GMRP failed to register;
b) The Port number of the Port on which the registration request was received.
c) The reason for the failure:
   1) Lack of Resources; or
   2) Registration Restricted.

12.11.1.3 Configure Restricted_Group_Registration parameters

12.11.1.3.1 Purpose

To configure the Restricted_Group_Registration parameter (10.3.2.3 in IEEE Std 802.1D, 1998 Edition) associated with one or more Ports.

12.11.1.3.2 Inputs

a) For each Port to be configured, a Port number and the value of the Restricted_Group_Registration parameter. The permissible values of this parameter are (as defined in 10.3.2.3 of IEEE Std 802.1D, 1998 Edition) as follows:
   1) TRUE;
   2) FALSE.

12.11.1.3.3 Outputs

None.

12.12 MST configuration entities

The following managed objects define the semantics of the management operations that can be performed upon the MST configuration in a Bridge:

a) The MSTI List object (12.12.1);
b) The FID to MSTID Allocation Table object (12.12.2);
c) The MST Configuration Table object (12.12.3).

12.12.1 The MSTI List

For MST Bridges, the MSTI List object models the operations that modify, or enquire about, the list of MST spanning tree instances supported by the Bridge. The object is modelled as a list of MSTIDs corresponding to the MSTIs supported by the Bridge.

The MSTID List object supports the following operations:

a) Read MSTI List (12.12.1.1);
b) Create MSTI (12.12.1.2);
c) Delete MSTI (12.12.1.3).

12.12.1.1 Read MSTI List

12.12.1.1.1 Purpose

To read the list of MSTIDs that are currently supported by the Bridge.

12.12.1.1.2 Inputs

None.

12.12.1.1.3 Outputs

a) MSTID list. The list of MSTID values that are currently supported by the Bridge.

12.12.1.2 Create MSTI

12.12.1.2.1 Purpose

To create a new MSTI and its associated state machines and parameters, and to add its MSTID to the MSTI List.

12.12.1.2.2 Inputs

a) The MSTID of the MSTI to be created.

12.12.1.2.3 Outputs

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

12.12.1.3 Delete MSTI

12.12.1.3.1 Purpose

To delete an existing MSTI and its associated state machines and parameters, and to remove its MSTID from the MSTI List.

12.12.1.3.2 Inputs

a) The MSTID of the MSTI to be deleted.

12.12.1.3.3 Outputs

a) Operation status. This takes one of the following values:
   1) Operation rejected as the MSTID value supplied in the input parameters is not present in the MSTI List.
   2) Operation rejected as the MSTID value supplied in the input parameter currently has one or more FIDs allocated to it in the FID to MSTID Allocation Table.
   3) Operation accepted.

12.12.2 The FID to MSTID Allocation Table

For MST Bridges, the FID to MSTID Allocation Table object models the operations that modify, or enquire about, the assignment of FIDs to spanning tree instances currently supported by the Bridge (8.11.3). The object is modelled as a fixed-length table in which each entry in the table corresponds to a FID, and the value of the entry specifies the MSTID of the spanning tree to which the set of VLANs supported by that FID are assigned. A value of zero in an entry specifies that the set of VLANs supported by that FID are assigned to the CST.

The MSTID Allocation Table object supports the following operations:

a) Read FID to MSTID allocations (12.12.2.1);
b) Set FID to MSTID allocation (12.12.2.2).

12.12.2.1 Read FID to MSTID allocations

12.12.2.1.1 Purpose

To read a range of one or more entries in the FID to MSTID Allocation Table.

12.12.2.1.2 Inputs

a) First FID—The FID of the first entry to be read;
b) Last FID—The FID of the last entry to be read.

If the value of Last FID is numerically equal to, or smaller than, the value of First FID, then a single table entry is read, corresponding to the value of First FID.

12.12.2.1.3 Outputs

a) List of entries—For each entry that was read:
   1) The FID of the entry: and
   2) The MSTID to which that FID is allocated.

12.12.2.2 Set FID to MSTID allocation

12.12.2.2.1 Purpose

To change the contents of an entry in the FID to MSTID Allocation Table.

12.12.2.2.2 Inputs

a) FID—The FID of the entry to be changed;
b) MSTID—The MSTID to which the FID is to be allocated.

12.12.2.2.3 Outputs

a) Operation status. This takes one of the following values:
   1) Operation rejected as the MSTID value supplied in the input parameters is not present in the MSTI List.
   2) Operation rejected as the FID value supplied in the input parameters is invalid or is not supported.
   3) Operation accepted.

12.12.2.2.4 Procedure

The Configuration Digest element of the MST Configuration Identifier is re-calculated, in accordance with the definition in 13.7, following any change in the allocations of FIDs to MSTIDs.

12.12.3 The MST Configuration Table

The MST Configuration Table managed object models the operations that can be performed on the MST Configuration Table for the Bridge (3.26, 8.11.1, and 13.7). Associated with the table is the MST Configuration Identifier for the Bridge (8.11.2, 13.7).

The MST Configuration Table is a read-only table, its elements derived from other configuration information held by the Bridge; specifically, the current state of the VID to FID allocation table (8.10.7.1, 12.10.3), and the FID to MSTID allocation table (8.11.3, 12.12.2). Hence, changes made to either of these tables can in turn affect the contents of the MST Configuration Table, and also affect the value of the “digest” element of the MST Configuration Identifier. The MST Configuration Table is modelled as a fixed table of 4096 elements, as described in 13.7.

The MST Configuration Table managed object supports the following operations:

a) Read MST Configuration Table Element (12.12.3.1);
b) Read VIDs assigned to MSTID (12.12.3.2);
c) Read MST Configuration Identifier (12.12.3.3);
d) Set MST Configuration Identifier Elements (12.12.3.4).

12.12.3.1 Read MST Configuration Table Element

12.12.3.1.1 Purpose

To read a single element of the current MST Configuration Table for the Bridge (13.7).

12.12.3.1.2 Inputs

a) A VID value, in the range 0 through 4094.

12.12.3.1.3 Outputs

a) A VID value, in the range 0 through 4094;
b) The MSTID value corresponding to that VID.

12.12.3.2 Read VIDs assigned to MSTID

12.12.3.2.1 Purpose

To read the list of VIDs that are currently assigned to a given MSTID in the MST Configuration Table for the Bridge (13.7).

12.12.3.2.2 Inputs

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

12.12.3.2.3 Outputs

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

12.12.3.3 Read MST Configuration Identifier

12.12.3.3.1 Purpose

To read the current value of the MST Configuration Identifier for the Bridge (13.7).

12.12.3.3.2 Inputs

None.

12.12.3.3.3 Outputs

a) The MST Configuration Identifier (13.7), consisting of:
   1) The Configuration Identifier Format Selector in use by the Bridge. This has a value of 0 to indicate the format specified in this Standard.
   2) The Configuration Name;
   3) The Revision Level;
   4) The Configuration Digest.

12.12.3.4 Set MST Configuration Identifier Elements

12.12.3.4.1 Purpose

To change the current values of the modifiable elements of the MST Configuration Identifier for the Bridge (13.7).

NOTE—The Configuration Digest element of the MST Configuration Identifier is read-only; its value is re-calculated whenever configuration changes occur that result in a change in the allocation of VIDs to MSTIs.

12.12.3.4.2 Inputs

a) The MST Configuration Identifier (13.7) Format Selector in use by the Bridge. This has a value of 0 to indicate the format specified in this Standard.

b) The Configuration Name;
c) The Revision Level.

12.12.3.4.3 Outputs

a) Operation Status. This can take the following values:
   1) Operation rejected due to unsupported Configuration Identifier Format Selector value.
   2) Operation accepted.

13. The Multiple Spanning Tree Protocol (MSTP)

The MSTP algorithm and protocol provides simple and full connectivity for frames assigned to any given VLAN throughout a Bridged Local Area Network comprising arbitrarily interconnected Bridges, each operating MSTP, STP (Clause 8 of IEEE Std 802.1D, 1998 Edition), or RSTP (Clause 17 of IEEE Std 802.1D, 1998 Edition). MSTP allows frames assigned to different VLANs to follow separate paths, each based on an independent Multiple Spanning Tree Instance (MSTI), within Multiple Spanning Tree (MST) Regions composed of LANs and or MST Bridges. These Regions and the other Bridges and LANs are connected into a single Common Spanning Tree (CST).

NOTE 1—The specification of RSTP is contained in IEEE Std 802.1w-2001, which is an amendment to IEEE Std 802.1D, 1998 Edition. References to IEEE Std 802.1D, 1998 Edition in this standard assume that this amendment, along with IEEE Std 802.1t-2001 (which documents technical and editorial corrections to IEEE Std 802.1D, 1998 Edition), have been applied to the base text of IEEE Std 802.1D, 1998 Edition.

MSTP connects all Bridges and LANs with a single Common and Internal Spanning Tree (CIST). The CIST supports the automatic determination of each MST Region, choosing its maximum possible extent. The connectivity calculated for the CIST provides the CST for interconnecting these Regions, and an Internal Spanning Tree (IST) within each Region. MSTP ensures that frames with a given VID are assigned to one and only one of the MSTIs or the IST within the Region, that the assignment is consistent amongst all the Bridges within the region, and that the stable connectivity of each MSTI and the IST at the boundary of the Region matches that of the CST. The stable active topology of the Bridged Local Area Network with respect to frames consistently classified as belonging to any given VLAN thus simply and fully connects all LANs and Bridges throughout the network, though frames belonging to different VLANs can take different paths within any MST Region.

NOTE 1—Readers of this specification are urged to begin by familiarizing themselves with the referenced specification of RSTP.

NOTE 2—Although the active topology determined by STP, RSTP, and MSTP fully connects the components of a Bridged Local Area Network, filtering (GVRP etc.) can restrict frames to a subset of the active topology where some VLANs are not present throughout.

13.1 Protocol Design Requirements

The Spanning Tree Algorithm and its associated Bridge Protocol operate in Bridged Local Area Networks of arbitrary physical topology comprising MSTP, RSTP, or STP Bridges connecting shared media or point to point LANs, so as to support, preserve, and maintain the quality of the MAC Service in all its aspects as specified by Clause 7. In order to do this the algorithm meets the following requirements:

a) It will configure the active topology into a single spanning tree for any given VLAN, such that there is at most one data route between any two end stations for frames consistently allocated to a given VID by Bridges conforming to this standard, eliminating data loops.
b) It will provide for fault tolerance by automatic reconfiguration of the spanning tree topology as a result of Bridge failure or a breakdown in a data path, within the confines of the available Bridged Local Area Network components, and for the automatic accommodation of any Bridge or Bridge Port added to the network without the formation of transient data loops.
c) The active topology will, with a high probability, stabilize within a short, known bounded interval in order to minimize the time for which the service is unavailable for communication between any pair of end stations.
d) The active topology will be predictable and reproducible, and may be selected by management of the parameters of the algorithm, thus allowing the application of Configuration Management, following traffic analysis, to meet the goals of Performance Management.

e) It will operate transparently to the end stations, such that they are unaware of their attachment to a single LAN or a Bridged LAN when using the MAC Service.

f) The communications bandwidth consumed by the Bridges in establishing and maintaining a spanning tree on any particular LAN will be a small percentage of the total available bandwidth and independent of the total traffic supported by the Bridged LAN regardless of the total number of Bridges or LANs.

g) It allows frames assigned to different VLANs to follow different data routes within administratively established regions of the network.

h) It will, with a high probability, continue to provide simple and full connectivity for frames even in the presence of administrative errors in the allocation of VLANs to spanning trees.

Additionally, the algorithm and protocol meet the following goals, which limit the complexity of Bridges and their configuration:

i) The memory requirements associated with each Bridge Port are independent of the number of Bridges and LANs in the Bridged LAN.

j) Bridges do not have to be individually configured before being added to the Bridged LAN, other than having their MAC Addresses assigned through normal procedures.

13.2 Protocol Support Requirements

MSTP does not require any additional configuration mechanisms beyond those specified in 17.1 of IEEE Std 802.1D, 1998 Edition in order to support the MAC Service. However to realize the improved throughput and associated frame loss and transit delay performance improvements made possible by the use of multiple spanning trees the following are required:

a) A means of consistently assigning VIDs to MSTIDs within each potential MST Region.

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

c) A means of assessing the probable distribution of traffic between sets of communicating end stations.

d) A choice of performance goals or the establishment of goals that the quality of service characteristics of some set of communications shall be independent of some other sets.

e) Choices of configuration parameters for the spanning trees that support these goals given the available physical components.

13.3 MSTP Overview

The Multiple Spanning Tree Protocol specifies:

a) An MST Configuration Identifier (13.7). This allows each Bridge to advertise its configuration for allocating frames with given VIDs to any of a number of distinct, fully and simply connected Multiple Spanning Tree Instances (MSTIs).

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

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

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

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

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

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

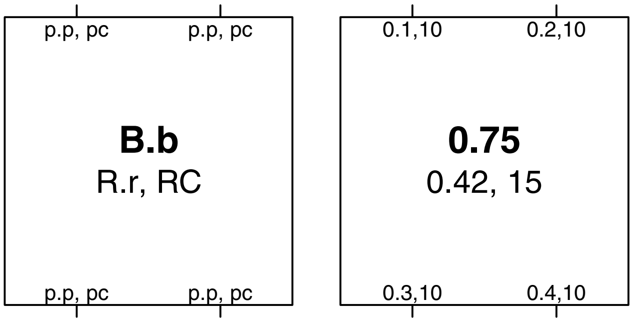
h) State machines and state variables associated with each spanning tree (CIST or MSTI), port, and port role, to select and change the Port State (8.4, 13.19 of this standard, 17.5 of IEEE Std 802.1D, 1998 Edition) that controls the processing and forwarding of frames allocated to that tree by a MAC Relay Entity (7.3).

13.3.1 Example Topologies

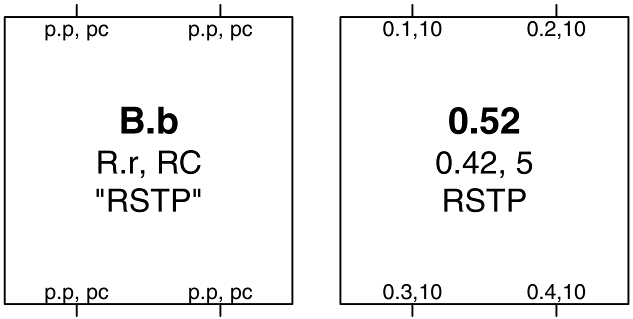
The examples shown in this clause make use of the diagrammatic conventions shown in Figure 13-1._

Figure 13-3 is an example Bridged Local Area Network, chosen to illustrate MSTP calculations rather than as an example of a common or desirable physical topology. Figure 13-3 is the same network showing Bridges and LANs with better CIST spanning tree priorities higher on the page, and including CIST priority vectors, port roles, and MST Regions. In this example:

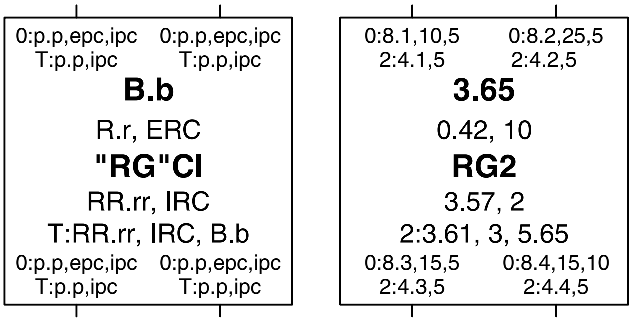
a) Bridge 0.42 has been chosen as the CIST Root because it has the best (numerically the lowest) Bridge Identifier of all the bridges in the network.



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



A template for and example of an RSTP Bridge.



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

A LAN

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

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

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

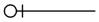


Figure 13-1—Diagrammatic conventions

b) Bridges 0.57 and 2.83 are in the same MST Region (1) as 0.42, because they have the same MST Configuration Identifier as the latter. Because they are in the same MST Region as the CIST Root, their External Root Path Cost is 0, and their CIST Regional Root is the CIST Root.

c) LANs A, B, C, and D are in Region 1 because a Region 1 MST Bridge is the CIST Designated Bridge for those LANs and there are no attached STP Bridges. LAN E is not in a Region (or is in a Region by itself, which is an equivalent view) because it is attached to Bridge 0.53, which is not an MST Bridge.

d) Bridges 0.77, 0.65, 0.97, 0.86, 3.84, and 3.72 are in the same MST Region (2) because they all have the same MST Configuration Identifier, and are all interconnected by LANs for which one of them is the CIST Designated Bridge.

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

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

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

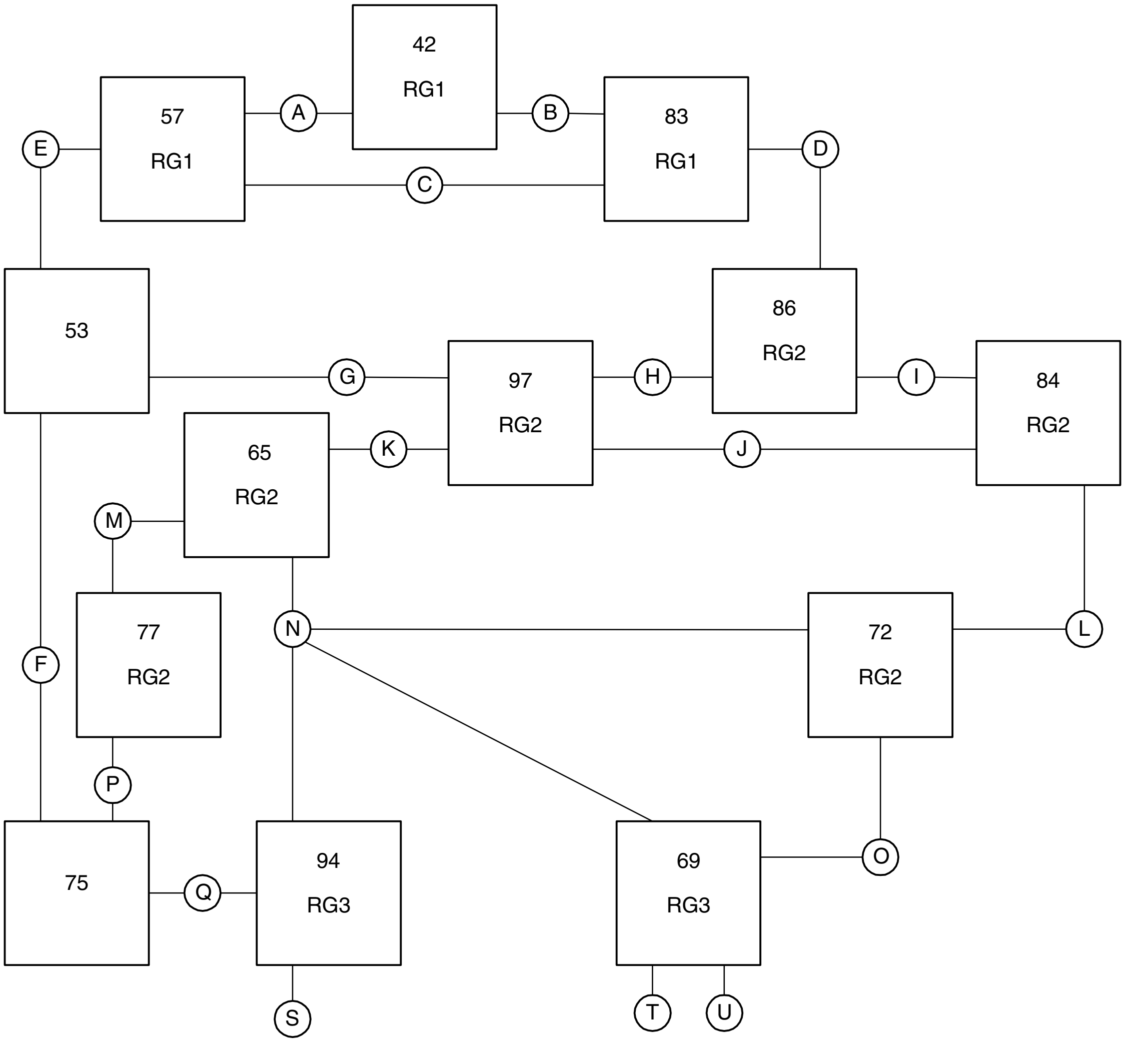


Figure 13-2—An Example Network

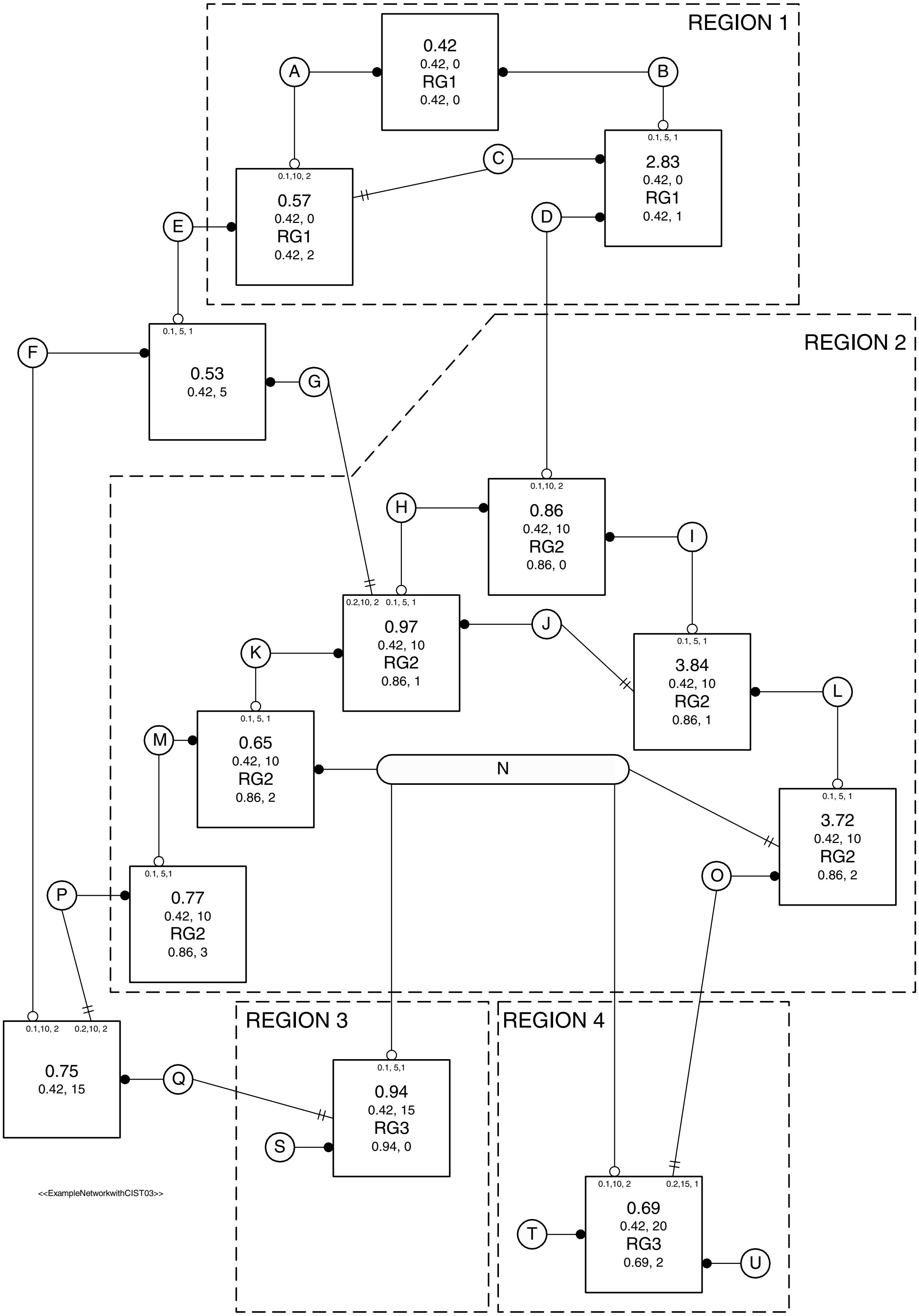


Figure 13-3—Example Network with CIST Priority Vectors, Port Roles, and MST Regions

Figure 13-4 shows a possible active topology of MSTI 2 within Region 2.

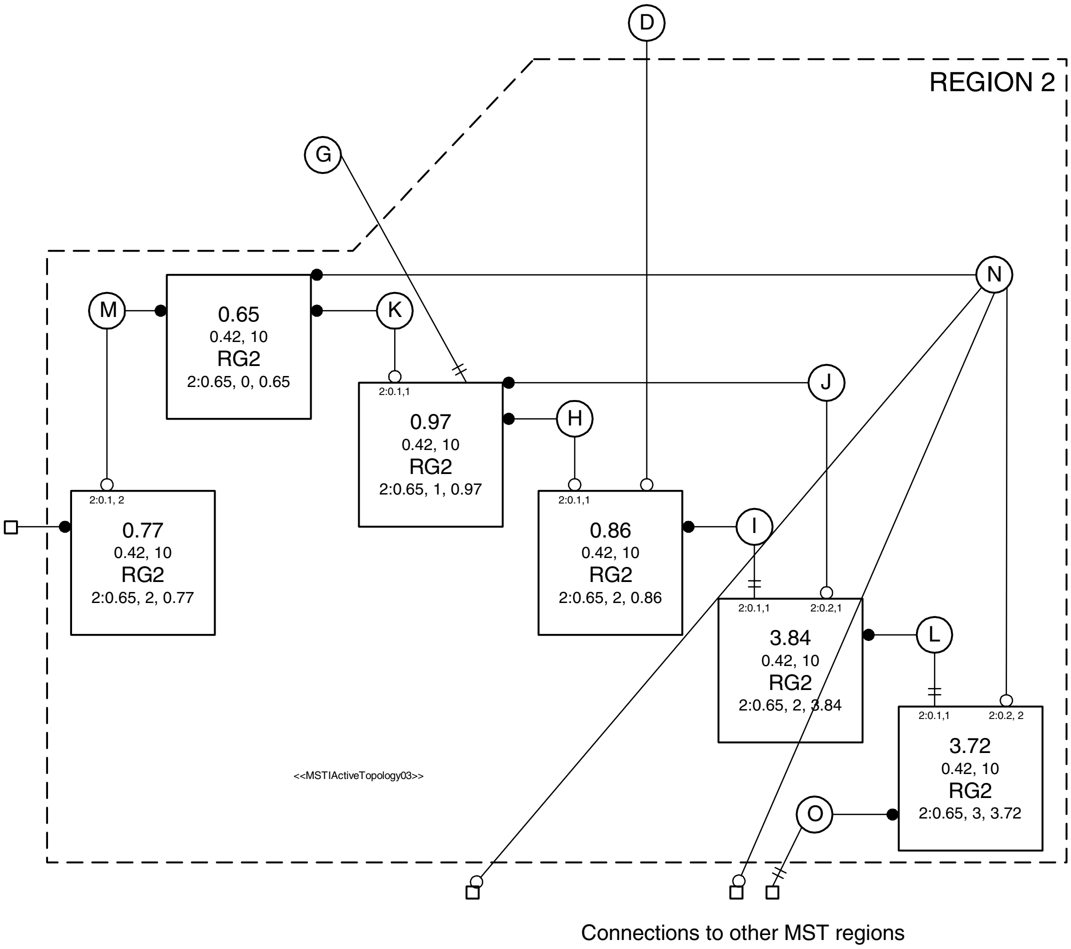


Figure 13-4—MSTI Active Topology in Region 2 of the example network

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

i) The connectivity between the whole of Region 2 and Region 1 is provided through a single Bridge Port, the Master Port on Bridge 0.86. This port was selected for this role because it is the CIST Root Port on the CIST Regional Root for the Region (see Figure 13-3).

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

13.4 Relationship of MSTP to RSTP

The design of the Multiple Spanning Tree Protocol is based on that of the Rapid Spanning Tree Protocol (Clause 17 of IEEE Std 802.1D, 1998 Edition) extended to provide the capability for frames assigned to different VLANs to be transmitted along different paths within MST Regions.

a) The selection of the CIST Root Bridge and the computation of port roles for the CIST uses the same fundamental algorithm (17.4.1 of IEEE Std 802.1D, 1998 Edition) but extended priority vector components and calculations (13.9, 13.10) within MST Regions as compared to RSTP (17.4.2 of IEEE

Std 802.1D, 1998 Edition). The effect of these extensions is to cause each region to resemble a single bridge from the point of view of the CST as calculated by STP or RSTP.

b) MST Configuration Identification is specific to MSTP.
c) The selection of the MSTI Regional Root Bridge and computation of port roles for each MSTI also uses the same fundamental spanning tree algorithm but modified priority vector components (13.11).
d) Different Bridges may be selected as the Regional Root for different MSTIs by modifying the manageable priority component of the Bridge Identifier differently for the MSTIs.
e) The port roles used by the CIST (Root, Designated, Alternate, Backup or Disabled Port) are the same as those of STP and RSTP (17.4 of IEEE Std 802.1D, 1998 Edition). The MSTIs use the additional port role Master Port. The Port States associated with each spanning tree and bridge port are the same as those of RSTP (17.5 of IEEE Std 802.1D, 1998 Edition).
f) The state variables associated with each port for each spanning tree and for the tree itself are those specified for RSTP as per bridge port and per bridge (17.13, 17.15, 17.17, 17.18 of IEEE Std 802.1D, 1998 Edition) with few exceptions, additions, and enhancements.
g) The state machine performance parameters specified for RSTP (17.6 of IEEE Std 802.1D, 1998 Edition) apply equally to the CIST. A simplified set of performance parameters apply to the MSTIs.
h) The state machine procedures of RSTP are used (17.9 of IEEE Std 802.1D, 1998 Edition) with detailed changes.

MSTP, like RSTP:

i) Cannot protect against temporary loops caused by the inter-connection of two LAN segments by devices other than Bridges (e.g., LAN repeaters) that operate invisibly with respect to support of the Bridges’ MAC Internal Sublayer Service.
j) Provides for rapid recovery of connectivity following the failure of a Bridge, Bridge Port, or a LAN. The timers used define worst case delays, only used to backup the normal operation of the protocol.
k) Provides a Force Protocol Version parameter, controlled by management and applicable to all Ports and trees supported by an MST bridge, to instruct MSTP to emulate aspects of early versions of spanning tree protocol. In particular the Force Protocol Version parameter allows rapid transitions to be disabled. This reduces the risk of an increase, as compared to STP, in the rates of frame duplication and misordering in the Bridged LAN, as discussed in F.2.4 of IEEE Std 802.1D, 1998 Edition.
l) Allows Bridge Ports to be configured such that they can transition directly to the Forwarding Port State on re-initialization of the Bridge. This may be appropriate where a specific Bridge Port is known to be connected to a LAN segment that is not connected to further Bridges. The per port operational control, operEdge, that supports this behavior applies equally to all the spanning trees of an MST Bridge.

13.5 Modelling an MST Region as a single RSTP Bridge

The specification of MSTP is such that the nominal replacement of an entire MST Region by a single RSTP Bridge leads to little change in the behavior of the remainder of the bridged local area network. This design is intended to assist those familiar with the STP and RSTP specifications to comprehend and verify MSTP, and to administer bridged local area networks using the MSTP specification.

In a network comprising STP Bridges, RSTP Bridges, and multiple MST Regions, treating the MST Regions as single Bridges provides the network administrator with a natural hierarchy. The internal management of MST Regions can be largely separated from the management of the active topology of the bridge local area network as a whole.

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

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

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

Within an MST Region, each MSTI operates in much the same way as an independent instance of RSTP with dedicated Regional Root Identifier, Internal Root Path Cost, and Internal Port Path Cost parameters. Moreover the overall spanning tree (the CIST) includes a fragment (the IST) within each MST Region that can be viewed as operating in the same way as an MSTI with the Regional Root as its root.

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

13.6 STP and RSTP compatibility

MSTP is designed to be STP and RSTP compatible and interoperable without additional operational management practice.

13.6.1 Designated Port Selection

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

A unique spanning tree priority (13.9) is required for each Bridge Port for STP, which has no other way of communicating port roles. Since port numbers on different bridges are not guaranteed to be unique, this necessitates the inclusion of the transmitting Bridge's Bridge Identifier in the STP BPDU. RSTP and MSTP's Port Protocol Migration state machines (13.29) ensure that all Bridges attached to any LAN with an attached STP Bridge send and receive STP BPDUs exclusively.

NOTE1 —This behavior satisfies the requirement for unique, agreed Designated Port for LANs with attached STP Bridges, but means that an MST Region cannot completely emulate a single Bridge since the transmitted Designated Bridge Identifier can differ on Bridge Ports at the Region's boundary.

MSTP transmits and receives the Regional Root Identifier and not the Designated Bridge Identifier in the BPDU fields recognized by RSTP (14.6) to allow both the MSTP and RSTP Bridges potentially connected to a single LAN to perform comparisons (13.9, 13.10) between all the spanning tree priority vectors transmitted that yield a single conclusion as to which RSTP Bridge or MST Region includes the Designated Port.

MST and RST BPDUs convey the transmitting port's CIST Port Role. This is checked on receipt by RSTP when receiving messages from a Designated Bridge (17.19.8 of IEEE Std 802.1D, 1998 Edition), thus ensuring that an RSTP Bridge does not incorrectly identify one MST Bridge Port as being Designated rather than another, even while omitting the competing Bridge Ports' Designated Bridge Identifiers from comparisons.

NOTE 2—This ability of MSTP Bridges to communicate the full set of MSTP information on shared LANs to which RSTP Bridges are attached avoids the need for the Port Protocol Migration machines to detect RSTP Bridges. Two or more MSTP and one or more RSTP Bridges may be connected to a shared LAN, with full MSTP operation. This includes the possibility of different MSTI Designated Ports (see 13.3.1).

13.6.2 Force Protocol Version

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

a) ST BPDUs, rather than MST BPDUs, are transmitted if Force Protocol Version is 0. RST BPDUs omit the MST Configuration Identifier and all MSTI Information.
b) RST BPDUs, rather than MST BPDUs, are transmitted if Force Protocol Version is 2. RST BPDUs omit the MST Configuration Identifier and all MSTI Information.
c) All received BPDUs are treated as being from a different MST Region if Force Protocol Version is 0 or 2.
d) The MSTP state machines disable rapid transitions if Force Protocol Version is 0. This allow MSTP Bridges to support applications and protocols that may be sensitive to the increased rates of frame duplication and misordering that can arise under some circumstances, as discussed in IEEE Std 802.1D F.2.4
e) The MSTP state machines allow full MSTP behavior if Force Protocol Version is 3 or more.

NOTE 1—Allowing for the case of a Force Protocol Version parameter value greater than 3 can simplify management of Bridges with different protocol versions.

NOTE 2—The Force Protocol Version parameter does not support multiple spanning trees with rapid transitions disabled.

13.7 MST Configuration Identification

It is essential that all Bridges within an MST Region agree on the allocation of VIDs to specific spanning trees. If the allocation differs, frames for some VIDs may be duplicated or not delivered to some LANs at all. MST Bridges check that they are allocating VIDs to the same spanning trees as their neighboring MST Bridges in the same Region by transmitting and receiving MST Configuration Identifiers along with the spanning tree information. These MST Configuration Identifiers, while compact, are designed so that two matching identifiers have a very high probability of denoting the same configuration even in the absence of any supporting management practice for identifier allocation.

NOTE 1—Suitable management practices for the deployment of equipment and the choice of Configuration Names and Revision Levels (see below) can be used to guarantee that the MST Configuration Identifiers will differ if the VID to spanning tree allocation differs within a single administrative domain.

Each MST Configuration Identifier contains the following components:

1) A Configuration Identifier Format Selector, the value 0 encoded in a fixed field of one octet to indicate the use of the following components as specified in this Standard.
2) The Configuration Name, a variable length text string encoded within a fixed field of 32 octets, conforming to RFC 2271's definition of SnmpAdminString.
3) The Revision Level, an unsigned integer encoded within a fixed field of 2 octets.

4) The Configuration Digest, a 16 octet signature of type HMAC-MD5 (see IETF RFC 2104) created from the MST Configuration Table (3.26, 8.11). For the purposes of calculating the Configuration Digest, the MST Configuration Table is considered to contain 4096 consecutive two octet elements, where each element of the table (with the exception of the first and last) contains an MSTID value encoded as a binary number, with the first octet being most significant. The first element of the table contains the value 0, the second element the MSTID value corresponding to VID 1, the third element the MSTID value corresponding to VID 2, and so on, with the next to last element of the table containing the MSTID value corresponding to VID 4094, and the last element containing the value 0. The key used to generate the signature consists of the 16 octet string specified in Table .

Table 13-1—Configuration Digest Signature Key

Parameter	Mandatory value
Configuration Digest Signature Key	0x13AC06A62E47FD51F95D2BA243CD0346

NOTE 2—The formulation of the signature as described above does not imply that a separate VID to MSTID translation table has to be maintained by the implementation; rather that it should be possible for the implementation to derive the logical contents of such a table, and the signature value as specified above, from the other configuration information maintained by the implementation, as described in Clause 12.

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

Table 13-2—Sample Configuration Digest Signature Keys

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

It is recommended that MST Bridge implementations provide an easily selectable or default configuration comprising a Configuration Name of the Bridge Address as a text string using the Hexadecimal Representation specified in IEEE Std 802-2001, a Revision Level of 0, and a Configuration Digest representing a VID to MSTID translation table containing the value 0 for every element. Such a table represents the mapping of all VLANs to the CIST. Since the Bridge Address is unique to each MST Bridge, no two MST Bridges using this default configuration will be identified as belonging to the same MST Region.

13.8 MST Regions

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

13.9 Spanning Tree Priority Vectors

All Bridges, whether they use STP, RSTP, or MSTP, send information to each other, in Configuration Messages (13.14 of this standard, 17.7 of IEEE Std 802.1D, 1998 Edition) to assign Port roles that determine each Port's participation in a fully and simply connected active topology based on one or more spanning trees. The information communicated is known as a *spanning tree priority vector*. Spanning tree priority vectors provide the basis for a concise specification of each protocol's computation of the active topology, in terms of both the entire Bridged LAN and the operation of individual Bridges in support of the distributed algorithm.

CIST priority vectors comprise the following components:

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

The CIST External Root Path Cost is significant throughout the Bridged LAN. It is propagated along each path from the CIST Root, and is added to at Bridge Ports that receive the priority vector from a Bridge in a different MST Region. The External Path Cost transmitted by a Bridge thus represents costs accumulated at the Root Ports of Bridges that are either not MST Bridges or are CIST Regional Roots, and is constant within a Region. The CIST Internal Root Path Cost is only significant and explicitly defined within a Region.

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

NOTE 2—The STP and RSTP specifications refer to the CIST Root Identifier and CIST External Root Path Cost simply as the Root Bridge Identifier and Root Path Cost respectively and omit the CIST Internal Root Path Cost. MSTP encodes the CIST Regional Root Identifier in the (External Designated) Bridge Identifier BPDU field used by RSTP to convey the Designated Bridge Identifier (14.3.3), so an entire MST Region appears to an RSTP capable Bridge as a single Bridge. However, this is not possible for STP, as the latter lacks the fields necessary for MST Bridges to communicate the Designated Bridge Identifier to resolve a potential priority vector tie, and MSTP BPDUs are not sent on a LAN to which an STP Bridge is attached.

MSTI priority vectors comprise the following components:

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

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

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

NOTE 3—The consistent use of lower numerical values to indicate better information is deliberate as the Designated Port that is closest to the Root Bridge, i.e. has a numerically lowest path cost component, is selected from amongst potential alternatives for any given LAN (13.9). Adopting the conventions that lower numerical values indicate better information, that where possible more significant priority components are encoded earlier in the octet sequence of a BPDU (14.3), and that earlier octets in the encoding of individual components are more significant (14.2) allow concatenated octets that compose a priority vector to be compared as if they were a multiple octet encoding of a single number, without regard to the boundaries between the encoded components. To reduce the confusion that naturally arises from having the lesser of two numerical values represent the better of the two, i.e. the one to be chosen all other factors being equal, this clause uses the following consistent terminology. Relative numeric values are described as “least”, “lesser”, “equal”, and “greater”, and their comparisons as “less than”, “equal to”, or “greater than”, while relative Spanning Tree priorities are described as “best”, “better”, “the same”, “different”, and “worse” and their comparisons as “better than”, “the same as”, “different from”, and “worse than”. The operators “<” and “=” represent less than and equal to respectively. The terms “superior” and “inferior” are used for comparisons that are not simply based on priority but include the fact that a priority vector can replace an earlier vector transmitted by the same Bridge Port. All these terms are defined for priority vectors in terms of the numeric comparison of components below (13.10, 13.11).

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

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

13.10 CIST Priority Vector Calculations

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

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

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

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

If B is not in the same MST Region as D, the Internal Root Path Cost is decoded as 0, as it has no meaning to B.

NOTE—If a Configuration Message is received in an RSTP or STP BPDU, both the Regional Root Identifier and the Designated Bridge Identifier are decoded from the single BPDU field used for the Designated Bridge Parameter (the MST BPDU field in this position encodes the CIST Regional Root Identifier). An STP or RSTP Bridge is always treated by MSTP as being in an MST Region of its own, so the Internal Root Path Cost is decoded as zero, and the tests below become the familiar checks used by STP and RSTP.

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

$$\begin{array}{l}(R_D == RootID)\ \&\&\ (ERC_D == ExtRootPathCost)\ \&\&\ (RR_D == RRootID)\ \&\& \\ (IRC_D == IntRootPathCost)\ \&\&\ (D == DesignatedBridgeID)\ \&\&\ (P_D == DesignatedPortID)\end{array}$$

and is better if:

$$\begin{array}{l}((R_D < RootID))\ || \\ ((R_D == RootID)\ \&\&\ (ERC_D < ExtRootPathCost))\ || \\ ((R_D == RootID)\ \&\&\ (ERC_D == ExtRootPathCost)\ \&\&\ (RR_D < RRootID))\ || \\ ((R_D == RootID)\ \&\&\ (ERC_D == ExtRootPathCost)\ \&\&\ (RR_D == RRootID) \\ \qquad \&\&\ (IRC_D < IntRootPathCost))\ || \\ ((R_D == RootID)\ \&\&\ (ERC_D == ExtRootPathCost)\ \&\&\ (RR_D == RRootID) \\ \qquad \&\&\ (IRC_D == IntRootPathCost)\ \&\&\ (D < DesignatedBridgeID))\ || \\ ((R_D == RootID)\ \&\&\ (ERC_D == ExtRootPathCost)\ \&\&\ (RR_D == RRootID) \\ \qquad \&\&\ (IRC_D == IntRootPathCost)\ \&\&\ (D == DesignatedBridgeID) \\ \qquad \&\&\ (P_D < DesignatedPortID))\end{array}$$

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

$$\begin{array}{l}(\{R_D : ERC_D : RR_D : IRC_D : D : P_D : P_B\} \\ \qquad \textit{is better than} \\ \ \{RootID : ExtRootPathCost : RRootID : IntRootPathCost : \\ \qquad DesignatedBridgeID : DesignatedPortID : RcvPortID\} \\ )\ ||\ ((D == DesignatedBridgeID)\ \&\&\ (P_D == DesignatedPortID))\end{array}$$

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

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

If the port priority vector was received from a bridge in a different MST Region (13.26.5), the External Port Path Cost EPC_{PB} is added to the External Root Path Cost component, and the Regional Root Identifier is set to the value of the Bridge Identifier for the receiving Bridge. The Internal Root Path Cost component will have been set to zero on reception.

$$\textit{root path priority vector} = \{R_D : ERC_D + EPC_{PB} : B : 0 : D : P_D : P_B)$$

If the port priority vector was received from a bridge in the same MST Region (13.26.5), the Internal Port Path Cost IPC_{PB} is added to the Internal Root Path Cost component.

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

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

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

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

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

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

13.11 MST Priority Vector Calculations

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

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

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

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

An MSTI message priority vector received from a Bridge that does not belong to the same MST Region is discarded.

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

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

and is better if:

$$\begin{aligned}&((RR_D < RRootID))\ \|\\ &((RR_D == RRootID)\ \&\&\ (IRC_D < IntRootPathCost))\ \|\\ &((RR_D == RRootID)\ \&\&\ (IRC_D == IntRootPathCost)\ \&\&\ (D < DesignatedBridgeID))\ \|\\ &((RR_D == RRootID)\ \&\&\ (IRC_D == IntRootPathCost)\ \&\&\ (D == DesignatedBridgeID)\\ &\qquad \&\&\ (P_D < DesignatedPortID))\end{aligned}$$

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

$$\begin{aligned}&(\{RR_D : IRC_D : D : P_D : P_B\}\\ &\qquad \textit{is better than}\\ &\{RRootID : IntRootPathCost : DesignatedBridgeID : DesignatedPortID : RcvPortID\}\\ &)\ ||\ ((D == DesignatedBridgeID)\ \&\&\ (P_D == DesignatedPortID))\end{aligned}$$

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

NOTE 1—the agreed flag (13.24.2) for the Port and this MSTI will be cleared if the CIST Root Identifier, CIST External Root Path Cost, and CIST Regional Root Identifier) in the received BPDU are not the same as those for the CIST designated priority vector for the port following processing of the received BPDU.

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

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

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

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

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

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

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

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

Figure 13-4 shows the priority vectors and the active topology calculated for an MSTI in a Region of the example network of Figure 13-3.

13.12 Port Role Assignments

Port Role assignments for Bridge Ports that are enabled are determined by each bridge in the Bridged Local Area Network (13.12) according to the source and relative priority of the spanning tree port priority vectors (13.9) selected for each Port following priority vector calculations (13.10, 13.11).

Each MST Bridge Port that is enabled is assigned a Port Role for each spanning tree. First one of the following roles: Root Port, Designated Port, Alternate Port, or Backup Port, is assigned for the CIST.

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

Then one of these roles, or the additional role of Master Port, is assigned for each MSTI.

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

If the Port is not enabled, it is assigned the Disabled Port role for the CIST and all MSTIs, to identify it as having no part in the operation of any of the spanning trees or the active topology of the network.

13.13 Stable Connectivity

This clause provides an analysis to show that MSTP meets its goal of providing full and simple connectivity for frames allocated to any given VLAN in a stable network, i.e. where the physical topology has remained constant for long enough that the spanning tree information communicated and processed by Bridges is not changing on any Bridge Port.

Each MST Region independently can allocate such frames to the IST or any given MSTI. Root Ports, Designated Ports, and Master Ports forward data frames, and Alternate, Backup, and Disabled Ports do not.

NOTE—The term Common Spanning Tree (CST) refers to the CIST connectivity between Regions, and the term Internal Spanning Tree (IST) to the CIST connectivity within each Region.

The CIST interconnects both individual LANs and Bridges, and complete MST Regions into a single spanning tree, each Region being part of the CIST as a whole. Frames with VIDs consistently allocated to the CIST within every MST Region follow an active topology determined by the minimum path costs to each Bridge and LAN provided by that single tree throughout the network, and thus enjoy full and simple connectivity.

Frames otherwise allocated follow the CIST outside and an MSTI within an MST Region. Simple and, in the absence of continual changes in physical connectivity, full connectivity of this composite active topology is ensured as follows:

a) Each Bridge or LAN is in one and only one Region.
(SST Bridges, LANs connected to STP Bridges, and LANs whose Designated Bridge is an SST Bridge, are all conveniently regarded as being in a Region of their own.)
b) Each and every frame is associated with one and only one VID.
c) Frames with any given VID are allocated either to the IST or to a given MSTI within any given Region, i.e. all frames are allocated to some tree and no frames are allocated to more than one tree.
d) The IST and each MSTI provides full and simple connectivity between all LANs and Bridges in an MST Region for frames allocated to the IST or that MSTI.

Hence full and simple connectivity is provided for all frames from any Bridge or LAN within an MST Region to any other within the Region.

Further:

e) All Bridges within an MST Region with ports connected to a given LAN reach a consistent agreement as to whether each of those ports is or is not a Boundary Port (i.e. attaches a Bridge to a LAN that is not in the same Region) prior to forwarding frames.
(MST Bridges make the determination on the basis of the CIST Designated Port for the LAN or the selection of the protocol by the Protocol Migration machines, both are necessarily complete prior to frame forwarding. SST Bridges being unaware of MST Regions behave as if each LAN is in a different Region to the Bridge.)
f) At a Boundary Port frames allocated to the CIST and all MSTIs are forwarded or not forwarded alike. This is because Port Role assignments are such that if the CIST Port Role is Root Port the MSTI Port Role will be Master Port, and if the CIST Port Role is Designated Port, Alternate Port, Backup Port, or Disabled Port, each MSTI's Port Role will be the same.
g) The CIST provides full and simple connectivity between all LANs and Bridges in the network, including the LANs and Bridges attached to the Boundary Ports of any MST Region.

Hence full and simple connectivity is provided for all frames between Bridges and LANs outside the MST Region since those frames will be carried across the MST Region if necessary, just as if they were allocated to the CIST whichever tree they are allocated to within the Region.

Similarly full and simple connectivity is provided for all frames between a Bridge or LAN inside the Region and a Bridge or LAN outside the region since the connectivity provided from within the Region by an MSTI to that outer Bridge or LAN is the same as that provided by the CIST.

Figure 13-5 illustrates the above connectivity with the simple example of Region 1 from the example network of Figure 13-3 and Figure 13-3. Bridge 0.42 has been selected as the CIST Root and Regional Root, Bridge 0.57 as the Regional Root for MSTI 1, and Bridge 2.83 for MSTI 2 by management of the per MSTI Bridge Identifier priority component. The potential loop through the three bridges in the Region is blocked at different Bridge Ports for the CIST, and each MSTI, but the connectivity across the Region and from each LAN and Bridge in the region through the boundaries of the Region is the same in all cases.

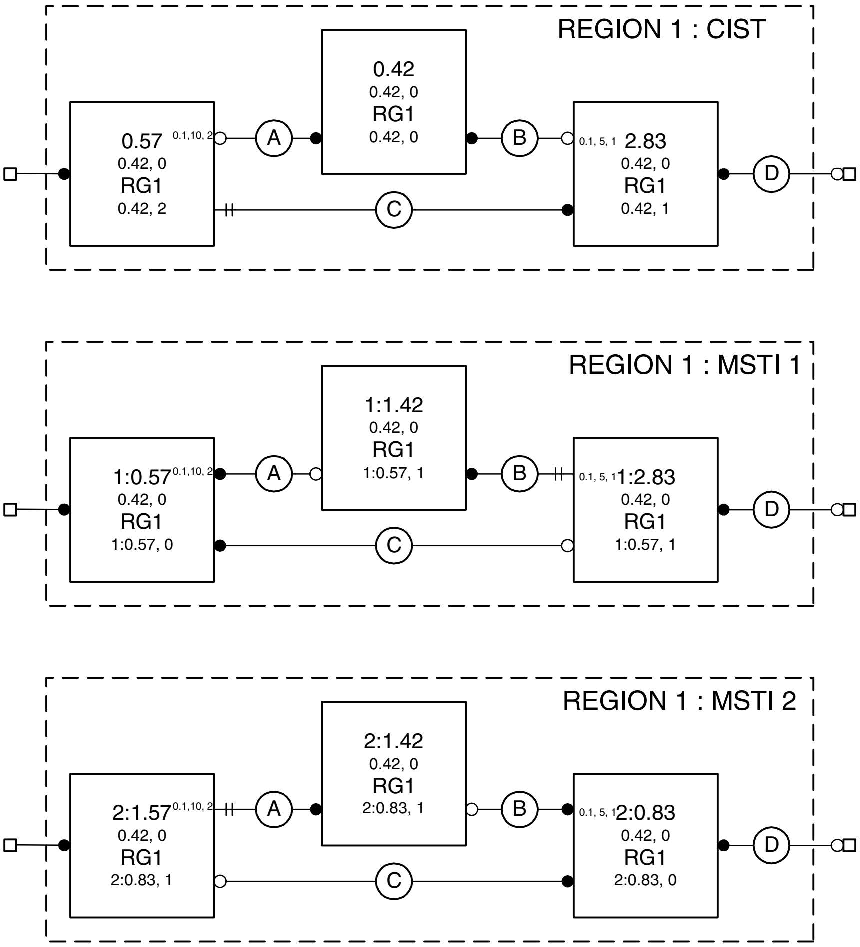


Figure 13-5—CIST and MSTI active topologies in Region 1 of the example network

13.14 Communicating Spanning Tree Information

Bridges transmit and receive MAC frames, each containing a Bridge Protocol Data Unit (BPDU) (Clauses 9 and 14 of IEEE Std 802.1D, 1998 Edition), to communicate Spanning Tree messages. A MAC frame conveying a BPDU carries the Bridge Group Address in the destination address field and is received by all the Bridges connected to the LAN on which the frame is transmitted. The Bridge Group Address is one of a small number of addresses that identify frames that are not directly forwarded by Bridges (7.12.6), but the information contained in the BPDU can be used by a Bridge in calculating its own BPDUs to transmit, and can stimulate that transmission.

BPDUs are used to convey three types of Spanning Tree message:

a) Configuration Messages;
b) Topology Change Notification (TCN) Messages;
c) MST Configuration Identifiers.

A Configuration Message for the CIST can be encoded and transmitted in an STP Configuration BPDU (9.3.1), an RST BPDU (9.3.3), or an MST BPDU (14). A TCN Message for the CIST can be encoded in an STP Topology Change Notification BPDU (9.3.2), an RST BPDU with the TC flag set, or an MST BPDU. Configuration and TCN Messages for the CIST and for all MSTIs in an MST Region are encoded in a single MST BPDU, as is the MST Configuration Identifier. No more than 64 MSTI Configuration Messages may be encoded in an MST BPDU, and no more than 64 MSTIs may be supported by an MST Bridge.

Configuration and Topology Change Notification BPDUs are distinguished from each other and from RST and MST BPDUs by their BPDU Type (Clause 9 of IEEE Std 802.1D, 1998 Edition). RST and MST BPDUs share the same BPDU Type and are distinguished by their version identifiers. Bridges implementing STP (Clause 8 of IEEE Std 802.1D, 1998 Edition) transmit and decode Configuration and Topology Change Notification BPDUs, and ignore RST and MST BPDUs on receipt. This ensures that connection of a Bridge Port of such a Bridge to a LAN that is also attached to by a Bridge implementing RSTP or MSTP is detected, as transmission of RSTP or MSTP BPDUs does not suppress regular transmissions by the STP Bridge. This functionality is provided by the Port Protocol Migration state machine for RSTP (17.26 of IEEE Std 802.1D, 1998 Edition) and MSTP (17.26). The Port Protocol Migration state machines selects the BPDU types used to encode Spanning Tree messages so that all Bridges attached to the same LAN participate in a spanning tree protocol, while maximizing the available functionality. If one or more attached Bridges only implement STP, only Configuration and Topology Change Notification BPDUs will be used and the functionality provided by the protocol will be constrained.

Each Configuration Message contains, among other parameters, a message priority vector (13.4.2, 13.4.3). This allows a receiving Bridge to determine the Port Role (13.4.4) including that of Designated Port. Configuration Messages are transmitted if the information to be transmitted by a Designated Port changes, or if a Root Port has an Agreement to convey. In addition Designated Ports transmit Configuration Messages at regular intervals to guard against loss and to assist in the detection of failed components (LANs, Bridges, or Bridge Ports). In both cases, message transmission is subject to a maximum transmission rate (see Transmission Limit in 13.28.2).

13.15 Changing Spanning Tree Information

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

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

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

If a Bridge Port's MAC_Operational parameter becomes FALSE, the Port becomes a Disabled Port and received spanning tree information is immediately discarded. Spanning tree information for the tree can be recomputed, the Bridge's Port Roles changed, and new spanning tree information transmitted if necessary.

Not all component failure conditions can be detected in this way, so each Designated Port transmits spanning tree information at regular intervals and a receiving Port will discard information and become a Designated Port if two transmissions are missed.

The Spanning Tree Protocol (STP, Clause 8 of IEEE Std 802.1D, 1998 Edition) and the Rapid Spanning Tree Protocol (RSTP, Clause 17 of IEEE Std 802.1D, 1998 Edition) do not use a hop count and detect both circulating aged information and loss of connectivity to a neighboring bridge by means of Message Age and Max Age (maximum message age) parameters. To ensure compatibility MSTP increments Message Age for information received at the boundary of an MST Region, discarding the information if necessary.

NOTE 1—MSTP's use of a separate hop count and message loss detection timer provides superior reconfiguration performance compared to STP and RSTP's use of Message Age and Max Age. Detection of loss of connectivity to a neighboring Bridge is not compromised by the need to allow for the overall diameter of the network, nor does the time allowed extend the number of hops permitted to aged recirculating information. Management calculation of the necessary parameters for custom topologies is also facilitated, as no allowance needs to be made for relative timer jitter and accuracy in different Bridges.

NOTE 2—The hop count applies to the spanning tree information for each tree. Message loss detection applies to all information transmitted by a given Bridge. The separate hop count can be more compactly encoded than the Message Age and Max Age timer values, and thus provides some per tree encoding efficiency in MST BPDUs.

13.16 Changing Port States

The Port State for each Bridge Port and spanning tree (CIST and MSTIs) is controlled by state machines whose goal is to maximize connectivity without introducing temporary data loops in the network. Root Ports, Master Ports, and Designated Ports are transitioned to the Forwarding Port State, and Alternate Ports and Backup Ports to the Discarding Port State, as rapidly as possible.

Transitions to the Discarding Port State can be simply effected without the risk of data loops. This clause describes the analysis used to determine the conditions for transitioning the Port State for a given spanning tree to Forwarding.

Starting with the assumption that any connected fragment of a network is composed of Bridges, Bridge Ports, and connected LANs that form a subtree of a spanning tree, this clause derives the conditions for transitioning ports with Root Port, Master Port, or Designated Port roles, such that the newly enlarged fragment continues to form either a subtree or the whole of the spanning tree. Since these conditions are applied every time a fragment is enlarged it is possible to trace the growth of a fragment from a single Bridge, which is clearly a consistent, if small, subtree of a spanning tree, to any sized fragment—thus justifying the initial assumption.

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

13.16.1 Subtree connectivity and priority vectors

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

13.16.2 Root Port transition to Forwarding

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

13.16.3 Designated Port transition to Forwarding

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

Figure 13-6 illustrates the generation of an Agreement at a Bridge's Root Port from an Agreement received or a Port State of Discarding at each of its Designated Ports, and a Port State of Discarding at each of its Alternate and Backup Ports. To solicit an Agreement each Designated Port that has been set to discard frames sends a Proposal. A Bridge receiving a Proposal transitions any Designated Port not already synchronized to Discarding, and solicits an Agreement by sending a Proposal in its turn.

NOTE 1—Agreements can be generated without prior receipt of a Proposal as soon as the conditions for the Agreement have been met. In that case subsequent receipt of a Proposal serves to elicit a further Agreement.

NOTE 2—If all Designated Ports have already been synchronized and the spanning priority vector received with the proposal does not convey worse information the synchronization is maintained and there is no need to transition Designated Ports to Discarding once more, or to transmit further Proposals.

13.16.4 Master Port transition to Forwarding

While the connectivity of the CIST from the CIST Regional Root through the Region to the rest of the CIST comprises a subtree rooted in the CIST Regional Root, the connectivity of the MSTI from the Master Port includes both a subtree below the CIST Regional Root and a subtree rooted in the MSTI Regional Root and connected to the CIST Regional Root by an MSTI Root Port. Figure 13-7 illustrates this connectivity for both part of the CIST and an MSTI through a Region in the example network of Figure 13-3. (In the example this latter subtree provides connectivity from the Master Port through LAN N to the subtree of the CIST outside the Region). Prior to the Master Port's transition to Forwarding it is possible that either MSTI subtree is providing connectivity to a prior Master Port. Before the Master Port can transition the connectivity of both subtrees has to agree with the new CIST Regional Root.

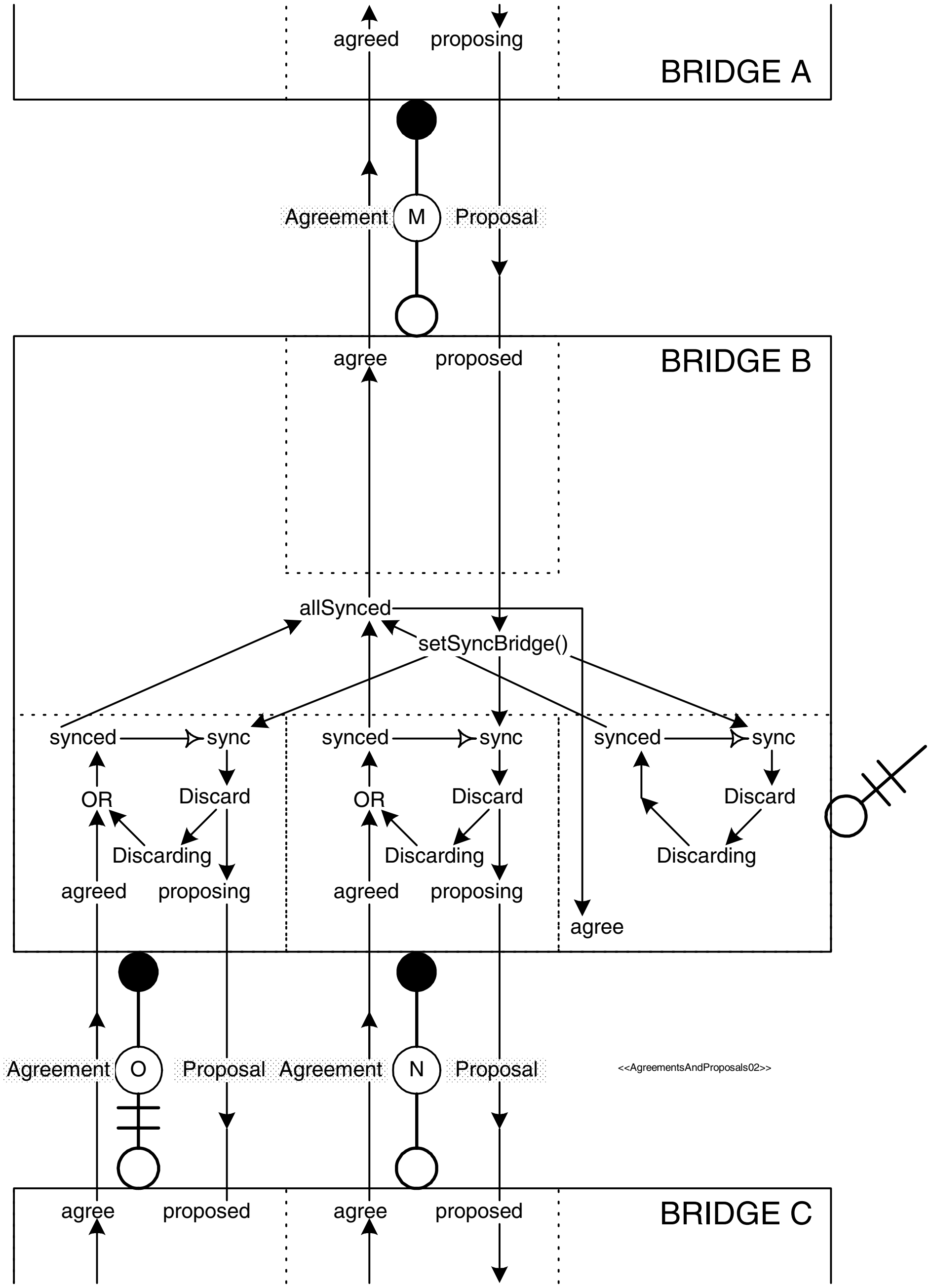
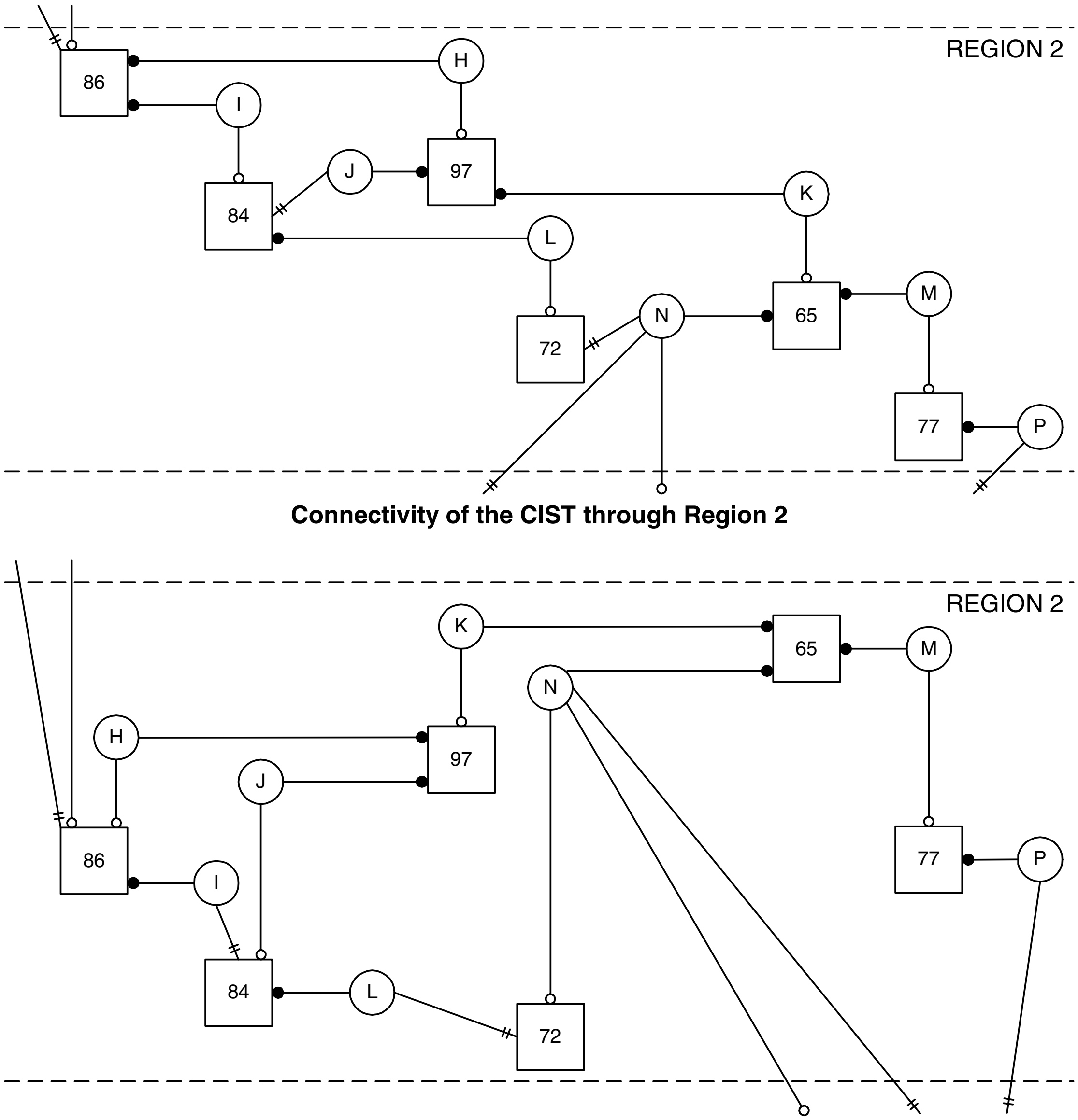


Figure 13-6—Agreements and Proposals

NOTE 1—The physical layout shown in the two halves of Figure 13-7 differs in order to reflect the different priorities and logical topologies for the two spanning tree instances. The layout convention used is that designated Ports are shown as horizontal lines, root Ports as vertical lines, and alternate Ports as diagonal lines.



Connectivity of the CIST through Region 2

Connectivity of an MSTI through Region 2

Figure 13-7—CSTI and MSTI Active Topologies in a Region

Figure 13-8 illustrates the extension of the Agreement mechanism to signal from Designated Ports to Root Ports as well as vice versa. To ensure that an MSTI does not connect alternate Master Ports, an Agreement is only recognized at an MSTI Port when the CIST Regional Root associated with the information matches that selected by the receiving port. Proposals, eliciting Agreements, necessarily flow from Designated Ports to Root Ports with the propagation of spanning tree information so a new CIST Regional Root cannot transmit a Proposal directly on its MSTI Root Ports. However updating of a CIST Designated Port's port priority vector with a new Regional Root Identifier forces the port to discard frames for all MSTIs, thus initiating the Proposal from the first Bridge nearer the MSTI Regional Root that learns of the new Regional Root.

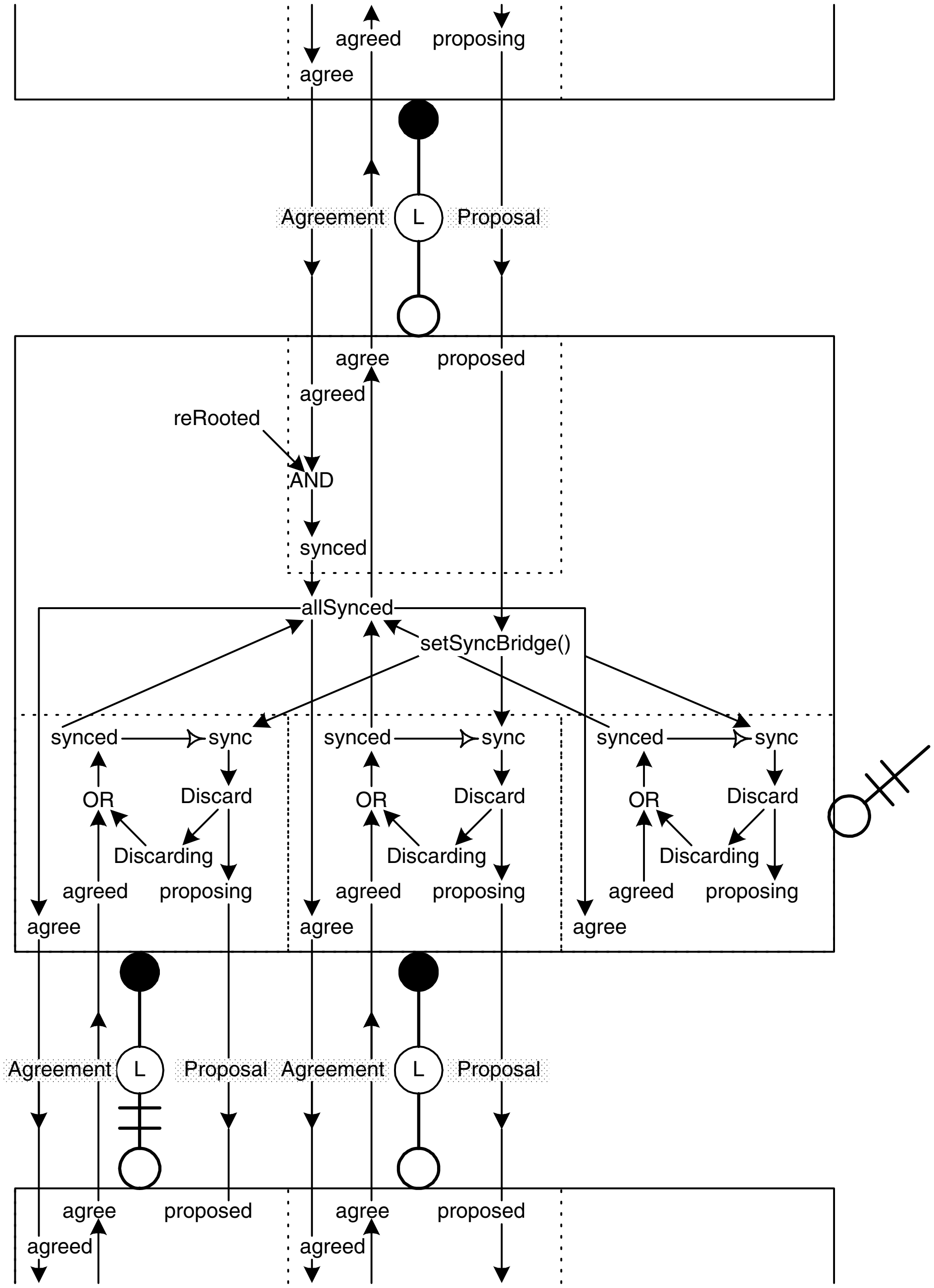


Figure 13-8—Enhanced Agreements

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

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

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

NOTE 4—In the important topology where the CIST Root Bridge is chosen to be within an MST Region, cuts are not transferred from the CIST to any MSTI in that Region. Thus the propagation of cuts in the CIST will not disrupt MSTI connectivity in the Region.

NOTE 5—Topology change detection by MSTI state machines is based on changes from Root, Master, and Designated Port Roles to Alternate, Backup, Disabled Port Roles, not on changes in Port State. Propagating 'cuts' designed to prevent temporary loops through a Region does not therefore require unnecessary changes to Filtering Databases with attendant temporary flooding of frames.

13.17 Updating Learned Station Location Information

A spanning tree reconfiguration can cause end stations to appear to move from the point of view of any given Bridge, even if that Bridge's Port States do not change, and is signaled from the Bridge whose Port Roles have changed to others using TCN messages. MST BPDUs encode separate TCN messages for the CIST and each MSTI, and MSTP supports the optimizations specified for RSTP (Clause 17.10 of IEEE Std 802.1D, 1998 Edition). Together these facilitate removal of entries for the minimum set of Ports from the Filtering Databases associated with the spanning trees whose active topology has changed. In addition MSTP only detects topology changes following a change of Port Role to Root Port, Master Port, or Designated Port. Temporary cuts in the active topology, introduced to ensure that rapid Port State transitions to Forwarding do not cause loops, and do not therefore cause Filtering Database entries to be flushed throughout the network, unless they are accompanied by Port Role changes.

Changes in the active topology of any MSTI do not change end station locations for the CIST or any other MSTI, unless the underlying changes in the physical topology that gave rise to the reconfiguration also cause those trees to reconfigure. Changes to the CST, i.e. the connectivity provided by the CIST between MST Regions, can cause end station location changes for all trees. Changes to an IST can cause CST end station location changes, but do not affect MSTIs in that Region unless those trees also reconfigure.

NOTE 1—The shorthand terms “end station locations for a given tree”, “the CST”, and “an IST”, are used to mean “the apparent location of end stations as recorded by filtering databases associated with the given tree”, “the connectivity provided by the CIST between and not internal to MST Regions”, and “the connectivity provided by the CIST internal to a given MST Region” respectively.

On receipt of a CIST TCN Message from a Bridge Port not internal to the Region, or on a change in Port Role for a Bridge Port not internal to the Region, TCN Messages are transmitted through each of the other Ports of the receiving Bridge for each MSTI, and the Filtering Databases for those ports are flushed.

NOTE 2—TCN Messages for the CIST are always encoded in the same way, irrespective of whether they are perceived to have originated from topology changes internal to the Region or outside it. This allows RSTP Bridges whose Root Ports attach to a LAN within an MST Region to receive these TCN Messages correctly.

NOTE 3—The Port receiving a CIST TCN Message from another Bridge Port external to the Region can be a Master Port, a Designated Port attached to the same LAN as an STP Bridge, or a Designated Port attached to a LAN that is within the Region but is attached to by the Root Ports of Bridges in other Regions.

13.18 MSTP and point-to-point links

MSTP uses the adminPointToPointMAC and operPointToPointMAC parameters (6.4.3 of IEEE Std 802.1D, 1998 Edition) to allow the point-to-point status of LANs to be manipulated administratively, and the operational state to be signalled to the MSTP state machines. This use, paralleling that of RSTP (17.11 of IEEE Std 802.1D, 1998 Edition), facilitates use of the Agreement mechanism (13.16) to enable rapid Forwarding Port State transitions.

13.19 Multiple Spanning Tree State Machines

The operation of the Multiple Spanning Tree Protocol is represented by the following common set of state machines:

a) A Port Timers state machine for the Bridge (13.27)
b) A Port Protocol Migration state machine for each Port (13.29)
c) A Port Receive state machine for each Port (13.28)
d) A Port Transmit state machine for each Port (13.30)

with the following set for the CIST and each MSTI:

e) A Port Information state machine for each Port (13.31)
f) A Port Role Selection state machine for the Bridge (13.32)
g) A Port Role Transitions state machine for each Port (13.33)
h) A Port State Transition state machine for each Port (13.34)
i) A Topology Change state machine for each Port (13.35)

The operation of each state machine and its associated variable and procedural definitions is specified in detail below. Each is modeled on the corresponding state machine for RSTP as described in Clause 17 of IEEE Std 802.1D, 1998 Edition. Modifications:

1) support both the CIST and the MSTIs

2) use the extended spanning tree priority vector definition and calculations for the CIST (13.10)
3) provide communication between the CIST state machines and the MSTI state machines as required by 13.16, 13.19, and 13.17.
4) enhance the operation of the topology change state machine to avoid unnecessary removal of end station location information following port state transitions without changes in port roles.

All references to named variables or procedures in the specification of the state machines are to those corresponding to the instance of the state machine using the function unless explicit reference is made to the CIST or given MSTIs.

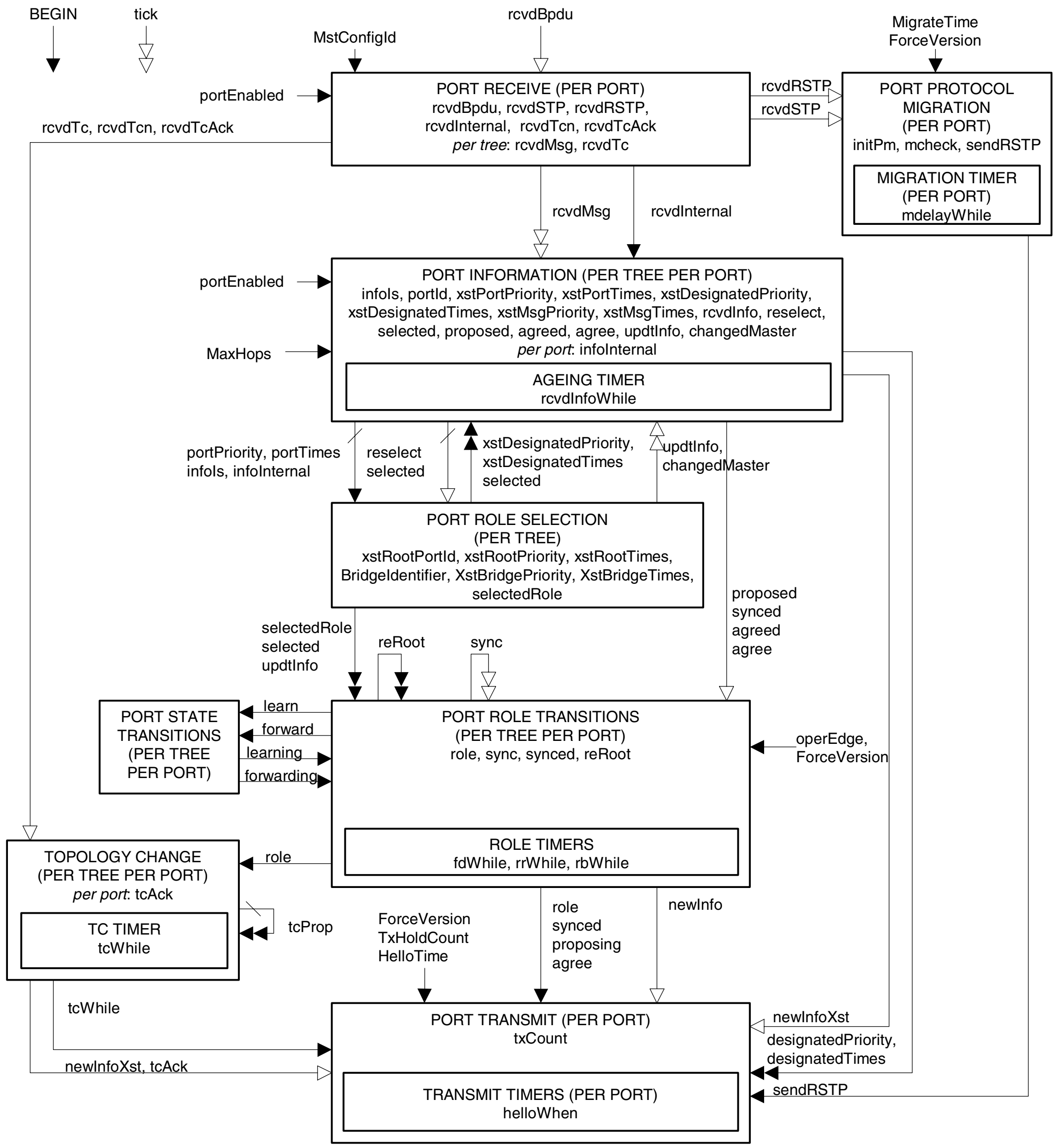
For further economy of specification, the names of CIST and MSTI conditions or procedures providing the same general functionality but differing in detail incorporate the prefix or suffix “Cist” or “Msti”. These are substituted for names beginning or ending in “Xst” in a single state machine description common to the CIST and any given MSTI.

NOTE 1—The specification of RSTP does not use a distinct Port Receive state machine, but combines the functionality into the Port Information state machine. Since separate instances of the latter are required for each tree, each operating on the messages contained in a single BPDU, it is convenient to separate out the aspects of reception that are purely per Port and per BPDU in this specification into a separate Port Receive state machine.

NOTE 2—Individual MSTIs do not implement their own Port Transmit state machine, but signal the need to transmit to a single Port Transmit State machine by setting the newInfoMsti variable. The procedures used by this machine transmit information for the CIST and all MSTIs in a single BPDU.

NOTE 3—A detailed list of the differences between the RSTP state machines specified in IEEE Std 802.1D, 1998 Edition as amended by IEEE Std 802.1D, 1998 Edition and the state machines specified in this standard is presented in Annex G.

Figure 13-9 illustrates the state machines, their state variables and communication between state machines. This overview diagram is not itself a state machine, but serves to illustrate the principal variables that are used to communicate between the individual machines and the variables local to each machine Figure 13-10 describes its notation.


BEGIN
tick
MstConfigId
rcvdBpdu
MigrateTime
ForceVersion
portEnabled
PORT RECEIVE (PER PORT)
rcvdBpdu, rcvdSTP, rcvdRSTP,
rcvdInternal, rcvdTcn, rcvdTcAck
per tree: rcvdMsg, rcvdTc
rcvdRSTP
rcvdSTP
PORT PROTOCOL MIGRATION (PER PORT)
initPm, mcheck, sendRSTP
MIGRATION TIMER (PER PORT)
mdelayWhile
rcvdTc, rcvdTcn, rcvdTcAck
rcvdMsg
rcvdInternal
portEnabled
MaxHops
PORT INFORMATION (PER TREE PER PORT)
infoIs, portId, xstPortPriority, xstPortTimes, xstDesignatedPriority,
xstDesignatedTimes, xstMsgPriority, xstMsgTimes, rcvdInfo, reselect,
selected, proposed, agreed, agree, updtInfo, changedMaster
per port: infoInternal
AGEING TIMER
rcvdInfoWhile
portPriority, portTimes
infoIs, infoInternal
reselect
selected
xstDesignatedPriority,
xstDesignatedTimes
selected
updtInfo,
changedMaster
PORT ROLE SELECTION (PER TREE)
xstRootPortId, xstRootPriority, xstRootTimes,
BridgeIdentifier, XstBridgePriority, XstBridgeTimes,
selectedRole
proposed
synced
agreed
agree
selectedRole
selected
updtInfo
reRoot
sync
PORT STATE TRANSITIONS (PER TREE PER PORT)
learn
forward
learning
forwarding
PORT ROLE TRANSITIONS (PER TREE PER PORT)
role, sync, synced, reRoot
operEdge,
ForceVersion
ROLE TIMERS
fdWhile, rrWhile, rbWhile
TOPOLOGY CHANGE (PER TREE PER PORT)
per port: tcAck
role
TC TIMER
tcWhile
tcProp
ForceVersion
TxHoldCount
HelloTime
role
synced
proposing
agree
newInfo
tcWhile
newInfoXst, tcAck
PORT TRANSMIT (PER PORT)
txCount
TRANSMIT TIMERS (PER PORT)
helloWhen
newInfoXst
designatedPriority,
designatedTimes
sendRSTP

NOTE: For convenience all timers are collected together into one state machine.

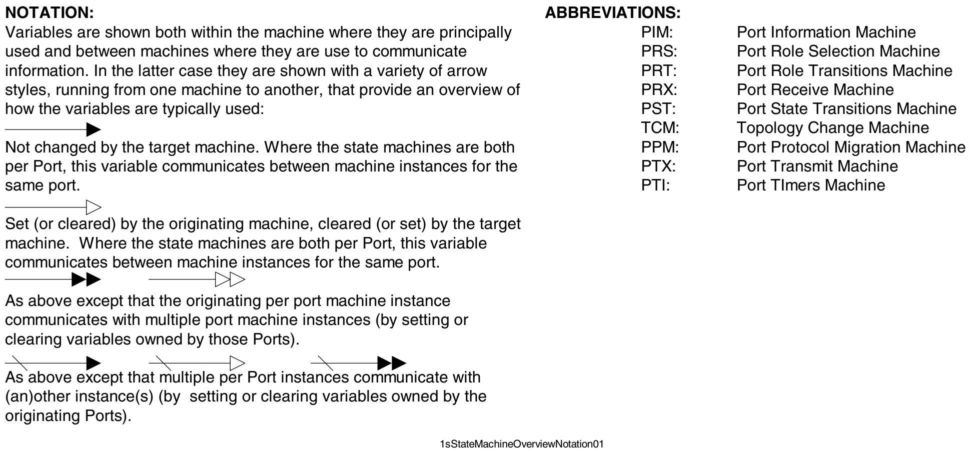
Figure 13-9—MSTP state machines—overview and relationships

13.20 Notational Conventions used in State Diagrams

The notational conventions used in the specification of MSTP are identical to those used in the specification of RSTP and defined in 17.14 of IEEE Std 802.1D, 1998 Edition.

13.21 State Machine Timers

Each of the state machine timers are as specified in 17.15 of IEEE Std 802.1D, 1998 Edition.



NOTATION:

Variables are shown both within the machine where they are principally used and between machines where they are use to communicate information. In the latter case they are shown with a variety of arrow styles, running from one machine to another, that provide an overview of how the variables are typically used:

Not changed by the target machine. Where the state machines are both per Port, this variable communicates between machine instances for the same port.

Set (or cleared) by the originating machine, cleared (or set) by the target machine. Where the state machines are both per Port, this variable communicates between machine instances for the same port.

As above except that the originating per port machine instance communicates with multiple port machine instances (by setting or clearing variables owned by those Ports).

As above except that multiple per Port instances communicate with (an)other instance(s) (by setting or clearing variables owned by the originating Ports).

ABBREVIATIONS:

PIM:	Port Information Machine
PRS:	Port Role Selection Machine
PRT:	Port Role Transitions Machine
PRX:	Port Receive Machine
PST:	Port State Transitions Machine
TCM:	Topology Change Machine
PPM:	Port Protocol Migration Machine
PTX:	Port Transmit Machine
PTI:	Port TImers Machine

Figure 13-10—MSTP overview notation

One instance of the following shall be implemented per-Port:

a) mdelayWhile
b) helloWhen

One instance per-Port of the following shall be implemented for the CIST and one per-Port for each MSTI:

c) fdWhile
d) rrWhile
e) rbWhile
f) tcWhile
g) rcvdInfoWhile

13.22 State Machine Performance Parameters

These parameters are treated as constants by the CIST and MSTI state machines; their values can be modified only by management action.

The following parameters are as specified in 17.16 of IEEE Std 802.1D, 1998 Edition for RSTP. A single value of each parameter applies to the MST Bridge as a whole, including all Ports and all CIST and MSTI state machines.

a) ForceVersion
b) FwdDelay
c) TxHoldCount
d) MigrateTime

The following parameter is as specified in 17.16 of IEEE Std 802.1D, 1998 Edition for RSTP, but may be managed separately for each Port.

e) HelloTime

The following parameter is additional to those specified for RSTP. A single value applies to all Spanning Trees within an MST Region (the CIST and all MSTIs) for which the Bridge is the Regional Root.

f) MaxHops

The following parameter is as specified for RSTP, but has been renamed for clarity in this specification. One value per-Port applies to the CIST.

g) ExternalPortPathCost (specified as PortPathCost in 17.16 of IEEE Std 802.1D, 1998 Edition)

The following parameter is additional to those specified for RSTP, and may be managed separately for the CIST and for each MSTI per-Port.

h) InternalPortPathCost

13.23 Per-Bridge Variables

Per-bridge variable(s) perform the functions described in 17.17 of IEEE Std 802.1D, 1998 Edition, but have enhanced or extended specifications or considerations.

A single instance of each of the following variables applies to the CIST and to all MSTIs.

a) BEGIN (13.23.1)
b) MstConfigId (13.23.8)

NOTE—MstConfigId is not specified in 17.17 of IEEE Std 802.1D, 1998 Edition.

There is one instance per-Bridge of each of the following for the CIST, and one for each MSTI.

c) BridgeIdentifier (13.23.2)

And one instance per-Bridge of each of the following for the CIST.

d) CistBridgePriority (13.23.3)
e) CistBridgeTimes (13.23.4)
f) cistRootPortId (13.23.5)
g) cistRootPriority (13.23.6)
h) cistRootTimes (13.23.7)

And one instance per-Bridge of each of the following for each MSTI.

i) MstiBridgePriority (13.23.9)
j) MstiBridgeTimes (13.23.10)
k) mstiRootPortId (13.23.11)
l) mstiRootPriority (13.23.12)
m) mstiRootTimes (13.23.13)

13.23.1 BEGIN

This variable is controlled by the system initialization process. A value of TRUE causes all CIST and MSTI state machines, including per Port state machines, to transit to their initial state. A value of FALSE allows all state machines to perform transitions out of their initial state, in accordance with the relevant state machine definitions.

Changes to any of the following parameters cause BEGIN to be asserted for the state machines for the Bridge, for all trees, and for each Port:

a) The MST Configuration Identifier.

13.23.2 BridgeIdentifier

The unique Bridge Identifier assigned to this Bridge for this tree (CIST or MSTI).

The 12-bit system ID extension component of a Bridge Identifier (9.2.5 of IEEE Std 802.1D, 1998 Edition) shall be set to zero for the CIST, and to the value of the MSTID for an MSTI, thus allocating distinct Bridge Identifiers to the CIST and each MSTI all based on the use of a single Bridge Address component value for the MST Bridge as a whole.

NOTE—This convention is used to convey the MSTID for each MSTI Configuration Message encoded in an MST BPDU.

The four most significant bits of the Bridge Identifier (the settable Priority component) for the CIST and for each MSTI can be modified independently of the setting of those bits for all other trees, as a part of allowing full and independent configuration control to be exerted over each Spanning Tree instance.

13.23.3 CistBridgePriority

The value of the CIST bridge priority vector, as defined in 13.10. The CIST Root Identifier, CIST Regional Root Identifier, and Designated Bridge Identifier components are all equal to the value of the CIST Bridge Identifier. The remaining components (External Root Path Cost, Internal Root Path Cost, Designated Port Identifier) are set to zero.

CistBridgePriority is used by updtCistRolesBridge() in determining the value of the cistRootPriority variable (see 13.23.6).

13.23.4 CistBridgeTimes

CistBridgeTimes comprises:

a) The current values of Bridge Forward Delay and Bridge Max Age (see Table 17-5 of IEEE Std 802.1D, 1998 Edition). These parameter values are determined only by management;
b) A Message Age value of zero.
c) The current value of MaxHops (13.22).

CistBridgeTimes is used by updtCistRolesBridge() in determining the value of the cistRootTimes variable (13.23.7).

13.23.5 cistRootPortId

The Port Identifier of the Root Port, and a component of the CIST root priority vector (13.10).

13.23.6 cistRootPriority

The CIST Root Identifier, CIST External Root Path Cost, CIST Regional Root Identifier, CIST Internal Root Path Cost, Designated Bridge Identifier, and Designated Port Identifier components of the Bridge's CIST root priority vector (13.10).

13.23.7 cistRootTimes

The Bridge's timer parameter values (Message Age, Max Age, Forward Delay, and remainingHops). The values of these timers are derived from the values stored in cistPortTimes (17.18.8 of IEEE Std 802.1D, 1998 Edition) for the Root Port. Max Age and Forward Delay are set equal to the values held by the Root Port. If the CIST root priority vector was received from a Bridge in a different MST Region (infoInternal

FALSE) Message Age is the value held by the Root Port incremented by the greater of (1/16 Max Age) and 1, rounded to the nearest whole second (see 17.19.21 of IEEE Std 802.1D, 1998 Edition), and remainingHops is set equal to MaxHops. Otherwise, if the CIST root priority vector was received from a Bridge in the same MST Region or the Bridge is itself the CIST Root (rcvdInternal TRUE), Message Age is the value held by the Root Port (in cistRootTimes) and remainingHops is the value held by the Root Port minus one.

13.23.8 MstConfigId

The value of the MST Configuration Identifier (13.7) corresponding to the Bridge's current MST Region Configuration.

13.23.9 MstiBridgePriority

The value of the MSTI bridge priority vector for a given MSTI (13.11). The MSTI Regional Root Identifier and Designated Bridge Identifier components are equal to the value of the MSTI Bridge Identifier (13.23.2). The remaining components (MSTI Internal Root Path Cost, Designated Port Identifier) are set to zero.

MstiBridgePriority is used by updtMstiRolesBridge() in determining the value of the mstiRootPriority variable (see 13.23.12).

13.23.10 MstiBridgeTimes

MstiBridgeTimes for a given MSTI is composed of a single component:

a) The current value of Bridge Max Hops (13.22). This parameter value is determined only by management.

MstiBridgeTimes is used by updtMstiRolesBridge() in determining the value of the mstiRootTimes variable (see 13.23.13).

13.23.11 mstiRootPortId

The Port Identifier of the Root Port for a given MSTI, and a component of the root priority vector (13.11).

13.23.12 mstiRootPriority

The MSTI Regional Root Identifier, MSTI Internal Root Path Cost, MSTI Designated Bridge Identifier, and MSTI Designated Port Identifier components of the Bridge's root priority vector (13.11) for a given MSTI.

13.23.13 mstiRootTimes

The value of remainingHops for a given MSTI as stored in mstiPortTimes (13.24.18) for the MSTI Root Port, derived from the value received in mstiMsgTimes by subtracting one.

13.24 Per-Port Variables

The following variables perform the function specified in 17.17 of IEEE Std 802.1D, 1998 Edition. A single per-Port instance applies to the CIST and to all MSTIs.

a) tick
b) txCount
c) operEdge
d) portEnabled

A single per-Port instance of the following variable(s) not specified in IEEE Std 802.1D, 1998 Edition applies to the CIST and or all MSTIs.

e) infoInternal (13.24.10)
f) newInfoCist (13.24.19)
g) newInfoMsti (13.24.20)
h) rcvdInternal (13.24.22)

The following variables perform the function specified in 17.17 of IEEE Std 802.1D, 1998 Edition. A single per-Port instance is used by all state machines.

i) initPm
j) rcvdBpdu
k) rcvdRSTP
l) rcvdSTP
m) rcvdTcAck
n) rcvdTcn
o) sendRSTP
p) tcAck

The following variables are as specified in 17.17 of IEEE Std 802.1D, 1998 Edition. There is one instance per-Port of each variable for the CIST, and one per-Port for each MSTI.

q) forward
r) forwarding
s) infoIs
t) learn
u) learning
v) proposed
w) proposing
x) rcvdTc
y) reRoot
z) reselect
aa) selected
ab) tcProp
ac) updtInfo

The following variables perform the functions described in 17.17 of IEEE Std 802.1D, 1998 Edition, but have enhanced or extended specifications or considerations. There is one instance per-Port of each variable for the CIST, and one per-Port for each MSTI.

ad) agreed (13.24.2)
ae) portId (13.24.21)
af) rcvdInfo (13.24.23) replaces the functionality of the variable previously named rcvdMsg;
ag) role (13.24.25)
ah) selectedRole (13.24.26)
ai) sync (13.24.27)
aj) synced (13.24.28)

The following variables perform the related functions described in 17.17 of IEEE Std 802.1D, 1998 Edition, but have extended specifications. There is one instance per-Port of each variable for the CIST.

ak) cistDesignatedPriority(13.24.4)
al) cistDesignatedTimes (13.24.5)

am) cistMsgPriority (13.24.6)
an) cistMsgTimes (13.24.7)
ao) cistPortPriority (13.24.8)
ap) cistPortTimes (13.24.9)

The following variables perform the related functions described in 17.17 of IEEE Std 802.1D, 1998 Edition, but have enhanced or extended specifications. There is one instance per-Port of each variable for each MSTI.

aq) mstiDesignatedPriority(13.24.11)
ar) mstiDesignatedTimes (13.24.12)
as) mstiMsgPriority (13.24.15)
at) mstiMsgTimes (13.24.16)
au) mstiPortPriority (13.24.17)
av) mstiPortTimes (13.24.18)

The following variable(s) are additional to those specified in 17.17 of IEEE Std 802.1D, 1998 Edition. There is one instance per-Port of each variable for the CIST, and one per-Port for each MSTI.

aw) agree (13.24.1)
ax) changedMaster (13.24.3)
ay) rcvdMsg (13.24.24) provides functionality distinct from the previous rcvdMsg variable, now renamed rcvdInfo (13.24.23)

The following variable(s) specified in 17.17 of IEEE Std 802.1D, 1998 Edition are not required or have been replaced by other variables in this specification.

az) newInfo (replaced by newInfoCist and newInfoMsti)
ba) tc (not required)

13.24.1 agree

This variable is used by the Port Transmit state machine to set the value of the Agreement flag in transmitted Configuration Messages for the given tree.

13.24.2 agreed

A Boolean value indicating that a Configuration Message has been received from another Bridge attached to the same LAN indicating Agreement that all the Port States for the given tree of all other Bridges attached to the same LAN as this Port are known to be likewise compatible with a loop free active topology determined by this Bridge's priority vectors and, in the absence of further communication with this Bridge, will remain compatible within the design probabilities of protocol failure due to repeated BPDU loss (13.16,13.19).

13.24.3 changedMaster

Set, for all Ports for all MSTIs, by the CIST Port Role Selection state machine using the updtRoleCist() procedure if the root priority vector selected has a different Regional Root Identifier than that previously selected, and has or had a non-zero CIST External Path Cost. changedMaster is always FALSE for the CIST.

NOTE—Changes in Regional Root Identifier will not cause loops if the Regional Root is within an MST Region, as is the case if and only if the MST Region is the Root of the CST. This important optimization allows the MSTIs to be fully independent of each other in the case where they compose the core of a network.

13.24.4 cistDesignatedPriority

The CIST Root Identifier, External Root Path Cost, Regional Root Identifier, Internal Root Path Cost, Designated Bridge Identifier, and Designated Port Identifier components of the Port's CIST designated priority vector, as defined in 13.10.

13.24.5 cistDesignatedTimes

The set of timer parameter values (Message Age, Max Age, Forward Delay, and remainingHops) that are used to update Port Times when updtInfo is set. The value of designatedTimes is copied from the cistRootTimes Parameter (13.23.7) by the operation of the updtRolesCist() procedure.

13.24.6 cistMsgPriority

The CIST Root Identifier, External Root Path Cost, Regional Root Identifier, Internal Root Path Cost, Designated Bridge Identifier, and Designated Port Identifier components of the CIST message priority vector conveyed in a received BPDU, as defined in 13.10.

13.24.7 cistMsgTimes

The timer parameter values (Message Age, Max Age, Forward Delay, Hello Time, and remainingHops) conveyed in a received BPDU. If the BPDU is an STP or RSTP without MSTP parameters, remainingHops is set to zero.

13.24.8 cistPortPriority

The CIST Root Identifier, External Root Path Cost, Regional Root Identifier, Internal Root Path Cost, Designated Bridge Identifier, and Designated Port Identifier components of the Port's port priority vector, as defined in 13.10.

13.24.9 cistPortTimes

The Port's timer parameter values (Message Age, Max Age, Forward Delay, Hello Time, and remainingHops). These timer values are used in BPDUs transmitted from the Port.

13.24.10 infoInternal

If infoIs is Received, indicating that the port has received current information from the Designated Bridge for the attached LAN, infoInternal is set if that Designated Bridge is in the same MST Region as the receiving Bridge, and clear otherwise.

13.24.11 mstiDesignatedPriority

The Regional Root Identifier, Internal Root Path Cost, Designated Bridge Identifier, and Designated Port Identifier components of the Port's designated priority vector, as defined in 13.11.

13.24.12 mstiDesignatedTimes

The value of remainingHops used to update mstiPortTimes when updtInfo is set. The value of mstiDesignatedTimes is copied from the mstiRootTimes Parameter (13.23.13) by the operation of the updtMstiRolesBridge() procedure.

13.24.13 mstiMaster

A Boolean variable used to determine the value of the Master flag for this MSTI and Port in transmitted MST BPDUs.

Set TRUE if the Port Role for the MSTI and Port is Root Port or Designated Port, and the Bridge has selected one of its Ports as the Master Port for this MSTI or the mstiMastered flag is set for this MSTI for any other Bridge Port with a Root Port or Designated Port Role. Set FALSE otherwise.

13.24.14 mstiMastered

A Boolean variable used to record the value of the Master flag for this MSTI and Port in MST BPDUs received from the attached LAN.

NOTE—mstiMaster and mstiMastered signal the connection of the MSTI to the CST via the Master Port throughout the MSTI. These variables and their supporting procedures do not affect the connectivity provided by this revision of this standard, but permit future enhancements to MSTP providing increased flexibility in the choice of Master Port without abandoning plug and play network migration. They are therefore omitted from the overviews of protocol operation, including Figure 13-19.

13.24.15 mstiMsgPriority

The Regional Root Identifier, Internal Root Path Cost, Designated Bridge Identifier, and Designated Port Identifier components of the MSTI message priority vector, as defined in 13.11 and conveyed in a received BPDU for this MSTI.

13.24.16 mstiMsgTimes

The value of remainingHops received with message priority components of the mstiMsgPriority for this MSTI.

13.24.17 mstiPortPriority

The Regional Root Identifier, Internal Root Path Cost, Designated Bridge Identifier, and Designated Port Identifier components of the Port's MSTI port priority vector, as defined in 13.11.

13.24.18 mstiPortTimes

The value of remainingHops for this MSTI in BPDUs transmitted through the Port.

13.24.19 newInfoCist

A Boolean variable set TRUE if a BPDU conveying changed CIST information is to be transmitted. It is set FALSE by the Port Transmit state machine.

13.24.20 newInfoMsti

A Boolean variable set TRUE if a BPDU conveying changed MSTI information is to be transmitted. It is value is set FALSE by the Port Transmit state machine.

13.24.21 portId

The Port Identifier for this Port. This variable forms a component of the port priority and designated priority vectors (13.10,13.11).

The four most significant bits of the Port Identifier (the settable Priority component) for the CIST and for each MSTI can be modified independently of the setting of those bits for all other trees, as a part of allowing full and independent configuration control to be exerted over each Spanning Tree instance.

13.24.22 rcvdInternal

A Boolean variable set TRUE by the Receive Machine if the BPDU received was transmitted by a Bridge in the same MST Region as the receiving Bridge.

13.24.23 rcvdInfo

Set to the result of the rcvInfoCist procedure for the CIST and the rcvInfoMsti procedure for an MSTI. It can take the values SuperiorDesignatedInfo, RepeatedDesignatedInfo, RootInfo, or OtherInfo.

13.24.24 rcvdMsg

A Boolean variable set TRUE by the Receive Machine if the BPDU received contains a message for this tree.

13.24.25 role

The assigned Port Role. The port is either a DisabledPort, a RootPort, a DesignatedPort, an AlternatePort, a BackupPort, or a MasterPort.

NOTE—The role of MasterPort is introduced for MSTIs for a Port where the CIST Port Role is a RootPort and the spanning tree information received is from another MST Region. An MSTI Master Port forms part of the stable active topology for frames allocated to that MSTI, just as the CIST Root Port forwards frames allocated to the CIST. The Port State for each MSTI may differ for each MSTI as required to suppress temporary loops.

13.24.26 selectedRole

A newly computed role for the Port.

13.24.27 sync

A Boolean value. Set TRUE to force the Port State to be compatible with the loop free active topology determined by the priority vectors held by this Bridge (13.16,13.19) for this tree (CIST, or MSTI), by transitioning the Port State to Discarding and soliciting an Agreement if possible, if the Port is not already synchronized (13.24.28).

13.24.28 synced

A Boolean value. TRUE only if the Port State is compatible with the loop free active topology determined by the priority vectors held by this Bridge for this tree (13.16,13.19).

13.25 State Machine Conditions

The following boolean variable evaluations are defined for notational convenience in the state machines. These definitions also serve to highlight those cases where a state transition for one tree (CIST or MSTI) depends on the state of the variables of one or more other trees.

13.25.1 allSynced

TRUE if and only if synced is TRUE for all Ports for the given Tree (CIST or MSTI).

13.25.2 cist

TRUE only for CIST state machines, i.e. FALSE for MSTI state machine instances.

13.25.3 cistRootPort

TRUE if the CIST role for the given Port is RootPort.

13.25.4 cistDesignatedPort

TRUE if the CIST role for the given Port is DesignatedPort.

13.25.5 mstiRootPort

TRUE if the role for any MSTI for the given Port is RootPort.

13.25.6 mstiDesignatedPort

TRUE if the role for any MSTI for the given Port is DesignatedPort.

13.25.7 rcvdAnyMsg

rcvdAnyMsg is TRUE for a given Port if rcvdMsg is TRUE for the CIST or any MSTI for that Port.

13.25.8 rcvdCistInfo

rcvdCistInfo is TRUE for a given Port if and only if rcvdMsg is TRUE for the CIST for that Port.

13.25.9 rcvdMstiInfo

rcvdMstiInfo is TRUE for a given Port and MSTI if and only if rcvdMsg is FALSE for the CIST for that Port and rcvdMsg is TRUE for the MSTI for that Port.

13.25.10 reRooted

TRUE if the rrWhile timer is clear (zero) for all Ports for the given Tree other than the given Port.

13.25.11 updtCistInfo

updtCistInfo is TRUE for a given Port if and only if updtInfo is TRUE for the CIST for that Port.

13.25.12 updtMstiInfo

updtMstiInfo is TRUE for a given Port and MSTI if and only if updtInfo is TRUE for the MSTI for that Port or either updtInfo or selected are TRUE for the CIST for that Port.

NOTE—The dependency of rcvdMstiInfo and updtMstiInfo on CIST variables for the Port reflects the fact that MSTIs exist in a context of CST parameters. The state machines ensure that the CIST parameters from received BPDUs are processed and updated prior to processing MSTI information.

13.26 State Machine Procedures

The following procedures perform the functions specified in 17.17 of IEEE Std 802.1D, 1998 Edition for the CIST state machines.

a) txTcn()

The following procedures perform the functions specified in 17.17 of IEEE Std 802.1D, 1998 Edition for the CIST or any given MSTI instance.

b) disableForwarding()
c) disableLearning()
d) enableForwarding()
e) enableLearning()
f) flush()
g) updtBPDUVersion()

The following procedures perform the general functions described in 17.17 of IEEE Std 802.1D, 1998 Edition for both the CIST and the MSTI state machines or specifically for the CIST or a given MSTI, but have enhanced or extended specifications or considerations.

h) clearReselectTree() (13.26.4)
i) newTcWhile() (13.26.6)
j) rcvInfoCist() (13.26.7) and rcvInfoMsti() (13.26.8)
k) recordProposalCist() (13.26.13) and recordProposalMsti() (13.26.14)
l) setReRootTree(13.26.16)
m) setSelectedTree() (13.26.17)
n) setSyncTree() (13.26.18)
o) setTcFlags() (13.26.19)
p) setTcPropTree() (13.26.20)
q) txConfig() (13.26.21)
r) txMstp() (13.26.22)

NOTE 1—the equivalent procedure to txMstp() in IEEE Std 802.1D, 1998 Edition is called txRstp().

s) updtRcvdInfoWhileCist() (13.26.23) and updtRcvdInfoWhileMsti() (13.26.24)
t) updtRolesCist() (13.26.25) and updtRolesMsti() (13.26.26)
u) updtRolesDisabledTree() (13.26.27)

NOTE 2—clearReselectBridge() has been replaced by clearReselectTree(); rcvBpdu() by rcvInfoCist() and rcvInfoMsti(); recordProposed() by recordProposalCist() and recordProposalMsti(); setSelectedBridge() by setSelectedTree(); setReRootBridge() by setReRootTree(); setSyncBridge() by setSyncTree(); setTcPropBridge() by setTcPropTree(); updtRcvdInfoWhile() by updtRcvdInfoWhileCist() and updtRcvdInfoWhileMsti(); updtRolesBridge() by updtRolesCist() and updtRolesMsti(); and updtRoleDisabledBridge() by updtRolesDisabledTree().

The following procedures perform functions additional to those described in 17.17 of IEEE Std 802.1D, 1998 Edition for both the CIST and the MSTI state machines or for the CIST or a given MSTI specifically:

v) betterorsameInfoCist() (13.26.1) and betterorsameInfoMsti() (13.26.2)
w) clearAllRcvdMsgs() (13.26.3)
x) fromSameRegion() (13.26.5)
y) recordAgreementCist() (13.26.9) and recordAgreementMsti() (13.26.10)
z) recordMasteredCist() (13.26.11)
aa) recordMasteredMSTI() (13.26.12)
ab) setRcvdMsgs() (13.26.15)

All references to named variables in the specification of procedures are to instances of the variables corresponding to the instance of the state machine using the function, i.e. to the CIST or the given MSTI as appropriate. References to the forwarding and learning functions for a Port apply to all and only those Filtering Databases associated with that specific tree.

13.26.1 betterorsameInfoCist()

Returns TRUE if the received CIST priority vector is better than or the same as (13.10) the CIST port priority vector.

13.26.2 betterorsameInfoMsti()

Returns TRUE if the MSTI priority vector is better than or the same as (13.11) the MSTI port priority vector.

NOTE—This procedure is not invoked if the received BPDU carrying the MSTI information was received from another MST Region. In that event, the Receive Machine (using setRcvdMsgs()) does not set rcvdMsg for any MSTI, and the Port Information Machine's SUPERIOR_DESIGNATED state is not entered.

13.26.3 clearAllRcvdMsgs()

Clears rcvdMsg for the CIST and all MSTIs, for all Ports.

13.26.4 clearReselectTree()

Clears reselect for the tree (the CIST or a given MSTI) for all Ports of the Bridge.

13.26.5 fromSameRegion()

Returns TRUE if rcvdRSTP is TRUE, and the received BPDU conveys an MST Configuration Identifier that matches that held for the Bridge. Returns FALSE otherwise.

13.26.6 newTcWhile()

This procedure sets the value of tcWhile, if and only if it is currently zero, to twice HelloTime on point-to-point links (i.e., links where the operPointToPointMAC parameter is TRUE; see 6.4.3) where the partner bridge port is RSTP capable, and to the sum of the Max Age and Forward Delay components of rootTimes otherwise (non-RSTP capable partners or shared media). The value of HelloTime is taken from cistPortTimes (13.24.9) for this Port.

13.26.7 rcvInfoCist()

Returns SuperiorDesignatedInfo if the received CIST message conveys a message priority (cistMsgPriority—13.24.6) that is superior (13.10) to the Port's port priority vector, or any of the received timer parameter values (cistMsgTimes—13.24.7) differ from those already held for the Port (cistPortTimes—13.24.9).

Returns RepeatedDesignatedInfo if the received CIST message conveys a message priority vector and timer parameters that are the same as the Port's port priority vector or timer values.

Returns RootInfo if the received CIST message conveys a Root Port Role and a CIST message priority that is the same or worse than the CIST port priority vector.

Otherwise, returns OtherInfo.

13.26.8 rcvInfoMsti()

Returns SuperiorDesignatedInfo if the received MSTI message conveys a message priority (mstiMsgPriority—13.24.15) that is superior (13.11) to the Port's port priority vector, or any of the received timer parameter values (mstiMsgTimes—13.24.16) differ from those already held for the Port (mstiPortTimes—13.24.18).

Returns RepeatedDesignatedInfo if the received MSTI message conveys a message priority vector and timer parameters that are the same as the Port's port priority vector or timer values.

Returns RootInfo if the received MSTI message conveys a Root Port Role and a MSTI message priority that is the same or worse than the MSTI port priority vector.

Otherwise, returns OtherInfo.

13.26.9 recordAgreementCist()

If the CIST Message was a Configuration Message received on a point to point link, and the CIST message has the Agreement flag set, and conveys either:

1) a Root Port Role with message priority the same as or worse than the port priority vector, or
2) a Designated Port Role with message priority the same as or better than the port priority vector,

the CIST agreed flag is set. Otherwise the CIST agreed flag is cleared.

Additionally, if the CIST message was received from a Bridge in a different MST Region i.e. the rcvdInternal flag is clear, the agreed flags for this Port for all MSTIs are set or cleared to the same value as the CIST agreed flag. If the CIST message was received from a Bridge in the same MST Region, the MSTI agreed flags are not changed.

13.26.10 recordAgreementMsti()

If the MSTI Message was received on a point to point link and:

a) the message priority vector of the CIST Message accompanying this MSTI Message (i.e. received in the same BPDU) has the same CIST Root Identifier, CIST External Root Path Cost, and Regional Root Identifier as the CIST port priority vector, and
b) the MSTI Message has the Agreement flag set, and conveys either
    1) a Root Port Role with message priority the same as or worse than the MSTI port priority vector, or
    2) a Designated Port Role with message priority the same as or better than the port priority vector,

the MSTI agreed flag is set. Otherwise the MSTI agreed flag is cleared.

NOTE—MSTI Messages received from Bridges external to the MST Region are discarded and not processed by recordAgreeementMsti() or recordProposalMsti().

13.26.11 recordMasteredCist()

If the CIST message was received from a Bridge in a different MST Region, i.e. the rcvdInternal flag is clear, the mstiMastered variable for this Port is cleared for all MSTIs.

13.26.12 recordMasteredMsti()

If the MSTI Message was received on a point to point link and the MSTI Message has the Master flag set, set the mstiMastered variable for the MSTI. Otherwise reset the mstiMastered variable.

13.26.13 recordProposalCist()

If the CIST Message was a Configuration Message received on a point to point link and has the Proposal flag set, the CIST proposed flag is set. Otherwise the CIST proposed flag is cleared.

Additionally, if the CIST message was received from a Bridge in a different MST Region i.e. the rcvdInternal flag is clear, the proposed flags for this Port for all MSTIs are set or cleared to the same value as the CIST proposed flag. If the CIST message was received from a Bridge in the same MST Region, the MSTI proposed flags are not changed.

13.26.14 recordProposalMsti()

If the MSTI Message was received on a point to point link and has the Proposal flag set, the MSTI proposed flag is set. Otherwise the MSTI proposed flag is cleared.

13.26.15 setRcvdMsgs()

Sets rcvdMsg for the CIST, and makes the received CST or CIST message available to the CIST Port Information state machines.

Additionally and if and only if rcvdInternal is set, sets rcvdMsg for each and every MSTI for which an MSTI message is conveyed in the BPDU, and makes available each MSTI message and the common parts of the CIST message priority (the CIST Root Identifier, External Root Path Cost, and Regional Root Identifier) to the Port Information state machine for that MSTI.

13.26.16 setReRootTree()

This procedure sets reRoot TRUE for this tree (the CIST or a given MSTI) for all Ports of the Bridge.

13.26.17 setSelectedTree()

Sets selected TRUE for this tree (the CIST or a given MSTI) for all Ports of the Bridge.

13.26.18 setSyncTree()

Sets sync TRUE for this tree (the CIST or a given MSTI) for all Ports of the Bridge.

13.26.19 setTcFlags()

If the received BPDU is a TCN BPDU, sets rcvdTcn TRUE and sets rcvdTc TRUE for each and every MSTI.

Otherwise, if the received BPDU is a ConfigBPDU or RST BPDU:

a) If the Topology Change Acknowledgment flag is set in the CST message, sets rcvdTcAck TRUE.
b) If rcvdInternal is clear and the Topology Change flag is set in the CST message, sets rcvdTc for the CIST and for each and every MSTI.
c) If rcvdInternal is set, sets rcvdTc for the CIST if the Topology Change flag is set in the CST message, and sets rcvdTc for each MSTI for which the Topology Change flag is set in the corresponding MSTI message.

13.26.20 setTcPropTree()

Sets tcProp TRUE for the given tree (the CIST or an MSTI) for all Ports except the Port that invoked the procedure.

13.26.21 txConfig()

Transmits a Configuration BPDU. The first four components of the message priority vector (13.24.6) conveyed in the BPDU are set to the value of the CIST Root Identifier, External Root Path Cost, Bridge Identifier, and Port Identifier components of the cistPortPriority (13.24.8) for this Port. The topology change flag is set if (tcWhile != 0) for the Port. The topology change acknowledgement flag is set to the value of TcAck for the Port. The remaining flags are set to zero. The value of the Message Age, Max Age, Fwd Delay, and Hello Time parameters conveyed in the BPDU are set to the values held in cistPortTimes (13.24.9) for the Port.

13.26.22 txMstp()

Transmits an MST BPDU (14.3.3), encoded according to the specification contained in 14.6. The first six components of the CIST message priority vector (13.24.6) conveyed in the BPDU are set to the value of cistPortPriority (13.24.8) for this Port. The Port Role in the BPDU (14.2.1) is set to the current value of the role variable for the transmitting port (13.24.25). The Agreement and Proposal flags in the BPDU are set to the values of the agreed (13.24.2) and proposing (13.24 of this standard, 17.18.20 of IEEE Std 802.1D, 1998 Edition) variables for the transmitting Port respectively. The CIST topology change flag is set if (tcWhile != 0) for the Port. The topology change acknowledge flag in the BPDU is never used and is set to zero. The learning flag is set if the learning variable for the CIST (13.24 of this standard, 17.18.9 of IEEE Std 802.1D, 1998 Edition) is TRUE. The forwarding flag is set if the forwarding variable for the CIST (13.24 of this standard, 17.18.5 of IEEE Std 802.1D, 1998 Edition) is TRUE. The value of the Message Age, Max Age, Fwd Delay, and Hello Time parameters conveyed in the BPDU are set to the values held in cstiPortTimes (13.24.9) for the Port.

If the value of the Force Protocol Version parameter is less than 3, no further parameters are encoded in the BPDU and the protocol version parameter is set to 2 (denoting an RST BPDU). Otherwise, the protocol version parameter is set to 3 and the remaining parameters of the MST BPDU are encoded:

a) The version 3 length.
b) The MST Configuration Identifier parameter of the BPDU is set to the value of the MstConfigId variable for the Bridge (13.23.8).
c) The CIST Internal Root Path Cost (13.24.8).
d) The CIST Bridge Identifier.
e) The CIST Remaining Hops.
f) The parameters of each MSTI message, encoded in MSTID order.

NOTE—No more than 64 MSTIs may be supported. The parameter sets for all of these can be encoded in a standard sized Ethernet frame.

13.26.23 updtRcvdInfoWhileCist()

Calculates the effective age, or remainingHops, to limit the propagation and longevity of received Spanning Tree information for the CIST, setting rcvdInfoWhile to the number of seconds that the information received on a Port will be held before it is either refreshed by receipt of a further configuration message or aged out.

If the information was received from a Bridge external to the MST Region (rcvdInternal FALSE), the effective age of the port information (cistPortPriority and cistPortTimes) is taken as the value of the Message Age parameter carried in a received BPDU (cistMsgTimes), incremented by the greater of (1/16th Max Age) and 1 second, and rounded to the nearest whole second. The value of Message Age and Max Age used in this calculation are taken from the cistPortTimes variable (17.18.18) and, the value assigned to rcvdInfoWhile is the lower of:

a) Max Age minus this effective age, and
b) 3 times the Hello Time,

or

c) zero, if the effective age exceeds Max Age.

and the value of the remainingHops component of portTimes is set to MaxHops.

If the information was received from a Bridge in the same MST Region (rcvdInternal TRUE), the Message Age component of cistPortTimes is set equal to the value received (cistMsgTimes) and the remainingHops component is set equal to the value received decremented by one. The value assigned to rcvdInfoWhile is:

d) 3 times the Hello Time,

or

e) zero, if either cistPortTimes Message Age effective exceeds Max Age or remainingHops is less than or equal to zero.

The value of Hello Time used by this procedure is taken from cistPortTimes.

13.26.24 updtRcvdInfoWhileMsti()

Calculates the remainingHops, to limit the propagation and longevity of received Spanning Tree information for an MSTI, setting rcvdInfoWhile to the number of seconds that the information received on a Port will be held before it is either refreshed by receipt of a further configuration message or aged out.

mstiPortTimes remainingHops is set equal to the value received (mstiMsgTimes) decremented by one. The value assigned to rcvdInfoWhile is:

a) 3 times the Hello Time,

or

b) zero, if mstiPortTimes remainingHops is less than or equal to zero.

The value of Hello Time used by this procedure is taken from cistPortTimes.

13.26.25 updtRolesCist()

This procedure calculates the following CIST Spanning Tree priority vectors (13.9, 13.10) and timer values:

a) The *root path priority vector* for each Bridge Port that is not Disabled and has a *port priority vector* (cistPortPriority plus portId—see 13.24.8 and 13.24.21) that has been recorded from a received message and not aged out (infoIs == Received); and
b) The Bridge's *root priority vector* (cistRootPortId, cistRootPriority—13.23.5, 13.23.6), chosen as the best of the CIST Spanning Tree priority vectors comprising the Bridge's own *bridge priority vector* (CistBridgePriority—13.23.3) plus all the calculated root path priority vectors whose DesignatedBridgeID component is not equal to the DesignatedBridgeID component of the Bridge's own bridge priority vector (13.10); and
c) The Bridge's *root times*, (cistRootTimes—13.23.7), determined as follows:
   1) If the chosen root priority vector is the bridge priority vector, *root times* is equal to CistBridgeTimes (13.23.4).
   2) If the chosen root priority vector is not the bridge priority vector, *root times* is equal to the value of cistPortTimes (13.24.9) for the port associated with the chosen root priority vector.
d) The *designated priority vector* (cistDesignatedPriority—13.24.4) for each port; and

e) The *designated times* for each Port (cistDesignatedTimes—13.24.5) set equal to the value of *root times*.

The CIST port role for each Port is assigned, and its port priority vector and Spanning Tree timer information are updated as follows:

f) If the Port is Disabled (infoIs = Disabled), selectedRole is set to DisabledPort. Otherwise:
g) If the port priority vector information was aged (infoIs = Aged), updtInfo is set and selectedRole is set to DesignatedPort;
h) If the port priority vector was derived from another port on the Bridge or from the Bridge itself as the Root Bridge (infoIs = Mine), selectedRole is set to DesignatedPort. Additionally, updtInfo is set if the port priority vector differs from the designated priority vector or the Port's associated timer parameters differ from those for the Root Port;
i) If the port priority vector was received in a Configuration Message and is not aged (infoIs == Received), and the root priority vector is now derived from it, selectedRole is set to RootPort and updtInfo is reset;
j) If the port priority vector was received in a Configuration Message and is not aged (infoIs == Received), the root priority vector is not now derived from it, the designated priority vector is not better than the port priority vector, and the designated bridge and designated port components of the port priority vector do not reflect another port on this bridge, selectedRole is set to AlternatePort and updtInfo is reset;
k) If the port priority vector was received in a Configuration Message and is not aged (infoIs == Received), the root priority vector is not now derived from it, the designated priority vector is not better than the port priority vector, and the designated bridge and designated port components of the port priority vector reflect another port on this bridge, selectedRole is set to BackupPort and updtInfo is reset.
l) If the port priority vector was received in a Configuration Message and is not aged (infoIs == Received), the root priority vector is not now derived from it, the designated priority vector is better than the port priority vector, selectedRole is set to DesignatedPort and updtInfo is set.

NOTE—The port role assignment is almost identical to that in 17.19.8 of IEEE Std 802.1D, 1998 Edition, with the addition of the final bullet, erroneously omitted from that specification.

13.26.26 updtRolesMsti()

This procedure calculates the following MST Spanning Tree priority vectors (13.9, 13.11) and timer values for an MSTI:

a) The *root path priority vector* for each Bridge Port that is not Disabled and has a *port priority vector* (mstiPortPriority plus portId—see 13.24.17 and 13.24.21) that has been recorded from a received message and not aged out (infoIs == Received); and
b) The Bridge's *root priority vector* (mstiRootPortId, mstiRootPriority—13.23.11, 13.23.12), chosen as the best of the MSTI Spanning Tree priority vectors for this MSTI comprising the Bridge's own *bridge priority vector* (MstiBridgePriority—13.23.9) plus all the calculated root path priority vectors whose DesignatedBridgeID component is not equal to the DesignatedBridgeID component of the Bridge's own bridge priority vector (13.11); and
c) The Bridge's *root times*, (mstiRootTimes—13.23.13), determined as follows:
    1) If the chosen root priority vector is the bridge priority vector, *root times* is equal to MstiBridgeTimes (13.23.10).
    2) If the chosen root priority vector is not the bridge priority vector, *root times* is equal to the value of mstiPortTimes (13.24.18) for the port associated with the chosen root priority vector.
d) The *designated priority vector* (mstiDesignatedPriority—13.24.11) for each port; and
e) The *designated times* for each Port (mstiDesignatedTimes—13.24.12) set equal to the value of *root times*.

The MSTI port role for each Port is assigned, and its port priority vector and Spanning Tree timer information are updated as follows:

f) If the Port is Disabled (infoIs = Disabled), selectedRole is set to DisabledPort;

g) If the Port is not Disabled, the selected CIST Port Role (calculated prior to invoking this procedure) is RootPort, and the CIST port priority information was received from a Bridge external to the MST Region (infoIs == Received and infoInternal == FALSE), selectedRole is set to MasterPort. Additionally, updtInfo is set if the MSTI port priority vector differs from the designated priority vector or the Port's associated timer parameters differ from those for the Root Port;

h) If the Port is not Disabled, the selected CIST Port Role (calculated prior to invoking this procedure) is AlternatePort, and the CIST port priority information was received from a Bridge external to the MST Region (infoIs == Received and infoInternal == FALSE), selectedRole is set to AlternatePort. Additionally, updtInfo is set if the MSTI port priority vector differs from the designated priority vector or the Port's associated timer parameters differ from those for the Root Port.

Otherwise, if the Port is not Disabled and the CIST port priority information was not received from a Bridge external to the Region (infoIs != Received or infoInternal == TRUE):

i) If the port priority vector information was aged (infoIs = Aged), updtInfo is set and selectedRole is set to DesignatedPort;

j) If the port priority vector was derived from another port on the Bridge or from the Bridge itself as the Root Bridge (infoIs = Mine), selectedRole is set to DesignatedPort. Additionally, updtInfo is set if the port priority vector differs from the designated priority vector or the Port's associated timer parameters differ from those for the Root Port;

k) If the port priority vector was received in a Configuration Message and is not aged (infoIs == Received), and the root priority vector is now derived from it, selectedRole is set to RootPort and updtInfo is reset;

l) If the port priority vector was received in a Configuration Message and is not aged (infoIs == Received), the root priority vector is not now derived from it, the designated priority vector is not better than the port priority vector, and the designated bridge and designated port components of the port priority vector do not reflect another port on this bridge, selectedRole is set to AlternatePort and updtInfo is reset;

m) If the port priority vector was received in a Configuration Message and is not aged (infoIs == Received), the root priority vector is not now derived from it, the designated priority vector is not better than the port priority vector, and the designated bridge and designated port components of the port priority vector reflect another port on this bridge, selectedRole is set to BackupPort and updtInfo is reset.

n) If the port priority vector was received in a Configuration Message and is not aged (infoIs == Received), the root priority vector is not now derived from it, the designated priority vector is better than the port priority vector, selectedRole is set to DesignatedPort and updtInfo is set.

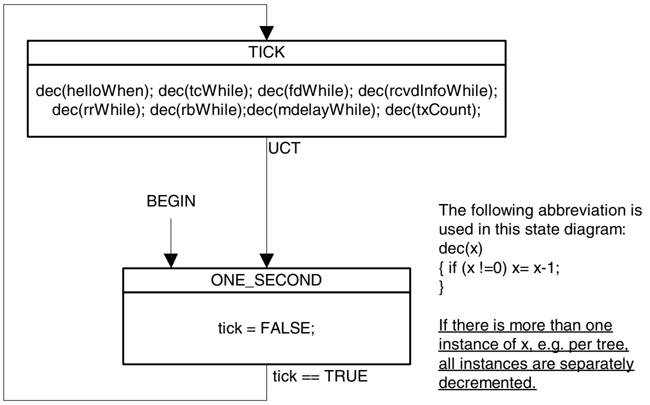
13.26.27 updtRolesDisabledTree()

This procedure sets selectedRole to DisabledPort for all Ports of the Bridge for a given tree (CIST or MSTI).

13.27 The Port Timers state machine

The Port Timers state machine for a given Port is responsible for decrementing the timer variables for the CIST and all MSTIs for that Port each second.

The Port Timers state machine shall implement the function specified by the state diagram contained in Figure 13-11.


TICK
dec(helloWhen); dec(tcWhile); dec(fdWhile); dec(rcvdInfoWhile); dec(rrWhile); dec(rbWhile);dec(mdelayWhile); dec(txCount);
UCT
BEGIN
ONE_SECOND
tick = FALSE;
tick == TRUE

The following abbreviation is used in this state diagram:
dec(x)
{ if (x !=0) x= x-1;
}

If there is more than one instance of x, e.g. per tree, all instances are separately decremented.

Figure 13-11—Port Timers state machine

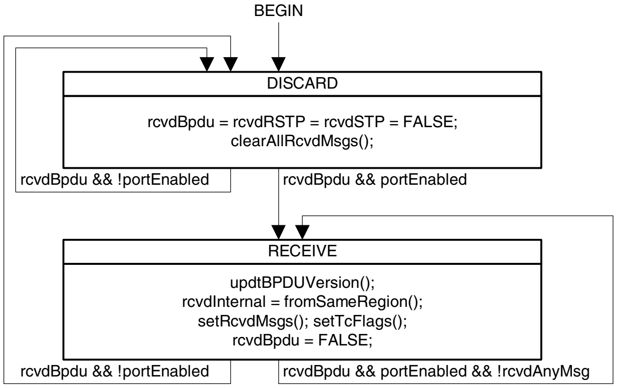


Figure 13-12—Port Receive state machine

13.28 Port Receive state machine

The Port Receive state machine shall implement the function specified by the state diagram contained in Figure 13-12 and the attendant definitions contained in 13.21 through 13.26.

This state machine is responsible for receiving BPDUs. Using the updtBPDUversion() procedure it sets either the rcvdRSTP or the rcvdSTP flag to reflect the type of the BPDU for use by the Port Protocol Migration machine. It sets the rcvdInternal flag if the transmitting Bridge and BPDU are internal to the MST Region, i.e. belong to the same MST Region as this Bridge (13.8). It sets rcvdMsg for the CIST, and additionally for each MSTI if the rcvdBPDU is internal. The next BPDU is not processed until all the rcvdMsg flags have been cleared by the per tree state machines.

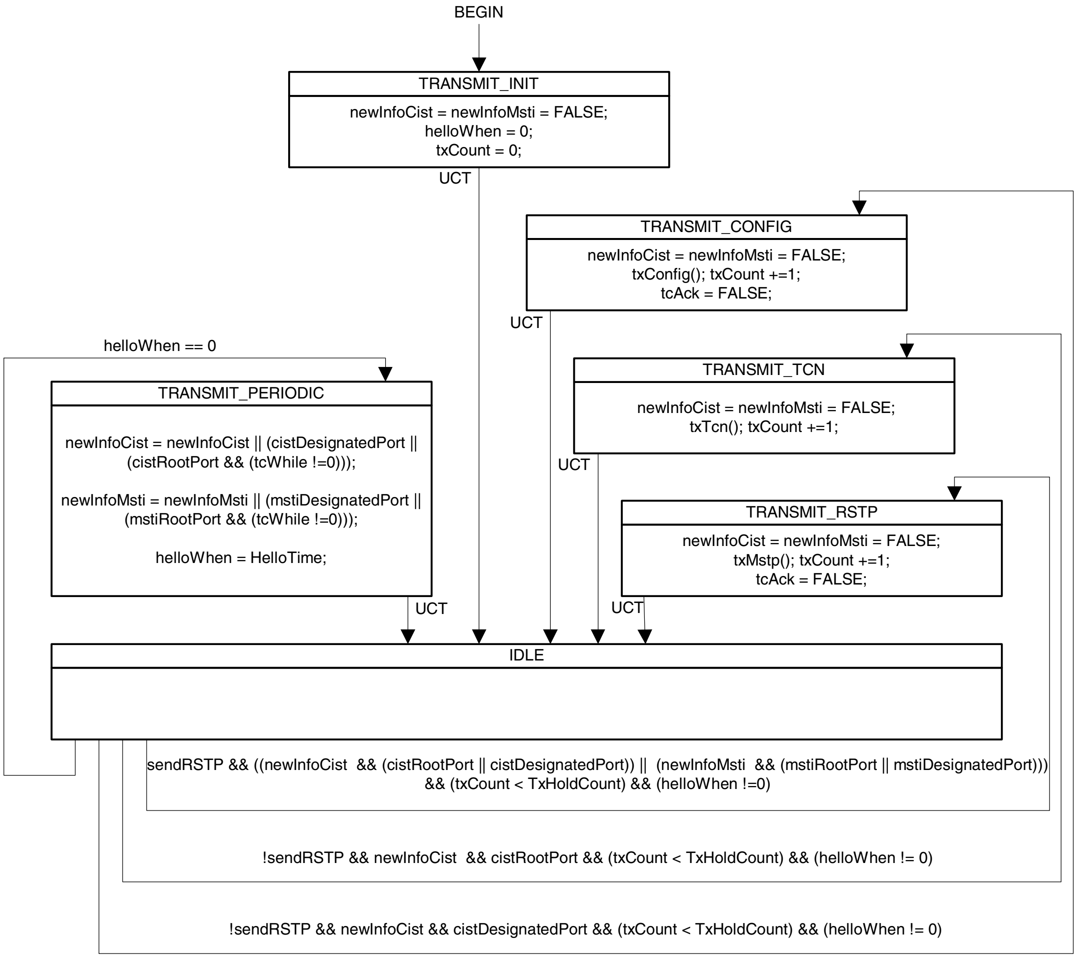
13.29 Port Protocol Migration state machine

The specification of this state machine is identical to that of the Port Protocol Migration state machine for RSTP (Clause 17 of IEEE Std 802.1D, 1998 Edition).

13.30 Port Transmit state machine

The Port Transmit state machine shall implement the function specified by the state diagram contained in Figure 13-13 and the attendant definitions contained in 13.21 through 13.26.

This state machine is responsible for transmitting BPDUs.


BEGIN
TRANSMIT_INIT
newInfoCist = newInfoMsti = FALSE;
helloWhen = 0;
txCount = 0;
UCT
TRANSMIT_CONFIG
newInfoCist = newInfoMsti = FALSE;
txConfig(); txCount +=1;
tcAck = FALSE;
UCT
helloWhen == 0
TRANSMIT_TCN
newInfoCist = newInfoMsti = FALSE;
txTcn(); txCount +=1;
TRANSMIT_PERIODIC
newInfoCist = newInfoCist || (cistDesignatedPort ||
(cistRootPort && (tcWhile !=0)));
newInfoMsti = newInfoMsti || (mstiDesignatedPort ||
(mstiRootPort && (tcWhile !=0)));
helloWhen = HelloTime;
UCT
TRANSMIT_RSTP
newInfoCist = newInfoMsti = FALSE;
txMstp(); txCount +=1;
tcAck = FALSE;
UCT
UCT
IDLE
sendRSTP && ((newInfoCist && (cistRootPort || cistDesignatedPort)) || (newInfoMsti && (mstiRootPort || mstiDesignatedPort)))
&& (txCount < TxHoldCount) && (helloWhen !=0)
!sendRSTP && newInfoCist && cistRootPort && (txCount < TxHoldCount) && (helloWhen != 0)
!sendRSTP && newInfoCist && cistDesignatedPort && (txCount < TxHoldCount) && (helloWhen != 0)

All transtions, except UCT, are qualified by "&& selected &&!updtInfo".

Figure 13-13—Port Transmit state machine

NOTE 1—Any single received BPDU that changes the CIST Root Identifier, CIST External Root Path Cost, or CIST Regional Root associated with MSTIs should be processed in their entirety, or not at all, before encoding BPDUs for transmission.This recommendation is made to minimize the number of BPDUs to be transmitted following receipt of a BPDU carrying new information. It is not required for correctness and has not therefore been incorporated into the state machines.

NOTE 2—If a CIST state machine sets newInfoCist, this machine will ensure that a BPDU is transmitted conveying the new CIST information. If MST BPDUs can be transmitted through the port this BPDU will also convey new MSTI information for all MSTIs. If an MSTI state machine sets newInfoMsti, and MST BPDUs can be transmitted though the port, this machine will ensure that a BPDU is transmitted conveying information for the CIST and all MSTIs. Separate newInfoCist and newInfoMsti variables are provided to avoid requiring useless transmission of a BPDU through a port that can only transmit STP BPDUs (as required by the ForceVersion parameter or Port Protocol Migration machine) following a change in MSTI information without any change to the CIST.

13.31 Port Information state machine

The Port Information state machine for each tree shall implement the function specified by the state diagram contained in Figure 13-14 and the attendant definitions contained in 13.21 through 13.26.

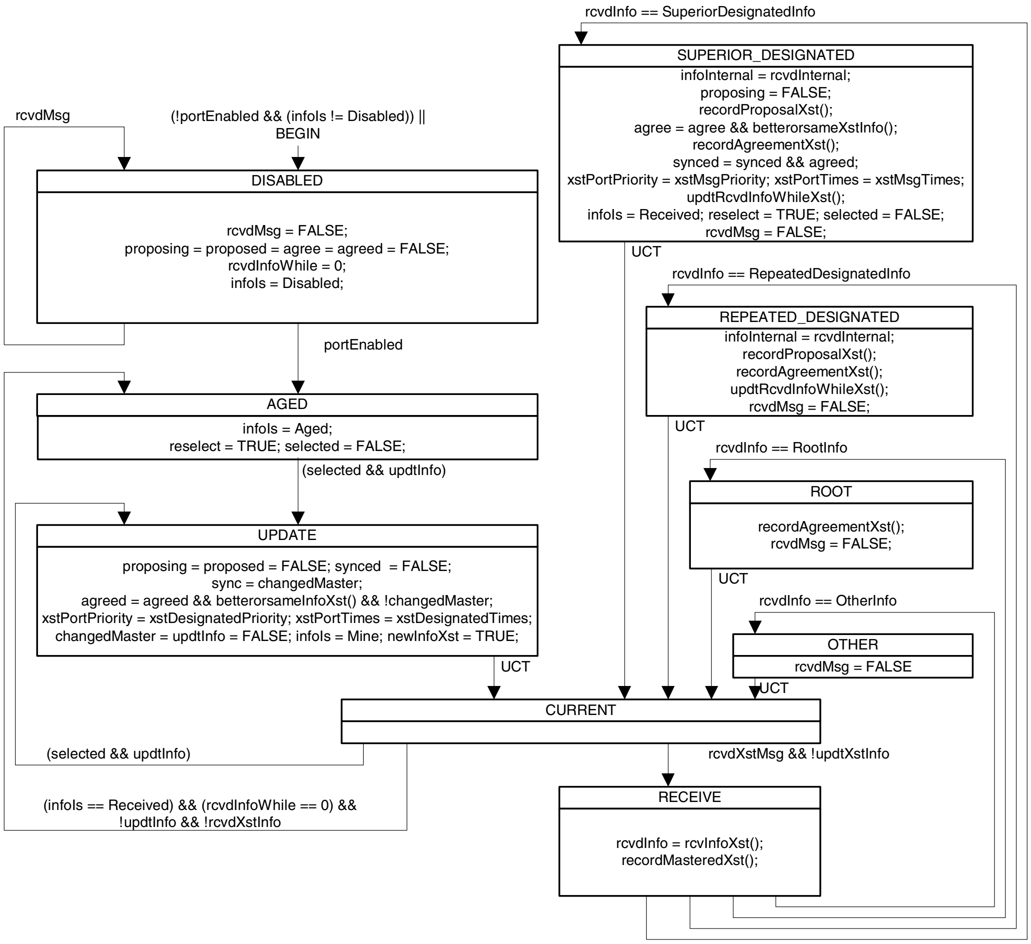

rcvdInfo == SuperiorDesignatedInfo
SUPERIOR_DESIGNATED
infoInternal = rcvdInternal;
proposing = FALSE;
recordProposalXst();
agree = agree && betterorsameXstInfo();
recordAgreementXst();
synced = synced && agreed;
xstPortPriority = xstMsgPriority; xstPortTimes = xstMsgTimes;
updtRcvdInfoWhileXst();
infoIs = Received; reselect = TRUE; selected = FALSE;
rcvdMsg = FALSE;
UCT
rcvdMsg
(!portEnabled && (infoIs != Disabled)) ||
BEGIN
DISABLED
rcvdMsg = FALSE;
proposing = proposed = agree = agreed = FALSE;
rcvdInfoWhile = 0;
infoIs = Disabled;
rcvdInfo == RepeatedDesignatedInfo
REPEATED_DESIGNATED
infoInternal = rcvdInternal;
recordProposalXst();
recordAgreementXst();
updtRcvdInfoWhileXst();
rcvdMsg = FALSE;
portEnabled
AGED
infoIs = Aged;
reselect = TRUE; selected = FALSE;
UCT
rcvdInfo == RootInfo
(selected && updtInfo)
ROOT
recordAgreementXst();
rcvdMsg = FALSE;
UPDATE
proposing = proposed = FALSE; synced = FALSE;
sync = changedMaster;
agreed = agreed && betterorsameInfoXst() && !changedMaster;
xstPortPriority = xstDesignatedPriority; xstPortTimes = xstDesignatedTimes;
changedMaster = updtInfo = FALSE; infoIs = Mine; newInfoXst = TRUE;
UCT
rcvdInfo == OtherInfo
OTHER
rcvdMsg = FALSE
UCT
UCT
CURRENT
(selected && updtInfo)
rcvdXstMsg && !updtXstInfo
(infoIs == Received) && (rcvdInfoWhile == 0) &&
!updtInfo && !rcvdXstInfo
RECEIVE
rcvdInfo = rcvInfoXst();
recordMasteredXst();

Figure 13-14—Port Information state machine

This state machine is responsible for recording the Spanning Tree information currently in use by the CIST or a given MSTI for a given Port, ageing that information out if it was derived from an incoming BPDU, and recording the origin of the information in the infoIs variable. The selected variable is cleared and reselect set to signal to the Port Role Selection machine that port roles need to be recomputed. The infoIs and portPriority variables from all ports are used in that computation and, together with portTimes, determine new values of designatedPriority and designatedTimes. The selected variable is set by the Port Role Selection machine once the computation is complete.

13.32 Port Role Selection state machine

The Port Role Selection machine for each tree shall implement the function specified by the state diagram contained in Figure 13-15 and the attendant definitions contained in 13.21 through 13.26.

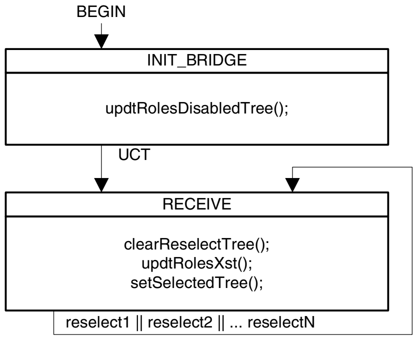


Figure 13-15—Port Role Selection

NOTE—The specification of this state machine is identical to that of the Port Role Selection state machine for RSTP (Clause 17 of IEEE Std 802.1D, 1998 Edition) but makes use of revised procedures as detailed in Clause 13.26 of this specification. These procedures have also been renamed relative to IEEE Std 802.1D, 1998 Edition; updtRoleDisabledBridge to updtRoleDisabledTree, clearReselectBridge to clearReselectTree, updtRolesBridge to updtRolesCist and updtRolesMsti, and setSelectedBridge to setSelectedTree.

13.33 Port Role Transitions state machine

The Port Role Transitions state machine shall implement the function specified by the state diagram contained in Figure 13-16 and Figure 13-17 and the attendant definitions contained in 13.21 through 13.26.

As Figure 13-16 and Figure 13-17 are component parts of the same state machine, the global transitions associated with both diagrams are possible exit transitions from the states shown in either of the diagrams. Figure 13-16 shows the Port Roles for Ports that do not form part of the active topology of the given Tree.

Figure 13-17 shows the Port Roles that form part of the active topology.

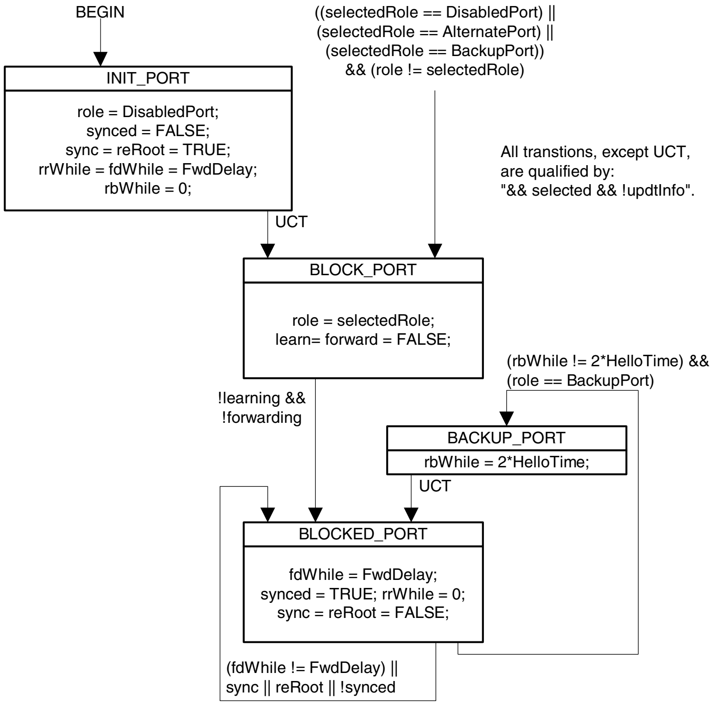
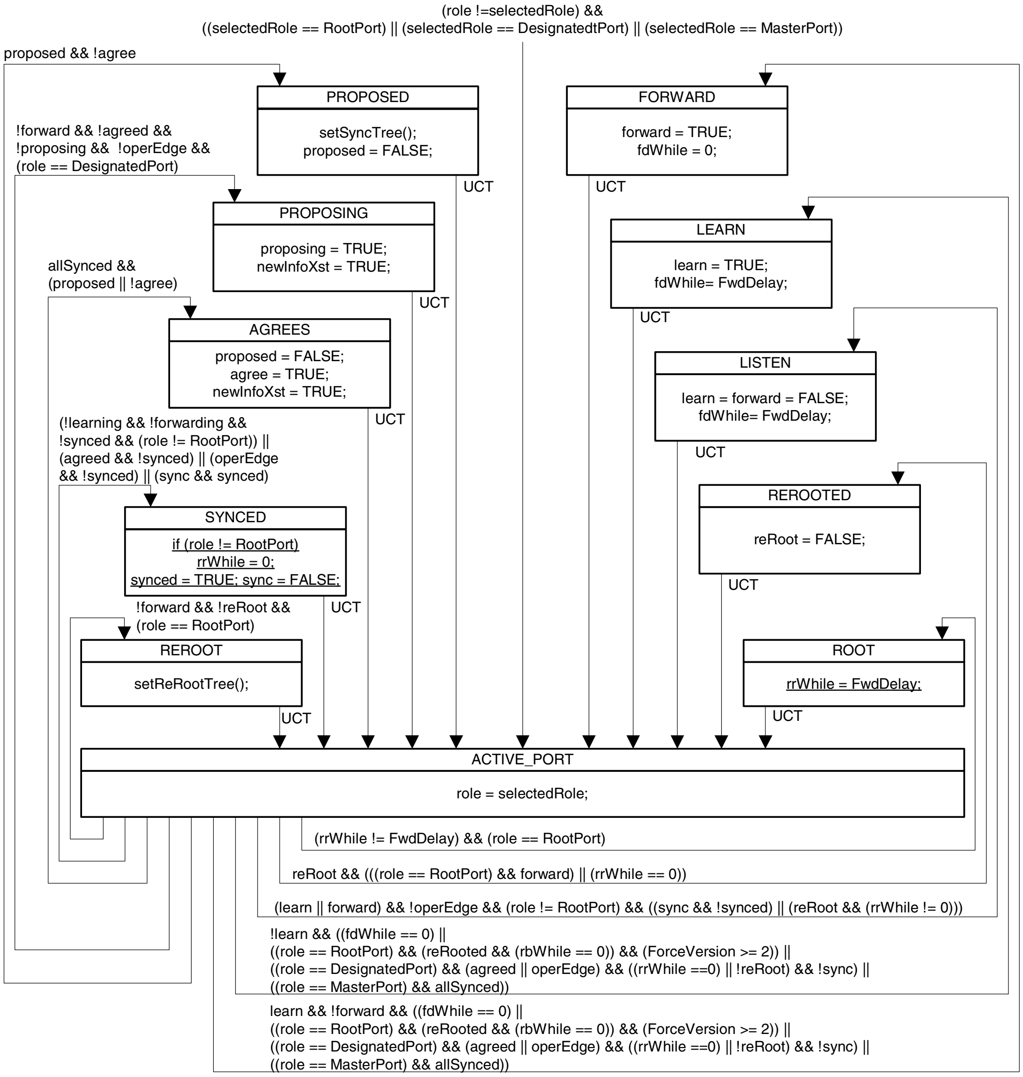


Figure 13-16—Port Role Transitions state machine— Part 1: Disabled, Alternate, and Backup Roles

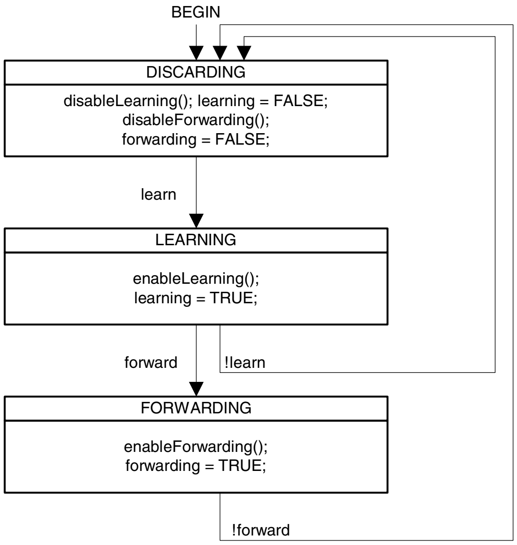

(role !=selectedRole) &&
((selectedRole == RootPort) || (selectedRole == DesignatedtPort) || (selectedRole == MasterPort))
proposed && !agree
PROPOSED
setSyncTree();
proposed = FALSE;
UCT
FORWARD
forward = TRUE;
fdWhile = 0;
UCT
!forward && !agreed &&
!proposing && !operEdge &&
(role == DesignatedPort)
PROPOSING
proposing = TRUE;
newInfoXst = TRUE;
UCT
LEARN
learn = TRUE;
fdWhile= FwdDelay;
UCT
allSynced &&
(proposed || !agree)
AGREES
proposed = FALSE;
agree = TRUE;
newInfoXst = TRUE;
UCT
LISTEN
learn = forward = FALSE;
fdWhile= FwdDelay;
UCT
(!learning && !forwarding &&
!synced && (role != RootPort)) ||
(agreed && !synced) || (operEdge
&& !synced) || (sync && synced)
REROOTED
reRoot = FALSE;
UCT
SYNCED
if (role != RootPort)
rrWhile = 0;
synced = TRUE; sync = FALSE;
!forward && !reRoot &&
(role == RootPort)
UCT
REROOT
setReRootTree();
UCT
ROOT
rrWhile = FwdDelay;
UCT
ACTIVE_PORT
role = selectedRole;
(rrWhile != FwdDelay) && (role == RootPort)
reRoot && (((role == RootPort) && forward) || (rrWhile == 0))
(learn || forward) && !operEdge && (role != RootPort) && ((sync && !synced) || (reRoot && (rrWhile != 0)))
!learn && ((fdWhile == 0) ||
((role == RootPort) && (reRooted && (rbWhile == 0)) && (ForceVersion >= 2)) ||
((role == DesignatedPort) && (agreed || operEdge) && ((rrWhile ==0) || !reRoot) && !sync) ||
((role == MasterPort) && allSynced))
learn && !forward && ((fdWhile == 0) ||
((role == RootPort) && (reRooted && (rbWhile == 0)) && (ForceVersion >= 2)) ||
((role == DesignatedPort) && (agreed || operEdge) && ((rrWhile ==0) || !reRoot) && !sync) ||
((role == MasterPort) && allSynced))

All transtions, except UCT, are qualified by "&& selected && !updtInfo".

Figure 13-17—Port Role Transitions state machine— Part 2: Root, Designated, and Master Roles

13.34 Port State Transition state machine

The Port State Transition state machine for each tree shall implement the function specified by the state diagram contained in Figure 13-18 and the attendant definitions contained in 13.21 through 13.26 above.

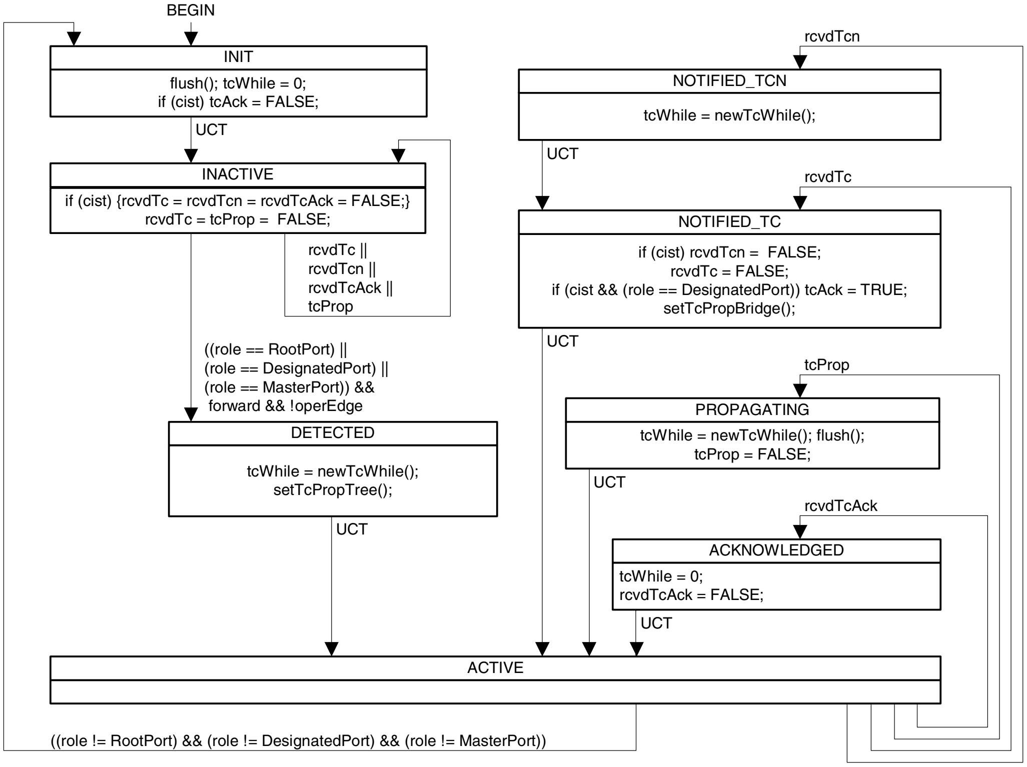


NOTE: A small system dependent delay may occur on each of the transitions shown.

Figure 13-18—Port State Transition state machine

13.35 Topology Change state machine

The Topology Change state machine for each tree shall implement the function specified by the state diagram contained in Figure 13-19 and the attendant definitions contained in 13.21 through 13.26.



NOTE 1 - rcvdTcn and rcvdTcAck conditions only occur for the CIST
NOTE 2 - The flush() procedure, when called on a transition from ACTIVE to INIT, should take account of any system dependent delay associated with port state transitions

Figure 13-19—Topology Change state machine

13.36 Performance

This subclause places requirements on the setting of the parameters of MSTP. It recommends default operational values for performance parameters. These have been specified in order to avoid the need to set values prior to operation, and have been chosen with a view to maximizing the ease with which Bridged LAN components interoperate.

The constraints on parameter values for correct operation are essentially the same as specified in IEEE Std for RSTP, and provide for the integration of MST Bridges into LANs that contain Bridges using STP. Bridges using MSTP shall conform to the parameter value requirements of 17.39 of IEEE Std 802.1D, 1998 Edition. Implementations of MSTP and managers of Bridged Local Area Networks should note the recommendations of that Clause. Maximum, minimum, default, and or applicable ranges are specified and recommended for values of the following parameters:

a) Maximum Bridge Diameter
b) Maximum Bridge Transit Delay
c) Maximum BPDU Transmission Delay
d) Maximum Message Age Increment Overestimate
e) Bridge Priority and Port Priority
f) External Port Path Cost (referred to as Path Cost in 17.39 of IEEE Std 802.1D, 1998 Edition)
g) Port Hello Time (HelloTime in 13.22)

NOTE—In MSTP Bridges, Hello Time is manageable on a per-Port basis, rather than per-Bridge in STP and RSTP. Port Hello Time therefore replaces the Bridge Hello Time parameter found in STP and RSTP.

h) Bridge Max Age
i) Bridge Forward Delay

This standard also makes recommendations on the applicable values of Internal Port Path Cost, a parameter specific to MSTP.

13.36.1 Internal Port Path Costs

It is recommended that default values of the Internal Port **Path Cost** parameter for each Bridge Port be based on the values shown in Table 13-3, the values being chosen according to the speed of the LAN segment to which each Port is attached.

Table 13-3—Internal Port Path Costs

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

Where intermediate link speeds are created as a result of the aggregation of 2 or more links of the same speed (see IEEE Std 802.3ad), it may be appropriate to modify the recommended values shown to reflect the change in link speed. However, as the primary purpose of the Path Cost is to establish the active topology of the network, it may be inappropriate for the Path Cost to track the effective speed of such links too closely, as the resultant active topology may differ from that intended by the network administrator. For example, if the network administrator had chosen an active topology that makes use of aggregated links for resilience (rather than for increased data rate), it would be inappropriate to cause a Spanning Tree topology change as a result of one of the physical links in an aggregation failing. Similarly, with links that can autonegotiate their data rate, reflecting such changes of data rate in changes to Path Cost may not be appropriate, depending upon the intent of the network administrator. Hence, as a default behavior, such dynamic changes of data rate should not automatically cause changes in Path Cost for the Port concerned.

NOTE 1—The values shown apply to both full duplex and half duplex operation. The intent of the recommended values and ranges shown is to minimize the number of Bridges in which path costs need to be managed in order to exert control over the topology of the Bridged LAN.

NOTE 2—The values shown are the same as those recommended in IEEE Std 802.1t-2001.

14. Use of BPDUs by MSTP

This clause specifies the BPDU formats, encoding, and decoding used to exchange protocol parameters with other Bridges operating MSTP, RSTP, or STP, by a Bridge Protocol Entity operating MSTP (Clause 13).

14.1 BPDU Structure

14.1.1 Transmission and representation of octets

All BPDUs shall contain an integral number of octets. The octets in a BPDU are numbered starting from 1 and increasing in the order they are put into a Data Link Service Data Unit (DLSDU).When bit positions in an octet or a sequence of octets encode a number, the number is encoded as an unsigned binary numeral with bit positions in lower octet numbers having more significance. Within an octet the bits are numbered from 8 to 1, where 1 is the low-order bit. Where sequences of bits are represented, high order bits are shown to the left of lower order bits in the same octet, and bits in lower octet numbers are shown to the left of bits in higher octet numbers.

14.1.2 Components

A Protocol Identifier is encoded in the initial octets of all BPDUs. The single Protocol Identifier value of 0000 0000 0000 0000 identifies the Spanning Tree family of protocols (the Spanning Tree Algorithm and Protocol, the Rapid Spanning Tree Algorithm and Protocol, and the Multiple Spanning Tree Protocol).

14.2 Encoding of parameter types

The following parameter types are encoded as specified in 9.1 of IEEE Std 802.1D, 1998 Edition.

a) Protocol Identifiers.
b) Protocol Version Identifiers.
c) BPDU Types.
d) Flags.
e) Port Identifiers.
f) Timer values.
g) Length values.

Additional considerations follow for encoding:

h) Port Roles
i) Bridge Identifiers
j) Port Identifiers
k) External Root Path Costs
l) Internal Root Path Costs

This standard specifies new or extended parameter types and encodings for:

m) Hop Counts

14.2.1 Encoding of Port Role values

Port Role values shall be encoded in two consecutive flag bits, taken to represent an unsigned integer, as follows:

a) A value of 0 indicates Master Port;
b) A value of 1 indicates Alternate or Backup;
c) A value of 2 indicates Root;
d) A value of 3 indicates Designated.

14.2.2 Allocation and encoding of Bridge Identifiers

The 12-bit system ID extension component of a Bridge Identifier (9.2.5 of IEEE Std 802.1D, 1998 Edition) is used to allocate distinct Bridge Identifiers to each Spanning Tree instance supported by the operation of MSTP, based on the use of a single Bridge Address component value for the MST Bridge as a whole. The system ID extension value zero shall be allocated to the Bridge Identifier used by MSTP in support of the CIST; the system ID extension value allocated to the Bridge Identifier used by a given MSTI shall be equal to the MSTID.

NOTE 1—This convention is used to convey the MSTID for each MSTI parameter set in an MST BPDU.

The four most significant bits of the Bridge Identifier for a given Spanning Tree instance (the settable Priority component) can be modified independently of the other Bridge Identifiers supported by the Bridge, allowing full configuration control to be exerted over each Spanning Tree instance with regard to bridge priority.

NOTE 2—Only these four bits of the transmitting Bridge's Bridge Identifier are encoded in BPDUs for each MSTI. The remainder of the Bridge Identifier is derived from the CIST Bridge Identifier and the MSTID using the system ID extension convention described above.

14.2.3 Allocation and encoding of Port Identifiers

The four most significant bits of the Port Identifier for a given Spanning Tree instance (the settable Priority component) can be modified independently for each Spanning Tree instance supported by the Bridge.

NOTE—Only these four bits of the transmitting Bridge's Port Identifier are encoded in a BPDU for each MSTI. The remainder of the Port Identifier is derived from the CIST Port Identifier.

14.2.4 Encoding of External Root Path Cost

The External Root Path Cost shall be encoded as specified by 9.2.6 of IEEE Std 802.1D, 1998 Edition for Root Path Cost in four octets, taken to represent a number of arbitrary cost units. Subclause 8.10.2 of IEEE Std 802.1D, 1998 Edition contains recommendations as to the increment to the Root Path Cost, in order that some common value can be placed on this parameter without requiring a management installation practice for Bridges in a Bridged LAN.

14.2.5 Encoding of Internal Root Path Cost

The Internal Root Path Cost shall be encoded in four octets, taken to represent a number of arbitrary cost units that may differ from those used for External Path Cost. Table 13-3 contains recommendations for the use of these units. These recommendations allow higher LAN speeds to be represented in support of both current and future technologies, while still allowing common values to be assigned without a management installation practice.

NOTE—This revision from the original 802.1D recommendations for STP Path Cost causes no operational difficulties because there was no installed base of Bridges using the Internal Root Path Cost parameter prior to approval of this standard.

14.2.6 Encoding of Hop Counts

The number of remaining Hops parameter shall be encoded in a single octet.

14.3 BPDU formats and parameters

14.3.1 STP BPDUs

The formats of STP BPDU Configuration and TCN BPDUs are as specified in Clause 9 of IEEE Std 802.1D, 1998 Edition.

14.3.2 RST BPDUs

The format of RST BPDUs is as specified in Clause 9 of IEEE Std 802.1D, 1998 Edition.

14.3.3 MST BPDUs

The format of MST BPDUs is compatible with that specified for RST BPDUs (Clause 9 of IEEE Std 802.1D, 1998 Edition), with the addition of fields to convey information for the IST and each MSTI, and is shown in Figure 14-1. Each transmitted MST BPDU shall contain the parameters specified and no others.

NOTE—The BPDU specified in this clause is carried in an LLC Type 1 frame following the DSAP, LSAP, and UI fields (7.12 on Addressing in IEEE Std 802.1D, 1998 Edition). The consequence of the inclusion of those 3 octets in an 802.3 or Ethernet MAC frame is that, if the MAC Addresses in the frame are aligned on an even octet boundary then so are the BPDU octet pairs 6 and 7, 14 and 15, 18 and 19, etc.

14.4 Validation of received BPDUs

An MST Bridge Protocol Entity shall examine Octets 1 and 2 (conveying the Protocol Identifier), Octet 3 (conveying the Protocol Version Identifier encoded as a number), Octet 4 (conveying the BPDU Type) and the total length of the received BPDU (including the preceding fields, but none prior to the Protocol Identifier) to determine the further processing required as follows:

a) If the Protocol Identifier is 0000 0000 0000 0000, the BPDU Type is 0000 0000, and the BPDU contains 35 or more octets, it shall be decoded as an STP Configuration BPDU.
b) If the Protocol Identifier is 0000 0000 0000 0000, the BPDU Type is 1000 0000 (where bit 8 is shown at the left of the sequence), and the BPDU contains 4 or more octets, it shall be decoded as an STP TCN BPDU (9.3.2 of IEEE Std 802.1D, 1998 Edition).
c) If the Protocol Identifier is 0000 0000 0000 0000, the Protocol Version Identifier is 2, and the BPDU Type is 0000 0010 (where bit 8 is shown at the left of the sequence), and the BPDU contains 36 or more octets, it shall be decoded as an RST BPDU.
d) If the Protocol Identifier is 0000 0000 0000 0000, the Protocol Version Identifier is 3 or greater, and the BPDU Type is 0000 0010, and the BPDU:
    1) contains 35 or more but less than 103 octets; or
    2) contains a Version 1 Length that is not 0; or
    3) contains a Version 3 length that does not represent an integral number, from 0 to 64 inclusive, of MSTI Configuration Messages;

    it shall be decoded as an RST BPDU.

e) If the Protocol Identifier is 0000 0000 0000 0000, the Protocol Version Identifier is 3 or greater, and the BPDU Type is 0000 0010, and the BPDU:
   1) contains 102 or more octets; and
   2) a Version 1 Length of 0; and
   3) a Version 3 length representing an integral number, from 0 to 64 inclusive, of MSTI Configuration Messages;

   it shall be decoded as an MST BPDU.

f) Otherwise the BPDU shall be discarded and not processed.

NOTE 1—The LLC LSAP that identifies BPDUs is reserved for standard protocols, no other protocols using that LSAP have been standardized though they may be at some future time. At that time BPDUs with different Protocol Identifiers may be processed according to the rules of those protocols, but will still be discarded from the point of view of MSTP.

NOTE 2—These validation rules are in accord with the approach to backward compatibility of future version enhancements set out in 9.3.4 of IEEE Std 802.1D, 1998 Edition. Tests (a) and (b) above do not check the Protocol Version Identifier.

NOTE 3—These validation rules do not contain a loopback check of the form specified in 9.3.4 of IEEE Std 802.1D, 1998 Edition.

14.5 Transmission of BPDUs

An MST Bridge Protocol Entity shall encode 0000 0000 0000 0000 in Octets 1 and 2 (conveying the Protocol Identifier), the remaining fields shall be encoded to convey an STP Configuration BPDU, an STP TCN BPDU, an RST BPDU, or an MST BDU as required by the Force Protocol Version parameter, the Port Protocol Migration state machine, and other protocol parameters, all as specified in Clause 13.

a) If transmission of an STP Configuration BPDU is required, the Protocol Version Identifier shall be 0, and the BPDU Type shall be 0000 0000.
b) If transmission of an STP TCN BPDU is required, the Protocol Version Identifier shall be 0, and the BPDU Type shall be 1000 0000.
c) If transmission of an RST BPDU is required, the Protocol Version Identifier shall be 2, and the BPDU Type shall be 0000 0010.
d) If transmission of an MST BPDU is required, the Protocol Version Identifier shall be 3, and the BPDU Type shall be 0000 0010.

The remaining parameters for STP Configuration, RST, and MST BPDUs shall be encoded as specified below.

14.6 Encoding and decoding of STP Configuration, RST, and MST BPDUs

STP Configuration, RST, and MST BPDU protocol parameters are encoded for transmission, and decoded, checked or ignored on receipt as follows:

a) Bit 1 of Octet 5 conveys the CIST Topology Change flag.
b) Bit 2 of Octet 5 conveys the CIST Proposal flag in RST and MST BPDUs. It is unused in STP Configuration BPDUs, and shall be transmitted as 0 and ignored on receipt.
c) Bits 3 and 4 of Octet 5 conveys the CIST Port Role in RST and MST BPDUs. It is unused in STP Configuration BPDUs, and shall be transmitted as 0 and ignored on receipt.
d) Bit 5 of Octet 5 conveys the CIST Learning flag in RST and MST BPDUs. It is unused in STP Configuration BPDUs, and shall be transmitted as 0 and ignored on receipt.
e) Bit 6 of Octet 5 conveys the CIST Forwarding flag in RST and MST BPDUs. It is unused in STP Configuration BPDUs, and shall be transmitted as 0 and ignored on receipt.

f) Bit 7 of Octet 5 conveys the CIST Agreement flag in RST and MST BPDUs. It is unused in STP Configuration BPDUs, and shall be transmitted as 0 and ignored on receipt.
g) Bit 8 of Octet 5 conveys the Topology Change Acknowledge Flag in STP Configuration BPDUs. It is unused in RST and MST BPDUs, and shall be transmitted as 0 and ignored on receipt.
h) Octets 6 through 13 convey the CIST Root Identifier.

NOTE 1—The 12 bit system id extension component of the CIST Root Identifier can be received and subsequently transmitted as an arbitrary value, even in MST BPDUs, since the CIST Root may be an STP Bridge.

i) Octets 14 through 17 convey the CIST External Root Path Cost.
j) Octets 18 through 25 shall take the value of the CIST Regional Root Identifier when transmitted in RST and MST BPDUs, and the value of the CIST Bridge Identifier of the transmitting Bridge when transmitted in STP Configuration BPDUs. On receipt of an STP Configuration or RST BPDU both the CIST Regional Root Identifier and the CIST Designated Bridge Identifier shall be decoded from this field. On receipt of an MST BPDU the CIST Regional Root Identifier shall be decoded from this field.
k) Octets 26 and 27 convey the CIST Port Identifier of the transmitting Bridge Port.
l) Octets 28 and 29 convey the Message Age timer value.
m) Octets 30 and 31 convey the Max Age timer value.
n) Octets 32 and 33 convey the Hello Time timer value used by the transmitting Bridge Port.
o) Octets 34 and 35 convey the Max Age timer value.

No further octets shall be encoded in STP Configuration BPDUs. Additional octets in received BPDUs identified by the validation procedure (14.4) as STP Configuration BPDUs shall be ignored. The specification of encoding or decoding of further octets in this subclause refers only to RST and MST BPDUs.

p) Octet 36 conveys the Version 1 Length. This shall be transmitted as 0. It is checked on receipt by the validation procedure (14.4).

No further octets shall be encoded in RST BPDUs. Additional octets in received BPDUs identified by the validation procedure (14.4) as RST BPDUs shall be ignored. The specification of encoding or decoding of further octets in this subclause refers only to MST BPDUs.

NOTE 2—As Version 2 does not specify any additional fields beyond the end of the Version 0 information, there is no Version 2 Length field specified in Version 2 of the protocol (see Clause 9 of IEEE Std 802.1D, 1998 Edition), and therefore no need for a Version 2 length field here.

q) Octets 37 and 38 convey the Version 3 Length. Its value is the number of octets taken by the parameters that follow in the BPDU. It is checked on receipt by the validation procedure (14.4).
r) Octets 39 through 89 convey the elements of the MST Configuration Identifier (13.7):
   1) The Configuration Identifier Format Selector is encoded in octet 39 and shall take the value 0000 0000;
   2) The Configuration Name is encoded in octets 40 through 71;
   3) The Revision Level is encoded as a number in octets 72 through 73;
   4) The Configuration Digest is encoded in octets 74 through 89.
s) Octets 90 through 93 convey the CIST Internal Root Path Cost.
t) Octets 94 through 101 convey the CIST Bridge Identifier of the transmitting Bridge. The 12 bit system id extension component of the CIST Bridge Identifier shall be transmitted as 0. The behavior on receipt is unspecified if it is non-zero.

NOTE 3—The 4 most significant bits of the Bridge Identifier constitute the manageable priority component for each MSTI and are separately encoded in MSTI Configuration Messages in the BPDU.

NOTE 4—The 4 most significant bits constitute the manageable priority component of each MSTI and are separately encoded in MSTI Configuration Messages in the BPDU.

u) Octet 102 encodes the value of remaining Hops for the CIST.
v) A sequence of zero or more, up to a maximum of 64, MSTI Configuration Messages follows, each encoded as specified below.

14.6.1 MSTI Configuration Messages

A single instance of the following set of parameters is encoded for each MSTI supported by the transmitting Bridge.

a) Bits 1, 2, 3 and 4, 5, 6, 7, and 8, respectively, of Octet 1 convey the Topology Change flag, Proposal flag, Port Role, Learning flag, Forwarding flag, Agreement flag, and Master flag for this MSTI.
b) Octets 2 through 9 convey the Regional Root Identifier. This includes the value of the MSTID for this Configuration Message encoded in bits 4 through 1 of Octet 1, and bits 8 through 1 of Octet 2.

NOTE—The 4 most significant bits of each MSTI's Regional Root Identifier constitute a manageable priority component.

c) Octets 10 through 13 convey the Internal Root Path Cost.
d) Bits 5 through 8 of Octet 14 convey the value of the Bridge Identifier Priority for this MSTI. Bits 1 through 4 of Octet 14 shall be transmitted as 0, and ignored on receipt.
e) Bits 5 through 8 of Octet 15 convey the value of the Port Identifier Priority for this MSTI. Bits 1 through 4 of Octet 15 shall be transmitted as 0, and ignored on receipt.
f) Octet 16 conveys the value of remainingHops for this MSTI.

	Octet
Protocol Identifier	1–2
Protocol Version Identifier	3
BPDU Type	4
CIST Flags	5
CIST Root Identifier	6–13
CIST External Path Cost	14–17
CIST Regional Root Identifier	18–25
CIST Port Identifier	26–27
Message Age	28–29
Max Age	30–31
Hello Time	32–33
Forward Delay	34–35
Version 1 Length = 0	36
Version 3 Length	37–38
MST Configuration Identifier	39–89
CIST Internal Root Path Cost	90–93
CIST Bridge Identifier	94–101
CIST Remaining Hops	102
MSTI Configuration Messages (may be absent)	103–*39 + Version 3 Length*

Figure 14-1—MST BPDU parameters and format

	Octet
MSTI Flags	1
MSTI Regional Root Identifier	2–9
MSTI Internal Root Path Cost	10–13
MSTI Bridge Priority	14
MSTI Port Priority	15
MSTI Remaining Hops	16

Figure 14-2—MSTI Configuration Message parameters and format

Annex A

(normative)

PICS proforma[11]

A.1 Introduction

The supplier of a protocol implementation which is claimed to conform to this standard shall complete the following Protocol Implementation Conformance Statement (PICS) proforma.

A completed PICS proforma is the PICS for the implementation in question. The PICS is a statement of which capabilities and options of the protocol have been implemented. The PICS can have a number of uses, including use

a) By the protocol implementor, as a checklist to reduce the risk of failure to conform to the standard through oversight;
b) By the supplier and acquirer—or potential acquirer—of the implementation, as a detailed indication of the capabilities of the implementation, stated relative to the common basis for understanding provided by the standard PICS proforma;
c) By the user—or potential user—of the implementation, as a basis for initially checking the possibility of interworking with another implementation (note that, while interworking can never be guaranteed, failure to interwork can often be predicted from incompatible PICSs);
d) By a protocol tester, as the basis for selecting appropriate tests against which to assess the claim for conformance of the implementation.

A.2 Abbreviations and special symbols

A.2.1 Status symbols

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

A.2.2 General abbreviations

N/A	not applicable
PICS	Protocol Implementation Conformance Statement

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

A.3 Instructions for completing the PICS proforma

A.3.1 General structure of the PICS proforma

The first part of the PICS proforma, implementation identification and protocol summary, is to be completed as indicated with the information necessary to identify fully both the supplier and the implementation.

The main part of the PICS proforma is a fixed-format questionnaire, divided into several subclauses, each containing a number of individual items. Answers to the questionnaire items are to be provided in the right-most column, either by simply marking an answer to indicate a restricted choice (usually Yes or No), or by entering a value or a set or range of values. (Note that there are some items where two or more choices from a set of possible answers can apply; all relevant choices are to be marked.)

Each item is identified by an item reference in the first column. The second column contains the question to be answered; the third column records the status of the item—whether support is mandatory, optional, or conditional: see also A.3.4 below. The fourth column contains the reference or references to the material that specifies the item in the main body of this standard, and the fifth column provides the space for the answers.

A supplier may also provide (or be required to provide) further information, categorized as either Additional Information or Exception Information. When present, each kind of further information is to be provided in a further subclause of items labelled *Ai* or *Xi,* respectively, for cross-referencing purposes, where *i* is any unambiguous identification for the item (e.g., simply a numeral). There are no other restrictions on its format and presentation.

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

NOTE—Where an implementation is capable of being configured in more than one way, a single PICS may be able to describe all such configurations. However, the supplier has the choice of providing more than one PICS, each covering some subset of the implementation's configuration capabilities, in case that makes for easier and clearer presentation of the information.

A.3.2 Additional information

Items of Additional Information allow a supplier to provide further information intended to assist the interpretation of the PICS. It is not intended or expected that a large quantity will be supplied, and a PICS can be considered complete without any such information. Examples might be an outline of the ways in which a (single) implementation can be set up to operate in a variety of environments and configurations, or information about aspects of the implementation that are outside the scope of this standard but that have a bearing upon the answers to some items.

References to items of Additional Information may be entered next to any answer in the questionnaire, and may be included in items of Exception Information.

A.3.3 Exception information

It may occasionally happen that a supplier will wish to answer an item with mandatory status (after any conditions have been applied) in a way that conflicts with the indicated requirement. No pre-printed answer will be found in the Support column for this: instead, the supplier shall write the missing answer into the Support column, together with an *Xi* reference to an item of Exception Information, and shall provide the appropriate rationale in the Exception item itself.

An implementation for which an Exception item is required in this way does not conform to this standard.

NOTE—A possible reason for the situation described above is that a defect in this standard has been reported, a correction for which is expected to change the requirement not met by the implementation.

A.3.4 Conditional status

A.3.4.1 Conditional items

The PICS proforma contains a number of conditional items. These are items for which both the applicability of the item itself, and its status if it does apply—mandatory or optional—are dependent upon whether or not certain other items are supported.

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

A conditional symbol is of the form "**pred:** S" where **pred** is a predicate as described in A.3.4.2 below, and S is a status symbol, M or 0.

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

A.3.4.2 Predicates

A predicate is one of the following:

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

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

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

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

A.3.4.3 References to the text of IEEE Std 802.1D, 1998 Edition

Many of the tables in the PICS Proforma refer to the text of IEEE Std 802.1D, 1998 Edition (ANSI/IEEE Std 802.1D). A short form reference, of the form {D}X, is used in the "References" columns of these tables to denote references to clauses, subclauses or tables in IEEE Std 802.1D, 1998 Edition, where X is the clause, subclause or table identifier.

A.4 PICS proforma for IEEE Std 802.1Q, 2003 Edition

A.4.1 Implementation identification

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

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

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

A.4.2 Protocol summary, IEEE Std 802.1Q, 2003 Edition

Identification of protocol specification	IEEE Std 802.1Q, 2003 Edition, IEEE Standards for Local and metropolitan area networks—Virtual Bridged Local Area Networks
Identification of amendments and corrigenda to the PICS proforma which have been completed as part of the PICS	Amd. : Corr. : Amd. : Corr. :
Have any Exception items been required? (See A.3.3: the answer Yes means that the implementation does not conform to IEEE Std 802.1Q, 2003 Edition)	No [ ] Yes [ ]

Date of Statement	

A.5 Major capabilities and options

Item	Feature	Status	References	Support
(1a)*	Communications Support		{D}6.5	
	Which MAC types are supported on Bridge Ports, implemented in conformance with the relevant MAC standards?			
(1a.1)*	CSMA/CD, IEEE Std 802.3	O.1		Yes [ ] No [ ]
(1a.2)*	Token Bus, IEEE Std 802.4	O.1		Yes [ ] No [ ]
(1a.3)*	Token Ring, IEEE Std 802.5	O.1		Yes [ ] No [ ]
(1a.4)*	FDDI, ISO 9314-2	O.1		Yes [ ] No [ ]
(1a.5)*	DQDB, IEEE Std 802.6	O.1		Yes [ ] No [ ]
(1a.6)*	ISLAN, IEEE Std 802.9	O.1		Yes [ ] No [ ]
(1a.7)*	ISLAN 16-T, IEEE Std 802.9a	O.1		Yes [ ] No [ ]
(1a.8)*	Demand Priority, IEEE Std 802.12 (IEEE Std 802.3 format)	O.1		Yes [ ] No [ ]
(1a.9)*	Demand Priority, IEEE Std 802.12 (ISO/IEC 8802-5 format)	O.1		Yes [ ] No [ ]
(1a.11)*	Wireless LAN, ISO/IEC 8802-11(IEEE Std 802.11)	O.1		Yes [} No [ ]
(1b)	Is LLC Type 1 supported on all Bridge Ports in conformance with IEEE Std 802.2?	M	8.2, 8.3, 8.14, IEEE Std 802.2	Yes [ ]
(1c)*	Is Source-Routing Transparent Bridge operation supported on any of the Bridge Ports? (If support is claimed, the PICS proforma detailed in IEEE Std 802.1D, Annex D, shall also be completed).	O	{D}Annex C	Yes [ ] No [ ]
(2)	Relay and filtering of frames (A.6)	M	8.5, 8.6, 8.7, 8.8, 8.9	Yes [ ]
(2a)	Does the Bridge support Basic Filtering Services?	M	{D}6.6.5, 8.7.2	Yes [ ]
(2b)*	Does the Bridge support Extended Filtering Services?	O	{D}6.6.5, 8.7.2	Yes [ ] No [ ]
	If item (2b) is not supported, mark “N/A” and continue at (2e).			N/A[ ]
(2c)*	Does the Bridge support dynamic Group forwarding and filtering behavior?	2b:M	{D}6.6.5	Yes [ ] No [ ]
(2d)	Does the Bridge support the ability for static filtering information for individual MAC Addresses to specify a subset of Ports for which forwarding or filtering decisions are taken on the basis of dynamic filtering information?	2b:O	{D}6.6.5	Yes [ ] No [ ]
(2e)*	Does the Bridge support expedited traffic classes on any of its Ports?	O	8.1.2, 8.6.5	Yes [ ] No [ ]
(4)*	Does the Bridge support management of the priority of relayed frames?	O	{D}6.5, 8.5.1, 8.6.5, 8.6.7, Table 8-1, Table 8-2, Table 8-3	Yes [ ] No [ ]
(5)	Maintenance of filtering information (A.7)	M	8.8, 8.10	Yes [ ]

A.5 Major capabilities and options (*Continued*)

Item	Feature	Status	References	Support
(7a)	Can the Filtering Database be read by management?	O	8.10	Yes [ ] No [ ]
(7c)*	Can Static Filtering Entries be created and deleted?	O	8.10.1	Yes [ ] No [ ]
(7g)	Can Static Filtering Entries be created and deleted in the Permanent Database?	O	8.10.10	Yes [ ] No [ ]
(7h)	Can Static Filtering Entries be created for a given MAC Address specification with a distinct Port Map for each inbound Port?	O	8.10.1	Yes [ ] No [ ]
(7i)	Can Group Registration Entries be dynamically created, updated and deleted by GMRP?	2c:M	8.10.4, {D}10	Yes [ ] N/A [ ]
(10)	Addressing (A.8)	M	8.14	Yes [ ]
(9a)*	Can the Bridge be configured to use 48-bit Universal Addresses?	O.3	8.14	Yes [ ] No [ ]
(9b)*	Can the Bridge be configured to use 48-bit Local Addresses?	O.3	8.14	Yes [ ] No [ ]
(13)*	Spanning Tree algorithm and protocol (A.9)	O.4	5, {D}8, {D}9	Yes [ ] N [ ]
(rst)*	Rapid Spanning Tree algorithm and protocol	O.4	5, {D}9, {D}17	Yes [ ] No [ ]
(mst)*	Multiple Spanning Tree algorithm and protocol	O.4	5, 7, 8.4.1, 8.4.2, 8.6.2, 8.10.7, 8.11, 8.12, 8.14.3, 11.2.3.4, 11.3.1, 13, 14	Yes [ ] No [ ]
(msti)	Support the CIST plus a stated maximum number of MSTIs, where that number is at least 2 and at most 64	mst:M	5.2, 8.10.7, 13.14	Yes [ ] MSTIs: _____ N/A [ ]
(16)*	Does the Bridge support management of the Spanning Tree topology?	O	{D}8.2	Yes [ ] No [ ]
(17)*	Does the Bridge support management of the protocol timers?	O	{D}8.10	Yes [ ] No [ ]
(19)*	VLAN Bridge Management Operations	O	12	Yes [ ] No [ ]
(20a)*	Are the Bridge Management Operations supported via a Remote Management Protocol?	19:O.5	{D}5	Yes [ ] No [ ] N/A [ ]
(20b)*	Are the Bridge Management Operations supported via a local management interface?	19:O.5	{D}5	Yes [ ] No [ ] N/A [ ]

A.5 Major capabilities and options (*Continued*)

Item	Feature	Status	References	Support
(23a)*	Does the implementation support, on each Port, one or more of the permissible combinations of values for the Acceptable Frame Types parameter?	M	5.1, 8.4.3	Yes [ ]
(23a.1)	State which Ports support: — Admit only VLAN-tagged frames; — Admit all frames.	M	5.1, 8.4.3	Ports: _____ Ports: _____
(23a.2)	On Ports that support both values, is the parameter configurable via management?	M	5.1, 8.4.3, 12.10	Yes [ ] N/A [ ]
(23b)	Does the implementation support the ability to insert tag headers into, modify tag headers in, and remove tag headers from relayed frames, as required by the capabilities of each Bridge Port?	M	5.1, 6.1, 9	Yes [ ]
(23c)	Does the implementation support the ability to perform automatic configuration and management of VLAN topology information by means of GVRP on all Ports?	M	5.1, 11	Yes [ ]
(23d)	Does the implementation support the ability for the Filtering Database to contain static and dynamic configuration information for at least one VLAN, by means of Static and Dynamic VLAN Registration Entries?	M	5.1, 8.10	Yes [ ]
(23d.1)	State the maximum number of VLANs supported by the implementation.	M	5.1, 8.10, 9.3.2.3	_____ VLANs
(23d.2)	State the range of VID values supported by the implementation.	M	8.10, 9.3.2.3	0 through _____
(23e)*	VLAN Learning support		5.1, 8.10.3, 8.10.7, 8.10.8	
(23e.1)	Does the implementation support at least one FID?	M		Yes [ ]
(23e.2)	Can the implementation allocate at least one VID to each FID supported?	M		Yes [ ]
(23e.4)	State the maximum number of FIDs that can be supported by the implementation.	M	8.10.7	____ FIDs
(23e.5)	State the maximum number of VIDs that can be allocated to each FID.	M	8.10.7	____ VIDs
(23e.6)	Does the implementation support configuration of VLAN Learning Constraints via management?	O	5.2, 8.10.7, 12.10.3	Yes [ ] No [ ]
(23e.7)	State the number of VLAN Learning Constraints that can be configured in the implementation.	23e.6:M	5.2, 8.10.7, 12.10.3	____ Constraints
(23e.8)	Does the implementation support configuration of VID to FID allocations via management?	O	5.2, 8.10.7.1, 12.10.3	Yes [ ] No [ ]

A.5 Major capabilities and options (*Continued*)

Item	Feature	Status	References	Support
(23e.9)	Does the implementation take account of the allocation of VIDs to FIDs when making forwarding decisions relative to group MAC Addresses?	O	8.10.8	Yes [ ] No [ ]
(23f)	On Ports that support untagged and priority-tagged frames, does the implementation support:		5.1, 8.4.4, 12.10	
(23f.1)	— A PVID value?	M		Yes [ ] N/A [ ]
(23f.2)	— The ability to configure one VLAN whose Untagged set includes that Port?	M		Yes [ ] N/A [ ]
(23f.3)	— Configuration of the PVID value via management operations?	M		Yes [ ] N/A [ ]
(23f.4)	— Configuration of Static Filtering Entries via management operations?	M		Yes [ ] N/A [ ]
(23f.5)	— The ability to configure more than one VLAN whose Untagged set includes that Port?	O		Yes [ ] No [ ] N/A [ ]
(23g)*	Does the implementation support the ability to enable and disable Ingress Filtering?	O	5.2, 8.4.5	
(23h)	Does the implementation support VLAN management operations?	19:O	5.2, 12.10.2, 12.10.3	Yes [ ] No [ ]
(23i)	Is the minimum tagged frame length that can be transmitted on IEEE Std 802.3 Ports less than 68 (but 64 or more) octets?	1a.1:O	6.5	Yes [ ] No [ ] N/A [ ]
(23j)*	When transmitting untagged frames and the canonical_format_indicator parameter indicates that the mac_service_data_unit may contain embedded MAC Addresses in a format inappropriate to the destination MAC method, which of the following procedures is adopted by the Bridge:		6.1, 6.4.2.2	
(23j.1)	Convert any embedded MAC Addresses in the mac_service_data_unit to the format appropriate to the destination MAC method.	O.7		Yes [ ] No [ ]
(23j.2)	Discard the frame without transmission on that Port.	O.7		Yes [ ] No [ ]
(23k)	Does the Bridge perform frame translations, where necessary, in accordance with the procedures described in ISO/IEC 11802-5, IETF RFC 1042, and IETF RFC 1390?	TB:M	6.1, 6.4.2.2	Yes [ ] No [ ] N/A [ ]
(23l)*	Does the implementation support Port-and-Protocol-based classification of frames on any or all Ports?	O	8.6	Yes [ ] No [ ]
(23m)*	Does the implementation support a Protocol Group Database?	23l:M	8.9.4	Yes [ ] N/A [ ]

Predicates:
TB = True if the Bridge supports translational Bridging; i.e., the Bridge supports 802.3/Ethernet MAC methods on one or more Ports and Token Ring/FDDI MAC methods on one or more Ports.

A.6 Relay and filtering of frames

Item	Feature	Status	References	Support
(2f)	Are received frames with MAC method errors discarded?	M	{D}6.4, 8.5	Yes [ ]
(2g)	Are correctly received frames submitted to the Learning Process?	M	8.5	Yes [ ]
(2h)	Are user data frames the only type of frame relayed?	M	8.5	Yes [ ]
(2i)	Are request with no response frames the only frames relayed?	M	8.5	Yes [ ]
(2j)	Are all frames addressed to the Bridge Protocol Entity submitted to it?	M	8.5	Yes [ ]
(2k)	Are user data frames the only type of frame transmitted?	M	8.7	Yes [ ]
(2l)	Are request with no response frames the only frames transmitted?	M	8.7	Yes [ ]
(ingr)	Are frames assigned to VLANs, and discarded or permitted ingress, in accordance with classification rules?	M	8.6.1, 8.9	Yes [ ]
(actve)	Are frames allocated to spanning trees, and forwarded or discarded in accordance with the active topology for their spanning tree instance?	M	8.6.2	Yes [ ]
(filter)	Are frames filtered as specified?	M	8.6.3	Yes [ ]
(egrs)	Are frames filtered and formatted in accordance with egress rules?	M	8.6.4	Yes [ ]
(pmap)	Is priority mapping performed as specified?	M	8.6.7	
(form)	Are frames formatted in accordance with the tagged frame format?	M	8.6.8, 9	Yes [ ]
(2m)	Are relayed frames queued for transmission only under the conditions in 8.6.5?	M	8.6.5, {D}8.4	Yes [ ]
(2n)	Is the order of relayed frames preserved in accordance with the requirements of the forwarding process?	M	8.6.5, 8.1.1	Yes [ ]
(2o)	Is a relayed frame submitted to a MAC Entity for transmission only once?	M	8.6.6, {D}6.3.4	Yes [ ]
(2p)	Is a maximum bridge transit delay enforced for relayed frames?	M	8.6.5	Yes [ ]
(2q)	Are queued frames discarded if a Port leaves the Forwarding State?	M	8.6.5	Yes [ ]
(2r)	Is the user priority of relayed frames preserved where possible?	M	{D}6.4	Yes [ ]
(2s)	Is the user priority set to the Default User Priority for the reception Port otherwise?	M	{D}6.4	Yes [ ]

A.6 Relay and filtering of frames (*Continued*)

Item	Feature	Status	References	Support
(2t)	Is the user priority regenerated by means of the User Priority Regeneration Table?	M	8.5.1, Table 8-1	Yes [ ]
(2u)	Is mapping of Regenerated User Priority to Traffic Class performed by means of the Traffic Class Table?	M	8.5.1, Table 8-2	Yes [ ]
(2v)	Is the access priority derived from the Regenerated User Priority as defined by the values in Table 8-3 for each outbound MAC method supported by the Bridge?	M	8.6.7, Table 8-3	Yes [ ]
(2w)	Does the Bridge generate an M_UNITDATA.indication primitive on receipt of a valid frame transmitted by the Bridge Port's local MAC entity?	MS1:X	{D}6.5.4, ISO 9314-2	No [ ] N/A [ ]
(2x)	Is only Asynchronous service used?	MS1:M	ISO 9314-2, 8.1.4	Yes [ ] N/A [ ]
(2y)	On receiving a frame from an FDDI ring for forwarding, does the bridge set the C indicator?	MS1:O	{D}6.5.4, ISO 9314-2, 7.3.8	Yes [ ] No [ ] N/A [ ]
(2z)	On receiving a frame from an FDDI ring for forwarding, does the bridge leave the C indicator unaltered?	MS1:O	{D}6.5.4, ISO 9314-2, 7.3.8	Yes [ ] No [ ] N/A [ ]
	If item 4 is not supported, mark "N/A" and continue at item (4d).			N/A [ ]
(4a)*	Can the Default User Priority parameter for each Port be set to any value in the range 0 through 7?	4:O.6	{D}6.4	Yes [ ] No [ ]
(4b)*	Can the entries in the User Priority Regeneration Table for each Port be set to the full range of values shown in Table 8-1?	4:O.6	8.5.1, Table 8-1	Yes [ ] No [ ]
(4c)*	Can the entries in the Traffic Class Table for each Port be set to the full range of values shown in Table 8-2?	MS2:O	8.6.5, Table 8-2	Yes [ ] No [ ] N/A [ ]
	If item 4 is supported, mark "N/A" and continue at item (4g)			N/A [ ]
(4d)	Does the Bridge support the recommended default value of the Default User Priority parameter for each Port?	¬ 4:M	{D}6.4	Yes [ ]
(4e)	Does the Bridge support the recommended default mappings between received user priority and Regenerated User Priority for each Port as defined in Table 8-1?	¬ 4:M	8.5.1, Table 8-1	Yes [ ]

A.6 Relay and filtering of frames (*Continued*)

Item	Feature	Status	References	Support
(4f)	Does the Bridge support the recommended default user_priority to traffic class mappings shown in Table 8-2 for each Port?	MS3:M	8.6.5, Table 8-2	Yes [ ] N/A [ ]
(4g)	Is the Bridge able to use any values other than those shown in Table 8-3 when determining the access priority for the MAC methods shown?	X	8.6.7, Table 8-3	No [ ]

Predicates:
MS1 = 1a.4 AND NOT (1a.1 OR 1a.2 OR 1a.3 OR 1a.5 OR 1a.6 OR 1a.7 OR 1a.8)
MS2 = 2e AND 4
MS3 = 2e AND NOT 4

A.7 Maintenance of filtering entries in the Filtering Database

Item	Feature	Status	References	Support
(5a)	Are Dynamic Filtering Entries created and updated if and only if the Port State permits?	M	8.10, 8.10.3, {D}8.4	Yes [ ]
(5b)	Are Dynamic Filtering Entries created on receipt of frames with a group source address?	X	8.10, 8.10.3	No [ ]
(5c)	Does the Filtering Database support Static Filtering Entries?	M	8.10.1	Yes [ ]
(5d)	Can a Dynamic Filtering Entry be created that conflicts with an existing Static Filtering Entry?	X	8.8, 8.10, 8.10.1, 8.10.3	No [ ]
(5e)	Does the Filtering Database support Dynamic Filtering Entries?	M	8.10.3	Yes [ ]
(5f)	Does the creation of a Static Filtering Entry remove any conflicting information in a Dynamic Filtering Entry for the same address?	M	8.10.1, 8.10.3	Yes [ ]
(5g)	Does each Static Filtering Entry specify a MAC Address specification and a Port Map?	M	8.10.1	Yes [ ]
(5h)	Are Dynamic Filtering Entries removed from the Filtering Database if not updated for the Ageing Time period?	M	8.10.3	Yes [ ]
(5i)	Does each Dynamic Filtering Entry specify a MAC Address specification and a Port Map?	M	8.10.3	Yes [ ]
(5j)	Is the Filtering Database initialized with the entries contained in the Permanent Database?	M	8.10.10	Yes [ ]
	If item (2c) is not supported, mark N/A and continue at item (6a).			N/A [ ]
(5k)	Does each Group Registration Entry specify a MAC Address specification and a Port Map?	2c:M	8.10.4	Yes [ ]

A.7 Maintenance of filtering entries in the Filtering Database (*Continued*)

Item	Feature	Status	References	Support
(5l)	Can the MAC Address specification in Group Registration Entries represent All Groups, All Unregistered Groups, or a specific group MAC Address?	2c:M	8.10.4	Yes [ ]
(5m)	Are Group Registration Entries created, updated and removed from the Filtering Database in accordance with the specification of GMRP?	2c:M	8.10.4, {D}10	Yes [ ]
(5n)	Are Group Registration Entries created, updated and removed from the Filtering Database by any means other than via the operation of GMRP?	2c:X	8.10.4, {D}10	No [ ]
(6a)	State the Filtering Database Size.	M	8.10	____ entries
(6b)	State the Permanent Database Size.	M	8.10	____ entries
	If item (7c) is not supported, mark N/A and continue at item (8a).			N/A [ ]
(7d)	Can Static Filtering Entries be made for individual MAC Addresses?	**7c:**M	8.10.1	Yes [ ]
(7e)	Can Static Filtering Entries be made for group MAC Addresses?	**7c:**M	8.10.1	Yes [ ]
(7f)	Can a Static Filtering Entry be made for the broadcast MAC Address?	**7c:**M	8.10.1	Yes [ ]
(8a)	Can the Bridge be configured to use the default value of Ageing Time recommended in Table 8-4?	O	8.10.3, Table 8-4	Yes [ ] No [ ]
(8b)	Can the Bridge be configured to use any of the range of values of Ageing Time specified in Table 7-4?	O	8.10.3, Table 8-4	Yes [ ] No [ ]

A.8 Addressing

Item	Feature	Status	References	Support
(10a)	Does each Port have a separate MAC Address?	M	8.14.3	Yes [ ]
(10b)	Are all BPDUs transmitted to the same group address?	M	8.14.3, {D}8.2	Yes [ ]
	If item (9a) is not supported, mark N/A and continue at item (10d1).			N/A [ ]
(10c)	Are all BPDUs transmitted to the Bridge Protocol Group Address when Universal Addresses are used?	**9a:**M	8.14.3 {D}8.2	Yes [ ]
(10d)	Is the source address of BPDUs the address of the transmitting Port?	**9a:**M	8.14.3	Yes [ ]

A.8 Addressing (*Continued*)

Item	Feature	Status	References	Support
(10d1)	Is the LLC address of BPDUs the standard LLC address identified for the Spanning Tree Protocol?	M	8.14.3, Table 8-9	Yes [ ]
(10e)	Is the Bridge Address a Universal Address?	**M**	8.14.5, {D}8.2	Yes [ ] N/A [ ]
(10f)	Are frames addressed to any of the Reserved Addresses relayed by the Bridge?	X	8.14.6	No [ ]
	If item (13) is not supported, mark N/A and continue at item (11c).			N/A [ ]
(11a)	Is Bridge Management accessible through each Port using the MAC Address of the Port and the LSAP assigned?	**13:**O	8.14.4	Yes [ ] No [ ]
(11b)	Is Bridge Management accessible through all Ports using the All LANs Bridge Management Group Address?	13:O	8.14.4	Yes [ ] No [ ]
(11c)	Is the Bridge Address the Address of Port 1?	**9a:**O	8.14.5	Yes [ ] No [ ] N/A [ ]
(11d)*	Are Group Addresses additional to the Reserved Addresses pre-configured in the Permanent Database?	O	8.14.6	Yes [ ] No [ ]
	If item (11d) is not supported, mark N/A and continue at item (12a).			N/A [ ]
(11e)	Can the additional pre-configured entries in the Filtering Database be deleted?	**11d:**O	8.14.6	Yes [ ] No [ ]
(12a)	Can a group MAC Address be assigned to identify the Bridge Protocol Entity?	**9b:**M	{D}8.2	Yes [ ] N/A [ ]
(12c)	Does each Port of the Bridge have a distinct identifier?	M	{D}8.2, {D}8.5.5.1	Yes [ ]

A.9 Spanning Tree Algorithm

Item	Feature	Status	References	Support
	If item (13) is not supported, mark N/A and continue at the start of A.10.			N/A [ ]
(13a)	Are all the following Bridge Parameters maintained?	13:M	{D}8.5.3	Yes [ ]
	Designated Root		{D}8.5.3.1	
	Root Cost		{D}8.5.3.2	
	Root Port		{D}8.5.3.3	
	Max Age		{D}8.5.3.4	
	Hello Time		{D}8.5.3.5	
	Forward Delay		{D}8.5.3.6	

A.9 Spanning Tree Algorithm (*Continued*)

Item	Feature	Status	References	Support
	Bridge Identifier		{D}8.5.3.7	
	Bridge Max Age		{D}8.5.3.8	
	Bridge Hello Time		{D}8.5.3.9	
	Bridge Forward Delay		{D}8.5.3.10	
	Topology Change Detected		{D}8.5.3.11	
	Topology Change		{D}8.5.3.12	
	Topology Change Time		{D}8.5.3.13	
	Hold Time		{D}8.5.3.14	
(13b)	Are all the following Bridge Timers maintained?	13:M	{D}8.5.4	Yes [ ]
	Hello Timer		{D}8.5.4.1	
	Topology Change Notification Timer		{D}8.5.4.2	
	Topology Change Timer		{D}8.5.4.3	
(13c)	Are all the following Port Parameters maintained for each Port?	13:M	{D}8.5.5	Yes [ ]
	Port Identifier		{D}8.5.5.1	
	State		{D}8.5.5.2, {D}8.4	
	Path Cost		{D}8.5.5.3	
	Designated Root		{D}8.5.5.4	
	Designated Cost		{D}8.5.5.5	
	Designated Bridge		{D}8.5.5.6	
	Designated Port		{D}8.5.5.7	
	Topology Change Acknowledge		{D}8.5.5.8	
	Configuration Pending		{D}8.5.5.9	
	Change Detection Enabled		{D}8.5.5.10	
(13d)	Are all the following Timers maintained for each Port?	13:M	{D}8.5.6	Yes [ ]
	Message Age Timer		{D}8.5.6.1	
	Forward Delay Timer		{D}8.5.6.2	
	Hold Timer		{D}8.5.6.3	
(13e)	Are Protocol Parameters and Timers maintained, and BPDUs transmitted, as required on each of the following events?	13:M	{D}8.7, {D}8.9, {D}8.5.3, {D}8.5.4, {D}8.5.5, {D}8.5.6	Yes [ ]
	Received Configuration BPDU		{D}8.7.1	

A.9 Spanning Tree Algorithm (*Continued*)

Item	Feature	Status	References	Support
	Received Topology Change Notification BPDU		{D}8.7.2	
	Hello Timer Expiry		{D}8.7.3	
	Message Age Timer Expiry		{D}8.7.4	
	Forward Delay Timer Expiry		{D}8.7.5	
	Topology Change Notification Timer Expiry		{D}8.7.6	
	Topology Change Timer Expiry		{D}8.7.7	
	Hold Timer Expiry		{D}8.7.8	
(13f)	Do the following operations modify Protocol Parameters and Timers, and transmit BPDUs as required?	13:M	{D}8.8, {D}8.9, {D}8.5.3, {D}8.5.4, {D}8.5.5, {D}8.5.6	Yes [ ]
	Initialization		{D}8.8.1	
	Enable Port		{D}8.8.2	
	Disable Port		{D}8.8.3	
	Set Bridge Priority		{D}8.8.4	
	Set Port Priority		{D}8.8.5	
	Set Path Cost		{D}8.8.6	
(13g)	Does the implementation support the ability to set the value of the Change Detection Enabled parameter to Disabled?	13:O	{D}8.5.5.10	Yes [ ] No [ ]
(14a)	Does the Bridge underestimate the increment to the Message Age parameter in transmitted BPDUs?	13:X	{D}8.10.1	No [ ]
(14b)	Does the Bridge underestimate Forward Delay?	13:X	{D}8.10.1	No [ ]
(14c)	Does the Bridge overestimate the Hello Time interval?	13:X	{D}8.10.1	No [ ]
(15a)	Does the Bridge use the specified value for Hold Time?	13:M	{D}8.10.2, {D}Table 8-3	Yes [ ]
	If item (16) is not supported, mark N/A and continue at (17a).			N/A [ ]
(16a)	Can the relative priority of the Bridge be set?	13 AND 16:M	{D}8.2, {D}8.5.3.7, {D}8.8.4	Yes [ ]
(16b)	Can the relative priority of the Ports be set?	13 AND 16:M	{D}8.2, {D}8.5.5.1, {D}8.8.5	Yes [ ]
(16c)	Can the path cost for each Port be set?	13 AND 16:M	{D}8.2, {D}8.5.5.3, {D}8.8.6	Yes [ ]

A.9 Spanning Tree Algorithm (*Continued*)

Item	Feature	Status	References	Support
	If item (17) is not supported, mark N/A and continue at (18a).			N/A [ ]
(17a)	Can Bridge Max Age be set to any of the range of values specified?	13 AND 17:M	{D}8.10.2, {D}8.5.3.8, {D}Table 8-3	Yes [ ]
(17b)	Can Bridge Hello Time be set to any of the range of values specified?	13 AND 17:M	{D}8.10.2, {D}8.5.3.9, {D}Table 8-3	Yes [ ]
(17c)	Can Bridge Forward Delay be set to any of the range of values specified?	13 AND 17:M	{D}8.10.2, {D}8.5.3.10, {D}Table 8-3	Yes [ ]
(18a)	Do all BPDUs contain an integral number of octets?	13:M	{D}9.1.1	Yes [ ]
(18b)	Are all the following BPDU parameter types encoded as specified?	13:M	{D}9.1.1, {D}9.2	Yes [ ]
	Protocol Identifiers		{D}9.2.1	
	Protocol Version Identifiers		{D}9.2.2	
	BPDU Types		{D}9.2.3	
	Flags		{D}9.2.4	
	Bridge Identifiers		{D}9.2.5	
	Root Path Cost		{D}9.2.6	
	Port Identifiers		{D}9.2.7	
	Timer Values		{D}9.2.8	
(18c)	Do Configuration BPDUs have the format and parameters specified?	13:M	{D}9.3.1	Yes [ ]
(18d)	Do Topology Change Notification BPDUs have the format and parameters specified?	13:M	{D}9.3.2	Yes [ ]
(18e)	Are received BPDUs validated as specified?	13:M	{D}9.3.3	Yes [ ]

A.10 Rapid Spanning Tree Algorithm

Item	Feature	Status	References	Support
	If item (rst) is not supported, mark N/A and continue at the start of A.11.			N/A [ ]
(ids)	Provision of identifiers for Bridge and Ports	rst:M	{D}17.2	Yes [ ]
(par1)	Not exceed the values in {D}17.28.2 for max Bridge transit delay, max message age increment overestimate and max BPDU transmission delay	rst:M	{D}5.1, {D}17.28.2	Yes [ ]
(par2)	Use the value given in {D}Table 17-5 for Transmission Limit	rst:M	{D}5.1, {D}Table 17-5	Yes [ ]

A.10 Rapid Spanning Tree Algorithm (*Continued*)

Item	Feature	Status	References	Support
(inc)	Inclusion of active Ports in computation of the active topology	rst:M	{D}17.5	Yes [ ]
(pro)	Processing of BPDUs received on Ports included in the computation of the active topology	rst:M	{D}17.5	Yes [ ]
(dis)	Discarding received frames in the Discarding state	rst:M	{D}17.5	Yes [ ]
(lrn)	Incorporating station location information to the Filtering Database in the Learning and Forwarding states	rst:M	{D}17.5	Yes [ ]
(nlrn)	Not incorporating station location information to the Filtering Database in the Discarding state	rst:M	{D}17.5	Yes [ ]
(rlrn)	Transfer learned MAC addresses from a retiring Root Port to a new Root Port	rst:O	{D}17.10	Yes [ ] No [ ]
(sm)	A single instance of the Port Role Selection state machine per Bridge and an instance of all other state machines per Port	rst:M	{D}17.13	Yes [ ]
(ptmr)	Port Timers state machine support	rst:M	{D}17.15, {D}17.20	Yes [ ]
(pism)	Port Information state machine support	rst:M	{D}17.15, {D}17.21	Yes [ ]
(prssm)	Port Role Selection state machine support	rst:M	{D}17.15, {D}17.22	Yes [ ]
(prtsm)	Port Role Transitions state machine support	rst:M	{D}17.15, {D}17.23	Yes [ ]
(pstsm)	Port State Transition state machine support	rst:M	{D}17.15, {D}17.24	Yes [ ]
(tcsm)	Topology Change state machine support	rst:M	{D}17.15, {D}17.25	Yes [ ]
(ppmsm)	Port Protocol Migration state machine support	rst:M	{D}17.15, {D}17.26	Yes [ ]
(ptsm)	Port Transmit state machine support	rst:M	{D}17.15, {D}17.27	Yes [ ]
(cde)	Not support Change Detection Enabled parameter	rst:M	{D}5.2	Yes [ ]
(estm)	Not: Underestimate the increment to the Message Age parameter in transmitted BPDUs. Underestimate Forward Delay. Overestimate the Hello Time interval when acting as the Root.	rst:M	{D}17.28.1	Yes [ ]
(htim)	Use of Transmission Limit	rst:M	{D}Table 17-5	Yes [ ]
(prel)	Enforcement of parameter relationships	rst:M	{D}17.28.2	Yes [ ]
(pcst)	No defaulting to use of automatic path cost changes	rst:M	{D}17.28.2	Yes [ ]
(prv)	Range and granularity of priority values	rst:M	{D}17.28.2	Yes [ ]

A.10 Rapid Spanning Tree Algorithm (*Continued*)

Item	Feature	Status	References	Support
(pcv)	Range and granularity of path cost values	rst:M	{D}17.28.2	Yes [ ]
	If item (16) is not supported, mark N/A and continue at (tmr1)			N/A [ ]
(mgt1)	Can the relative priority of the Bridge be set?	rst AND **16**:M	{D}17.2, {D}17.4, {D}17.13	Yes [ ]
(mgt2)	Can the relative priority of the Ports be set?	rst AND **16**:M	{D}17.2, {D}17.4, {D}17.13	Yes [ ]
(mgt3)	Can the path cost for each Port be set?	rst AND **16**:M	{D}17.2, {D}17.4, {D}17.13	Yes [ ]
	If item (17) is not supported, mark N/A and continue at (pdu1)			N/A [ ]
*(tmr1)	Can Bridge Max Age be set to any of the range of values specified?	rst AND **17**:M	{D}17.2, {D}17.13, {D}Table 17-5	Yes [ ]
(tmr2)	Can Bridge Hello Time be set to any of the range of values specified?	rst AND **17**:M	{D}17.2, {D}17.13, {D}Table 17-5	Yes [ ]
(tmr3)	Can Bridge Forward Delay be set to any of the range of values specified?	rst AND **17**:M	{D}17.2, {D}17.13, {D}Table 17-5	Yes [ ]
*(pdu1)	Do all BPDUs contain an integral number of octets?	rst:M	{D}9.1.1	Yes [ ]
(pdu2)	Are all the following BPDU parameter types encoded as specified?	rst:M	{D}9.1.1, {D}9.2	Yes [ ]
	Protocol Identifiers		{D}9.2.1	
	Protocol Version Identifiers		{D}9.2.2	
	BPDU Types		{D}9.2.3	
	Flags		{D}9.2.4	
	Bridge Identifiers		{D}9.2.5	
	Root Path Cost		{D}9.2.6	
	Port Identifiers		{D}9.2.7	
	Timer Values		{D}9.2.8	
(pdu3)	Do Configuration BPDUs have the format, parameters, and parameter values specified?	rst:M	{D}9.3.1	Yes [ ]

A.10 Rapid Spanning Tree Algorithm (*Continued*)

Item	Feature	Status	References	Support
(pdu4)	Do Topology Change Notification BPDUs have the format, parameters, and parameter values specified?	rst:M	{D}9.3.2	Yes [ ]
(pdu5)	Do Rapid Spanning Tree BPDUs have the format, parameters, and parameter values specified?	rst:M	{D}9.3.3	Yes [ ]
(pdu6)	Are received BPDUs validated as specified?	rst:M	{D}9.3.4	Yes [ ]

A.11 Multiple Spanning Tree Algorithm

Item	Feature	Status	References	Support
	If item (mst) is not supported, mark N/A and continue at the start of A.12.			N/A [ ]
(maxfid)	Support at least as many FIDs as MSTIs	mst:M	5.1, 5.2, 8.10.7	Yes [ ]
(assfid)	Associate each FID to a spanning tree	mst:M	5.1, 8.11.3	Yes [ ]
(confid)	Transmit and receive MST Configuration Identifier information	mst:M	5.1, 8.11.2	Yes [ ]
(pstate)	Support a set of port state information per spanning tree per port	mst:M	5.1, 8.4.1, 8.4.2, 13.34	Yes [ ]
(proto)	Support an instance of spanning tree protocol per spanning tree per port	mst:M	5.1, 8.12, 13	Yes [ ]
(baddr)	Use the Bridge Group Address as specified	mst:M	5.1, 8.14.3	Yes [ ]
(default)	Support default Bridge Forward Delay and Bridge Priority parameter values as specified	mst:M	5.1, 13.23	Yes [ ]
(gvrpctx)	Support GVRP in multiple spanning tree contexts	mst:M	5.1, 11.2.3.3, 11.2.3.4	Yes [ ]
(manage)	Support VLAN bridge management for the bridge protocol entity in all supported spanning trees	mst:M	5.1, 12.8	Yes [ ]
(manag1)	Support independent management of bridge and port priority and path cost per spanning tree	mst:M	5.1, 12.8.1	Yes [ ]
(manag2)	Support VLAN management per spanning tree	mst:M	5.1, 12.10.1	Yes [ ]
(manag3)	Support MSTI configuration management	mst:M	5.1, 12.12	Yes [ ]
(mids)	Provision of identifiers for Bridge and Ports	mst:M	13.23.2, 13.24.21, {D}17.2	Yes [ ]
(mpar1)	Not exceed the values in {D}17.28.2 for max Bridge transit delay, max message age increment overestimate and max BPDU transmission delay	mst:M	13.36, {D}5.1, {D}17.28.2	Yes [ ]
(mpar2)	Use the value given in {D}Table 17-5 for Transmission Limit	mst:M	13, {D}5.1, {D}Table 17-5	Yes [ ]
(minc)	Inclusion of active Ports in computation of the active topology for a given spanning tree	mst:M	13, {D}17.5	Yes [ ]

A.11 Multiple Spanning Tree Algorithm (*Continued*)

Item	Feature	Status	References	Support
(mpro)	Processing of BPDUs received on Ports included in the computation of the active topology for a given spanning tree	mst:M	13, {D}17.5	Yes [ ]
(mdis)	Discarding received frames in the Discarding state for a given spanning tree	mst:M	13, {D}17.5	Yes [ ]
(mlrn)	Incorporating station location information to the Filtering Database in the Learning and Forwarding states for a given spanning tree	mst:M	13, {D}17.5	Yes [ ]
(mnlrn)	Not incorporating station location information to the Filtering Database in the Discarding state for a given spanning tree	mst:M	13, {D}17.5	Yes [ ]
(mrlrn)	Transfer learned MAC addresses from a retiring Root Port to a new Root Port for a given spanning tree	mst:O	13, {D}17.10	Yes [ ] No [ ]
(msm)	Instances of state machines per Bridge, per Port and per spanning tree instance, as specified	mst:M	13.19	Yes [ ]
(mptmr)	Port Timers state machine support	mst:M	13.21, 13.27	Yes [ ]
(mprsm)	Port Receive state machine support	mst:M	13.21, 13.28	Yes [ ]
(mmsm)	Port Protocol Migration state machine support	mst:M	13.21, 13.29	Yes [ ]
(mptsm)	Port Transmit state machine support	mst:M	13.21, 13.30	Yes [ ]
(mpism)	Port Information state machine support	mst:M	13.21, 13.31	Yes [ ]
(mprssm)	Port Role Selection state machine support	mst:M	13.21, 13.32	Yes [ ]
(mprtsm)	Port Role Transitions state machine support	mst:M	13.21, 13.33	Yes [ ]
(mpstsm)	Port State Transition state machine support	mst:M	13.21, 13.34	Yes [ ]
(mtcsm)	Topology Change state machine support	mst:M	13.21, 13.35	Yes [ ]
(mcde)	Not support Change Detection Enabled parameter	mst:M	{D}5.2	Yes [ ]
(mestm)	Not: Underestimate the increment to the Message Age parameter in transmitted BPDUs. Underestimate Forward Delay. Overestimate the Hello Time interval.	mst:M	13.35, {D}17.28.1	Yes [ ]
(mhtim)	Use of Transmission Limit	mst:M	13.35, {D}Table 17-5	Yes [ ]
(mprel)	Enforcement of parameter relationships	mst:M	13.35, {D}17.28.2	Yes [ ]
(mprv)	Range and granularity of priority values	mst:M	13.35, {D}17.28.2	Yes [ ]
(mpcv)	Range and granularity of path cost values	mst:M	13.35, {D}17.28.2	Yes [ ]
	If item (16) is not supported, mark N/A and continue at (mtmr1)			N/A [ ]

A.11 Multiple Spanning Tree Algorithm (*Continued*)

Item	Feature	Status	References	Support
(mmgt1)	Can the relative priority of the Bridge be set?	mst AND **16:**M	{D}17.2, {D}17.4, {D}17.13	Yes [ ]
(mmgt2)	Can the relative priority of the Ports be set?	mst AND **16:**M	{D}17.2, {D}17.4, {D}17.13	Yes [ ]
(mmgt3)	Can the path cost for each Port be set?	mst AND **16:**M	{D}17.2, {D}17.4, {D}17.13	Yes [ ]
	If item (17) is not supported, mark N/A and continue at (mpdu1)			N/A [ ]
*(mtmr1)	Can Bridge Max Age be set to any of the range of values specified?	mst AND **17:**M	{D}17.2, {D}17.13, {D}Table 17-5	Yes [ ]
(mtmr2)	Can Port Hello Time (13.22) be set to any of the range of values specified?	mst AND **17:**M	{D}17.2, {D}17.13, {D}Table 17-5	Yes [ ]
(mtmr3)	Can Bridge Forward Delay be set to any of the range of values specified?	mst AND **17:**M	{D}17.2, {D}17.13, {D}Table 17-5	Yes [ ]
*(mpdu1)	Do all BPDUs contain an integral number of octets?	mst:M	14.1.1	Yes [ ]
(mpdu2)	Are all the BPDU parameter types encoded as specified?	mst:M	14.2, {D}9.1.1, {D}9.2	Yes [ ]
(mpdu3)	Do Configuration BPDUs have the format, parameters, and parameter values specified?	mst:M	{D}9.3.1	Yes [ ]
(mpdu4)	Do Topology Change Notification BPDUs have the format, parameters, and parameter values specified?	mst:M	{D}9.3.2	Yes [ ]
(mpdu5)	Do Rapid Spanning Tree BPDUs have the format, parameters, and parameter values specified?	mst:M	{D}9.3.3	Yes [ ]
(mpdu6)	Do Multiple Spanning Tree BPDUs have the format, parameters, and parameter values specified?	mst:M	14.3	Yes [ ]
(mpdu7)	Are received BPDUs validated as specified?	mst:M	14.4, 14.6	Yes [ ]
(mpdu8)	Are Configuration, TCN, RST and MST BPDUs encoded and transmitted as specified?	mst:M	14.5, 14.6	Yes [ ]

A.12 Bridge Management

Item	Feature	Status	References	Support
	If item (19) is not supported, mark N/A and continue at (20c).			N/A [ ]
(19a)	Discover Bridge	**19:**M	12.4.1.1	Yes [ ]
(19b)	Read Bridge	**19:**M	12.4.1.2	Yes [ ]
(19c)	Set Bridge Name	**19:**M	12.4.1.3	Yes [ ]
(19d)	Reset Bridge	**19:**M	12.4.1.4	Yes [ ]
(19e)	Read Port	**19:**M	12.4.2.1	Yes [ ]
(19f)	Set Port Name	**19:**M	12.4.2.2	Yes [ ]
(19g)	Read Forwarding Port Counters	**19:**M	12.6.1.1	Yes [ ]
(19g.1)	Are the Forwarding Port Counters maintained per VLAN?	19:O		Yes [ ] No [ ]
(19g.2)	Does the implementation support the Discard on Error Details parameter?	19:O		Yes [ ] No[ ]
(19h)	Read Filtering Database	**19:**M	12.7.1.1	Yes [ ]
(19i)	Set Filtering Database Ageing Time	**19:**M	12.7.1.2	Yes [ ]
(19j)	Read Permanent Database	**19:**M	12.7.6.1	Yes [ ]
(19k)	Create Filtering Entry	**19:**M	12.7.7.1	Yes [ ]
(19l)	Delete Filtering Entry	**19:**M	12.7.7.2	Yes [ ]
(19m)	Read Filtering Entry	**19:**M	12.7.7.3	Yes [ ]
(19n)	Read Filtering Entry Range	**19:**M	12.7.7.4	Yes [ ]
(19o)	Read CIST Bridge Protocol Parameters	**19:**M	12.8.1.1	Yes [ ]
(m19o)	Read MSTI Bridge Protocol Parameters	19 AND mst:M	12.8.1.2	Yes [ ] N/A [ ]
(19p)	Set CIST Bridge Protocol Parameters	**19:**M	12.8.1.3	Yes [ ]
(m19p)	Set MSTI Bridge Protocol Parameters	**19** AND mst**:**M	12.8.1.4	Yes [ ] N/A [ ]
(19q)	Read CIST Port Parameters	**19:**M	12.8.2.1	Yes [ ]
(m19q)	Read MSTI Port Parameters	**19** AND mst**:**M	12.8.2.2	Yes [ ] N/A [ ]
(19r)	Force CIST Port State	**19:**M	12.8.2.3	Yes [ ]
(m19r)	Force MSTI Port State	**19** AND mst**:**M	12.8.2.4	Yes [ ] N/A [ ]
(19s)	Set CIST Port Parameters	**19:**M	12.8.2.5	Yes [ ]
(m19s)	Set MSTI Port Parameters	**19** AND mst**:**M	12.8.2.6	Yes [ ] N/A [ ]
(migr)	Force BPDU Migration Check	19 AND (rst OR mst):M	12.8.2.7	Yes [ ] N/A [ ]

A.12 Bridge Management (*Continued*)

Item	Feature	Status	References	Support
(19t)	Read Port Default User Priority	MS4:M	12.6.2.1	Yes [ ] N/A [ ]
(19u)	Set Port Default User Priority	MS4:M	12.6.2.2	Yes [ ] N/A [ ]
(19v)	Read Port User Priority Regeneration Table	MS5:M	12.6.2.3	Yes [ ] N/A [ ]
(19w)	Set Port User Priority Regeneration Table	MS5:M	12.6.2.4	Yes [ ] N/A [ ]
(19x)	Read Port Traffic Class Table	MS7:M	12.6.3.1	Yes [ ] N/A [ ]
(19y)	Set Port Traffic Class Table	MS7:M	12.6.3.2	Yes [ ] N/A [ ]
(19z)	Read Outbound Access Priority Table	MS6:M	12.6.2.5	Yes [ ] N/A [ ]
(19aa)	Read GARP Timers	MS8:M	12.9.1.1	Yes [ ] N/A [ ]
(19ab)	Set GARP Timers	MS8:M	12.9.1.2	Yes [ ] N/A [ ]
(19ac)	Read GARP Protocol Controls	MS8:M	12.9.2.1	Yes [ ] N/A [ ]
(19ad)	Set GARP Protocol Controls	MS8:M	12.9.2.2	Yes [ ] N/A [ ]
(19ae)	Read GARP State	MS8:M	12.9.3.1	Yes [ ] N/A [ ]
(19af)	Read Bridge VLAN Configuration	19:M	12.10.1.1	Yes [ ] N/A [ ]
(19ah)	Configure PVID values	19:M	12.10.1.2	Yes [ ] N/A [ ]
(19ai)	Configure Acceptable Frame Types parameter	23a.2:M	12.10.1.3	Yes [ ] N/A [ ]
(19aj)	Configure Enable Ingress Filtering parameters	23g:M	12.10.1.4	Yes [ ] N/A [ ]
(19ak)	Reset Bridge VLAN Bridge.	19:M	12.10.1.5	Yes [ ] N/A [ ]
(19al)	Notify VLAN Registration Failure	19:M	12.10.1.6	Yes [ ] N/A [ ]
(19am)	Read VLAN Configuration	19:M	12.10.2.1	Yes [ ] N/A [ ]
(19an)	Create VLAN Configuration	19:M	12.10.2.2	Yes [ ] N/A [ ]
(19ao)	Delete VLAN Configuration	19:M	12.10.2.3	Yes [ ] N/A [ ]
	If item (mst) is not supported, mark N/A and continue at (19ap)			N/A [ ]
(rmst)	Read MSTI List	19 AND mst:M	12.12.1.1	Yes [ ]
(cmst)	Create MSTI	19 AND mst:M	12.12.1.2	Yes [ ]
(dmst)	Delete MSTI	19 AND mst:M	12.12.1.3	Yes [ ]
(rfmst)	Read FID to MSTI allocation	19 AND mst:M	12.12.2.1	Yes [ ]
(sfmst)	Set FID to MSTI allocation	19 AND mst:M	12.12.2.2	Yes [ ]
(rmste)	Read MST Configuration Table Element	19 AND mst:M	12.12.3.1	Yes [ ]
(rmstv)	Read VIDs assigned to MSTID	19 AND mst:M	12.12.3.2	Yes [ ]

A.12 Bridge Management (*Continued*)

Item	Feature	Status	References	Support
(rmstc)	Read MSTI Configuration Identifier	19 AND mst:M	12.12.3.3	Yes [ ]
(smstc)	Set MSTI Configuration Identifier	19 AND mst:M	12.12.3.4	Yes [ ]
	If Item (23e.6) is not supported, mark N/A and continue at Item (19at).			N/A
(19ap)	Read VLAN Learning Constraints	23e.6:M	12.10.3.1	Yes [ ]
(19aq)	Read VLAN Learning Constraints for VID	23e.6:M	12.10.3.2	Yes [ ]
(19aq)	Set VLAN Learning Constraint	23e.6:M	12.10.3.3	Yes [ ]
(19ar)	Delete VLAN Learning Constraint	23e.6:M	12.10.3.4	Yes [ ]
(19as)	Notify Learning Constraint Violation	23e.6:M	12.10.3.10	Yes [ ]
	If Item (23e.8) is not supported, mark N/A and continue at Item (20c).			N/A
(19at)	Read VID to FID allocations	23e.8:M	12.10.3.5	Yes [ ]
	Read FID allocation for VID	23e.8:M	12.10.3.6	Yes [ ]
	Read VIDs allocated to FID	23e.8:M	12.10.3.7	Yes [ ]
	Set VID to FID allocation	23e.8:M	12.10.3.8	Yes [ ]
	Delete VID to FID allocation	23e.8:M	12.10.3.9	Yes [ ]
	If item (20a) is not supported, mark N/A and continue at (20e).			N/A [ ]
(20c)	What Management Protocol standard(s) or specification(s) are supported?	**20a:**M	{D}5	
(20d)	What standard(s) or specifications for Managed Objects and Encodings are supported?	**20a:**M	{D}5	
	If item (20b) is not supported, mark N/A and continue at A.11.			N/A [ ]
(20e)	What specification of the local management interface is supported?	**20b:**M	{D}5	

Predicates:
MS4=19 AND 4a
MS5=19 AND 4b
MS6=19 AND 4
MS7=19 AND 4c
MS8=19 AND 2b

A.13 Performance

Item	Feature	Status	References	Support
(21a)	Specify a Guaranteed Port Filtering Rate, and the associated measurement interval *TF*, for each Bridge Port in the format specified below.	M	{D}16.1	
(21b)	Specify a Guaranteed Bridge Relaying Rate, and the associated measurement interval *TR*, in the format specified below. Supplementary information shall clearly identify the Ports.	M	{D}16.2	

Guaranteed Bridge Relaying Rate	TR
_ _ _ _ _ _ _ _ _ _ frames per second	_ _ _ _ _ _ second(s)

Port number(s) or other identification	Guaranteed port filtering rate (specify for all ports)	T_F (specify for all ports)
	_ _ _ _ _ _ _ _ _ _ frames per second	_ _ _ _ _ _ second(s)
	_ _ _ _ _ _ _ _ _ _ frames per second	_ _ _ _ _ _ second(s)
	_ _ _ _ _ _ _ _ _ _ frames per second	_ _ _ _ _ _ second(s)
	_ _ _ _ _ _ _ _ _ _ frames per second	_ _ _ _ _ _ second(s)
	_ _ _ _ _ _ _ _ _ _ frames per second	_ _ _ _ _ _ second(s)
	_ _ _ _ _ _ _ _ _ _ frames per second	_ _ _ _ _ _ second(s)
	_ _ _ _ _ _ _ _ _ _ frames per second	_ _ _ _ _ _ second(s)
	_ _ _ _ _ _ _ _ _ _ frames per second	_ _ _ _ _ _ second(s)

A.14 GARP and GMRP

Item	Feature	Status	References	Support
	If Item 2b is not supported, mark N/A and continue at item (22i).			N/A [ ]
(22a)	Is the GMRP Application address used as the destination MAC Address in all GMRP protocol exchanges?	2b:M	{D}10.4.1, {D}Table 12-1	Yes [ ]
(22b)	Are GMRP protocol exchanges achieved by means of LLC Type 1 procedures, using the LLC address for Spanning Tree protocol?	2b:M	{D}12.4, {D}12.5, {D}Table 7-8	Yes [ ]

A.14 GARP and GMRP (*Continued*)

Item	Feature	Status	References	Support
(22c)	Are GMRP protocol exchanges achieved using the GARP PDU formats, and the definition of the attribute type and value encodings defined for GMRP?	2b:M	10, {D}10.3.1, {D}12.4, {D}12.5, {D}12.11	Yes [ ]
(22d)	Does the implementation support the operation of the Applicant, Registrar, and Leave All state machines?	2b:M	{D}12.8	Yes [ ]
(22e)	Does the Bridge propagate registration GMRP information only on Ports that are part of the active topology of the GIP Context for the VLAN on which the registration was received?	2b:M	10, {D}12.3.3, {D}12.3.4	Yes [ ]
(22f)	Are GARP PDUs received on Ports that are in the Forwarding State forwarded, filtered or discarded in accordance with the requirements for handling GARP Application addresses?	2b:M	{D}7.12.3, {D}12.5	Yes [ ]
(22g)	Does the GMRP application operate as defined in Clause 10 of IEEE Std 802.1D, 1998 Edition, as modified by Clause 10 of this standard?	2b:M	10, {D}10, {D}10.3	Yes [ ]
(22h)	Are received GARP PDUs that are not well formed for any GARP Applications supported, discarded?	2b:M	10, {D}10.3.1, {D}12.4, {D}12.5, {D}12.10, {D}12.11	Yes [ ]
(22i)	Does the implementation support the use of the Restricted Group Registration parameter for each Port?	2b:O	5.2, {D}10.3.2	Yes [ ] No [ ]
(22j)	Are all GARP PDUs that are (a) Received on Ports that are in the Forwarding State, and are (b) Destined for GARP applications that the Bridge does not support, forwarded on all other Ports that are in Forwarding?	M	8.14.3, {D}12.5	Yes [ ]
(22k)	Are any GARP PDUs that are (a) Received on any Port, and (b) Destined for GARP applications that the Bridge does not support, submitted to any GARP Participants?	X	8.14.3, {D}12.5	No [ ]
(22l)	Are any GARP PDUs that are (a) Received on any Ports that are not in the Forwarding State, and are (b) Destined for GARP applications that the Bridge does not support, forwarded on any other Ports of the Bridge?	X	8.14.3, {D}12.5	No [ ]

A.14 GARP and GMRP (*Continued*)

Item	Feature	Status	References	Support
(22m)	Are any GARP PDUs that are (a) Received on any Ports that are in the Forwarding State, and are (b) Destined for GARP applications that the Bridge supports, forwarded on any other Ports of the Bridge?	X	8.14.3, {D}12.5	No [ ]
(22n)	Are all GARP PDUs that are: (a) Received on any Port, and (b) Destined for GARP applications that the Bridge supports, submitted to the appropriate GARP Participants?	M	8.14.3, {D}12.5	Yes [ ]
22o	Are all GARP PDUs received on disabled Ports discarded?	M	{D}12.2	Yes [ ]

A.15 VLAN support

Item	Feature	Status	References	Support	
	Ingress rules				
(24a)	Can the PVID or the VID in any member of the VID Set for any Port be assigned the value of the null VLAN ID?	X	8.4.4, Table 9-2	No [ ]	
(24b)	Are frames discarded (or not discarded) in accordance with the settings of the Acceptable Frame Types parameters?	M	8.6.1	Yes [ ]	
(24c)	Are all frames received classified as belonging to exactly one VLAN, as defined in the ingress rules?	M	8.6.1	Yes [ ]	
(24d)	Is Ingress Filtering performed in accordance with the value of the Enable Ingress Filtering parameter?	M	8.6.1	Yes [ ]	
(24e)	Are all frames that are not discarded as a result of the application of the ingress rules submitted to the Forwarding Process and to the Learning Process?	M	8.6.1	Yes [ ]	
(24f)	State which Ports support Port-and-Protocol-based classification rules.	23l:M	8.4.4	Ports:	_____
(24f.1)	For each Port that supports Port-and-Protocol-based classification rules, is a VID Set supported?	23l:M	8.4.4	Port: Yes [ ]	_____ N/A [ ]
(24f.2)	For each Port that supports Port-and-Protocol-based classification rules, state how many entries are supported in the VID Set.	23l:M	8.4.4	Port: _____	_____ Entries
(24f.3)	For each Port that supports Port-and-Protocol-based classification rules, is the VID Set configurable via management?	23l:M	12.10.1.2	Port: Yes [ ]	_____ N/A [ ]

A.15 VLAN support (*Continued*)

Item	Feature	Status	References	Support	
(24g.1)	State how many entries are supported in the Protocol Group Database.	23m:M	8.9.4	Entries	_____
(24g.2)	Is the Protocol Group Database configurable via management?	23m:O	12.10.2.1	Yes [ ]	No [ ]
(24g.3)	Does the Protocol Group Database support entries of format Ethernet?	23m:O	8.9.4	Yes [ ]	No [ ]
(24g.4)	Does the Protocol Group Database support entries of format RFC_1042?	23m:O	8.9.4	Yes [ ]	No [ ]
(24g.5)	Does the Protocol Group Database support entries of format SNAP_8021H?	23m:O	8.9.4	Yes [ ]	No [ ]
(24g.6)	Does the Protocol Group Database support entries of format SNAP_Other?	23m:O	8.9.4	Yes [ ]	No [ ]
(24g.7)	Does the Protocol Group Database support entries of format LLC_Other?	23m:O	8.9.4	Yes [ ]	No [ ]
(24g.8)	Does the Protocol Group Database support entries of at least one of the following formats: Ethernet, RFC_1042, SNAP_8021H, SNAP_Other, LLC_Other?	23m: M	8.9.4	Yes [ ]	
	Egress rules				
(25a)	Are frames discarded if the transmission Port is not present in the Member set for the frame’s VID?	M	8.6.4, 8.10.9	Yes [ ]	
(25b)	Are frames discarded if the value of the include_tag parameter is False, and the Bridge does not support the ability to translate embedded MAC Address information from the format indicated by the canonical_format_indicator parameter to the format appropriate to the media type on which the data request will be carried?	23j.2:M	8.6.4	Yes [ ]	N/A [ ]
(25c)	Are frames transmitted as VLAN-tagged frames or as untagged frames in accordance with the value of the untagged set for the frame’s VID?	M	8.6.4	Yes [ ]	
	Filtering Database				
(26a)	Does the implementation support Static VLAN Registration Entries as defined in 8.11.2?	M	8.10.2	Yes [ ]	
(26b)	Does the implementation support the creation of a separate Static VLAN Registration Entry with a distinct Port Map for each VLAN from which frames are received by the Forwarding Process?	O	8.10.2	Yes [ ]	No [ ]
(26c)	Does the implementation support Dynamic VLAN Registration Entries as defined in 8.11.5?	M	8.10.5	Yes [ ]	

A.15 VLAN support (*Continued*)

Item	Feature	Status	References	Support
(26d)	Does the implementation support the creation of a separate Dynamic VLAN Registration Entry with a distinct Port Map for each VLAN from which frames are received by the Forwarding Process?	O	8.10.5	Yes [ ] No [ ]
(26e)	Does the implementation allocate VIDs to FIDs in accordance with the specification in 8.11.7?	M	8.10.7, 8.10.7.2	Yes [ ]
(26f)	Does the implementation correctly detect Learning Constraint violations?	M	8.10.7.3	Yes [ ]
(26g)	Is determination of the Member set and the untagged set for a given VLAN achieved as defined in 8.11.9?	M	8.10.9	Yes [ ]
	Tagged frames			
(27a)	Do VLAN-tagged frames transmitted by the Bridge conform to the format defined in Clause 9 for the MAC type on which they are transmitted?	M	9	Yes [ ]
(27b)	Are all BPDUs transmitted untagged?	M	8.14.7	Yes [ ]
	VLAN use of GMRP. If item (2b) is not supported, mark N/A and continue at item (29a).			N/A [ ]
(28a)	Does the implementation of GMRP recognize the use of VLAN Contexts for the transmission and reception of GMRP PDUs?	2b:M	10, 10.1, 10.2 10.3	Yes [ ]
(28b)	Does the implementation of GMRP support the creation of distinct GMRP Participants for each VLAN context?	2b:M	10.2	Yes [ ]
(28c)	Does the implementation support the identification of VLAN contexts in transmitted GMRP PDUs by means of VLAN-tagged or untagged frames, in accordance with the member set and untagged set for the VLAN Context concerned?	2b:M	10.3	Yes [ ]
(28d)	Are GMRP PDUs transmitted only on Ports that are part of the active topology for the VLAN Context concerned?	2b:M	10.1	Yes [ ]
	VLAN Topology Management			
(29a)	Does the implementation support the creation, updating and removal of Dynamic VLAN Registration Entries in the Filtering Database under the control of GVRP?	M	11	Yes [ ]
(29b)	Does the Permanent Database contain an entry for the Default VID that defines Registration Fixed on all Ports?	O	11.2.1.3	Yes [ ] No [ ]
(29c)	Is the GVRP Application address used as the destination MAC Address in all GVRP protocol exchanges?	M	11, Table 11-1	Yes [ ]

A.15 VLAN support (*Continued*)

Item	Feature	Status	References	Support
(29d)	Are GVRP protocol exchanges achieved by means of LLC Type 1 procedures, using the LLC address for Spanning Tree protocol?	M	11, {D}12.4, {D}12.5, {D}Table 7-8	Yes [ ]
(29e)	Are GVRP protocol exchanges achieved using the GARP PDU formats, and the definition of the attribute type and value encodings defined for GVRP?	M	11, 11.2.3.1, {D}12.4, {D}12.5, {D}12.11	Yes [ ]
(29f)	Does the implementation support the operation of the Applicant, Registrar, and Leave All state machines?	M	{D}12.8	Yes [ ]
(29g)	Does the Bridge propagate registration GVRP information only on Ports that are part of the active topology of the base Spanning Tree Context?	M	11, {D}12.3.3, {D}12.3.4	Yes [ ]
(29h)	Does the GVRP application operate as defined in Clause 11?	M	11	Yes [ ]
(29i)	Does the implementation support the use of the Restricted VLAN Registration parameter?	O	5.2, 11.2.3.2.2, 11.2.3.2.3	Yes [ ] No [ ]

Annex B

(informative)

Shared and Independent VLAN Learning

This standard provides for a variety of approaches to the implementation of VLAN Bridges from the point of view of the way that individual MAC Addresses are learned, and how that learned information is used in subsequent forwarding/filtering decisions. There are two mechanisms that are used as a basis for these variants:

a) Making use of address information learned across a number of VLANs in order to make learning decisions relative to any one of those VLANs. This is referred to as *Shared VLAN Learning* (SVL, 3.9);

b) Making use of address information learned in one VLAN only in order to make learning decisions relative to that VLAN, and ensuring that it is not used in learning decisions relative to any other VLAN. This is referred to as *Independent VLAN Learning* (IVL, 3.5).

These mechanisms lead to the SVL/IVL model for how a VLAN Bridge implements learning and filtering for MAC Addresses. Using the terminology of 8.11.7, an SVL/IVL Bridge supports multiple FIDs (which effectively equates to supporting multiple Filtering Databases), and multiple VLANs can use each FID. By varying the number of FIDs supported, and the number of VLANs that can share each FID, the following simplifications of the SVL/IVL model can be created:

c) *Shared VLAN Learning (SVL) only.* The implementation supports only one FID, so all VLANs share the same learned MAC Address information, regardless of which VLAN the information was learned in;

d) *Independent VLAN Learning (IVL) only.* Multiple FIDs are supported, but each FID can support only one VID, so each VLAN makes use only of MAC Address information learned within that VLAN.

All three approaches are permitted by this standard, and each has advantages in particular circumstances. The remainder of this annex discusses

e) The requirements for Independent VLAN Learning, Shared VLAN Learning, or both;

f) How Bridges are made aware of the requirement for particular VLANs to be “shared” or “independent”;

g) How Bridges based on one of these models can interoperate with Bridges based on a different model, in the same Bridged LAN.

B.1 Requirements for Shared and Independent Learning

Under most circumstances, the SVL and IVL approaches work equally well, and Bridges adopting either approach can be freely intermixed within a Bridged LAN. There are, however, a small number of configuration cases where, in order to prevent undue flooding of unicast frames, and in some cases, to make communication between the affected end systems possible, it is necessary to make specific choices as to how Bridges that adopt these different learning models are deployed in a Bridged LAN. The following subclauses give examples of some of these configurations, and also provide a generic statement of the requirements that must be met in order for each learning model to be successfully deployed.

B.1.1 Connecting independent VLANs

Figure B-1 illustrates how a device that connects two VLANs together, and which therefore itself shares learning between those VLANs, creates a need for those VLANs to be independent in other Bridges.

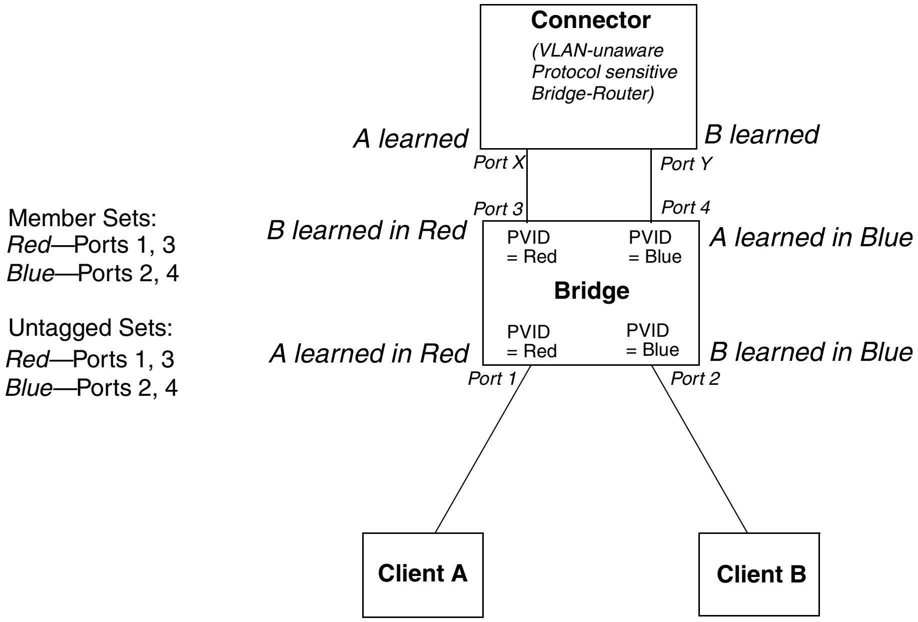


Figure B-1—Connecting independent VLANs—1

Clients A and B are connected via the protocol sensitive Bridge-Router (Connector), with an intervening VLAN-aware Bridge. The fact that all the Ports of the Bridge carry untagged traffic neatly conceals the fact that the Connector has the effect of connecting VLANs Red and Blue together with regard to bridged traffic. The Connector itself learns A and B in the same database, as it has no knowledge of VLANs Red and Blue. This prevents any traffic transmitted on the Red VLAN (Port X of the Connector) that is destined for A, from being bridged to Port Y and transmitted on the Blue VLAN.

The VLAN-aware Bridge must keep its learning separate for Red and Blue; otherwise, the addresses of the two clients would be alternately learned on diagonally opposite Ports as, for example, traffic sourced by A reenters the Bridge on Port 4 having previously been seen on Port 1.

NOTE—This example assumes that Spanning Tree is disabled in the Connector, so that the VLAN-aware Bridge does not attempt to suppress the loop that apparently exists if VLANs are not taken into account.

A simpler example can be constructed, with a single Port connecting the Connector and the VLAN-aware Bridge, if the Connector is itself VLAN-aware and transmits and receives only VLAN-tagged traffic. In this case, the Connector would allocate a single FID for use by Red and Blue. This is shown in Figure B-2.

B.1.2 Duplicate MAC Addresses

The simplest example of a need for Independent VLAN Learning occurs where two (or more) distinct devices in different parts of the network reuse the same individual MAC Address, or where a single device is connected to multiple LAN segments, and all of its LAN interfaces use the same individual MAC Address. This is shown in Figure B-3.

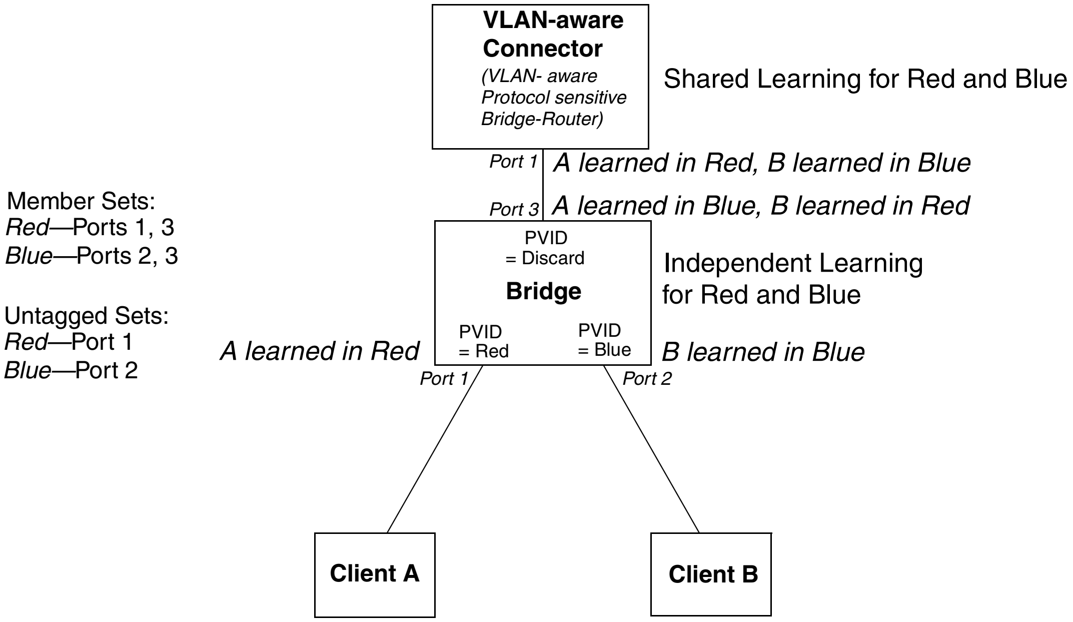


Figure B-2—Connecting independent VLANs—2

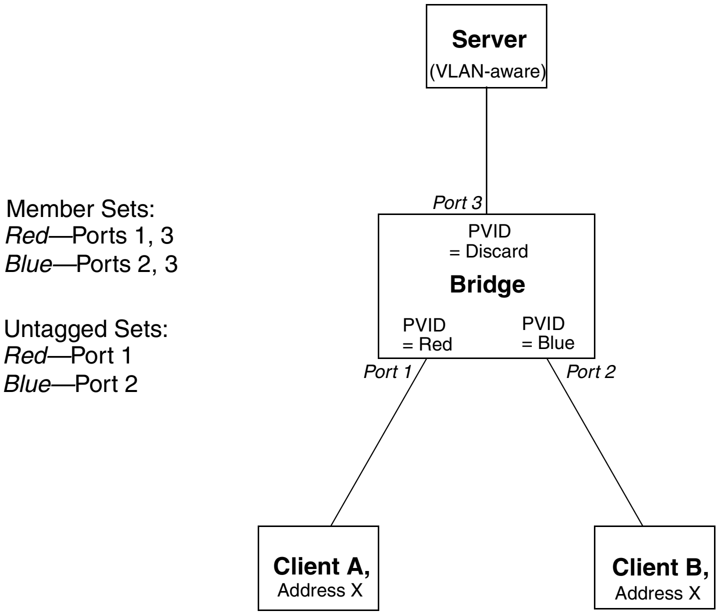


Figure B-3—Duplicate MAC Addresses

The example shows two clients with access to the same server; both clients are using the same individual MAC Address, X. If the Bridge shares learning between VLAN Red (which serves Client A) and VLAN Blue (which serves Client B), i.e., the Bridge uses the same FID for both VLANs, then Address X will appear to move between Ports 1 and 2 of the Bridge, depending upon which client has most recently transmitted a frame. Communication between these Clients and the server will therefore be seriously disrupted. Assignment of distinct FIDs for Red and Blue ensures that communication can take place correctly.

Hence, in order to construct this particular VLAN configuration, either an IVL Bridge or an SVL/IVL Bridge would be required.

B.1.3 Asymmetric VLANs

A primary example of the requirement for Shared VLAN Learning is found in "asymmetric" uses of VLANs. Under normal circumstances, a pair of devices communicating in a VLAN environment will both send and receive using the same VLAN; however, there are some circumstances in which it is convenient to make use of two distinct VLANs, one used for A to transmit to B: the other used for B to transmit to A. An example of such an application of VLANs is shown in Figure 11-1. Note that:

a) In the example, the server and both clients are assumed to be VLAN-unaware devices, i.e., they transmit and receive untagged frames only;
b) The ingress classification rules assumed by the example are as defined in this standard, i.e., Port-based classification only;
c) The configuration shown can only be achieved by management configuration of appropriate values in Static VLAN Registration Entries (8.11.9) in order to configure the indicated member sets and untagged sets.

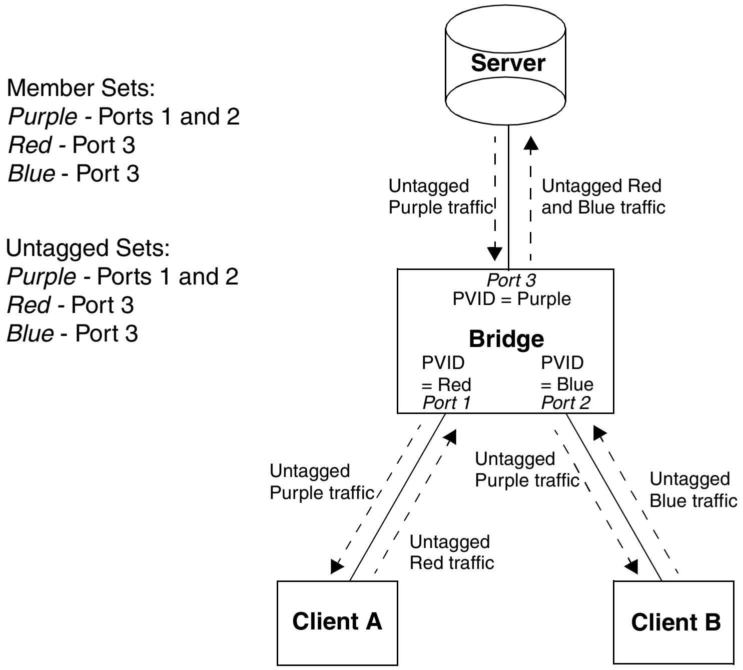


Figure B-4—Asymmetric VLAN use: "multi-netted server"

In the example, Port-based tagging and an asymmetric VLAN configuration is used in order to permit Clients A and B access to a common server, but to prohibit Clients A and B from talking to each other. Examples of where this type of configuration might be required are if the clients are on distinct IP subnets, or if there is some confidentiality-related need to segregate traffic between the clients.

Client A transmits to the server via Port 1, which will classify this traffic as belonging to VLAN Red; the Bridge therefore learns Client A's MAC Address on Port 1 in VLAN Red. The Server transmits its responses to Client A via Port 3, which classifies the return traffic as belonging to VLAN Purple. If individual MAC Address learning is configured in the Bridge such that learning is independent between Red and Purple (Red and Purple are allocated to distinct FIDs), then the Bridge will have no knowledge of A in VLAN Purple, and will therefore flood the server's responses to Client A on both Port 1 and Port 2. Conversely, if Red and Purple are defined to share the same FID, then the address information learned in Red will be available for use in forwarding the Purple traffic, and the responses to Client A are forwarded only through Port 1.

Similarly, there is a need in this configuration for Blue and Purple to share learning information; hence, in order for this configuration to achieve its objectives, the Red, Blue, and Purple VIDs must be allocated to the same FID in the Bridge.

Hence, in order to construct this particular VLAN configuration, either an SVL Bridge or an SVL/IVL Bridge would be required.

NOTE—The example has been deliberately simplified; in practical applications, the central Bridge would likely be replaced by a number of VLAN-aware Bridges, interconnected with links that would carry the traffic between clients and server as VLAN-tagged frames, with VLAN-tagging and untagging occurring only at the "edge" Ports of the Bridged LAN. An alternative approach to the one described here could also be achieved either by using a VLAN-aware server, or by use of more sophisticated Ingress classification rules.

B.1.4 Generic constraints on SVL and IVL use

This subclause describes the general constraints on the mapping of VLANs to FIDs, from the point of view of a given Bridge that learns from or forwards frames on a set of VLANs (the Bridge's "active set" of VLANs). If

a) The individual MAC Addresses associated with each point of attachment to the active set of VLANs are unique (i.e., the "Duplicate MAC Address problem" is not present), and
b) There is no Bridge or Bridge-like device that takes frames from one VLAN in the active set and subsequently transmits them on another VLAN in the active set, then

every VLAN in the active set may share the same FID; in other words, individual MAC Address information learned in any one VLAN may be used in forwarding decisions taken relative to any of the others, so the SVL approach can be used in that Bridge.

Further, if

c) Each bidirectional, individual MAC-Addressed, conversation between pairs of end stations makes use of the same VLAN (ID) in both directions, then

every VLAN in the active set may be allocated a distinct FID (with the possibility of a little extra flooding as learning of addresses in one VLAN does not contribute to forwarding decisions for that address in any other VLAN). Under these circumstances, rule b) may also be relaxed and restated as follows:

d) Frames on one VLAN in the active set may be received by (up to) one Bridge and transmitted on another VLAN in the active set, provided that there is no loop in such VLAN to VLAN forwarding, e.g., for a set of VLANs Red, Blue, Green,...etc., there is no logical loop in copying frames between VLANs, such as copying from Red to Green by one Bridge, Green to Blue by another, and Blue back to Red by a third.

So

e) If rules a), b), and c) are true, and d) is false, for all VLANs in the active set, then either an SVL or an IVL Bridge can be deployed;
f) If rules a) and b) are true, and c) and d) are false for all VLANs in the active set, then only an SVL Bridge can be deployed;

g) If rules a) or b) or d) are false, and c) is true for all VLANs in the active set, then only an IVL Bridge can be deployed.

The above conditions are all on the basis that they apply "for all VLANs in the active set." Clearly, in more complex scenarios, some VLANs in the active set will have requirements that dictate SVL behavior on the part of a given Bridge, while others will have requirements that dictate IVL behavior. Under such circumstances, an SVL/IVL Bridge is required, allowing those VLANs that need to be shared to be mapped to a single FID, while those that need to be independent are mapped to distinct FIDs. Needless to say, wherever an SVL or IVL Bridge can be deployed, it can successfully be replaced by an appropriately configured SVL/IVL Bridge.

B.2 Configuring the Global VLAN Learning Constraints

Clause B.1 described the requirements that exist for the two approaches to learning in VLAN Bridges, closing with some generic rules for how to determine whether, for a given Bridge, SVL, IVL, or SVL/IVL can be successfully deployed. In VLAN Bridges, the set of requirements for Independent and/or Shared VLAN Learning is configured as a set of global VLAN Learning Constraint specifications, using the management tools defined in 12.10.3. Two types of VLAN Learning Constraint are defined in 8.11.7.2, which also defines how the set of constraints is used in order to derive a valid mapping of VIDs to FIDs.

The constraint specifications can be constructed on a modular basis. For example, the configuration shown in Figure 11-1 has a requirement for Shared VLAN Learning between VLANs Red, Blue, and Purple. This could be expressed as follows:

{Red S Purple};
{Blue S Purple}

with {Red S Blue} being implied by the transitive nature of the S Constraint.

If we add a similar server access configuration in the same network that requires Red to share with Yellow and Orange, then this could be expressed as

{Red S Yellow};
{Orange S Yellow}

with {Red S Orange}, {Yellow S Blue}, {Yellow S Purple}, {Orange S Blue}, and {Orange S Purple} being implied by the transitive nature of the S Constraint.

Hence, Red, Blue, Purple, Yellow and Orange are all required to map to the same FID in order for the set of S Constraints (both explicit and implied) to be met. The constraints that express that requirement are built up from their constituent requirements; namely, for Red and Blue to share with Purple to meet one configuration need, and for Red to share with Yellow and Orange to meet another.

NOTE 1—The five VLANs in this example can be viewed as forming a Shared Set; i.e., a set of VLANs that have a mutual requirement to share learned information—all members of a Shared Set must map to the same FID. Any sequence of S Constraints defines one or more such Shared Sets. Any two Shared Sets can also map to the same FID as long as, for any pair of VLANs, one selected from each Shared Set, there are no I Constraints that require that pair of VLANs to learn independently. Hence, if there are only S Constraints defined, then all VLANs can be mapped to a single FID.

Similarly, the I Constraints can be added on a modular basis. Continuing from the above example, a Bridge-Router (Figure B-1) might be present in the Bridged LAN, which has the effect of connecting VLANs Indigo and Green together, thus creating a requirement for Indigo and Green to be independent. This could be expressed as

{Indigo I 1};
{Green I 1}

A separate independence requirement might be imposed by the fact that three stations, attached to Indigo, Vermilion, and Red VLANs, all make use of the same individual MAC Address. This could be expressed as:

{Indigo I 2};
{Vermilion I 2};
{Red I 2}

Hence, {Indigo, Vermilion, Red} have to be mutually independent (assigned to distinct FIDs), {Indigo, Green} have to be mutually independent, and {Red, Blue, Purple, Yellow, Orange} have to be shared.

The minimum number of FIDs required to satisfy this total constraint specification is three, e.g.:

FID A: Red, Blue, Purple, Yellow, Orange
FID B: Indigo
FID C: Green, Vermilion

although an equally valid allocation for 3 FIDs is

FID A: Green, Red, Blue, Purple, Yellow, Orange
FID B: Indigo
FID C: Vermilion

and an equally valid allocation, using the maximum number of FIDs that could be used for this set of constraints and VLANS is:

FID A: Red, Blue, Purple, Yellow, Orange
FID B: Indigo
FID C: Green
FID D: Vermilion

NOTE 2—It can clearly be seen from this example that it is possible to add further constraints that result in impossible VID to FID mappings; for example, if we were to add {Indigo S Red} or ({Yellow I 3}, {Blue I 3}), then the result is at least one pair of VLANs that have a requirement to both share the same FID and to use distinct FIDs at the same time. Such configurations are examples of Learning Constraint inconsistencies (8.11.7.3).

The assumption behind these constraint specifications is that they are applied globally, in the sense that all VLAN Bridges in a given Bridged LAN are configured with the same set of constraints. This is important, in order to ensure that each Bridge is in a position to determine whether or not, given its current active set of VLANs, it is capable of adopting a VID to FID mapping that will satisfy the specified constraints. If it cannot achieve such a mapping (for any of the reasons identified in 8.11.7.3), then it has detected a network misconfiguration that can only be resolved by management intervention. The managed object specification 12.10.3 provides a Notification for use in such circumstances, to alert a management station to the existence of the problem.

B.3 Interoperability

If the configuration of the Bridged LAN is such that it is not necessary to configure any VLAN Learning Constraints into the Bridges, i.e.:

a) There are no instances where two (or more) points of attachment to different LAN segments (and different VLANs) make use of the same individual MAC Address;
b) There are no instances where a Bridge receives frames on one VLAN and transmits them on another VLAN;
c) There is no asymmetric VLAN use, i.e., there is no pair of end stations for which bidirectional, unicast conversations make use of different VLANs for each direction of transmission,

then it is possible to freely intermix SVL, IVL, and SVL/IVL Bridges in that Bridged LAN, and they can all successfully interoperate.

If the configuration of the Bridged LAN requires one or more S Constraints (and no I Constraints) to be configured into the Bridges, then SVL and SVL/IVL Bridges can be used freely; however, IVL Bridges may only be used in locations where their active set of VLANs does not include any pair of VLANs for which an S Constraint (either explicit or implied) has been defined.

If the configuration of the Bridged LAN requires two or more I Constraints (and no S Constraints) to be configured into the Bridges, then IVL and SVL/IVL Bridges can be used freely; however, SVL Bridges may only be used in locations where their active set of VLANs does not include any pair of VLANs for which I Constraints with the same Independent Set Identifier have been defined.

If the configuration of the Bridged LAN requires both I Constraints and S Constraints to be configured into the Bridges, then SVL/IVL Bridges can be used freely; however,

d) SVL Bridges may only be used in locations where their active set of VLANs does not include any pair of VLANs for which I Constraints with the same Independent Set Identifier have been defined, and
e) IVL Bridges may only be used in locations where their active set of VLANs does not include any pair of VLANs for which an S Constraint (either explicit, or implied) has been defined.

Annex C

(informative)

MAC method dependent aspects of VLAN support

This annex examines the set of services, frame formats, and MAC methods involved in the provision of VLAN services across IEEE 802 LANs using different MAC methods, and the mapping/bridging functions necessary for that provision.

C.1 The variables

End station MAC Service users make use of MAC data transmission services that convey the following types of information:

a) Ethernet Type-encoded (E) and LLC-encoded (L) information (see 3.1 and 3.2);
b) Frames (either E or L) in which any MAC Addresses embedded in the MAC data are carried in Canonical (C) or Non-canonical (N) format;

NOTE 1—The terms *Canonical format* and *Non-canonical format* are described in Annex F.

c) Frames that carry source-routing information (R), or frames that are Bridged transparently (T).

Hence, there are potentially eight combinations of these variables, corresponding to eight distinct services, as follows:

d) E-C-T (Ethernet Type-encoded, Canonical, transparent),
e) E-C-R (Ethernet Type-encoded, Canonical, source-routed),
f) E-N-T (Ethernet Type-encoded, Non-canonical, transparent),
g) E-N-R (Ethernet Type-encoded, Non-canonical, source-routed),
h) L-C-T (LLC-encoded, Canonical, transparent),
i) L-C-R (LLC-encoded, Canonical, source-routed),
j) L-N-T (LLC-encoded, Non-canonical, transparent),
k) L-N-R (LLC-encoded, Non-canonical, source-routed),

These services are supported over two basic LAN types:

l) 802.3/Ethernet (C);
m) Token Ring/FDDI (R).

There are two VLAN environments involved:

n) Untagged frames (U);
o) Tagged frames (T).

This leads to a total of 32 potential frame/encapsulation formats to consider (8 X 2 X 2). There are 96 possible one-way heterogeneous bridging functions between these various LAN/VLAN environments; 48 symmetrical (2-way) functions.

The combination of services and environments is illustrated in Figure 9-3; italics indicate services that have no untagged representation on the MAC method concerned.

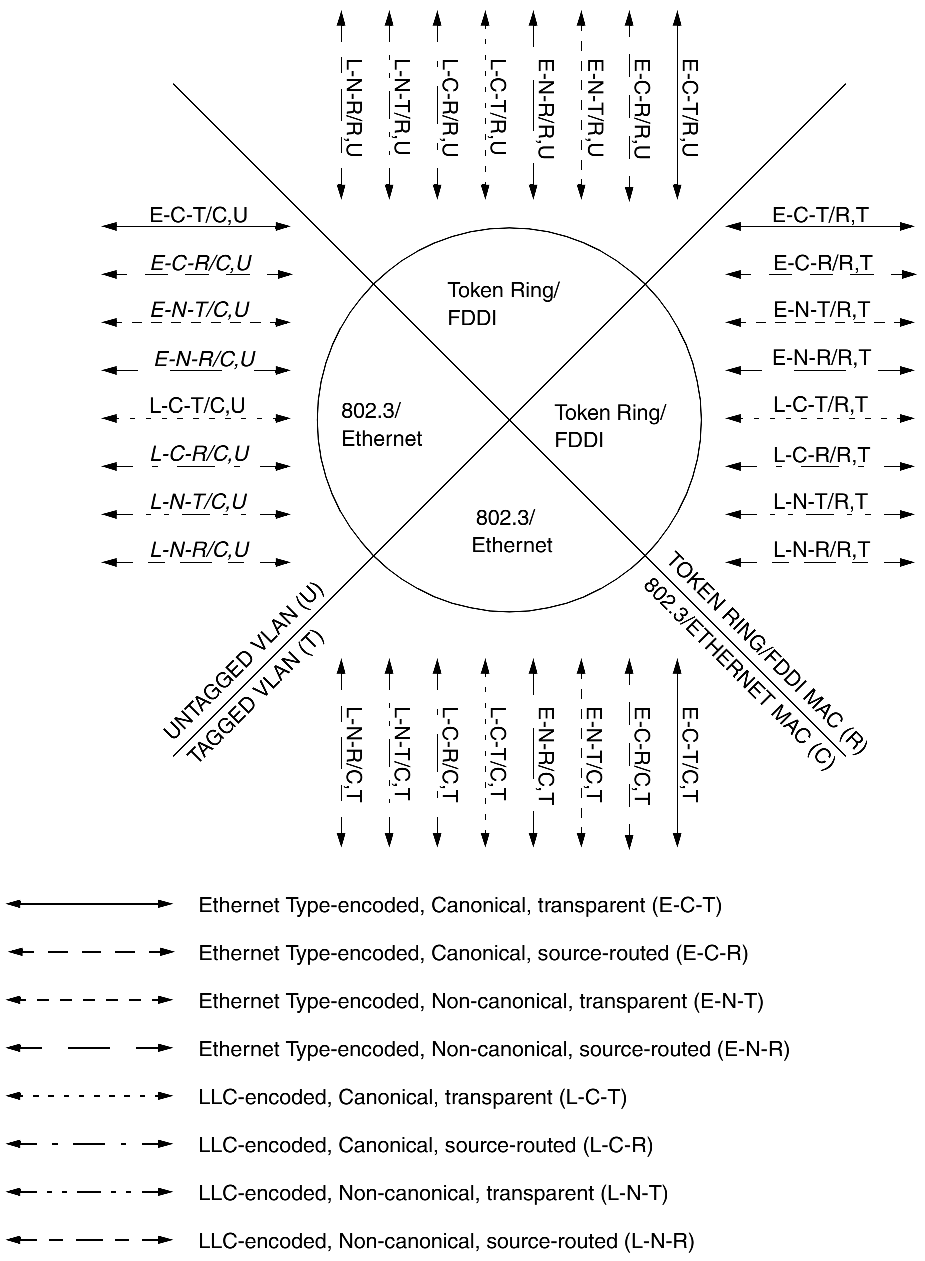


Figure C-1—Services and environments

Figure C-2 illustrates the heterogeneous Bridging functions involved. In this diagram, the bridging functions are labelled H, Q or Q+H depending upon whether the function involves IEEE Std 802.1H, IETF RFC 1042, and IETF RFC 1390 frame translation, 802.1Q VLAN-tagging/untagging/tag translation, or a combination of 802.1Q VLAN-tagging/untagging/tag translation and IEEE Std 802.1H, IETF RFC 1042, and IETF RFC 1390 frame translation.

NOTE 2—Figure C-2 is not intended to represent a real LAN, simply to illustrate the various Bridging functions.

In both diagrams, the frame formats involved are identified by three initial letters that identify the service provided, from the list of 8 services above. The fourth letter indicates the MAC method that carries the frame (C or R), and the fifth letter indicates the type of VLAN (U or T).

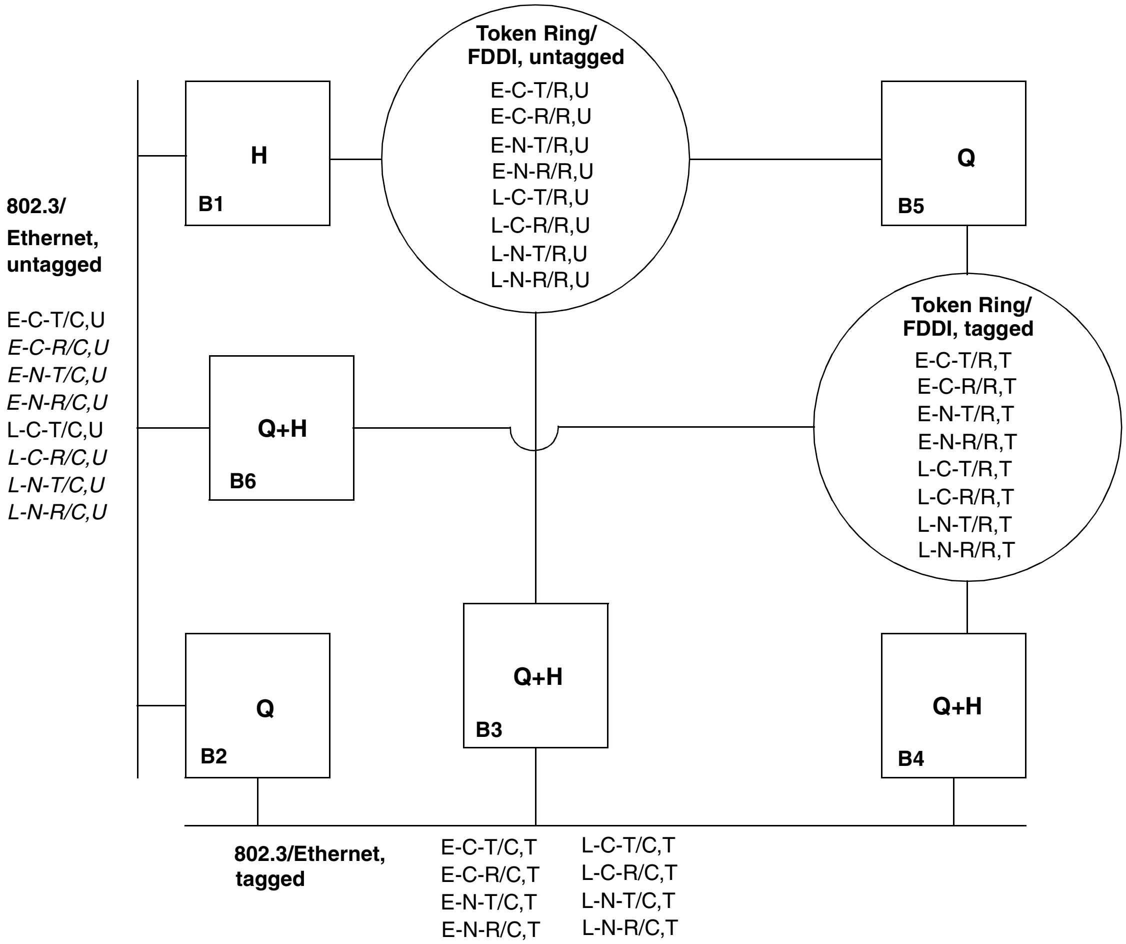


Figure C-2—Heterogeneous Bridging functions

C.2 Bridging functions

C.2.1 Bridging function B1

This function bridges between heterogeneous untagged VLAN environments. The following frame translations are involved:

a) IEEE Std 802.1H, IETF RFC 1042, and IETF RFC 1390 encapsulation/decapsulation for frames carrying Ethernet Type-encoded information;
b) IEEE Std 802.1D, 1998 Edition conversion for frames carrying LLC-encoded information.
c) Any requirements for translation from Canonical to Non-canonical address format, or vice versa, must be met if communication is to be maintained between end stations separated by this Bridging function.
d) Any source-routed traffic cannot be relayed between these environments, as there is no representation for source-routing information in untagged frames on 802.3/Ethernet LANs.

C.2.2 Bridging function B2

This function involves VLAN entry (I to T) and exit (T to I); it bridges between untagged and tagged 802.3/ Ethernet environments. The frame translations involved are as follows:

a) Insertion of Ethernet-encoded tag headers on VLAN entry;
b) Removal of Ethernet-encoded tag headers on VLAN exit;
c) Translation of Non-canonical MAC Addresses to Canonical on VLAN exit;
d) Any source-routed traffic cannot be relayed between these environments, as there is no representation for source-routing information in untagged frames on 802.3/Ethernet LANs.

NOTE—VLAN entry in Non-canonical format does not occur, as the native representation on 802.3/Ethernet is Canonical format. VLAN exit of Non-canonical information can occur only if the Bridge is capable of translating the representation of embedded MAC Addresses to their Canonical format.

C.2.3 Bridging function B3

This function involves VLAN entry and exit; it bridges between tagged 802.3/Ethernet and untagged Ring environments. The following frame translations are involved:

a) For untagged Ethernet Type-encoded information on Token Ring/FDDI (VLAN entry): Removal of IEEE Std 802.1H, IETF RFC 1042, and IETF RFC 1390 encapsulation, and insertion of Ethernet-encoded tag header;
b) For tagged Ethernet Type-encoded information on 802.3/Ethernet (VLAN exit): Removal of tag header and insertion of IEEE Std 802.1H, IETF RFC 1042, and IETF RFC 1390 encapsulation;
c) For VLAN entry/exit with frames carrying LLC-encoded information: Insertion/removal of Ethernet-encoded tag header;
d) Translation of MAC Addresses to the format appropriate for the destination Ring on VLAN exit;
e) Any source-routing information present in the frame is preserved; the Token Ring/FDDI RIF is copied into the E-RIF in the Ethernet-encoded tag header on VLAN entry (with the CFI/NCFI flags set appropriately), and copied back on exit.

NOTE—VLAN entry (tagged 802.3/Ethernet from untagged Token Ring/FDDI) in Canonical format normally occurs only from ISO/IEC 9314-2, as the native representation on IEEE Std 802.5 is Non-canonical format, and for ISO/IEC 9314-2 is Canonical format. Hence, VLAN exit in Canonical format onto IEEE Std 802.5 can occur only if the Bridge is capable of translating the representation of embedded MAC Addresses; i.e., of converting the frame from Canonical on 802.3/Ethernet to Non-canonical on IEEE Std 802.5. Similarly, VLAN exit in Non-canonical format onto ISO/IEC 9314-2 can occur only if the Bridge is capable of converting the frame from Non-canonical on 802.3/Ethernet to Canonical on ISO/IEC 9314-2.

C.2.4 Bridging function B4

This function bridges between tagged 802.3/Ethernet and Ring environments. The following frame translations are involved:

a) For tagged Ethernet Type-encoded information on Token Ring/FDDI to 802.3/Ethernet: Removal of IEEE Std 802.1H, IETF RFC 1042, and IETF RFC 1390 encapsulation, and conversion of the tag header to the Ethernet-encoded form;
b) For tagged Ethernet Type-encoded information on 802.3/Ethernet to Token Ring/FDDI: IEEE Std 802.1H, IETF RFC 1042, and IETF RFC 1390 encapsulation, and conversion of the tag header to the SNAP-encoded form;
c) For tagged frames carrying LLC-encoded information: conversion of the tag header between the SNAP-encoded and Ethernet-encoded forms;

d) Any source-routing information is preserved between these environments by copying between the Token Ring/FDDI RIF and the E-RIF in the Ethernet-encoded tag header.

NOTE 1—This Bridging function is not required to modify the format of embedded MAC Addresses.
NOTE 2—In FDDI LANs, source-routing information may be present either in an E-RIF within the tag header or in the normal position for a source-routed frame.

C.2.5 Bridging function B5

This function involves VLAN entry and exit; it bridges between tagged and untagged Ring environments. The frame translations involved are as follows:

a) Insertion of SNAP-encoded tag header on VLAN entry;

b) Removal of SNAP-encoded tag header on VLAN exit;

c) Translation of MAC Addresses to the format appropriate for the destination Ring on VLAN exit;

d) Any source-routing information present in the frame is preserved.

NOTE 1—VLAN entry in Canonical format normally occurs only from ISO/IEC 9314-2, as the native representation on IEEE Std 802.5 is Non-canonical format, and for ISO/IEC 9314-2 is Canonical format. Hence, VLAN exit in Canonical format onto IEEE Std 802.5 can occur only if the Bridge is capable of translating the representation of embedded MAC Addresses; i.e., of converting the frame from Canonical on 802.3/Ethernet to Non-canonical on IEEE Std 802.5. Similarly, VLAN exit in Non-canonical format onto ISO/IEC 9314-2 can occur only if the Bridge is capable of converting the frame from Non-canonical on 802.3/Ethernet to Canonical on ISO/IEC 9314-2.
NOTE 2—In FDDI LANs, source-routing information may be present either in an E-RIF within the tag header or in the normal position for a source-routed frame.

C.2.6 Bridging function B6

This function involves VLAN entry and exit; it bridges between untagged 802.3/Ethernet and tagged Ring environments. The following frame translations are involved:

a) For untagged Ethernet Type-encoded information on 802.3/Ethernet (VLAN entry): IEEE Std 802.1H, IETF RFC 1042, and IETF RFC 1390 encapsulation, and insertion of SNAP-encoded tag header;

b) For tagged Ethernet Type-encoded information on Token Ring/FDDI (VLAN exit): Removal of SNAP-encoded tag header and removal of IEEE Std 802.1H, IETF RFC 1042, and IETF RFC 1390 encapsulation;

c) For VLAN entry/exit with frames carrying LLC-encoded information: Insertion/removal of SNAP-encoded tag header;

d) Any source-routed traffic cannot be relayed between these environments, as there is no representation for source-routing information in untagged frames on 802.3/Ethernet LANs.

NOTE 1—VLAN entry in Non-canonical format does not occur, as the native representation on 802.3/Ethernet is Canonical format. VLAN exit of Non-canonical format can occur only if the Bridge is capable of translating the representation of embedded MAC Addresses; i.e., of converting the frame from Non-canonical format to Canonical format on 802.3/Ethernet.
NOTE 2—In FDDI LANs, source-routing information may be present either in an E-RIF within the tag header or in the normal position for a source-routed frame.

C.3 Frame formats

The following abbreviations are used in the descriptions of the frame formats in this annex, with the following meanings:

AC	Access Control field—in Token Ring frames only (see ISO/IEC 15802-5 and Note below)
RCI	Ring Control Information—AC (if present) plus FC fields
DA	Destination MAC Address
SA	Source MAC Address
PT	Ethernet Protocol Type
SPT	SNAP-encoded Ethernet Protocol Type (C.6.1)
TPID	Tag Protocol ID (9.3.1)
ETPID	Ethernet-encoded TPID (9.3.1.1)
STPID	SNAP-encoded TPID (9.3.1.2)
TCI	Tag Control Information (9.3.2)
CFI	Canonical Format Indicator (9.3.2.2)
NCFI	Non-canonical Format Indicator (9.3.3.5)
C	Canonical
N	Non-canonical
R	E-RIF present
VID	VLAN Identifier (9.3.2.3)
Len	IEEE Std 802.3-style Length/Type field (C.6.2)
LLC	LLC addressing and control information, as defined in IEEE Std 802.1B
RIF	Source-Routing Information Field (C.6.4)
E-RIF	Embedded RIF (9.3.3, C.6.4)
C-Data	MAC user data in which any embedded MAC Addresses are in Canonical format (C.6.3)
N-Data	MAC user data in which any embedded MAC Addresses are in Non-canonical format
PAD	Padding (C.6.5)
FCS	Frame Check Sequence

In C.3.2, the possible frame formats are categorized by service type; in C.3.3 they are categorized by bearer MAC method and tagging method.

NOTE—The text in this annex makes the generalization of treating the FC fields in 8802-5 and FDDI as if they are the same, in order to simplify the descriptions as much as possible. In reality, there are detailed differences between FC fields in the two MAC methods. When translating between 8802-5 and FDDI, the most likely behavior is to propagate the "LLC frame" indication and the User Priority field between the FC octets on input and output.

C.3.1 Structure of the tagged frame

Figure C-3 illustrates the frame formats used for carrying tagged Ethernet Type-encoded information and LLC-encoded information using 8802-5 Token Ring MAC methods.

Figure C-4 illustrates the frame formats used for carrying tagged Ethernet Type-encoded information and LLC-encoded information using FDDI MAC methods. Two forms of tagged frame are shown:

a) The *source-routed form*, in which the frame carries a RIF in the normal position, following the source MAC Address. This form can only be used on FDDI LANs that support source routing; and
b) The *transparent form*, in which an E-RIF is present in the tag header if the frame carries Non-canonical or source-routed information.

Figure C-5 illustrates the frame formats used for carrying tagged Ethernet Type-encoded information and LLC-encoded information on 802.3/Ethernet MAC methods.

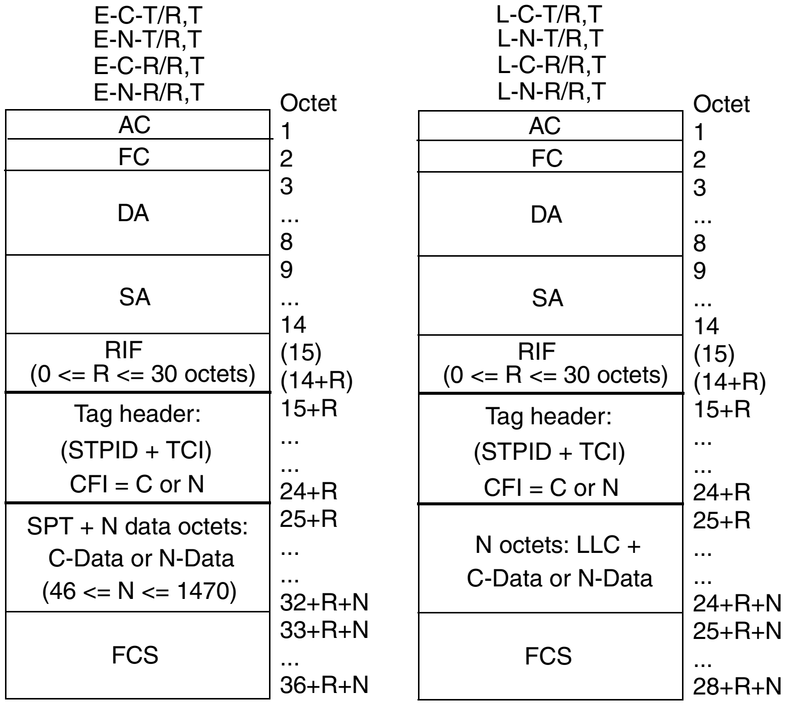


Figure C-3—Tagged frames on 8802-5 Token Ring LANs

As can be seen from these diagrams, the major differences between the tagged frame formats in 802.3/Ethernet and Token Ring/FDDI MAC methods are

c) The presence/absence of RCI (Ring Control Information);
d) The position of the RIF and E-RIF fields;
e) The encoding used to carry the Tag Protocol Identifier (2 octets for ETPID vs. 8 octets for STPID);
f) The encoding used to carry Ethernet Protocol Types (2 octets for PT vs. 8 octets for SPT);
g) The presence/absence of the Length/Type field;
h) The presence/absence of the PAD field.

The diagrams also illustrate the similarities between:

i) The format of tagged frames on 8802-5 and the source-routed form of tagged frames on FDDI;
j) The format of tagged frames on 802.3/Ethernet and the transparent form of tagged frames on FDDI.

C.3.2 Frame formats by service type

C.3.2.1 Frame formats for Ethernet Type-encoded service

C.3.2.1.1 Ethernet Type-encoded, Canonical, transparent

E-C-T/C,U:	DA, SA, PT, C-Data, FCS
E-C-T/C,T:	DA, SA, ETPID, TCI (CFI=C), PT, C-Data, FCS
E-C-T/R,U:	RCI, DA, SA (RII reset), SPT, C-Data, FCS
E-C-T/R,T:	RCI, DA, SA (RII reset), STPID, TCI (CFI=C), SPT, C-Data, FCS

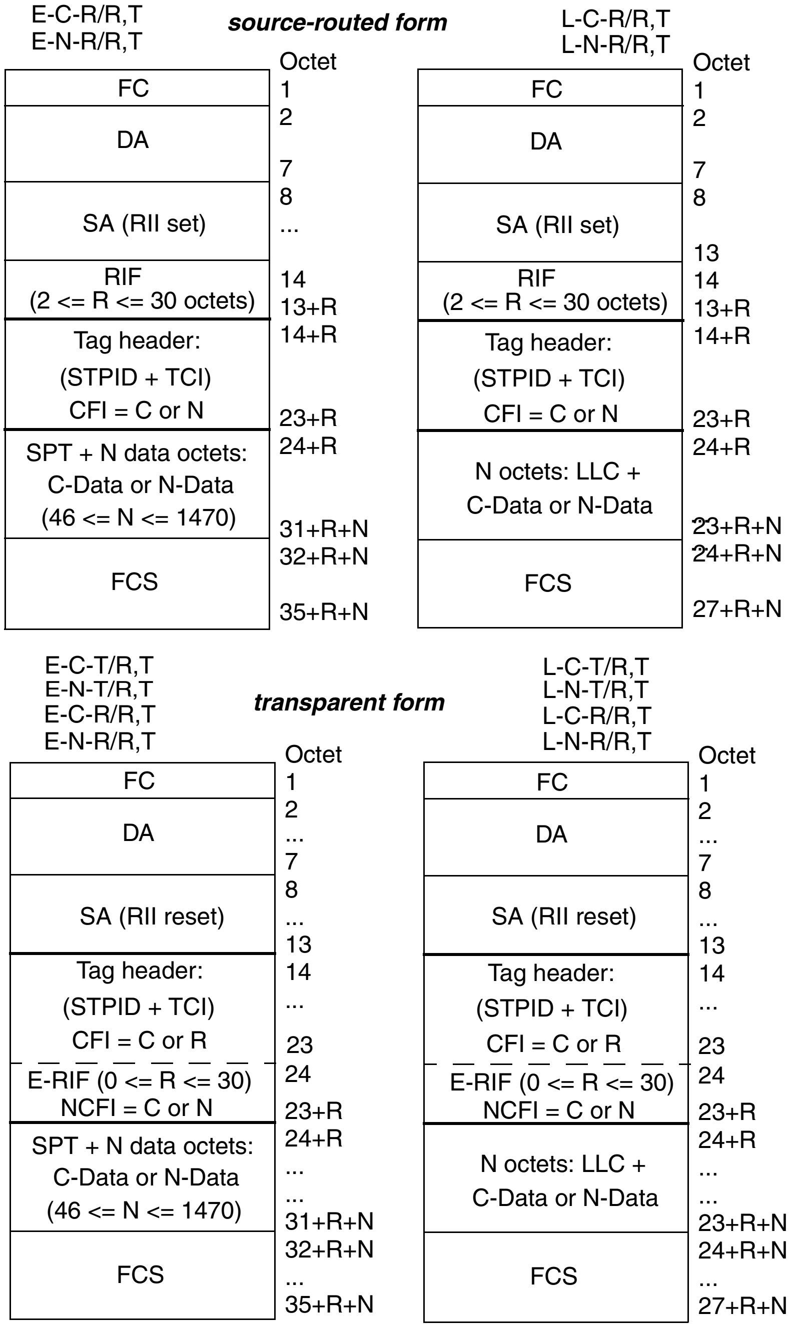


Figure C-4—Tagged frames on FDDI LANs

C.3.2.1.2 Ethernet Type-encoded, Canonical, source-routed

E-C-R/C,U:	No representation possible
E-C-R/C,T:	DA, SA, ETPID, TCI (CFI=R), PT, E-RIF (NCFI=C), C-Data, FCS
E-C-R/R,U:	RCI, DA, SA (RII set), RIF, SPT, C-Data, FCS
E-C-R/R,T:	RCI, DA, SA (RII set), RIF, STPID, TCI (CFI=C), SPT, C-Data, FCS *(source-routed form)*
E-C-R/R,T:	RCI, DA, SA (RII reset), STPID, TCI (CFI=R), E-RIF (NCFI=C), SPT, C-Data, FCS *(transparent form)*

C.3.2.1.3 Ethernet Type-encoded, Non-canonical, transparent

E-N-T/C,U: No representation possible
E-N-T/C,T: DA, SA, ETPID, TCI (CFI=R), PT, E-RIF (NCFI=N), N-Data, FCS
E-N-T/R,U: RCI, DA, SA (RII reset), SPT, N-Data, FCS
E-N-T/R,T: RCI, DA, SA (RII reset), STPID, TCI (CFI =N), SPT, N-Data, FCS *(8802-5 Token Ring form)*
E-N-T/R,T: RCI, DA, SA (RII reset), STPID, TCI (CFI=R), E-RIF (NCFI=N), SPT, N-Data, FCS *(FDDI form)*

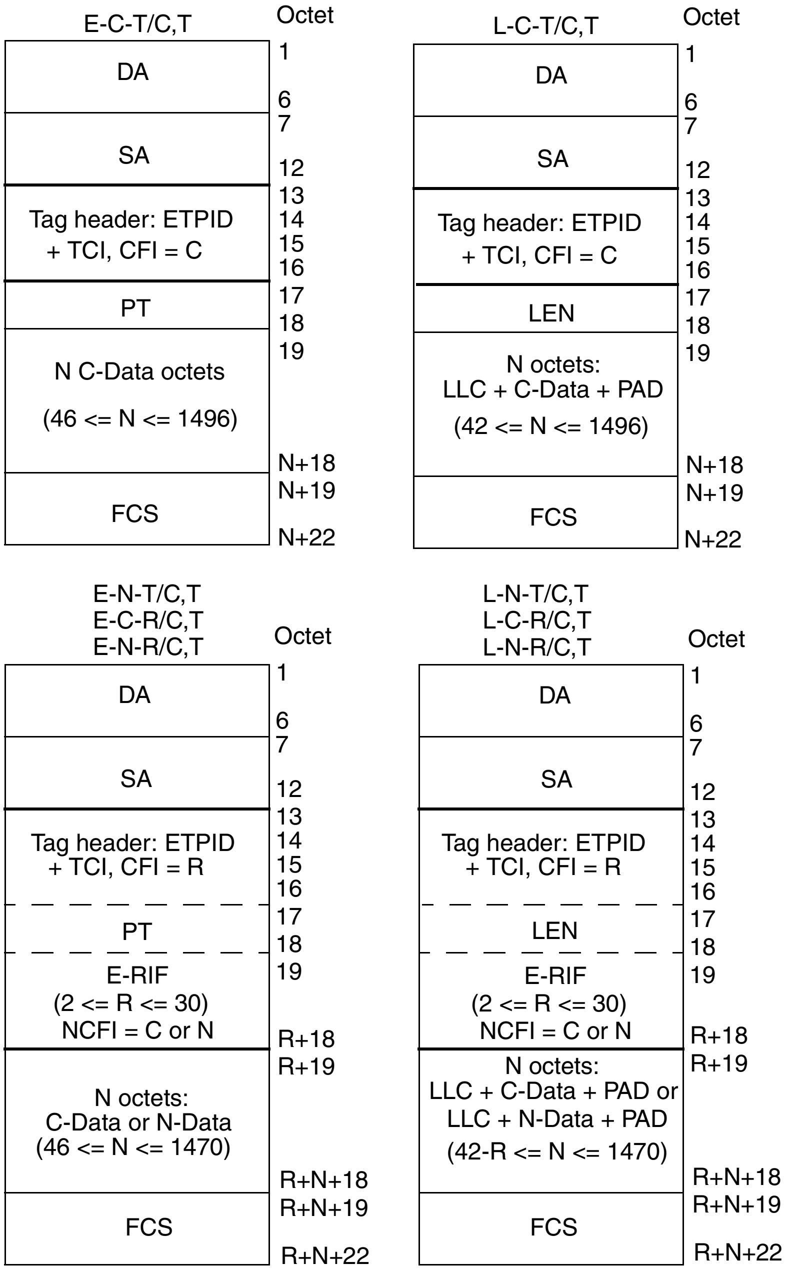


Figure C-5—Tagged frames on 802.3/Ethernet LANs

C.3.2.1.4 Ethernet Type-encoded, Non-canonical, source-routed

E-N-R/C,U:	No representation possible
E-N-R/C,T:	DA, SA, ETPID, TCI (CFI=R), PT, E-RIF (NCFI=N), N-Data, FCS
E-N-R/R,U:	RCI, DA, SA (RII set), RIF, SPT, N-Data, FCS
E-N-R/R,T:	RCI, DA, SA (RII set), RIF, STPID, TCI (CFI=N), SPT, N-Data, FCS *(source-routed form)*
E-N-R/R,T:	RCI, DA, SA (RII reset), STPID, TCI (CFI=R), E-RIF (NCFI=N), N-Data, FCS *(transparent form)*

C.3.2.2 Frame formats for LLC-encoded service

C.3.2.2.1 LLC-encoded, Canonical, transparent

L-C-T/C,U:	DA, SA, LEN, LLC, C-Data, PAD, FCS
L-C-T/C,T:	DA, SA, ETPID, TCI (CFI=C), LEN, LLC, C-Data, PAD, FCS
L-C-T/R,U:	RCI, DA, SA (RII reset), LLC, C-Data, FCS
L-C-T/R,T:	RCI, DA, SA (RII reset), STPID, TCI (CFI=C), LLC, C-Data, FCS

C.3.2.2.2 LLC-encoded, Canonical, source-routed

L-C-R/C,U:	No representation possible
L-C-R/C,T:	DA, SA, ETPID, TCI (CFI=R), LEN, E-RIF (NCFI=C), LLC, C-Data, PAD, FCS
L-C-R/R,U:	RCI, DA, SA (RII set), RIF, LLC, C-Data, FCS
L-C-R/R,T:	RCI, DA, SA (RII set), RIF, STPID, TCI (CFI=C), LLC, C-Data, FCS *(source-routed form)*
L-C-R/R,T:	RCI, DA, SA (RII reset), STPID, TCI (CFI=R), E-RIF (NCFI=C), LLC, C-Data, FCS *(transparent form)*

C.3.2.2.3 LLC-encoded, Non-canonical, transparent

L-N-T/C,U:	No representation possible
L-N-T/C,T:	DA, SA, ETPID, TCI (CFI=R), LEN, E-RIF (NCFI=N), LLC, N-Data, PAD, FCS
L-N-T/R,U:	RCI, DA, SA (RII reset), LLC, N-Data, FCS
L-N-T/R,T:	RCI, DA, SA (RII reset), STPID, TCI (CFI=N), LLC, N-Data, FCS *(8802-5 Token Ring form)*
L-N-T/R,T:	RCI, DA, SA (RII reset), STPID, TCI (CFI=R), E-RIF (NCFI=N), LLC, N-Data, FCS *(FDDI form)*

C.3.2.2.4 LLC-encoded, Non-canonical, source-routed

L-N-R/C,U:	No representation possible
L-N-R/C,T:	DA, SA, ETPID, TCI (CFI=R), LEN, E-RIF (NCFI=N), LLC, N-Data, PAD, FCS
L-N-R/R,U:	RCI, DA, SA (RII set), RIF, LLC, N-Data, FCS
L-N-R/R,T:	RCI, DA, SA (RII set), RIF, STPID, TCI (CFI=N), LLC, N-Data, FCS *(source-routed form)*
L-N-R/R,T:	RCI, DA, SA (RII reset), STPID, TCI (CFI=R), E-RIF (NCFI=N), LLC, N-Data, FCS *(transparent form)*

C.3.3 Frame formats by MAC method type and tagging method

C.3.3.1 Frame formats for 802.3/Ethernet MAC methods

C.3.3.1.1 802.3/Ethernet, untagged

E-C-T/C,U:	DA, SA, PT, C-Data, FCS
E-C-R/C,U:	No representation possible
E-N-T/C,U:	No representation possible
E-N-R/C,U:	No representation possible
L-C-T/C,U:	DA, SA, LEN, LLC, C-Data, PAD, FCS
L-C-R/C,U:	No representation possible
L-N-T/C,U:	No representation possible
L-N-R/C,U:	No representation possible

C.3.3.1.2 802.3/Ethernet, tagged

E-C-T/C,T:	DA, SA, ETPID, TCI (CFI=C), PT, C-Data, FCS
E-C-R/C,T:	DA, SA, ETPID, TCI (CFI=R), PT, E-RIF (NCFI=C), C-Data, FCS
E-N-T/C,T:	DA, SA, ETPID, TCI (CFI=R), PT, E-RIF (NCFI=N), N-Data, FCS
E-N-R/C,T:	DA, SA, ETPID, TCI (CFI=R), PT, E-RIF (NCFI=N), N-Data, FCS
L-C-T/C,T:	DA, SA, ETPID, TCI (CFI=C), LEN, LLC, C-Data, PAD, FCS
L-C-R/C,T:	DA, SA, ETPID, TCI (CFI=R), LEN, E-RIF (NCFI=C), LLC, C-Data, PAD, FCS
L-N-T/C,T:	DA, SA, ETPID, TCI (CFI=R), LEN, E-RIF (NCFI=N), LLC, N-Data, PAD, FCS
L-N-R/C,T:	DA, SA, ETPID, TCI (CFI=R), LEN, E-RIF (NCFI=N), LLC, N-Data, PAD, FCS

C.3.3.2 Frame formats for Token Ring/FDDI MAC methods

C.3.3.2.1 Token Ring/FDDI, untagged

E-C-T/R,U:	RCI, DA, SA (RII reset), SPT, C-Data, FCS
E-C-R/R,U:	RCI, DA, SA (RII set), RIF, SPT, C-Data, FCS
E-N-T/R,U:	RCI, DA, SA (RII reset), SPT, N-Data, FCS
E-N-R/R,U:	RCI, DA, SA (RII set), RIF, SPT, N-Data, FCS
L-C-T/R,U:	RCI, DA, SA (RII reset), LLC, C-Data, FCS
L-C-R/R,U:	RCI, DA, SA (RII set), RIF, LLC, C-Data, FCS
L-N-T/R,U:	RCI, DA, SA (RII reset), LLC, N-Data, FCS
L-N-R/R,U:	RCI, DA, SA (RII set), RIF, LLC, N-Data, FCS

C.3.3.2.2 Token Ring/FDDI, tagged

E-C-T/R,T:	RCI, DA, SA (RII reset), STPID, TCI (CFI=C), SPT, C-Data, FCS
E-C-R/R,T:	RCI, DA, SA (RII set), RIF, STPID, TCI (CFI=C), SPT, C-Data, FCS *(source-routed form)*
E-C-R/R,T:	RCI, DA, SA (RII reset), STPID, TCI (CFI=R), E-RIF (NCFI=C), SPT, C-Data, FCS *(transparent form)*
E-N-T/R,T:	RCI, DA, SA (RII reset), STPID, TCI (CFI=N), SPT, N-Data, FCS *(8802-5 Token Ring form)*
E-N-T/R,T:	RCI, DA, SA (RII reset), STPID, TCI (CFI=R), E-RIF (NCFI=N), SPT, N-Data, FCS *(FDDI form)*
E-N-R/R,T:	RCI, DA, SA (RII set), RIF, STPID, TCI (CFI=N), SPT, N-Data, FCS *(source-routed form)*
E-N-R/R,T:	RCI, DA, SA (RII reset), STPID, TCI (CFI=R), E-RIF (NCFI=N), N-Data, FCS *(transparent form)*

L-C-T/R,T:	RCI, DA, SA (RII reset), STPID, TCI (CFI=C), LLC, C-Data, FCS
L-C-R/R,T:	RCI, DA, SA (RII set), RIF, STPID, TCI (CFI=C), LLC, C-Data, FCS *(source-routed form)*
L-C-R/R,T:	RCI, DA, SA (RII reset), STPID, TCI (CFI=R), E-RIF (NCFI=C), LLC, C-Data, FCS *(transparent form)*
L-N-T/R,T:	RCI, DA, SA (RII reset), STPID, TCI (CFI=N), LLC, N-Data, FCS *(8802-5 Token Ring form)*
L-N-T/R,T:	RCI, DA, SA (RII reset), STPID, TCI (CFI=R), E-RIF (NCFI=N), LLC, N-Data, FCS *(FDDI form)*
L-N-R/R,T:	RCI, DA, SA (RII set), RIF, STPID, TCI (CFI=N), LLC, N-Data, FCS *(source-routed form)*
L-N-R/R,T:	RCI, DA, SA (RII reset), STPID, TCI (CFI=R), E-RIF (NCFI=N), LLC, N-Data, FCS *(transparent form)*

C.4 Procedures for tagging, untagging, and relaying tagged frames

The formal definition of the procedures whereby tag headers are added and removed, and tagged frames are relayed are embodied in Clauses 7 and 8. This informal description is included in order to add clarity to the formal definition of the process.

C.4.1 Tagging

The following subclauses describe the translations that are performed when an untagged frame is relayed in tagged form.

C.4.1.1 MAC header information

The RCI, DA, SA and RIF fields (if supported in the source frame and/or destination MAC methods) are translated from their representation in the source frame into the equivalent representation in the destination frame in accordance with the procedures described in IEEE Std 802.1D, 1998 Edition. This will result in

a) Preservation of the AC (Token Ring to Token Ring only) and FC fields (Token Ring/FDDI to Token Ring/FDDI only);
b) Translation of the DA and SA into their equivalent representation in the destination MAC methods;
c) Preservation of the RIF field, if present; either in its conventional position (Token Ring/FDDI to Token Ring/FDDI, source-routed form) or within the tag header (Token Ring/FDDI to 802.3/Ethernet or FDDI, transparent form).

NOTE—The ability of the tag header to carry source-routing information across 802.3/Ethernet LANs does not imply a requirement on the part of a pure 802.3/Ethernet Bridge to support source routing. This capability is provided simply to allow traffic that originates in, and is destined for, a source-routed environment to transit as tagged traffic across a non-source-routed environment. Similarly, this capability allows source-routed traffic to transit an FDDI network that is otherwise unable to support source routing.

C.4.1.2 Tag header insertion

The tag header is inserted immediately following the SA field (if no RIF is present in the destination frame) or immediately following the RIF field (if RIF is present in the destination frame). The header contains

a) An Ethernet-encoded TPID (destination MAC method is 802.3/Ethernet) or a Snap-encoded TPID (destination MAC method is Token Ring/FDDI);
b) A TCI field, as follows:

1) The user_priority field is set in accordance with the procedure described in IEEE Std 802.1D, 1998 Edition;
2) The CFI flag, indicating C/N (8802-5 Token Ring, and source-routed FDDI MAC methods), or C/[RIF present] (802.3/Ethernet and transparent FDDI MAC methods), in accordance with the format of the MAC user data;
3) The VID field is set to the VID of the VLAN to which the source frame belongs.

c) An E-RIF field, immediately following the Length/Type field (802.3/Ethernet and transparent FDDI MAC methods), if the frame is carrying Non-canonical data and/or source-routing information. The NCFI in the RIF indicates C or N, in accordance with the format of the MAC user data.

C.4.1.3 Ethernet Type-encoded data

If the MAC user data carries Ethernet Type-encoded data, i.e., the protocol identifier is an Ethernet Type value or an Ethernet Type value that has been SNAP encoded as described in IEEE Std 802.1D, 1998 Edition and IEEE Std 802.1H, and if the frame is being relayed between differing MAC methods (802.3/Ethernet to or from Token Ring/FDDI), then the data is translated from its source format to the format appropriate to the destination MAC method in accordance with the procedures described in IEEE Std 802.1D, 1998 Edition and IEEE Std 802.1H.

C.4.1.4 FCS

When tagging a frame and performing the attendant field translations, it is necessary to recompute the Frame Check Sequence (FCS) field of the tagged frame. As stated in IEEE Std 802.1D, 1998 Edition, 6.3.7, the Bridge shall not introduce additional undetected frame errors as a result of such FCS recomputation.

NOTE—Where necessary in order to preserve the protection afforded by the original FCS, it is possible to incrementally compute the new FCS value, based on the original FCS, adjusted for the new fields added and the frame length. This technique, and other techniques for preserving FCS integrity, are discussed in IEEE Std 802.1D, 1998 Edition, Annex G.

C.4.2 Untagging

The following subclauses describe the frame translations that are performed when a received tagged frame is relayed in untagged format.

C.4.2.1 MAC header information

The RCI, DA, SA and RIF or E-RIF fields (if supported in the source frame and/or destination MAC methods) are translated from their representation in the source frame into the equivalent representation in the destination frame in accordance with the procedures described in IEEE Std 802.1D, 1998 Edition. This will result in

a) Preservation of the AC (Token Ring to Token Ring only) and FC fields (Token Ring/FDDI to Token Ring/FDDI only);
b) Translation of the DA and SA into their equivalent representation in the destination MAC method;
c) Preservation of any source-routing information carried in the source frame, if present, and if the destination MAC method is a Token Ring/FDDI LAN that supports source routing. (If the source MAC method is 802.3/Ethernet or transparent FDDI and the tag header carries an E-RIF in which the RT field indicates a transparent frame, then the E-RIF is not considered to be carrying any source-routing information.)

C.4.2.2 Tag header

The tag header is removed.

C.4.2.3 Ethernet Type-encoded data

If the frame carries Ethernet Type-encoded data, i.e., the protocol identifier is an Ethernet Type value or an Ethernet Type value that has been SNAP encoded as described in IEEE Std 802.1D, 1998 Edition and IEEE Std 802.1H, and if the frame is being relayed between differing MAC methods (802.3/Ethernet to or from Token Ring/FDDI), then the data is translated from its source format to the format appropriate to the destination MAC method in accordance with the with the procedures described in IEEE Std 802.1D, 1998 Edition and IEEE Std 802.1H.

C.4.2.4 Address translation

If the CFI/NCFI information in the tagged frame indicates that embedded addresses are being carried in a format inappropriate to the destination MAC method, then it is necessary either to translate the addresses from C to N or vice versa, or to discard the frame if such translation is not supported by the Bridge.

C.4.2.5 FCS

When removing a frame's VLAN tag and performing the attendant field translations, it is necessary to recompute the Frame Check Sequence (FCS) field of the tagged frame. As stated in IEEE Std 802.1D, 1998 Edition, 6.3.7, the Bridge shall not introduce additional undetected frame errors as a result of such FCS recomputation.

NOTE—Where necessary in order to preserve the protection afforded by the original FCS, it is possible to incrementally compute the new FCS value, based on the original FCS, adjusted for the new fields added and the frame length. This technique, and other techniques for preserving FCS integrity, are discussed in IEEE Std 802.1D, 1998 Edition, Annex G.

C.4.3 Relaying tagged frames

The following subclauses describes the frame translations that are performed when a received tagged frame is relayed in tagged format.

C.4.3.1 MAC header information

The RCI, DA, and SA (if supported in the source frame and/or destination frame formats) are translated from their representation in the source frame into the equivalent representation in the destination frame in accordance with the procedures described in IEEE Std 802.1D, 1998 Edition.

For source-routed Token Ring/FDDI to 802.3/Ethernet or transparent FDDI, the RIF field (if present) is translated into the E-RIF field of the destination frame.

For relay between source-routed Token Ring/FDDI environments, the RIF (if present) is copied into the RIF field of the destination frame.

For 802.3/Ethernet or transparent FDDI to source-routed Token Ring/FDDI, the E-RIF field, if present, is translated into the RIF of the destination frame, with the NCFI bit reset, unless the E-RIF indicates that the frame is a transparent frame, in which case, the E-RIF is discarded.

This will result in

a) Preservation of the AC (Token Ring to Token Ring only) and FC fields (Token Ring/FDDI to Token Ring/FDDI only);
b) Translation of the DA and SA into their equivalent representation in the destination MAC method;
c) Preservation of any information carried in the E-RIF or RIF field, if present and if it carries source-routing information.

C.4.3.2 Tag header

If the source and destination MAC methods differ, the tag header is modified as follows:

a) The TPID field is set in accordance with the destination MAC method. An Ethernet-encoded TPID is used where the destination MAC method is 802.3/Ethernet; a Snap-encoded TPID is used where the destination MAC method is Token Ring/FDDI;
b) The information carried in the User Priority and VID fields in the TCI are copied unchanged into the destination frame's TCI.
c) If the source and destination MAC methods are of the same type, then the CFI (and RIF, if present in 802.3/Ethernet) are copied unchanged into the destination tag header.
d) If the source and destination MAC methods differ, then the CFI information in the source tag header is translated into the format appropriate for the destination tag header.

C.4.3.3 Ethernet Type-encoded data

If the frame carries Ethernet Type-encoded data, i.e., the protocol identifier is an Ethernet Type value or an Ethernet Type value that has been SNAP encoded as described in IEEE Std 802.1D, 1998 Edition and IEEE Std 802.1H, and if the frame is being relayed between differing MAC methods (802.3/Ethernet to or from Token Ring/FDDI), then the data is translated from its source format to the format appropriate to the destination MAC method in accordance with the with the procedures described in IEEE Std 802.1D, 1998 Edition and IEEE Std 802.1H.

C.4.3.4 FCS

When relaying tagged frames, if it is necessary to perform any attendant field translations, then it is necessary to recompute the Frame Check Sequence (FCS) field of the tagged frame. As stated in IEEE Std 802.1D, 1998 Edition, 6.3.7, the Bridge shall not introduce additional undetected frame errors as a result of such FCS recomputation.

NOTE—Where necessary in order to preserve the protection afforded by the original FCS, it is possible to incrementally compute the new FCS value, based on the original FCS, adjusted for the new fields added and the frame length. This technique and other techniques for preserving FCS integrity are discussed in IEEE Std 802.1D, 1998 Edition, Annex G.

C.4.4 Padding and frame size considerations

C.4.4.1 Treatment of PAD fields in IEEE Std 802.3 frames

The minimum frame size constraint placed on 802.3/Ethernet frames requires frames to carry zero or more pad octets following the MAC client data, in order to ensure that no frame of total length less than 64 octets is transmitted on the medium. This means that frames whose overall length would otherwise be less than 64 octets in length have (64-len) octets of padding added after the MAC client data, where len is the size of the frame before padding.

When tagged frames are transmitted by a Bridge on an IEEE Std 802.3 MAC, there are two permissible approaches (7.2):

a) Keep the minimum frame size generated by the Bridge equal to 64 octets. This implies that the number of pad octets in a received untagged IEEE Std 802.3 frame would be reduced by up to 4 octets when that frame was tagged;
b) Adopt a minimum tagged frame length of 68 octets. This implies that the number of pad octets in a received untagged IEEE Std 802.3 frame would not be adjusted when tagging such frames; equally, if subsequently untagged, no pad adjustment would be necessary before transmission on 802.3/Ethernet.

There is a similar choice to be made in end stations that generate tagged frames:

c) In some existing implementations, the decision as to whether pad octets are needed will be made at a point where it is impractical to distinguish between tagged and untagged frames. In these cases, the end station will use a minimum frame size of 64 octets for all frames;
d) In other cases, the padding decision will be taken at a point before it is known whether the frame will be transmitted tagged or untagged. In these cases, the end station will use a minimum tagged frame size of 68 octets, and a minimum of 64 octets for untagged frames.

The above approaches are all consistent with the IEEE Std 802.3 frame specification, as amended by IEEE Std 802.3ac-1998.

The implication of this is that, for correct operation on 802.3/Ethernet, all devices have to be capable of correctly handling tagged frames of less than 68 octets in length (C.4.4.3).

C.4.4.2 Maximum PDU size

VLAN tagging of an untagged frame, or relaying frames in tagged frame format, can result in an increase in the length of the original frame. If transmission of a given frame in tagged frame format through a given destination Port would result in violation of the maximum PDU size for the destination MAC method, the Bridge discards the frame for that destination Port.

NOTE—Violation of the maximum PDU sizes for destination MAC methods can produce undefined results in Bridged LANs that contain devices that adhere strictly to these maxima, or in MAC methods where these maxima are inherently constrained by the operation of the MAC method itself (e.g., constrained by timing considerations in the MAC state machines).

IEEE Std 802.3ac-1998 defines an extension to the normal 802.3 maximum frame size for the specific purpose of accommodating the additional octets of the VLAN tag header. The example frame translations in this annex make use of this extension to the 802.3 frame size.

C.4.4.3 Minimum PDU size

VLAN untagging of a tagged frame results in the original frame decreasing in length.

Where the destination MAC is CSMA/CD:

a) If untagging a given frame would result in violation of the minimum frame length requirements of CSMA/CD, the Bridge is required to adjust the PAD field to ensure that the frame length equals the minimum length of 64 octets (7.2 and C.4.4.1);
b) If a frame is transmitted in tagged frame format, the Bridge may adopt a minimum tagged frame length of either 64 or 68 octets, as an implementation option. If the latter is chosen, the Bridge adjusts the size of the PAD field on transmission for any tagged frame that is less than 68 octets in length (7.2, C.4.4.1).

C.5 Frame translations for different MAC methods

Examples of the frame translations that can occur when an untagged frame is translated into a tagged frame, and when tagged frames are relayed, are illustrated in the following clauses.

Subclauses C.5.1 and C.5.2 describe the translations that can occur when untagged frames on 802.3/Ethernet, and Token Ring/FDDI are translated into the tagged frame format. C.5.3 describes the translations that can occur when a tagged frame is relayed between differing MAC methods in tagged format. In each subclause, the following cases are shown:

a) The untagged frame carried Ethernet Type-encoded information;
b) The untagged frame carried LLC-encoded information.

NOTE—In developing the example translations, the field sizes on 802.3/Ethernet have been calculated using the IEEE Std 802.3ac-1998 extension to the standard maximum frame size (normally 1518 octets). IEEE Std 802.3ac-1998 allows the maximum frame size to be extended by 4 octets for the specific purpose of accommodating the tag header.

C.5.1 Tagging of untagged 802.3/Ethernet frames

C.5.1.1 Ethernet Type-encoded information on 802.3/Ethernet LAN to tagged frame format

Figure 8-1 illustrates the translation between an untagged Ethernet Type-encoded frame on 802.3/Ethernet (E-C-T/C,U) and a tagged frame on 802.3/Ethernet (E-C-T/C,T).

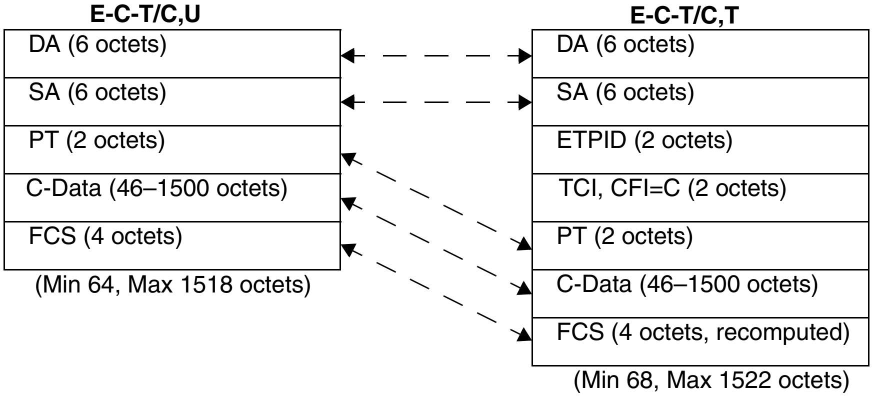


Figure C-6—Translation between E-C-T/C,U and E-C-T/C,T

The following translations are required in order to tag an E-C-T/C,U frame on 802.3/Ethernet:

a) The SA and DA fields are copied unchanged;
b) The ETPID and TCI are inserted, with CFI=C;
c) The PT and C-Data fields are copied unchanged;
d) The FCS is recomputed.

Removal of the tag involves the reverse of this process.

This form of tagging causes the original frame size to be increased by 4 octets.

Figure C-7 illustrates the translation between an untagged Ethernet Type-encoded frame on 802.3/Ethernet (E-C-T/C,U) and a tagged frame on Token Ring/FDDI (E-C-T/R,T).

The following translations are required in order to tag an E-C-T/C,U frame on a Token Ring/FDDI LAN:

e) The appropriate variant of the RCI field is added;
f) The DA and SA fields carry the same MAC Addresses as in the original frame;

NOTE—The meaning of the wording used in f) (and in other instances in this annex where this form of words is used) is that the MAC Addresses in the original and translated frames, when represented using the hexadecimal notation defined in Clause 5 of IEEE Standard 802, are the same.

g) The STPID and TCI are inserted, with CFI=C;
h) The PT is translated into the IEEE Std 802.1H, IETF RFC 1042, and IETF RFC 1390-encoded form (SPT);
i) The C-Data field is copied unchanged;
j) The FCS is recomputed.

Removal of the tag involves the reverse of this process.

This form of tagging causes the original frame size to be increased by 17 octets for FDDI or 18 octets for Token Ring.

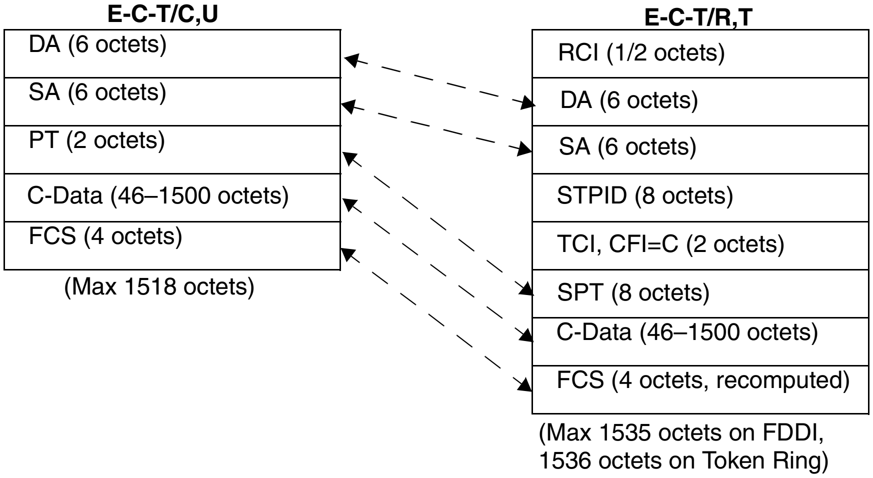


Figure C-7—Translation between E-C-T/C,U and E-C-T/R,T

NOTE—In translational (VLAN-unaware) bridging between 802.3/Ethernet and Ring LANs, an Ethernet Type-encoded frame increases in size by 7 octets on FDDI, and 8 octets on Token Ring.

Translations for E-C-R/C,U, E-N-T/C,U, and E-N-R/C,U to their equivalent tagged frame formats (E-C-R/C,T, E-N-T/C,T, and E-N-R/C,T on 802.3/Ethernet, and E-C-R/R,T, E-N-T/R,T and E-N-R/R,T on Token Ring/FDDI) cannot be shown, as there is no representation for such untagged frames on 802.3/Ethernet LANs. Similarly, translation of the tagged frames E-C-R/C,T, E-N-R/C,T, E-N-R/R,T, and E-C-R/R,T to untagged frames on 802.3/Ethernet is not possible, as it involves loss of the source-routing information. Translation of the remaining Non-canonical, transparent tagged frame formats into E-C-T/C,U is possible, but only if the Bridge is capable of translating Non-canonical data to its Canonical form.

C.5.1.2 LLC-encoded information on 802.3/Ethernet to tagged frame format

Figure C-8 illustrates the translation between an untagged frame on 802.3/Ethernet carrying LLC-encoded information (L-C-T/C,U) and a tagged frame on 802.3/Ethernet (L-C-T/C,T).

Tagging an L-C-T/C,U frame on 802.3/Ethernet LANs requires the following frame translations:

a) The DA and SA fields are copied unchanged;
b) Insert ETPID and TCI fields, with CFI=C;

c) Len, LLC and C-Data fields are copied unchanged;
d) The PAD may either be copied unchanged (giving a minimum tagged frame size of 68 octets), or reduced by up to 4 octets (giving a minimum tagged frame size of 64 octets), as an implementation option;

NOTE—If the actual length of the data portion of the frame is inconsistent with the value held in LEN, then this inconsistency is not corrected by the tagging process. This is done in order that protocols which generate such inconsistency, and which require that inconsistency to be maintained for their correct operation, are not broken by this aspect of tagging.

e) Recompute the FCS.

Removal of the tag involves the reverse of this process.

This form of tagging causes the original frame size to be increased by 4 octets.

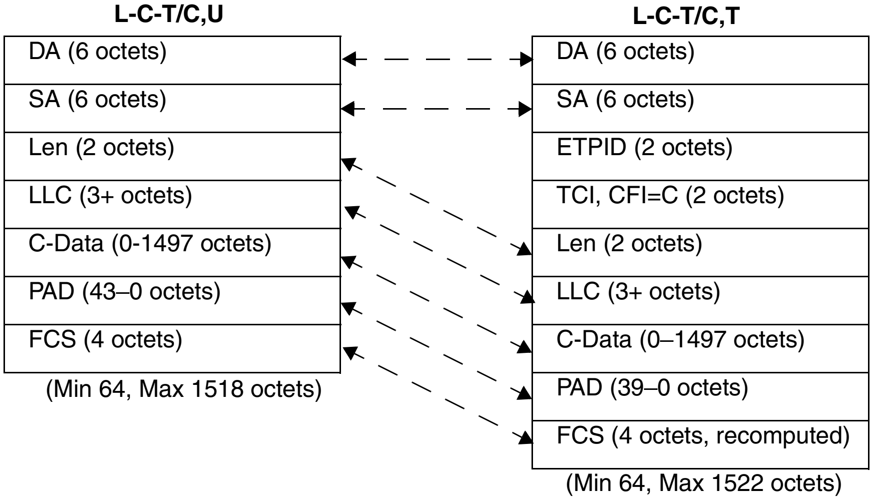


Figure C-8—Translation between L-C-T/C,U and L-C-T/C,T

Figure C-9 illustrates the translation between an untagged LLC-encoded frame on 802.3/Ethernet (L-C-T/C,U) and a tagged frame on Token Ring/FDDI (L-C-T/R,T).

Tagging in LLC-encoded format consists of the following frame translations:

f) The appropriate RCI field for the Ring MAC method concerned is added;
g) The DA and SA fields carry the same MAC Addresses as in the original frame;
h) Insert STPID and TCI fields, with CFI=C;
i) The Len field is removed;
j) Copy the LLC field unchanged;
k) The C-Data field is copied unchanged;
l) The PAD field is removed;
m) Recompute the FCS.

Removal of the tag (tagged frame to native 802.3/Ethernet frame) involves the reverse of this process.

This form of tagging causes the original frame size to be increased by 9 octets for FDDI or 10 octets for Token Ring.

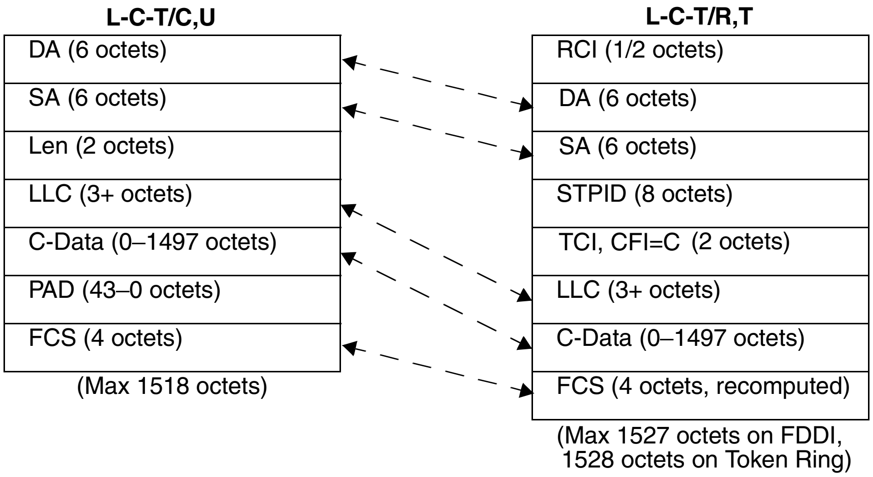


Figure C-9—Translation between L-C-T/C,U and L-C-T/R,T

NOTE—In translational (VLAN-unaware) bridging between 802.3/Ethernet and Ring LANs, an LLC-encoded frame reduces in size by 1 octet on FDDI, and does not change in length on Token Ring.

Translations for L-C-R/C,U, L-N-T/C,U, and L-N-R/C,U to their equivalent tagged frame formats (L-C-R/C,T, L-N-T/C,T, and L-N-R/C,T on 802.3/Ethernet, and L-C-R/R,T, L-N-T/R,T, and L-N-R/R,T on Token Ring/FDDI) cannot be shown, as there is no representation for such untagged frames on 802.3/Ethernet LANs. Similarly, translation of L-C-R/C,T, L-N-R/C,T, L-N-R/R,T, and L-C-R/R,T to untagged frames on 802.3/Ethernet is not possible, as it involves loss of the source-routing information. Translation of the remaining Non-canonical, transparent tagged frame formats into L-C-T/C,U is possible, but only if the Bridge is capable of translating Non-canonical data to its Canonical form.

C.5.2 Translation of untagged Token Ring/FDDI frames

C.5.2.1 Ethernet Type-encoded information on Token Ring/FDDI to tagged frame format

Figure C-10 illustrates the translation between an untagged Ethernet Type-encoded frame on Token Ring/FDDI (E-C-T/R,U, E-N-T/R,U, E-C-R/R,U or E-N-R/R,U) and a tagged frame on 802.3/Ethernet (E-C-T/C,T, E-N-T/C,T, E-C-R/C,T, or E-N-R/C,T).

Tagging requires the following frame translations:

a) Remove the RCI field;
b) The DA and SA fields carry the same MAC Addresses as in the original frame, with the RII bit reset;
c) Insert ETPID and TCI, with CFI=C (E-C-T/R,U) or R (all other frame types);
d) If the RII bit was set in the original frame, translate the RIF into the tag header E-RIF. For E-N-T/R,U, create a RIF with frame type = transparent. Set the E-RIF NCFI to C or N appropriately;
e) Translate the SPT into its corresponding PT;
f) Copy the Data field;
g) Recompute the FCS.

Removal of the tag involves the reverse of this process.

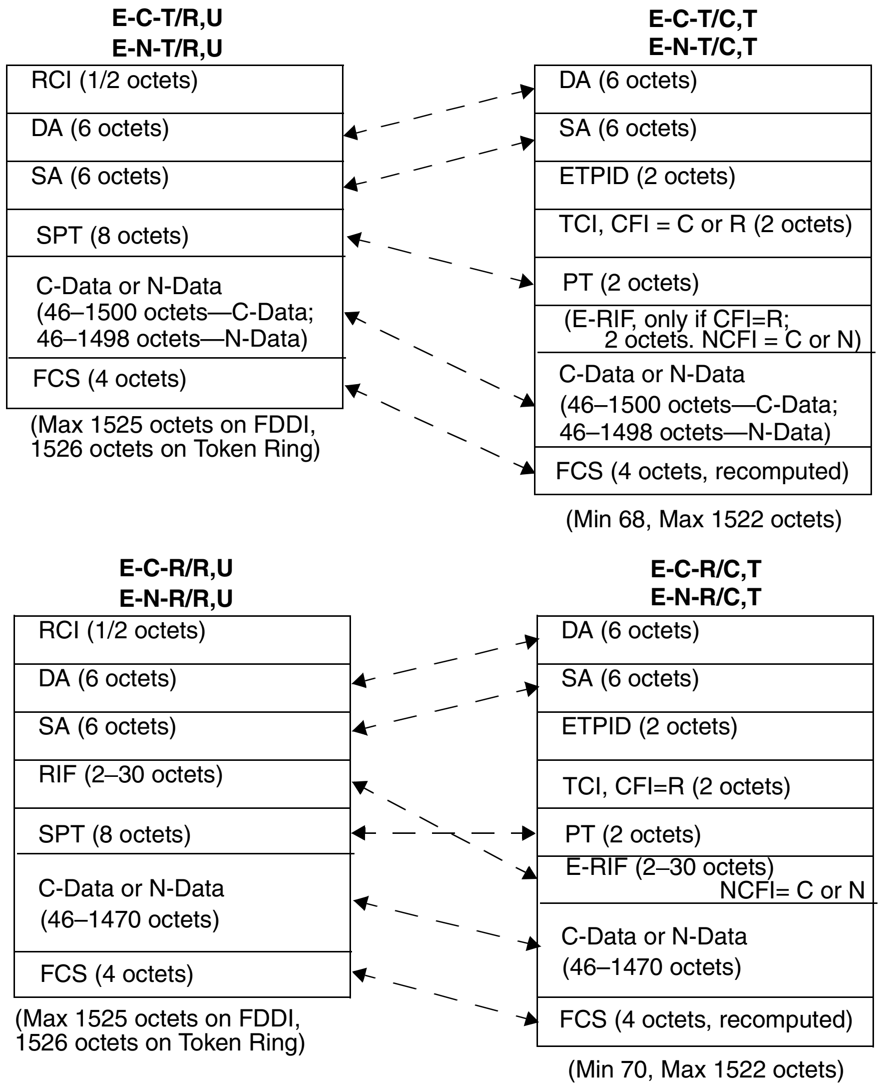


Figure C-10—Translation between E-X-X/R,U and E-X-X/C,T

NOTE 1—When removing the tag, if the CFI/NCFI indicates that embedded address information is in a form inappropriate for the destination MAC method, then it is necessary either to translate the address information or to discard the frame.

This form of tagging causes the original frame size to be reduced by 3 or 4 octets.

Figure C-11 illustrates the translation between an untagged Ethernet Type-encoded frame on Token Ring/ FDDI (E-C-T/R,U, E-N-T/R,U, E-C-R/R,U, or E-N-R/R,U) and a tagged frame on 8802-5 Token Ring (E-C-T/R,T, E-N-T/R,T, E-C-R/R,T, or E-N-R/R,T). Figure C-11 also illustrates the translation of E-C-T/R,U, E-C-R/R,U, and E-N-R/R,U to tagged frames on FDDI media, the latter two translations illustrating the source-routed form of the tagged frame on FDDI.

Tagging requires the following frame translations:

h) Copy the RCI field;
i) The DA and SA fields carry the same MAC Addresses as in the original frame, with RII in the same state as in the original frame;

j) Copy the RIF field if present (RII set);
k) Insert STPID and TCI, setting the CFI to N or C appropriately;
l) Copy the SPT field;
m) Copy the Data field;
n) Recompute the FCS.

Removal of the tag (tagged frame to native Token Ring/FDDI frame) involves the reverse of this process.

NOTE 2—When removing the tag, if the CFI indicates that embedded address information is in a form inappropriate for the destination MAC method, then it is necessary either to translate the address information or to discard the frame.

This form of tagging causes the original frame size to be increased by 10 octets.

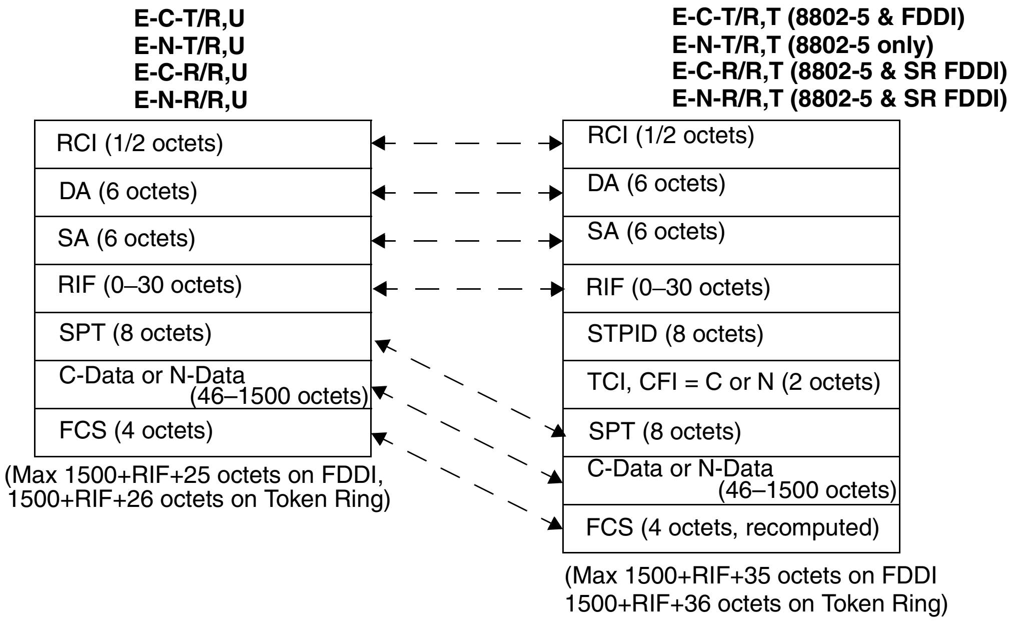

E-C-T/R,U
E-N-T/R,U
E-C-R/R,U
E-N-R/R,U
RCI (1/2 octets)
DA (6 octets)
SA (6 octets)
RIF (0–30 octets)
SPT (8 octets)
C-Data or N-Data (46–1500 octets)
FCS (4 octets)
(Max 1500+RIF+25 octets on FDDI, 1500+RIF+26 octets on Token Ring)
E-C-T/R,T (8802-5 & FDDI)
E-N-T/R,T (8802-5 only)
E-C-R/R,T (8802-5 & SR FDDI)
E-N-R/R,T (8802-5 & SR FDDI)
RCI (1/2 octets)
DA (6 octets)
SA (6 octets)
RIF (0–30 octets)
STPID (8 octets)
TCI, CFI = C or N (2 octets)
SPT (8 octets)
C-Data or N-Data (46–1500 octets)
FCS (4 octets, recomputed)
(Max 1500+RIF+35 octets on FDDI 1500+RIF+36 octets on Token Ring)

Figure C-11—Translation between E-X-X/R,U and E-X-X/R,T (8802-5 & SR FDDI)

Figure C-12 illustrates the translation between an untagged Ethernet Type-encoded frame on Token Ring/ FDDI (E-N-T/R,U, E-C-R/R,U or E-N-R/R,U) and a tagged frame on FDDI (E-N-T/R,T, E-C-R/R,T or E-N-R/R,T). Note that the translation of E-C-T/R,U to E-C-T/R,T was dealt with in Figure C-11.

Tagging requires the following frame translations:

o) Copy the RCI field;
p) The DA and SA fields carry the same MAC Addresses as in the original frame, but with RII reset regardless of its state in the original frame;
q) Insert STPID and TCI, setting the CFI to R;
r) Translate the RIF field if present (RII set in source frame) to the E-RIF, otherwise create an E-RIF with RT indicating a transparent frame. Set the NCFI to C or N appropriately;
s) Copy the SPT field;
t) Copy the Data field;
u) Recompute the FCS.

Removal of the tag (FDDI tagged frame to native Token Ring/FDDI frame) involves the reverse of this process.

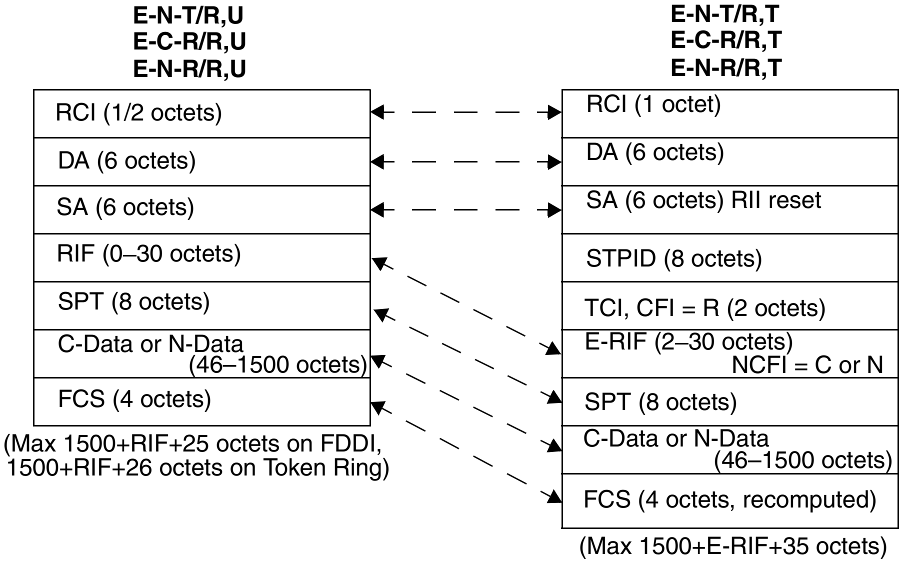


Figure C-12—Translation between E-X-X/R,U and E-X-X/R,T (transparent FDDI)

NOTE 3—When removing the tag, if the NCFI indicates that embedded address information is in a form inappropriate for the destination MAC method, then it is necessary either to translate the address information or to discard the frame.

This form of tagging causes the original frame size to be increased by 10 or 12 octets.

C.5.2.2 LLC-encoded information on Token Ring/FDDI to tagged frame format

Figure C-13 illustrates the translation between an untagged LLC-encoded frame on Token Ring/FDDI (L-C-T/R,U, L-N-T/R,U, L-C-R/R,U, or L-N-R/R,U) and a tagged frame on 802.3/Ethernet (L-C-T/C,T, L-N-T/C,T, L-C-R/C,T, or L-N-R/C,T).

Tagging requires the following frame translations:

a) Remove the RCI field;
b) The DA and SA fields carry the same MAC Addresses as in the original frame, with the RII bit reset;
c) Insert ETPID and TCI, with CFI=C (L-C-T/R,U) or R (all other frame types);
d) If the RII bit was set in the original frame, translate the RIF into the tag header E-RIF. For L-N-T/R,U, create an E-RIF with frame type = transparent. Set the E-RIF NCFI to C or N appropriately;
e) Insert the Len field, with value equal to the number of LLC+Data octets;
f) Copy the LLC field;
g) Copy the Data field;
h) PAD field is inserted if Len is less than 46 (if a minimum tagged frame size of 68 is implemented) or if less than 42 (if a minimum tagged frame size of 64 is implemented);
i) Recompute the FCS.

Removal of the tag involves the reverse of this process.

NOTE 1—When removing the tag, if the CFI/NCFI information indicates that embedded address information is in a form inappropriate for the destination MAC method, then it is necessary either to translate the address information or to discard the frame.

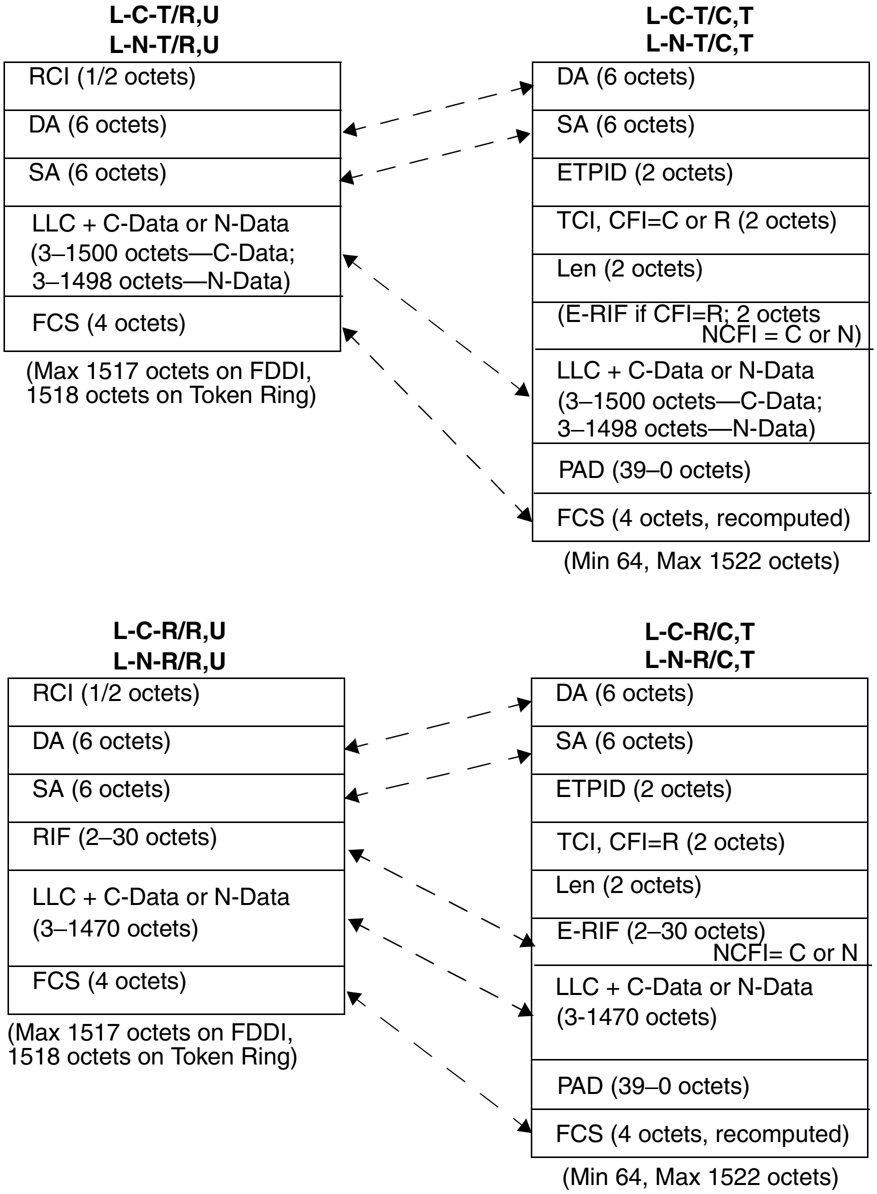

L-C-T/R,U
L-N-T/R,U
RCI (1/2 octets)
DA (6 octets)
SA (6 octets)
LLC + C-Data or N-Data (3–1500 octets—C-Data; 3–1498 octets—N-Data)
FCS (4 octets)
(Max 1517 octets on FDDI, 1518 octets on Token Ring)
L-C-T/C,T
L-N-T/C,T
DA (6 octets)
SA (6 octets)
ETPID (2 octets)
TCI, CFI=C or R (2 octets)
Len (2 octets)
(E-RIF if CFI=R; 2 octets NCFI = C or N)
LLC + C-Data or N-Data (3–1500 octets—C-Data; 3–1498 octets—N-Data)
PAD (39–0 octets)
FCS (4 octets, recomputed)
(Min 64, Max 1522 octets)
L-C-R/R,U
L-N-R/R,U
RCI (1/2 octets)
DA (6 octets)
SA (6 octets)
RIF (2–30 octets)
LLC + C-Data or N-Data (3–1470 octets)
FCS (4 octets)
(Max 1517 octets on FDDI, 1518 octets on Token Ring)
L-C-R/C,T
L-N-R/C,T
DA (6 octets)
SA (6 octets)
ETPID (2 octets)
TCI, CFI=R (2 octets)
Len (2 octets)
E-RIF (2–30 octets) NCFI= C or N
LLC + C-Data or N-Data (3-1470 octets)
PAD (39–0 octets)
FCS (4 octets, recomputed)
(Min 64, Max 1522 octets)

Figure C-13—Translation between L-X-X/R,U and L-X-X/C,T

This form of tagging causes the original frame size to be increased by 5 or 6 octets.

Figure C-14 illustrates the translation between an untagged LLC-encoded frame on Token Ring/FDDI (L-C-T/R,U, L-N-T/R,U, L-C-R/R,U, or L-N-R/R,U) and a tagged frame on 8802-5 Token Ring (L-C-T/R,T, L-N-T/R,T, L-C-R/R,T, or L-N-R/R,T). Figure C-14 also illustrates the translation of L-C-T/R,U, L-C-R/R,U, and L-N-R/R,U to tagged frames on FDDI media, the latter two translations illustrating the source-routed form of the tagged frame on FDDI.

Tagging requires the following frame translations:

j) Copy the RCI field;
k) The DA and SA fields carry the same MAC Addresses as in the original frame;

l) Copy the RIF field if present;
m) Insert STPID and TCI, setting the CFI to N or C appropriately;
n) Copy the LLC field;
o) Copy the Data field;
p) Recompute the FCS.

Removal of the tag (tagged frame to native Token Ring/FDDI frame) involves the reverse of this process.

NOTE 2—When removing the tag, if the CFI indicates that embedded address information is in a form inappropriate for the destination MAC method, then it is necessary either to translate the address information or to discard the frame.

This form of tagging causes the original frame size to be increased by 10 octets for FDDI and Token Ring.

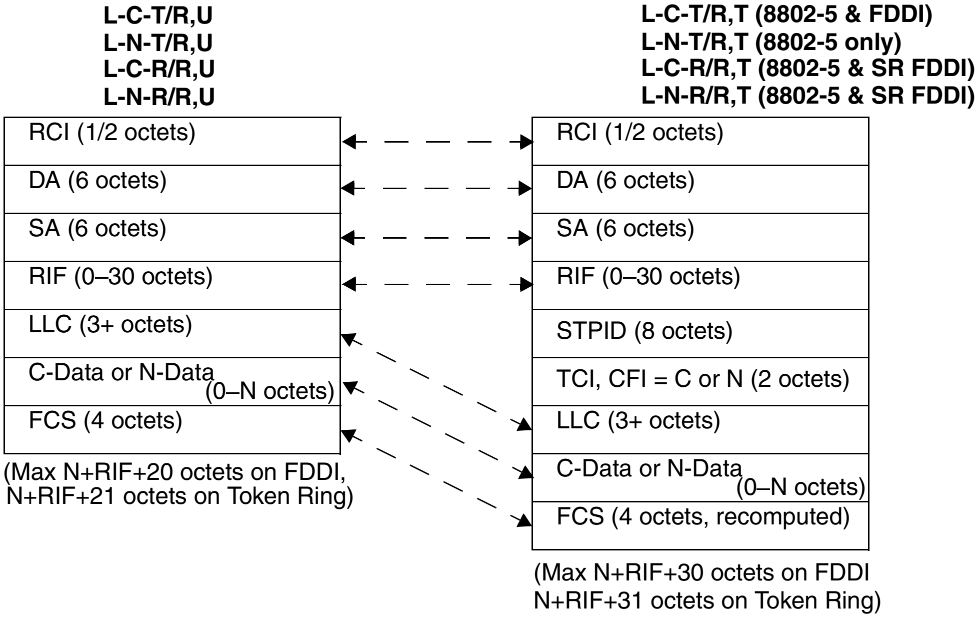

L-C-T/R,U
L-N-T/R,U
L-C-R/R,U
L-N-R/R,U
L-C-T/R,T (8802-5 & FDDI)
L-N-T/R,T (8802-5 only)
L-C-R/R,T (8802-5 & SR FDDI)
L-N-R/R,T (8802-5 & SR FDDI)
RCI (1/2 octets)
DA (6 octets)
SA (6 octets)
RIF (0–30 octets)
LLC (3+ octets)
C-Data or N-Data (0–N octets)
FCS (4 octets)
(Max N+RIF+20 octets on FDDI, N+RIF+21 octets on Token Ring)
RCI (1/2 octets)
DA (6 octets)
SA (6 octets)
RIF (0–30 octets)
STPID (8 octets)
TCI, CFI = C or N (2 octets)
LLC (3+ octets)
C-Data or N-Data (0–N octets)
FCS (4 octets, recomputed)
(Max N+RIF+30 octets on FDDI N+RIF+31 octets on Token Ring)

Figure C-14—Translation between L-X-X/R,U and L-X-X/R,T (8802-5 & SR FDDI)

Figure C-15 illustrates the translation between an untagged LLC-encoded frame on Token Ring/FDDI (L-N-T/R,U, L-C-R/R,U, or L-N-R/R,U) and a tagged frame on 8802-5 Token Ring (L-C-T/R,T, L-N-T/R,T, L-C-R/R,T, or L-N-R/R,T). Note that the translation of L-C-T/R,U to L-C-T/R,T was dealt with in Figure C-14.

Tagging requires the following frame translations:

q) Copy the RCI field;
r) The DA and SA fields carry the same MAC Addresses as in the original frame, but with RII reset regardless of its state in the original frame;
s) Insert STPID and TCI, setting the CFI to R;
t) Translate the RIF field if present (RII set in source frame) to the E-RIF, otherwise create an E-RIF with RT indicating a transparent frame. Set the NCFI to C or N appropriately;
u) Copy the LLC field;
v) Copy the Data field;
w) Recompute the FCS.

Removal of the tag (tagged frame to native Token Ring/FDDI frame) involves the reverse of this process.

NOTE 3—When removing the tag, if the NCFI indicates that embedded address information is in a form inappropriate for the destination MAC method, then it is necessary either to translate the address information or to discard the frame.

This form of tagging causes the original frame size to be increased by 10 or 12 octets.

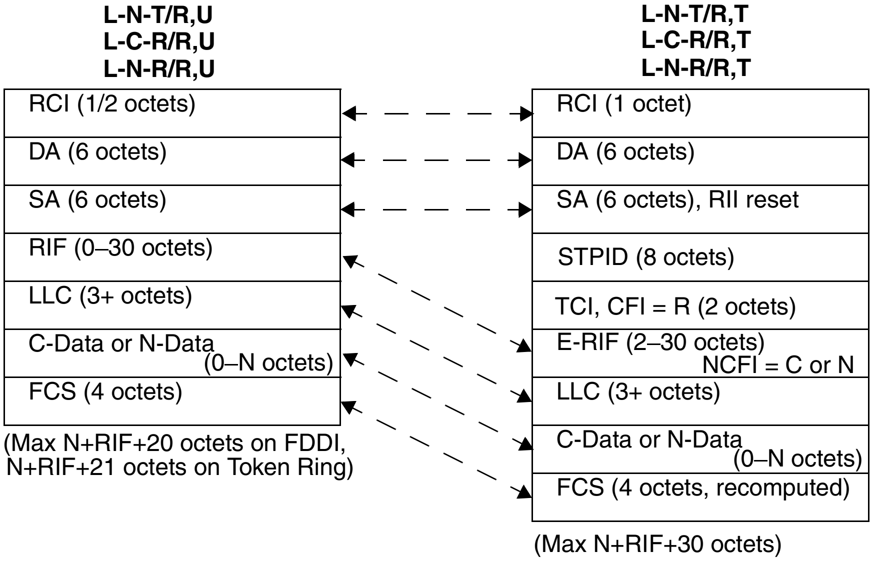


Figure C-15—Translation between L-X-X/R,U and L-X-X/R,T (transparent FDDI)

C.5.3 Translation of tagged frames during relaying

The following subclauses show the frame translations that can occur when a tagged frame is relayed from 802.3/Ethernet to Token Ring/FDDI and vice versa. The translations that occur between the transparent FDDI tagged frame format and the SR form on Token Ring/FDDI are also shown.

C.5.3.1 Tagged frames carrying Ethernet Type-encoded information

Figure C-16 illustrates the translation of tagged frames carrying Ethernet Type-encoded information between Token Ring/FDDI LANs and 802.3/Ethernet LANs.

Relaying Ethernet Type-encoded tagged frames from Token Ring/FDDI (SR form) to 802.3/Ethernet requires the following frame translations:

a) Remove the RCI field;
b) The DA and SA fields carry the same MAC Addresses as in the original frame, with the RII bit reset;
c) Replace the STPID with an ETPID;
d) The TCI field carries the same VID and Priority values as in the original frame (unless the relay function causes changes to user_priority or VID values). For E-C-T/C,T, CFI = C, otherwise CFI = R;
e) Convert the SPT field to a PT (IEEE Std 802.1H, IETF RFC 1042, and IETF RFC 1390 translation);
f) Copy the RIF, if present, into the tag header E-RIF. Create an E-RIF if the data type being carried is E-N-T/C,T. Set the NCFI in the E-RIF to C or N appropriately;
g) Copy the Data field;
h) Recompute the FCS.

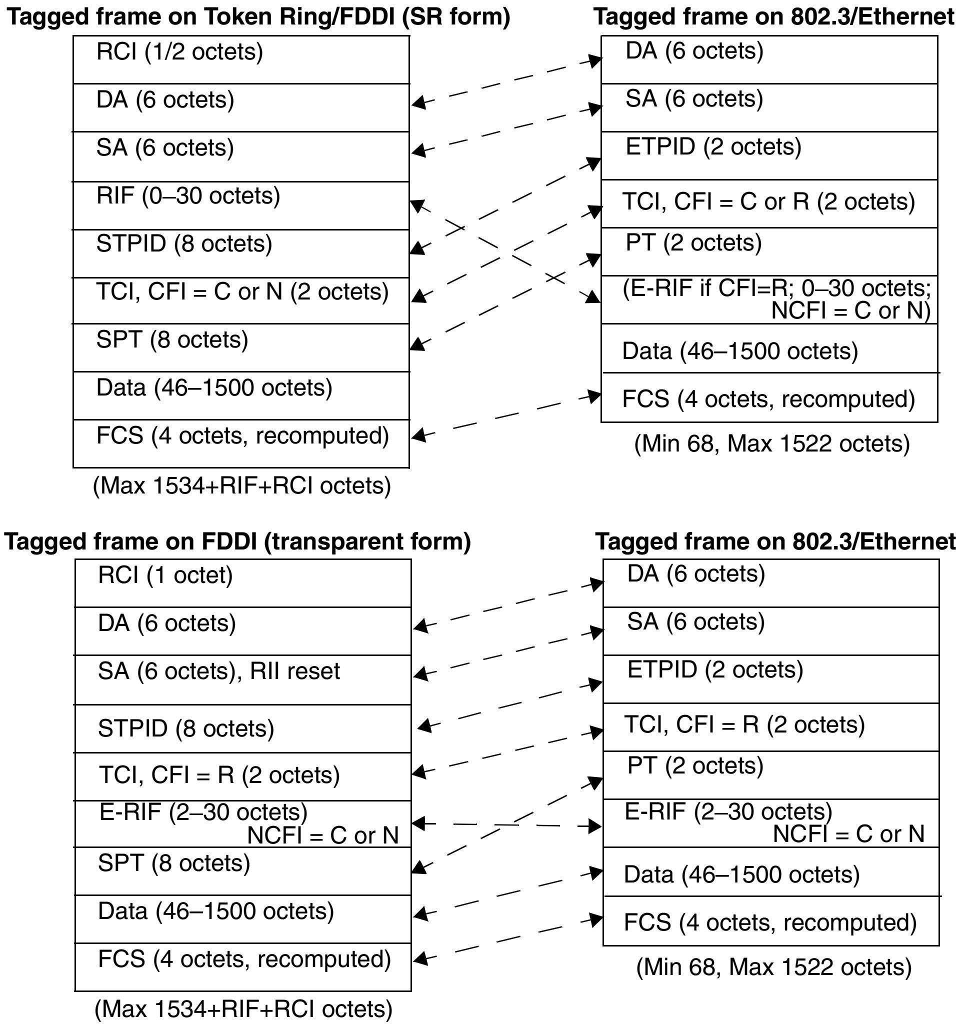


Figure C-16—Relaying Ethernet Type-encoded tagged frames

Relaying Ethernet Type-encoded tagged frames from FDDI (transparent form) to 802.3/Ethernet requires the following frame translations:

i) Remove the RCI field;
j) The DA and SA fields carry the same MAC Addresses as in the original frame, with the RII bit reset;
k) Replace the STPID with an ETPID;
l) The TCI field carries the same VID, Priority and CFI values as in the original frame (unless the relay function causes changes to user_priority or VID values);
m) Convert the SPT field to a PT (IEEE Std 802.1H, IETF RFC 1042, and IETF RFC 1390 translation);
n) Copy the E-RIF;
o) Copy the Data field;
p) Recompute the FCS.

Relaying from 802.3/Ethernet to Token Ring/FDDI involves the reverse of these processes.

C.5.3.2 Tagged frames carrying LLC-encoded information

Figure C-17 illustrates the translation of tagged frames carrying LLC-encoded information between Token Ring/FDDI LANs and 802.3/Ethernet LANs.

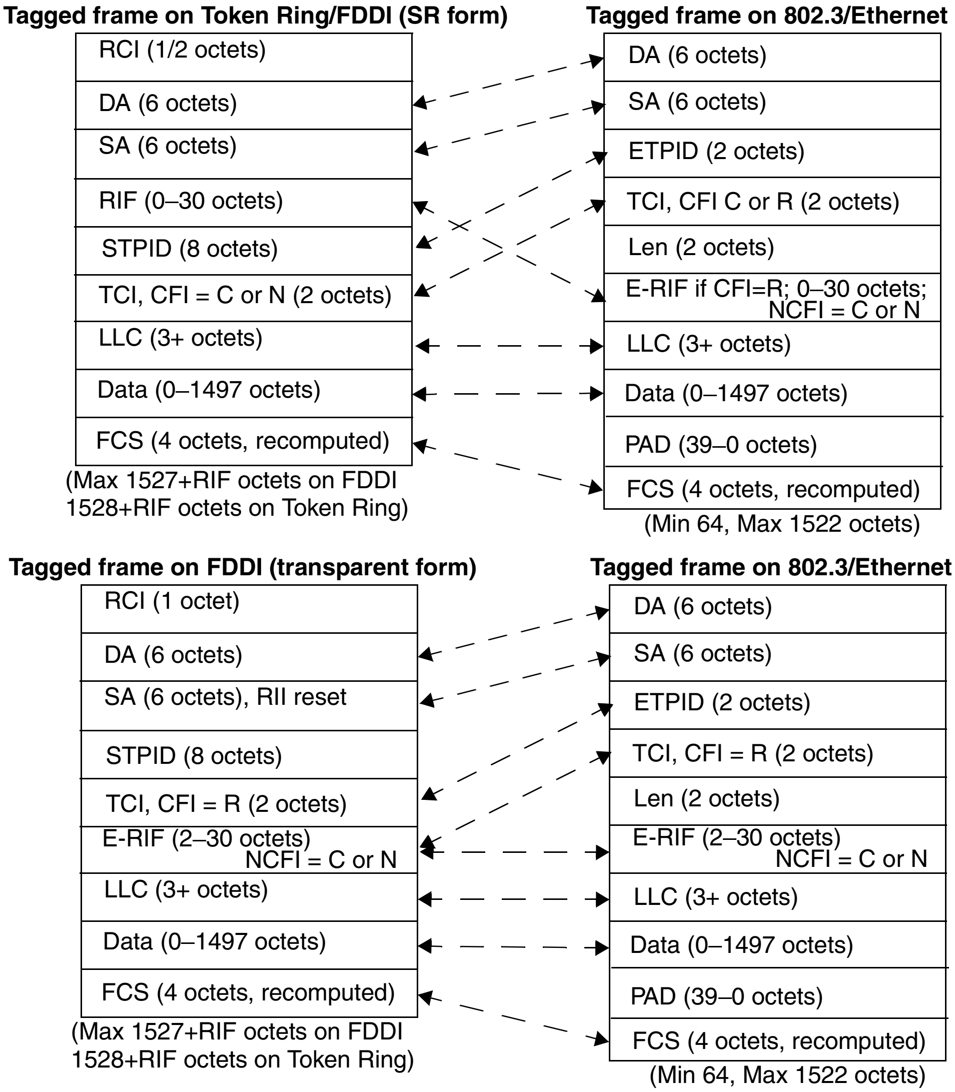


Figure C-17—Relaying LLC-encoded tagged frames

Relaying LLC-encoded frames in tagged format from Token Ring/FDDI (SR form) to 802.3/Ethernet requires the following frame translations:

a) Remove the RCI field;
b) The DA and SA fields carry the same MAC Addresses as in the original frame, with the RII bit reset;
c) Replace the STPID with an ETPID;
d) The TCI field carries the same VID and Priority values as in the original frame (unless the relay function causes changes to user_priority or VID values). For L-C-T/C,T, the CFI = C, otherwise CFI = R;
e) Insert a LEN field, equal to LLC+RIF (if present) +Data;

f) Copy the RIF, if present, into the tag header E-RIF. Create an E-RIF if the data type being carried is E-N-T/C,T. Set the NCFI bit to C or N appropriately;

g) Copy the LLC field;

h) Copy the Data field;

i) Recompute the FCS.

Relaying LLC-encoded frames in tagged format from FDDI (transparent form) to 802.3/Ethernet requires the following frame translations:

j) Remove the RCI field;

k) The DA and SA fields carry the same MAC Addresses as in the original frame, with the RII bit reset;

l) Replace the STPID with an ETPID;

m) The TCI field carries the same VID, Priority and CFI values as in the original frame (unless the relay function causes changes to user_priority or VID values);

n) Insert a LEN field, equal to E-RIF + LLC +Data;

o) Copy the E-RIF;

p) Copy the LLC field;

q) Copy the Data field;

r) Recompute the FCS.

Relaying from 802.3/Ethernet to Token Ring/FDDI involves the reverse of these process.

C.5.3.3 Translation between transparent FDDI format and SR format

Figure C-18 illustrates the translation of tagged frames between transparent FDDI format and the corresponding SR format on Token Ring/FDDI LANs. The translation shown applies to X-N-T/R,T, X-C-R/R,T and X-N-R/R,T frames only; other than translation of the RCI field, there is no translation required for X-C-T/R,T frames between Token Ring/FDDI LANs.

Relaying tagged frames from FDDI (transparent form) to Token Ring/FDDI (SR form) requires the following frame translations:

a) Translate the RCI field;

b) The DA and SA fields carry the same MAC Addresses as in the original frame, with the RII bit set or reset to reflect the presence or absence of source-routing information in the E-RIF;

c) Translate source-routing information, if any, from the E-RIF form to the RIF, with the NCFI bit reset;

d) Copy the STPID;

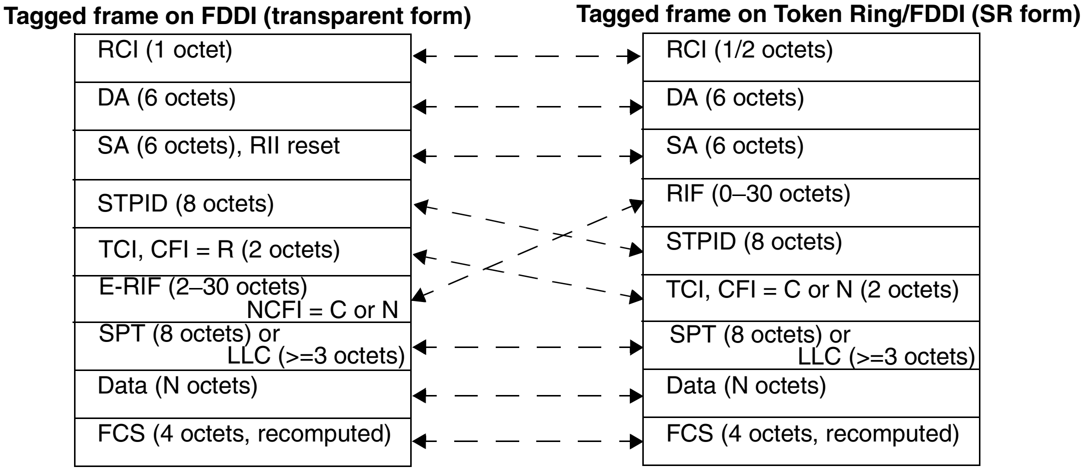
e) The TCI field carries the same VID and Priority values as in the original frame (unless the relay function causes changes to user_priority or VID values). The CFI is set to C or N to match the NCFI value in the E-RIF;

f) Copy the SPT or LLC field;

g) Copy the Data field;

h) Recompute the FCS.

Relaying from Token Ring/FDDI (SR form) to FDDI (transparent form) involves the reverse of these processes.



NOTE—Applies to X-N-T/R,T, X-C-R/R,T, and X-N-R/R,T frames only.

Figure C-18—Relaying tagged frames between transparent and SR forms

C.6 Field definitions

Subclauses C.6.1 through C.6.5 describe the field structures that correspond to some of the field names that appear in abbreviated form in the frame format diagrams in this standard.

NOTE—These fields are defined in other standards, and are not part of the additional specification required for the tagged frame format. They are included here in order to simplify the frame descriptions that appear in this standard, not in order to redefine their structure.

C.6.1 SNAP-encoded Protocol Type

The SNAP-encoded Protocol Type is eight octets in length, encoded in SNAP format. It consists of the standard SNAP header in the first three octets, followed by a SNAP PID consisting of the 00-00-00 OUI, followed by the Ethernet Type value to be encoded, as shown in Figure C-19.

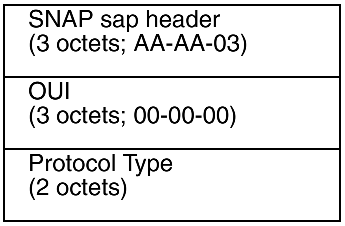


Figure C-19—SNAP-encoded Protocol Type format

C.6.2 Len

This is the IEEE Std 802.3 Length/Type field; for the Length interpretation, it may take any value that is less than or equal to 1500. Values that exceed 1535 are interpreted as Ethernet Types. Values that exceed 1500 but are less than 1535 are undefined.

C.6.3 C-Data and N-Data

This is the data field of the encapsulated frame:

a) N-Data refers to a data field that is carried in Canonical format regardless of the MAC method carrying the frame;
b) C-Data refers to a data field that is carried in Non-canonical format regardless of the MAC method carrying the frame.

C.6.4 RIF and E-RIF

The RIF is the Source-Routing Information Field, as defined in IEEE Std 802.1D, 1998 Edition, C.3.3.2. If the original (untagged) frame had a RIF, then the RIF field of the tagged frame takes its value.

The E-RIF is a modified form of the RIF that appears within the tag header in tagged frames on transparent LANs (802.3/Ethernet, and FDDI when used as a transparent LAN). The structure of the E-RIF is defined in 9.3.3.

C.6.5 PAD

Zero or more padding octets, as required in order for the minimum frame size to be at least 64 octets.

Annex D

(informative)

Background to VLANs

The term VLAN has many different definitions throughout the communications industry. The model of VLANs defined in this standard supports most of these views, although it may not be immediately obvious as to how they are supported. The goal of this annex it to take some of the common terms that have been used in the description of VLANs and relate them to the model presented in this standard.

D.1 Basic VLAN concepts

Figure 9-3 shows a simple example of a Port-based VLAN. For untagged traffic, VLAN membership is determined by one of several methods:

a) For a port on an access link that is using Port-based VLAN classification, membership is determined by the PVID assigned to the receiving Port.

b) For a port on an access link that is using Port-and-Protocol-based VLAN classification, membership is determined by the VID that is mapped from the link-layer protocol carried in the frame and is assigned to the receiving Port.

NOTE—Other criteria for VLAN membership, such as MAC Address, could be used, but these are beyond the scope of this discussion.

For this configuration there needs to be a way to convey the VLAN information between the two bridges. This is done by adding a VLAN tag to every frame that is sent between the two bridges; such frames are known as VLAN-tagged frames. This connection between the two bridges is commonly known as a *Trunk Link*.

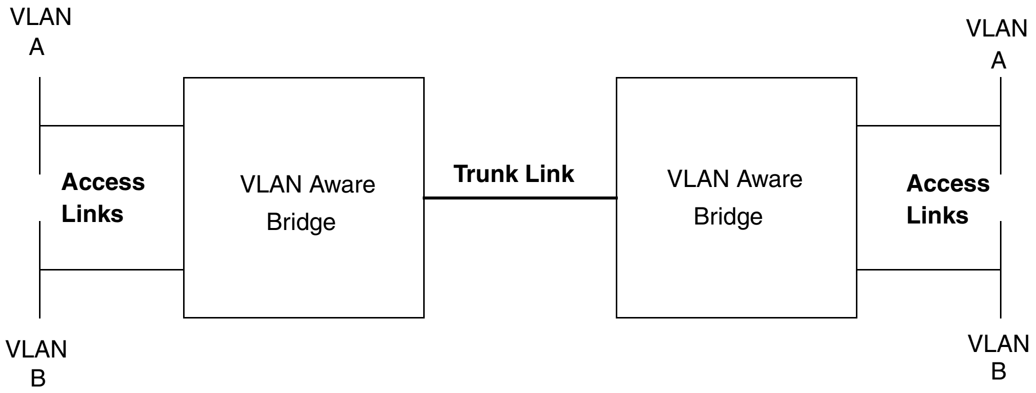


Figure D-1—Port-based VLANs

D.1.1 Trunk Links

A Trunk Link is a LAN segment used for multiplexing VLANs between VLAN Bridges. All the devices that connect to a Trunk Link must be *VLAN-aware*. VLAN-aware devices are devices that are able to understand VLAN membership and VLAN frame formats. Conversely, *VLAN-unaware* devices do not have an understanding of VLAN membership and VLAN frame formats. All frames, including end station frames, on a Trunk Link are VLAN-tagged, i.e., they carry a tag header that contains a non-null VLAN ID. Consequently, there are no VLAN-unaware end stations on an a Trunk Link. The Trunk Link in Figure 9-3 is a point-to-point LAN segment; there are therefore exactly two VLAN-aware Bridges attached to this Trunk. A Trunk Link could also be a shared medium LAN segment that has many VLAN-aware Bridges attached to it.

D.1.2 Access Links

The other links in Figure 9-3 are commonly known as *Access Links*. An Access Link is a LAN segment used to multiplex one or more VLAN-unaware devices into a Port of a VLAN Bridge. In simple terms this is an 802 LAN segment (IEEE Std 802.3, IEEE Std 802.5, etc.) with end stations attached, that is connected into a VLAN-aware Bridge. All frames on an Access Link carry no VLAN identification; i.e., there are no VLAN-tagged frames on an Access Link. Typically the Access Link is viewed as being on the edge of the VLAN network. The Access Link itself could consist of a number of LAN segments interconnected by IEEE Std 802.1D, 1998 Edition-conformant Bridges (this is termed a *legacy region* in E.1.2). Like the Access Link, there are no VLAN-tagged frames transmitted in this legacy region.

D.1.3 Hybrid Links

When VLAN-unaware end stations are added to a Trunk Link, the resultant link is commonly known as a Hybrid Link. A Hybrid Link is a LAN segment that has both VLAN-aware and VLAN-unaware devices attached to it. Consequently, a Hybrid Link can carry both VLAN-tagged frames and other (untagged or priority-tagged) frames. It must be borne in mind that, for a given VLAN, all frames transmitted by a given Bridge on a given hybrid link must be tagged the same way on that link. They must be either

a) All untagged; or
b) All tagged, carrying the same VLAN ID.

Note that a Bridge can transmit a mix of VLAN-tagged frames and untagged frames but they must be for different VLANs. In Figure D-2 all the frames for VLANs A and B are tagged on the hybrid link. All frames for VLAN C on the hybrid link are untagged.

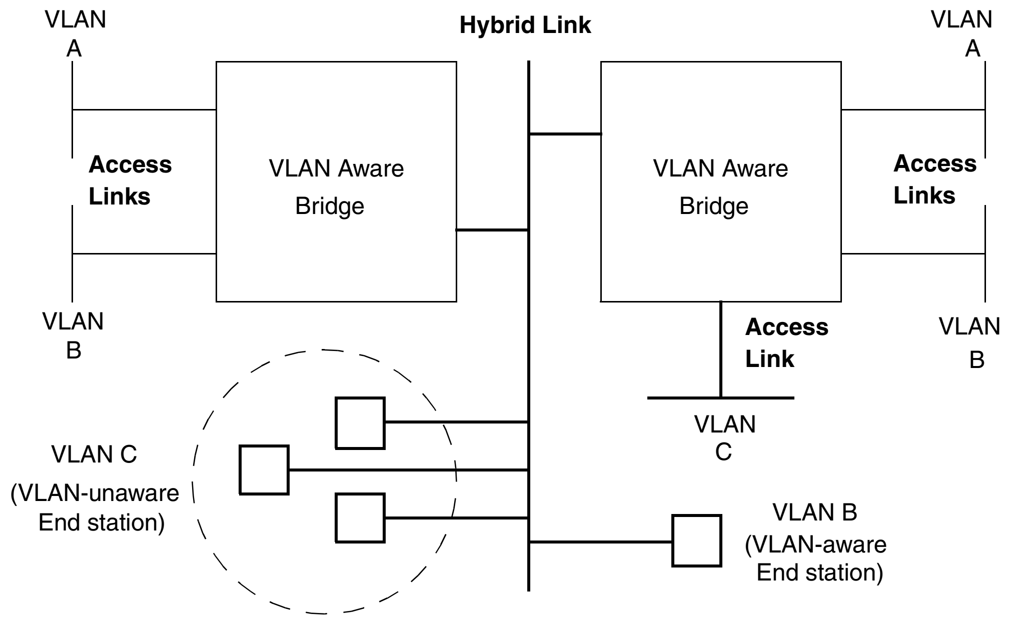


Figure D-2—Hybrid Links

On a Hybrid link the decision to tag or not to tag a frame is a function of the VLAN and not a function of the link itself, since both formats are allowed. The Hybrid link can be thought of as the general case of both Access and Trunk links.

D.2 Relationship of other VLAN styles to the Port-based VLAN model

D.2.1 Link types

A Hybrid link in which all frames are VLAN-tagged is a Trunk Link. Conversely, a Hybrid Link that has no VLAN-tagged frames is an Access Link. The distinction between Access and Trunk becomes less important in an actual implementation where all types of frames have to be handled. The Acceptable frame Types parameter (8.4.3) allows control to be exerted over the reception of frames that do not carry VLAN identification information (i.e., untagged and priority-tagged frames). Depending upon the value of the parameter, the decision is taken as to whether to discard the frame or to add in VLAN information. The more general implementation will allow all types of frames (VLAN-tagged, priority-tagged, and untagged) to be present, hence all links are conceptually Hybrid Links.

The VLAN model defined in this standard essentially takes the view that all links are Hybrid Links. It is then up to the system administrator to determine, through appropriate application of the management functions available in the Bridge, whether all links remain operating as Hybrid Links, or whether particular links need to be configured as Access Links or Trunk Links, in other words, to explicitly configure some Ports to discard untagged and priority-tagged frames. This allows the description of the functionality of the Bridge to be kept relatively simple, while retaining the practical implementation and configuration mechanisms necessary in order for the other link types to be derived. The Acceptable frame Types parameter does not allow configuration of a Port such that VLAN-tagged frames are discarded; hence, from the practical point of view, the distinction between an Access Link and a Hybrid Link is not one of Port configuration; it is simply determined by the presence or absence of other devices on that link that generate VLAN-tagged frames.

D.2.2 Use of other VLAN styles

This standard defines the following classification rules:

a) A Port-based tagging rule, whereby all untagged and priority-tagged frames received by a Port are classified as belonging to the VLAN whose VID (the PVID) is associated with that Port.
b) A Port-and-Protocol-based rule, whereby all untagged frames received by a Port are classified, by inspecting the data-link layer framing and upper-layer protocol type, as belonging to one of the VLANs whose VID is associated with that Port.

The Port-based style of operation should be viewed as the base level of a possible hierarchy of VLAN styles, each one able to classify untagged frames according to particular ingress rules. Examples of some other ingress rules might include the following:

c) MAC Address-based classification; e.g., associating a set of MAC Addresses with a given VLAN ID in order to define the membership of the VLAN;
d) Subnet-based classification; e.g., allocating VLAN membership on the basis of IP subnet addressing characteristics of frames.

For a given implementation, such rules might form a natural hierarchy; e.g., IP Subnet-based tagging might take the highest priority. If the packet is not an IP packet, then tagging is based on the protocol being used: IPX or LAT. If some other protocol is in use (not IP, IPX, or LAT), then the classification is based on MAC Addresses. If the addresses in the frame do not match the address-based classifications that are configured, then the Port-based rule is applied.

The result of such a hierarchy is that a given ingress rule defines the default that is applied if the higher priority rule fails to classify the frame, with the Port-based rule forming the lowest level, “catch-all” default.

NOTE—Clearly, if a given rule in the hierarchy is able to classify all possible frames, then all rules below that point in the hierarchy are effectively disabled.

The addition of further ingress rules in 802.1Q Bridges could be achieved

e) As proprietary extensions to the existing specification;
f) As future standardized extensions.

Given that the starting point for this standard is that all links are Hybrid Links, there is no need for such additional classification and tagging functionality to exist within the Bridges themselves; it would, for example, be possible to develop “tagging engines” that are capable of implementing more complex classifications than Port-based classification, and which are placed between the 802.1Q Bridge Port and an Access Link. Such a device would provide a richer functionality in terms of VLAN classification style, while remaining compatible with Port-based VLAN operation.

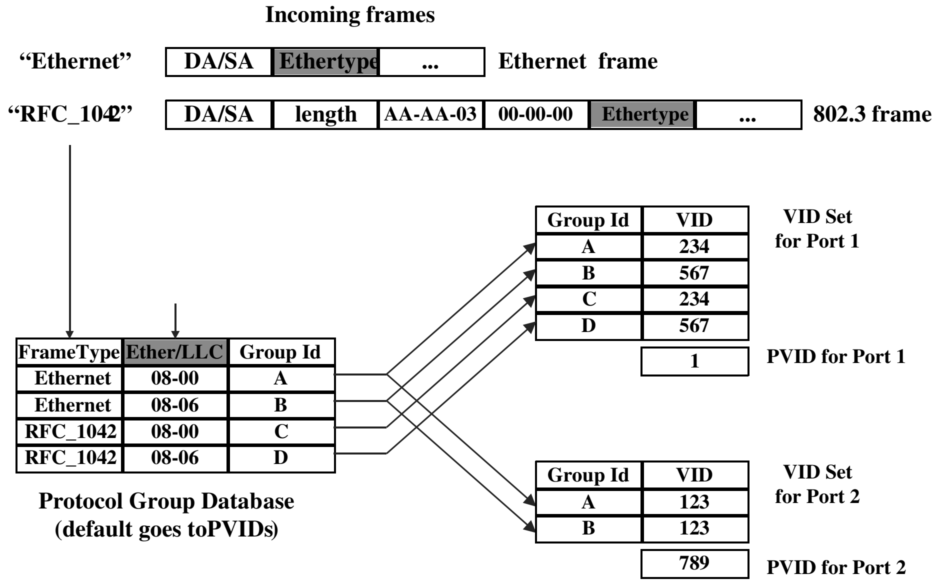
D.3 Example configurations for the Port-and-protocol-based VLAN model

One example scenario for the configuration of Port-and-Protocol based VLAN classification was shown in Figure 8-8. That example showed the use of Protocol Groups to tie together Protocol Templates for common treatment on several Ports of a bridge. Some further examples are presented here.

D.3.1 Example of per-protocol control

In the example in Figure D-3, Protocol Templates are used individually without any groupings. This method allows maximum granularity of control by mapping individual IEEE 802.3 Type values to their own VIDs. Here, each Protocol Template is assigned its own Protocol Group Identifier; each port's VID Set is set up to group protocols together into the same VLAN, as required. This allows a different grouping of protocols on different Ports of the bridge: here, Port 1 classifies Type-encapsulated 802.3-format IP and RFC 1042-format IP packets to the same VLAN (VID 234) and it classifies Type-encapsulated 802.3-format ARP and RFC 1042-format ARP packets to a different VLAN (VID 567). At the same time, Port 2 treats Type-encapsulated 802.3-format ARP and IP packets the same and classifies them to the same VLAN (VID 123) whilst treating RFC 1042 IP and ARP packets the same and using the default treatment to classify them to a different VLAN (VID 789, the PVID for this Port). The tradeoff here is that there is more per-Port configuration required to achieve this level of control and it can use up a larger number of VID Set entries, which may be a scarce resource in a Bridge, to achieve the desired result.

–



FrameType	Ether/LLC	Group Id
Ethernet	08-00	A
Ethernet	08-06	B
RFC_1042	08-00	C
RFC_1042	08-06	D

Group Id	VID
A	234
B	567
C	234
D	567
	1

Group Id	VID
A	123
B	123
	789

Figure D-3—.Example of control down to the individual protocol level

D.3.2 Protocol-based frame filtering

Port and Protocol-based VLAN classification mechanism can also be used to set up protocol-based filters for selectively discarding frames based upon their upper-layer protocol: either of the approaches described above, using Protocol Groups or else using individual Protocol Templates, can be used to classify frames to a particular VLAN, created specially for the purpose. If the VLAN is configured to have no members in its Member Set, then any frames directed there will effectively be discarded. Note that this approach does use up a VID Set entry on each Port for which the filter is desired; this may favor using the Protocol Group approach.

Additionally, this mechanism can be extended for security analysis. Instead of totally discarding the frames that match protocol-based filters, a "mirroring" Port can be added to the Member Set for the filtering VLAN: this has the effect of trapping all frames which fail some sort of protocol-based access control test for future analysis by an analysis probe located on the mirroring Port.

Annex E

(informative)

Interoperability considerations

VLAN-aware Bridges that conform to this standard are able to interoperate in Bridged LANs with other VLAN-aware Bridges. However, the VLAN-based filtering service defined in this standard, as provided in the context of a single spanning tree for the Bridged LAN, involves some constraints on the network topology and individual device configurations that differ from the set of constraints that apply to the building and configuration of Bridged LANs based only on IEEE Std 802.1D, 1998 Edition.

In addition, VLAN-aware Bridges are able to interoperate with Bridges conformant with the IEEE Std 802.1D, 1998 Edition specification (or with the earlier IEEE Std 802.1D specification), as well as with both priority-aware and VLAN-aware end systems. Both the VLAN based filtering service and the tag insertion and removal service of 802.1Q cause constraints on intermixed network topologies and device configurations that again differ from the building and configuration of IEEE Std 802.1D, 1998 Edition standard networks.

The implications of certain device configurations may not be immediately apparent from the technical detail of this standard. In order to clarify the nature of the additional constraints, the following subclauses

a) Describe the basic requirements for interoperability;
b) Discuss those requirements in the context of homogeneous and heterogeneous configurations, with examples of some of the problems that can occur if these requirements are not adhered to.

E.1 Requirements for interoperability

There are two primary aspects of the configuration of a Bridged LAN that are of concern from the point of view of interoperability:

a) Establishing a consistent view of the static filtering configuration of Bridges in the Bridged LAN;
b) Ensuring that untagged frames are VLAN-tagged (and that the tag is subsequently removed) consistently regardless of Spanning Tree reconfigurations.

E.1.1 Static filtering requirements

Static filtering controls allow the network administrator to impose a level of control over the permitted connectivity in the Bridged LAN, by setting static MAC Address filters in the Filtering Databases of Bridges, and by controlling the extent of particular VLANs by manipulation of Static VLAN Registration Entries (8.11.2).

In order to ensure that end station to end station connectivity (or the lack of it) is consistent in all possible Spanning Tree configurations, any static filters need to be established taking account of the full mesh topology of the physical interconnections between Bridges in the Bridged LAN, not just the “normal” Spanning Tree topology to which the network is configured when all Bridges and LAN segments are operating correctly. An example of the consequences of failure to establish consistent controls for static VLAN filtering is given in E.2.1.

E.1.2 Configuration requirements for VLAN-tagging

802.1Q Bridges classify incoming untagged frames by applying either a Port-based tagging rule on ingress that uses the PVID for the receiving Port as the VLAN classification for such frames or a Port-and-Protocol-based rule that uses the frame's upper-layer protocol to select one of a port's VIDs. Maintaining consistent connectivity between any pair of end stations that are on the same VLAN, and where one or both of those end stations is VLAN-unaware, requires that

a) All VLAN-aware Bridge Ports that are connected to the same LAN segment apply a consistent set of ingress rules (8.6);
b) All VLAN-aware Bridge Ports that are connected to the same *legacy region* of a Bridged LAN apply a consistent set of ingress rules;
c) All VLAN-aware Bridge Ports that serve LAN segments to which members of the same VLAN are (or can be) attached apply a consistent set of ingress rules.

A legacy region of a Bridged LAN consists of any set of LAN segments that are physically interconnected via VLAN-unaware, IEEE Std 802.1D, 1998 Edition Bridges. A legacy region has the property that, by appropriate configuration of the Spanning Tree, a Spanning Tree path could be created between any pair of LAN segments in the region such that the path would pass only through VLAN-unaware Bridges.

NOTE—In case b), Spanning Tree reconfiguration within the legacy region can change the logical connectivity between the VLAN Ports and the LAN segments that they (directly or indirectly) serve. Hence, a Spanning Tree reconfiguration could result in any end stations connected to the legacy region being serviced via any of the VLAN-aware Ports. In effect, such a reconfiguration reduces case b) to case a). Figure E-2 and Figure E-3 give examples of this type of configuration. In Figure E-2, the legacy region consists of all three LAN segments and both IEEE Std 802.1D, 1998 Edition Bridges. In Figure E-3, the legacy region consists of the IEEE Std 802.1D, 1998 Edition Bridge and both LAN segments to which it is attached. An example of case c) is where an end station attached to a leaf LAN segment is in the same VLAN as a server that is attached to a distinct LAN segment, i.e., all possible Spanning Tree paths between the two stations pass through a VLAN-aware region of the Bridged LAN.

The essence of what these rules express is that if a given untagged frame belongs on a given VLAN, then the classification and tagging behavior of any VLAN-aware Bridges that are required to tag that frame needs to be the same, regardless of the logical connectivity that is created by the Spanning Tree configuration of the Bridged LAN. Examples of the consequences of failure to apply these rules appear in E.3 and E.6.

E.2 Homogenous 802.1Q Bridged LANs

This standard requires new considerations in building a Bridged LAN in which all Bridges are VLAN-aware. The arbitrary plug and play capability of IEEE Std 802.1D, 1998 Edition in creating a network topology is restricted when making use of the VLAN extensions defined in this standard.

E.2.1 Consistency of static VLAN filtering

In order for stations that are members of a given VLAN to be able to reach other members of the same VLAN elsewhere in the Bridged LAN, all Ports that are part of the Spanning Tree active topology (i.e., all Ports that are in a forwarding state) connecting the stations must be included in the Member Set (8.11.9) for the given VLAN. In order for this connectivity to be independent of any reconfiguration of the Spanning Tree topology, all paths among those stations, both forwarding and blocked, must have this characteristic. Use of management controls to manipulate the Member Set (e.g., filters for security) must be applied in a manner consistent with requirements of the full mesh topology of the Bridged LAN.

An inconsistency occurs, for example, if a VLAN is restricted from an active path, but not from a redundant path currently blocked by the operation of Spanning Tree. Should a Spanning Tree reconfiguration enable the previously blocked path, the restriction will no longer be in place. In the reverse, a Spanning Tree reconfiguration may suddenly impose a restriction that had not previously existed. A common use of such management restriction will likely arise from managers who make use of an "access" port construct. An access port may be a port which is absent from the Member Set (8.11.9) in all VLANs but the untagged, default VLAN. Should such an access port become the active connection between two portions of the Bridged LAN as a result of a Spanning Tree reconfiguration, all VLANs but that one will be partitioned at that point in the topology.

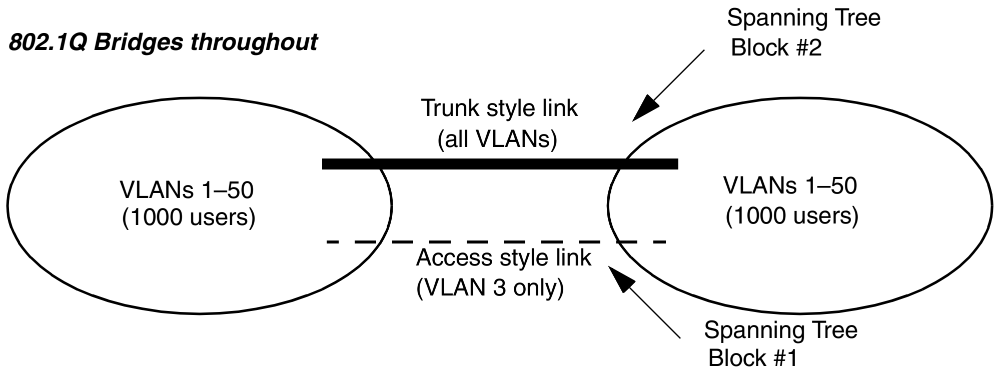


Figure E-1—Static filtering inconsistency

In Figure 6-1, the trunk style link and access style link cause a loop through the left and right portions of the network. STP will block one or the other. Should the Spanning Tree block at point #1, all 2000 users may communicate on any of the 50 VLANs. However, should the Spanning Tree block at point #2, the left and right portions of the network will be partitioned on all VLANs excepting VLAN 3 (the VLAN carried via the access style link.)

E.2.2 Consistent view of the "untagged VLAN(s)" on a given LAN segment

In the Port-based VLAN model defined in this standard, the PVID for a Port provides the VLAN classification for all frames on the attached LAN segment that are received in untagged form. Any LAN segment with more than one 802.1Q Bridge attached has such an "untagged VLAN" for each Bridge. No explicit requirement that these be consistent for all Bridges on the same LAN segment, nor mechanism to assure such, has been included in this standard.

Similarly, in the Port-and-Protocol-based VLAN model defined in this standard, one of the members of the VID Set for a Port provides the VLAN classification for all frames on the attached LAN segment that are received in untagged form with a particular upper-layer protocol. Any LAN segment with more than one such 802.1Q Bridge attached has such a set of "untagged VLANs" for each Bridge. No explicit requirement that these be consistent for all Bridges on the same LAN segment, nor mechanism to assure such, has been included in this standard.

Consider the case of a LAN segment to which are attached three VLAN-aware Bridges, each of which is using Port-based classification and is capable of transmission of untagged frames onto the LAN segment. An untagged frame placed on that segment by any one of the Bridges will be associated by each of the other two Bridges with their own configured PVID for their receiving port on that LAN. The 802.1Q VLAN model requires that each frame have a unique VLAN association, and that association is represented by a single, global VID value. Therefore, it follows that all 802.1Q Bridges on that LAN segment must make use of the same classification rules (in this case, the same PVID) for their ports connected to that LAN segment.

It has been suggested that in the special case of a direct point-to-point connection between two 802.1Q Bridges or other VLAN-aware devices, other rules might apply. No mechanism for identifying such links has yet been suggested.

This creates a configuration challenge for installers of Bridges that conform to this standard. Initial management configuration of the Bridges (the setting of PVIDs, VID Sets and Protocol Group Databases) must be made consistent among the Bridges, in a manner that takes into account the actual physical topology. Changes to the physical topology may require specific changes to the configuration of all affected switches. These requirements effectively disallow a plug-and-play installation as supported by IEEE Std 802.1D Bridged LANs, unless all Bridges are left with their default Port-based classification rules and with each PVID = 1.

E.3 Heterogeneous Bridged LANs: intermixing IEEE Std 802.1D (D) and 802.1Q (Q) Bridges

This clause discusses networks in which VLAN-aware Bridges that conform to this standard are intermixed with VLAN-unaware Bridges conformant to IEEE Std 802.1D, 1998 Edition.

A principal limitation in intermixing Q Bridges with D Bridges is that the VLAN filtering services are not universally available throughout the Bridged LAN. Also, services for the insertion and removal of tags are not universally available. Further, spanning tree reconfigurations may cause filtering services, as well as tag insertion and removal services, to become available or become unavailable independent of actions of affected users.

E.3.1 Example: Adding an 802.1Q Bridge to provide filtering to an IEEE Std 802.1D network

Example problems can be shown with the following topology diagrams. Figure E-2 includes one Q Bridge and two D Bridges:

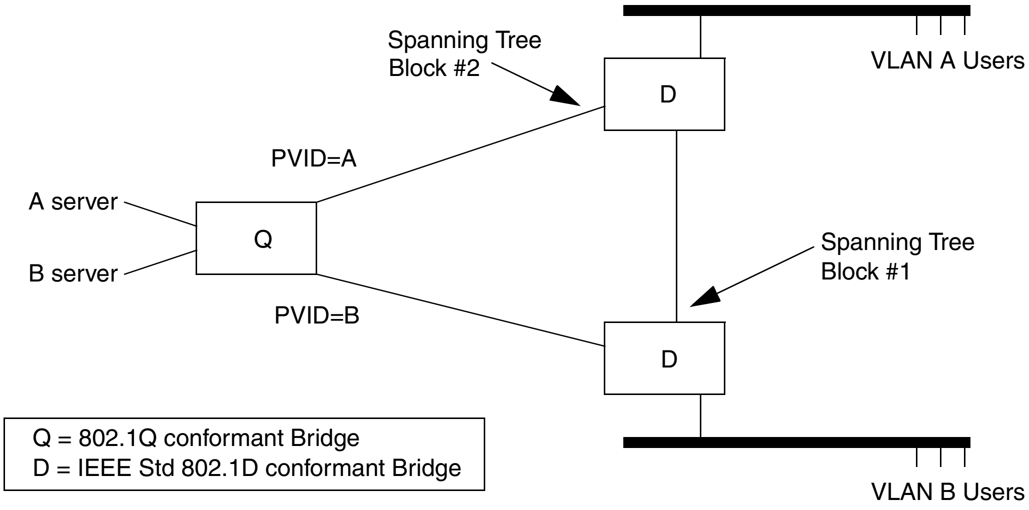


Figure E-2—Interoperability with IEEE Std 802.1D Bridges: example 1

If the Spanning Tree protocol determines to break the loop among the three Bridges by blocking at point #1, connectivity within each VLAN is as desired. However, should the block occur at point #2, traffic from VLAN A users will pass through both D Bridges, and be treated as VLAN B traffic upon arrival in the Q Bridge. Connectivity to the A server will be lost for the A users.

E.3.2 Example: Adding an IEEE Std 802.1D Bridge to a (previously) Homogenous 802.1Q Network

A similar problem, demonstrating the impact of placing a D Bridge within an otherwise homogenous Q topology, can be shown by the configuration in Figure E-3. Here we include two Q Bridges and add a single redundant D Bridge:

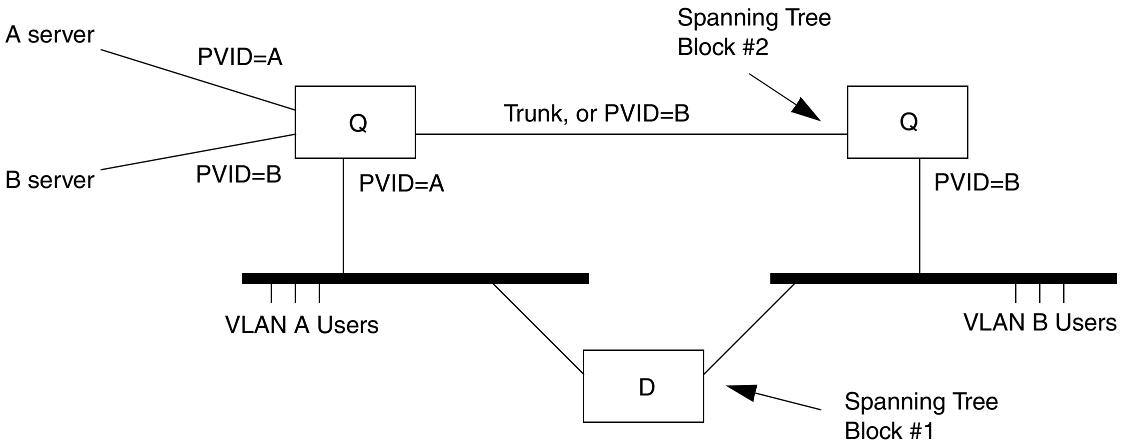


Figure E-3—Interoperability with IEEE Std 802.1D Bridges: example 2

If STP determines to break the loop among the three Bridges by blocking at point #1, connectivity within each VLAN is as desired. The two Q switches operate as expected. A and B VLAN frames are VLAN-tagged on arrival in either Q Bridge, and forwarded only to the appropriate servers. Now suppose an STP reconfiguration results in a block at point #2, but not at #1. This redirects VLAN B user traffic through the IEEE Std 802.1D Bridge. VLAN B users no longer have their traffic identifiably distinct from VLAN A. An immediate consequence is that the VLAN B users will no longer have access to the “B server.”

E.4 Heterogeneous Bridged LANs: intermixing IEEE Std 802.1H and 802.1Q Bridges

Translating Bridges (i.e., Bridges that relay between Token Ring/FDDI and 802.3/Ethernet LANs) that implement the encapsulation techniques described in IEEE Std 802.1H, IETF RFC 1042, and IETF RFC 1390 can be intermixed with 802.1Q Bridges under certain limited conditions. In order to understand the limitations involved, it is necessary to describe what happens to the various tagged frame formats when passed through a Translating Bridge. The frame formats shown in the following subclauses use the notation that is defined in Annex C.

NOTE—The examples shown are not exhaustive; in particular, the examples do not make use of the transparent tagged frame format on FDDI. However, the examples illustrate the nature of the problems that can occur with such translations.

E.4.1 LLC-encoded tagged frames relayed from 802.3/Ethernet to Token Ring or source-routed FDDI

A Transparent LLC-encoded frame on 802.3/Ethernet has the following form:

L-C-T/C,T: DA, SA, ETPID, TCI (CFI reset), LEN, LLC, C-Data, PAD, FCS

The Translating Bridge will recognize the ETPID as an Ethernet Type that requires translation into an SPT. This effectively translates the ETPID to an STPID. The translated frame therefore appears as follows:

RCI, DA, SA (RII reset), STPID, TCI (CFI reset), LEN, LLC, C-Data, PAD, FCS

Whereas a true translation of the tagged frame via an 802.1Q Bridge would result in

L-C-T/R,T: RCI, DA, SA (RII reset), STPID, TCI (CFI reset), LLC, C-Data, FCS

As can be seen, the resultant frame superficially appears to be a valid tagged frame; however, it includes spurious LEN and PAD fields, and is therefore not a valid tagged frame.

A source-routed LLC-encoded frame on 802.3/Ethernet has the following form:

L-C-R/C,T: DA, SA, ETPID, TCI (CFI set), LEN, RIF (NCFI=C), LLC, C-Data, PAD, FCS

The Translating Bridge generates

RCI, DA, SA (RII reset), STPID, TCI (CFI set), LEN, RIF (NCFI=C), LLC, C-Data, PAD, FCS

Whereas a true translation of the tagged frame via an 802.1Q Bridge would result in

L-C-R/R,T: RCI, DA, SA (RII set), RIF, STPID, TCI (CFI reset), LLC, C-Data, FCS

Again, the resultant frame superficially appears to be a valid tagged frame; however, it includes spurious LEN, RIF and PAD fields, the RIF information is not visible to any source routing Bridges attached to the Ring medium, and the CFI indicates Non-canonical data where the actual data is Canonical.

NOTE—Similar problems exist for the other two LLC-encoded formats, L-N-R and L-N-T, but the CFI correctly indicates Non-canonical data in the translated frame.

In both cases, the effect of the Translating Bridge is symmetrical; passing this invalid frame back through the Translating Bridge restores it to a valid tagged frame format on 802.3/Ethernet.

Consequently,

a) The translated frames cannot be correctly interpreted by VLAN-aware end stations attached to the Ring medium;
b) In both cases, any 802.1Q Bridge that attempts to untag the translated frames will generate invalid untagged frames;
c) Any 802.1Q Bridge that attempts to relay the translated frames back onto Token Ring/FDDI will generate invalid tagged frames;
d) In the case of the source-routed frame, any source routing Bridges attached to the Ring will treat the frame as a transparent frame;

e) As the effect of the Translating Bridge is symmetric, passing the frame through an even number of such translations before any 802.1Q device attempts to interpret the frame format results in correct operation of the 802.1Q devices.

E.4.2 Ethernet Type-encoded tagged frames relayed from 802.3/Ethernet to Token Ring or source-routed FDDI

A Transparent Ethernet Type-encoded frame on 802.3/Ethernet has the following form:

E-C-T/C,T: DA, SA, ETPID, TCI (CFI reset), PT, C-Data, FCS

The Translating Bridge generates:

RCI, DA, SA (RII reset), STPID, TCI (CFI reset), PT, C-Data, FCS

Whereas a true translation of the tagged frame via an 802.1Q Bridge would result in

E-C-T/R,T: RCI, DA, SA (RII reset), STPID, TCI (CFI reset), SPT, C-Data, FCS

The resultant frame is superficially a valid tagged frame, but carries an untranslated Ethernet Type where there should be a SNAP-encoded Ethernet Type.

A source-routed Ethernet Type-encoded frame on 802.3/Ethernet has the following form:

E-C-R/C,T: DA, SA, ETPID, TCI (CFI set), PT, RIF (NCFI=C), C-Data, FCS

The Translating Bridge generates

RCI, DA, SA, STPID, TCI (CFI set), PT, RIF (NCFI=C), C-Data, FCS

Whereas a true translation of the tagged frame via an 802.1Q Bridge would result in

E-C-R/R,T: RCI, DA, SA (RII set), RIF, STPID, TCI (CFI reset), SPT, C-Data, FCS

Again, the resultant frame is superficially a valid tagged frame, but carries an untranslated Ethernet Type where there should be a SNAP-encoded Ethernet Type, and carries a spurious RIF field. The CFI is also incorrect.

NOTE—Similar problems exist for the other two Ethernet Type-encoded formats, E-N-R and E-N-T, but the CFI correctly indicates Non-canonical data in the translated frame.

In both cases, the effect of the Translating Bridge is symmetrical; passing this invalid frame back through the Translating Bridge restores it to a valid tagged frame format on 802.3/Ethernet.

Consequently,

a) The translated frames cannot be correctly interpreted by VLAN-aware end stations attached to the Ring medium;
b) Any 802.1Q Bridge that attempts to untag the translated frames will generate invalid untagged frames;
c) Any 802.1Q Bridge that attempts to relay the translated frames back onto Token Ring/FDDI will generate invalid tagged frames;

d) In the case of the source-routed frame, any source routing Bridges attached to the Ring will treat the frame as a transparent frame;
e) As the effect of the Translating Bridge is symmetric, passing the frame through an even number of such translations before any 802.1Q device attempts to interpret the frame format results in correct operation of the 802.1Q devices.

E.4.3 LLC-encoded tagged frames relayed from Token Ring or source-routed FDDI to 802.3/Ethernet

A Transparent LLC-encoded frame on Token Ring/FDDI has the following form:

L-C-T/R,T: RCI, DA, SA (RII reset), STPID, TCI (CFI reset), LLC, C-Data, FCS

The Translating Bridge generates

DA, SA, ETPID, TCI (CFI reset), LLC, C-Data, PAD, FCS

Whereas a true translation of the tagged frame via an 802.1Q Bridge would result in

L-C-T/C,T: DA, SA, ETPID, TCI (CFI reset), LEN, LLC, C-Data, PAD, FCS

As can be seen, the resultant frame superficially appears to be a valid tagged frame; however, it is missing the LEN field, and is therefore not a valid tagged frame.

A source-routed LLC-encoded frame on Token Ring/FDDI has the following form:

L-C-R/R,T: RCI, DA, SA (RII set), RIF, STPID, TCI (CFI reset), LLC, C-Data, FCS

The Translating Bridge generates

DA, SA, ETPID, TCI (CFI reset), LLC, C-Data, PAD, FCS

Whereas a true translation of the tagged frame via an 802.1Q Bridge would result in

L-C-R/C,T: DA, SA, ETPID, TCI (CFI set), LEN, RIF (NCFI=C), LLC, C-Data, PAD, FCS

Again, the resultant frame superficially appears to be a valid tagged frame, but is missing the LEN field. The RIF information has been lost.

Similar problems exist for the other two LLC-encoded formats, L-N-R and L-N-T, but in addition, the CFI will be set, indicating the presence of a RIF in the tag header when no such RIF field exists.

In both cases, the effect of the Translating Bridge is almost symmetrical; passing the invalid frame back through the Translating Bridge restores it to a valid tagged frame format on 802.3/Ethernet, but with the loss of any source-routing information that may have been present, and the inclusion of a PAD field if the original frame on the Ring medium had been small.

Consequently,

a) The translated frames cannot be correctly interpreted by VLAN-aware end stations attached to the Ring medium;
b) Any 802.1Q Bridge that attempts to untag the translated frames will generate invalid untagged frames;

c) Any 802.1Q Bridge that attempts to relay the translated frames back onto Token Ring/FDDI will generate invalid tagged frames;

d) Any source-routing information is lost;

e) As the effect of the Translating Bridge is almost symmetric (the RIF is lost, and there may be a PAD included), passing the frame through an even number of such translations before any 802.1Q device attempts to interpret the frame format results in correct operation of the 802.1Q devices, as long as those devices are not sensitive to the presence of spurious PAD information.

E.4.4 Ethernet Type-encoded tagged frames relayed from Token Ring or source-routed FDDI to 802.3/Ethernet

A Transparent Ethernet Type-encoded frame carrying Canonical data on Token Ring/FDDI has the following form:

E-C-T/R,T: RCI, DA, SA (RII reset), STPID, TCI (CFI reset), SPT, C-Data, FCS

The Translating Bridge generates

DA, SA, ETPID, TCI (CFI reset), SPT, C-Data, FCS

Whereas a true translation of the tagged frame via an 802.1Q Bridge would result in

E-C-T/C,T: DA, SA, ETPID, TCI (CFI reset), PT, C-Data, FCS

As can be seen, the resultant frame superficially appears to be a valid tagged frame; however, the SPT has not undergone translation to a PT. End stations encountering this frame on 802.3/Ethernet would only be capable of interpreting it if they were able to recognize the SPT as an embedded Ethernet Type.

Translating Canonical source-routed Ethernet Type-encoded information produces similar results, but with the additional loss of the source-routing information.

A Transparent Ethernet Type-encoded frame carrying Non-canonical information on Token Ring/FDDI has the following form:

E-N-T/R,T: RCI, DA, SA (RII reset), STPID, TCI (CFI set), SPT, N-Data, FCS

The Translating Bridge generates:

DA, SA, ETPID, TCI (CFI set), SPT, N-Data, FCS

Whereas a true translation of the tagged frame via an 802.1Q Bridge would result in:

E-N-T/C,T: DA, SA, ETPID, TCI (CFI set), PT, RIF (NCFI=N), N-Data, FCS

Again, the resultant frame superficially appears to be a valid tagged frame, but the SPT has not been translated to a PT, and the RIF is not present in the tag header. Any 802.1Q device will therefore interpret the first N octets of the N-Data field as if it was the RIF.

Translating Non-canonical source-routed Ethernet Type-encoded information produces similar results, but with the additional loss of the source-routing information.

In both cases, the effect of the Translating Bridge is almost symmetrical; passing this invalid frame back through the Translating Bridge restores it to a valid tagged frame format on 802.3/Ethernet, but with the loss of any source-routing information that may have been present.

Consequently,

a) The translated frames cannot be correctly interpreted by VLAN-aware end stations attached to the Ring medium;
b) Any 802.1Q Bridge that attempts to untag translated frames carrying Non-canonical information will generate invalid untagged frames;
c) Any 802.1Q Bridge that attempts to relay untag translated frames carrying Non-canonical information back onto Token Ring/FDDI will generate invalid tagged frames;
d) Frames carrying Canonical information can be successfully untagged or relayed in tagged form using other MAC methods, as long as the 802.1Q Bridge is capable of correctly handling the embedded SPT;
e) Any source-routing information is lost;
f) As the effect of the Translating Bridge is symmetric (apart from the loss of RIF), passing the frame through an even number of such translations before any 802.1Q device attempts to interpret the frame format results in correct operation of the 802.1Q devices.

E.4.5 Conclusions

Except for the limited case of relaying Canonical Ethernet Type-encoded information from 802.3/Ethernet to Token Ring/FDDI, the translation that a tagged frame undergoes when passing through a Translating Bridge renders the frame uninterpretable by any 802.1Q device (either end station or Bridge), unless the frame has passed through an even number of Translating Bridges before the 802.1Q device attempts to interpret the frame. In regions where an odd number of translations have occurred, source-routing information is rendered invisible to source routing Bridges in some cases. In the other cases, source-routing information is lost after the first translation.

Consequently, the use of Translating Bridges intermixed with 802.1Q Bridges is feasible only if

a) An even number of translations (or zero translations) is experienced by any tagged frame that is transmitted between any pair of 802.1Q-aware devices in the Bridged LAN;
b) The loss of source routing capability across some regions of the Bridged LAN is acceptable; specifically across regions where the first Translating Bridge encountered by a correctly formatted tagged frame will relay the frame from Token Ring/FDDI to 802.3/Ethernet;
c) End stations are not sensitive to receiving LLC-encoded frames that have PAD octets added to the LLC user data.

E.5 Heterogeneous Bridged LANs: intermixing 802.1Q Bridges with IEEE Std 802.1D Bridges

The specification in this standard for the use of GMRP in VLANs (11.2) makes use of VLAN-tagged frames to signal the GIP Context that applies to the registration information carried in GMRP PDUs. Devices that implement GMRP as specified in IEEE Std 802.1D, 1998 Edition will regard such frames as badly formed GMRP frames, and will therefore discard them on receipt. Using an IEEE Std 802.1D Bridge to interconnect two or more LAN regions containing 802.1Q devices that implement GMRP will therefore prevent GMRP information propagation between the 802.1Q regions, with attendant effects upon the forwarding behavior of both the 802.1D and 802.1Q Bridges in the LAN. This configuration can be made to work if the IEEE Std 802.1D Bridge is statically configured with the following:

a) An All Groups entry in the Filtering Database, specifying Registration Fixed on all Ports, and
b) The GMRP Protocol Administrative Control parameters set to disable GMRP on all Ports.

As the Bridge no longer supports the GMRP application, it will forward GMRP PDUs on all Ports that are in Forwarding. The effect of this is to configure the IEEE Std 802.1D Bridge to behave in the same manner as an ISO/IEC 10038 Bridge.

Placing IEEE Std 802.1D Bridges around the periphery of an 802.1Q-based Bridged LAN works correctly, as long as, for a given 802.1D Bridge, the 802.1Q Bridges connected to the same segment(s) are configured to untag any VLANs that are relevant to the GMRP operation of the 802.1D Bridge. The IEEE Std 802.1D Bridge generates untagged GMRP frames, which the 802.1Q Bridges classify according to the value of the PVID for the reception Port; in a simple configuration of the 802.1Q Bridges, the Ports that connect to the 802.1D Bridge are configured for the PVID VLAN to be untagged on egress.

NOTE 1—There may be situations where more complex configurations are required, in which VLANs other than the PVID are configured untagged in order to maintain the correct IEEE Std 802.1D Bridge filtering behavior.

NOTE 2—For bridges that make VLAN assignments on untagged frames according to Port-and-Protocol-based classification rules, special care is necessary in configuration: since GMRP does not carry information about the particular protocol for which the Group membership is intended, it must be taken to apply to all protocols for which untagged traffic is carried on a particular link. Therefore, the VLAN indicated by a port’s PVID is used to carry the GMRP frames associated with all of the VLANs which are members of the VID Set of that Port; in addition, the PVID must be configured to egress untagged on any other bridge port where any of the set of VIDs also egresses untagged and requires GMRP operation beyond that egress port.

The effect of this type of configuration is that all registrations propagated by a given IEEE Std 802.1D Bridge on a given (Port-based or Port-and-Protocol-based) VLAN are seen by all other 802.1D Bridges served by 802.1Q Bridges for which that VLAN is configured for untagged egress. The filtering behavior of the IEEE Std 802.1D Bridges is therefore governed only by the behavior of other devices (both 802.1D and 802.1Q) that are attached to the same VLAN.

E.6 Intermixing Port-based classification and Port-and-Protocol-based classification or future enhancements in 802.1Q

The discussion above on intermixing Q Bridges with D Bridges has a direct analogue in the mixing of bridges implementing only Port-based and Port-and-Protocol-based classification of frames in 802.1Q networks. This and potential subsequent editions of 802.1Q extend the VLAN classification capabilities to support more sophisticated ingress rules for frame classification.

In VLAN configurations that use both Port-based and Port-and-Protocol-based VLAN classification, a Bridge that supports only Port-based VLAN classification will merge VLANs that would otherwise be classified separately by a Bridge that supports Port-and-Protocol-based VLAN classification. To get around this problem, it may be possible to dedicate specific Ports to specific protocols in Bridges that support only Port-based VLAN classification, as in the example shown in Figure E-4. However, such solutions may not be possible where there are multiprotocol end stations in the network.

E.6.1 Example: Intermixing Protocol-based Ingress Rules

Consider the case where Bridges implement a configuration mechanism to select between Port-based classification rules (a “Q-Port Bridge”) and Port-and-Protocol-based rules (a “Q-Port/Protocol Bridge”). This would allow, for example, for support for IP and IPX as distinct VLANs. The topology shown in Figure E-4 might apply when a Q-Port/Protocol Bridge is added to a topology, otherwise using only Q-Port bridges, to allow users of two protocols to participate in two separate VLANs.

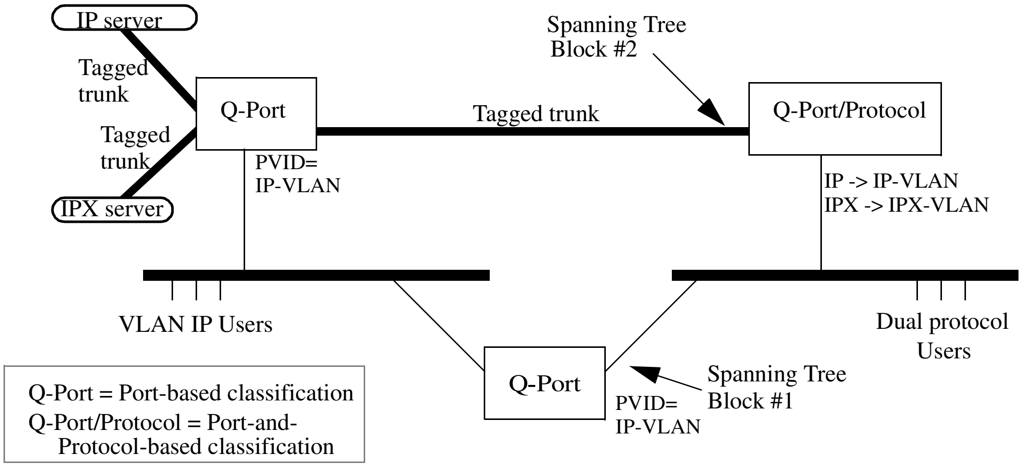


Figure E-4—Interoperability between Port-based and Port-and-Protocol-based classification

Consider this network, when STP has blocked at point #1, and not at #2. The upper Q-Port Bridge operates as expected, and the Q-Port/Protocol Bridge provides Port-and-Protocol-based classification for the frames received from the dual protocol users on the right-hand segment. IP-VLAN and IPX-VLAN frames are VLAN-tagged on the trunks to and from the uppermost Bridges and servers. But if a STP reconfiguration should result in a block at point #2, but not at #1, activating traffic through the lower Q-Port Bridge, dual protocol users will have all their traffic treated as part of the IP-VLAN. An immediate consequence is that the uppermost Bridge will no longer provide them access to the “IPX server”. It is the fact that the lower Q-Port Bridge must have just one of the two VLANs configured for all untagged traffic, regardless of its protocol, that leads to this lack of connectivity.

E.6.2 Differing views of untagged traffic on a given LAN segment

Further challenges arise when one considers the case where several Q Bridges, some implementing Port-based and some implementing Port-and-Protocol-based classification all attach to the same LAN segment. Again, the rule that any given frame exists in exactly one VLAN requires that all of these Bridges be configured with consistent ingress rules. In this case, the Q-Port Bridges will provide a least common capability, and this further requires common configuration of the PVID in the Q-Port and Q-Port/Protocol Bridges.

Annex F

(informative)

Frame translation considerations

When relaying frames between different MAC methods, there are a number of frame translations that may be required, depending upon the source and destination MAC methods concerned, the tagging state of the received and transmitted frames, and the data carried within the frames:

a) If the source and destination MAC methods differ, then the overall format of the received frame must be translated into the frame format required for the MAC onto which the frame will be transmitted. The details of the frame translation at this level is defined by the definition of the Internal Sublayer Service (IEEE Std 802.1D, 1998 Edition, 6.4), the support of the internal sublayer service by specific MAC procedures (IEEE Std 802.1D, 1998 Edition, 6.5), and the standards that define the specific MAC procedures concerned.
b) If the received frame is being relayed in tagged format between differing MAC methods, or if the tagging state is to change, then the tag header may require translation, insertion, or removal as defined in this standard.
c) If Ethernet Type-encoded data is being carried in the frame, then the format of that data may require translation as defined in IEEE Std 802.1H, IETF RFC 1042, and IETF RFC 1390. These translations are essentially concerned with providing a standardized means of representing Ethernet Type-encoded frames on LANs that have no inherent ability within their MAC procedures to represent Ethernet Type values (i.e., LAN MACs where the “native” link layer protocol is LLC), and ensuring that the frame translation in the reverse direction results in the correct format on the 802.3/Ethernet LAN. The mechanisms specified in these standards are based around the use of SNAP encoding to carry Ethernet Type values in native LLC-encoded environments, and the use of translation tables to control the reverse translation if this should be necessary.

One aspect that is not fully covered in the standards mentioned is the issue of how differences in bit ordering between MAC methods can affect the data translation requirements. For the majority of data relayed by a Bridge, these differences in bit ordering are not an issue; although the bit ordering of MAC data “on the wire” may differ between MAC methods, the representation of that data within the relay function of the Bridge is independent of the order of transmission or reception adopted by the various MACs.

An exception can occur when the data carried in a frame includes MAC Address values; for example, in IP ARP packets. There are two formats used in LANs when representing MAC Address values in MAC user data:

d) Canonical format;
e) Non-canonical format.

The octet ordering is identical in both cases (the first octet put into the mac_service_data_unit is the left-most octet of the address when written down using hexadecimal notation as defined in Section 5 of IEEE Std 802); however, the bit ordering within each octet differs:

f) Canonical format: The least significant bit of each octet of the standard hexadecimal representation of the address (see IEEE Std 802-1990) represents the least-significant (LS) bit of the corresponding octet of the Canonical format of the address;
g) Non-canonical format: The most significant bit of each octet of the standard hexadecimal representation of the address (see IEEE Std 802-1990) represents the least-significant bit of the corresponding octet of the Canonical format of the address.

Ideally, a single format would be used in all places; however, both can be used, depending on the MAC method concerned. The native format used in MACs where the bit transmission order is LS bit first (e.g., IEEE Std 802.3 and IEEE Std 802.4), and also in FDDI, is the Canonical format; in MACs where the bit transmission order is most-significant (MS) bit first (e.g., IEEE Std 802.5), Non-canonical format is used. A further complication here is that the format used is not consistent across all higher layer protocols (some use a single format regardless of MAC method, others use the native format for the underlying MAC method), and there is no single standard that specifies which protocols carry embedded MAC Address information or the format in which they carry it.

Clearly, in order for the recipients of frames carrying embedded address information to be able to interpret MAC Address information correctly, it is necessary either to include information in the frame that specifies which format is in use, or for Bridges to modify the format of the information appropriately when relaying frames between regions of the network where the expected format differs for the protocol being carried.

In “Translating Bridges” (Bridges that relay between differing MAC methods, based on IEEE Std 802.1D, 1998 Edition, IEEE Std 802.1H, IETF RFC 1042, and IETF RFC 1390—i.e., in Bridging Function B1 in Figure C-2), the latter approach is the only option, as there is no way that additional information can be carried in the frame in order to identify the embedded format. Consequently, in order for such a Bridge to successfully translate embedded address information, it needs to be able to recognize the higher layer protocol carried in the frame, and act accordingly.

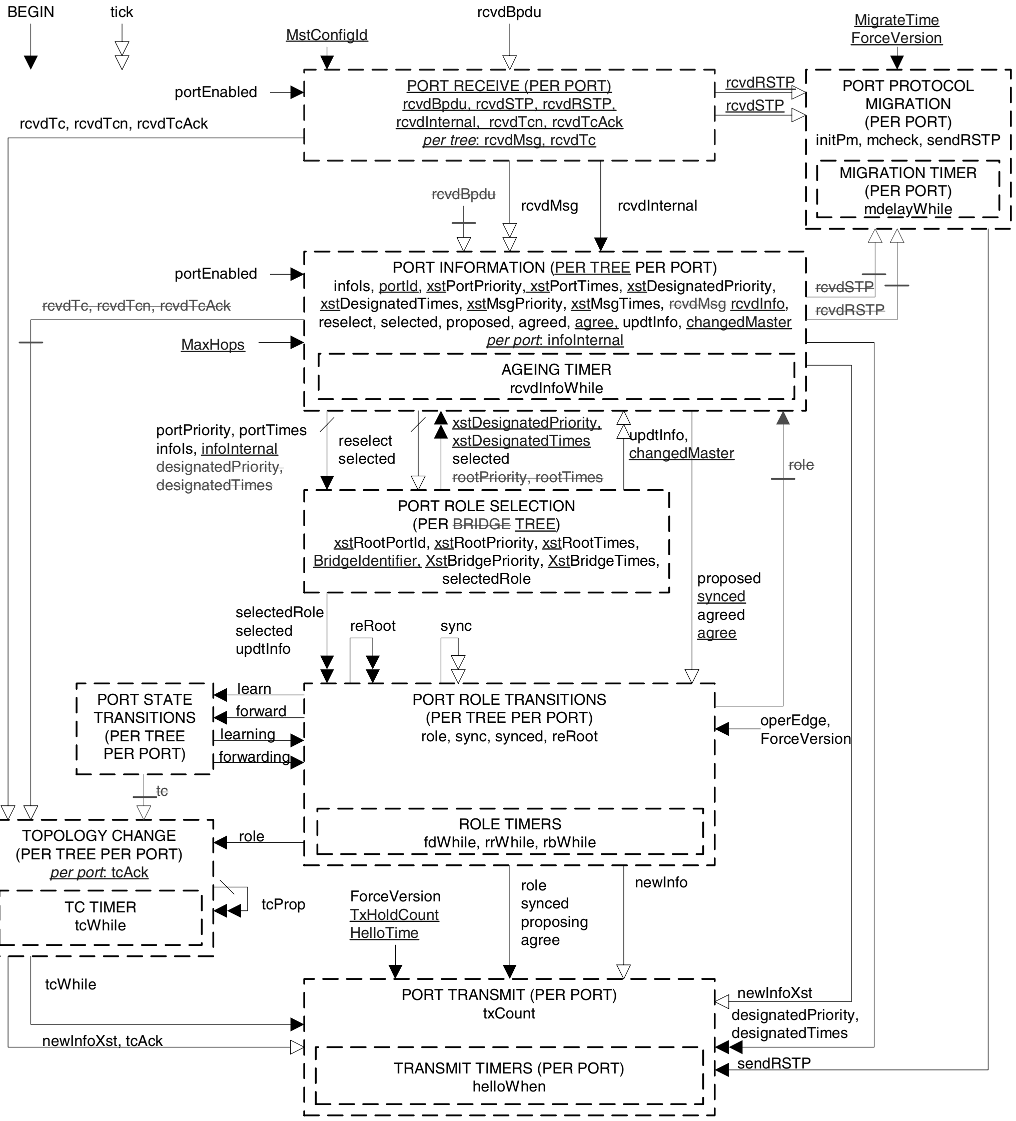
In VLAN-aware Bridges, the information carried in the CFI (and in the NCFI bit of the RIF, in 802.3/Ethernet frames) provides a direct indication of the format of embedded addresses. Hence, any considerations related to translating the format of embedded addresses can be confined to Bridges where a frame is received untagged and the tag header is inserted on transmission, or received tagged and the tag header is removed on transmission.

Annex G

(informative)

Differences between MSTP and RSTP (802.1w) state machines

This annex documents the differences between the RSTP state machines defined in Clause 17 of IEEE Std 802.1w-2001 and the MSTP state machines defined in Clause 13 of this standard. The differences are illustrated by means of redline and ~~strikeout~~ markings, showing the changes made to the RSTP state machines in order to create the MSTP state machines.



NOTE 1: For convenience all timers are collected together into one state machine.

NOTE 2: This overview diagram is not itself a state machine, but serves to illustrate the principal variables that are used to communicate between the individual RST state machines and the variables local to each machine.

Figure G-1—MSTP state machines - overview and relationships

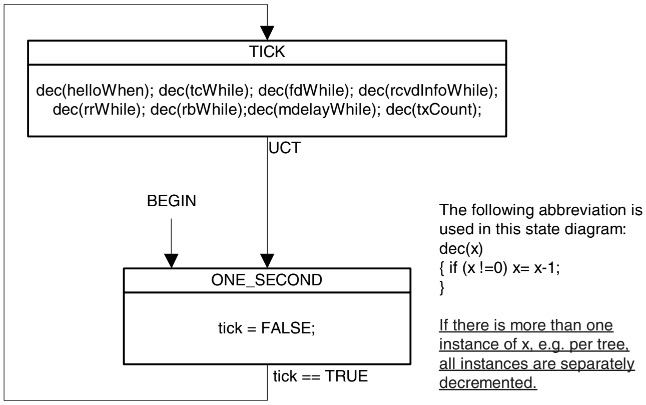

TICK
dec(helloWhen); dec(tcWhile); dec(fdWhile); dec(rcvdInfoWhile);
dec(rrWhile); dec(rbWhile);dec(mdelayWhile); dec(txCount);
UCT
BEGIN
ONE_SECOND
tick = FALSE;
tick == TRUE
The following abbreviation is
used in this state diagram:
dec(x)
{ if (x !=0) x= x-1;
}
If there is more than one
instance of x, e.g. per tree,
all instances are separately
decremented.

Figure G-2—Port Timers state machine

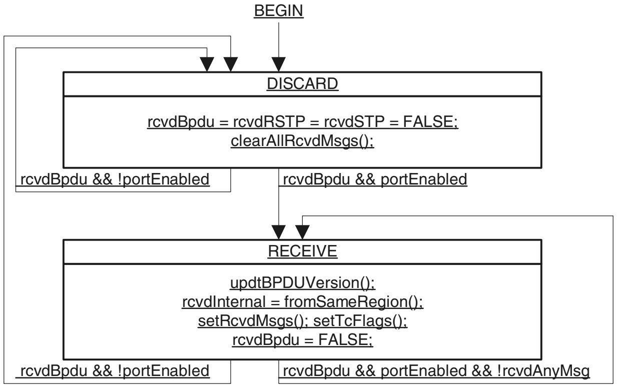
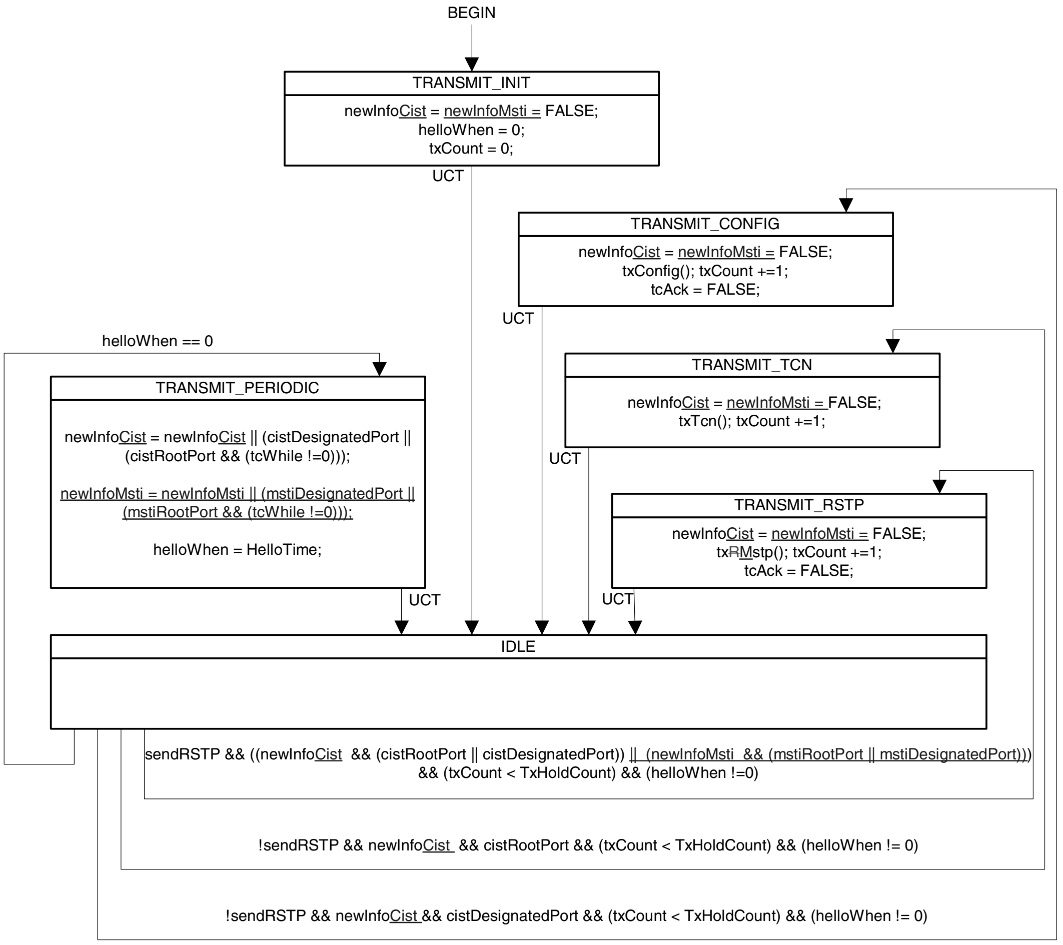


Figure G-3—Port Receive state machine


BEGIN
TRANSMIT_INIT
newInfoCist = newInfoMsti = FALSE;
helloWhen = 0;
txCount = 0;
UCT
TRANSMIT_CONFIG
newInfoCist = newInfoMsti = FALSE;
txConfig(); txCount +=1;
tcAck = FALSE;
UCT
helloWhen == 0
TRANSMIT_TCN
newInfoCist = newInfoMsti = FALSE;
txTcn(); txCount +=1;
TRANSMIT_PERIODIC
newInfoCist = newInfoCist || (cistDesignatedPort ||
(cistRootPort && (tcWhile !=0)));
newInfoMsti = newInfoMsti || (mstiDesignatedPort ||
(mstiRootPort && (tcWhile !=0)));
helloWhen = HelloTime;
UCT
TRANSMIT_RSTP
newInfoCist = newInfoMsti = FALSE;
txRMstp(); txCount +=1;
tcAck = FALSE;
UCT
UCT
IDLE
sendRSTP && ((newInfoCist && (cistRootPort || cistDesignatedPort)) || (newInfoMsti && (mstiRootPort || mstiDesignatedPort)))
&& (txCount < TxHoldCount) && (helloWhen !=0)
!sendRSTP && newInfoCist && cistRootPort && (txCount < TxHoldCount) && (helloWhen != 0)
!sendRSTP && newInfoCist && cistDesignatedPort && (txCount < TxHoldCount) && (helloWhen != 0)

All transtions, except UCT, are qualified by "&& selected &&!updtInfo".

Figure G-4—Port Transmit state machine

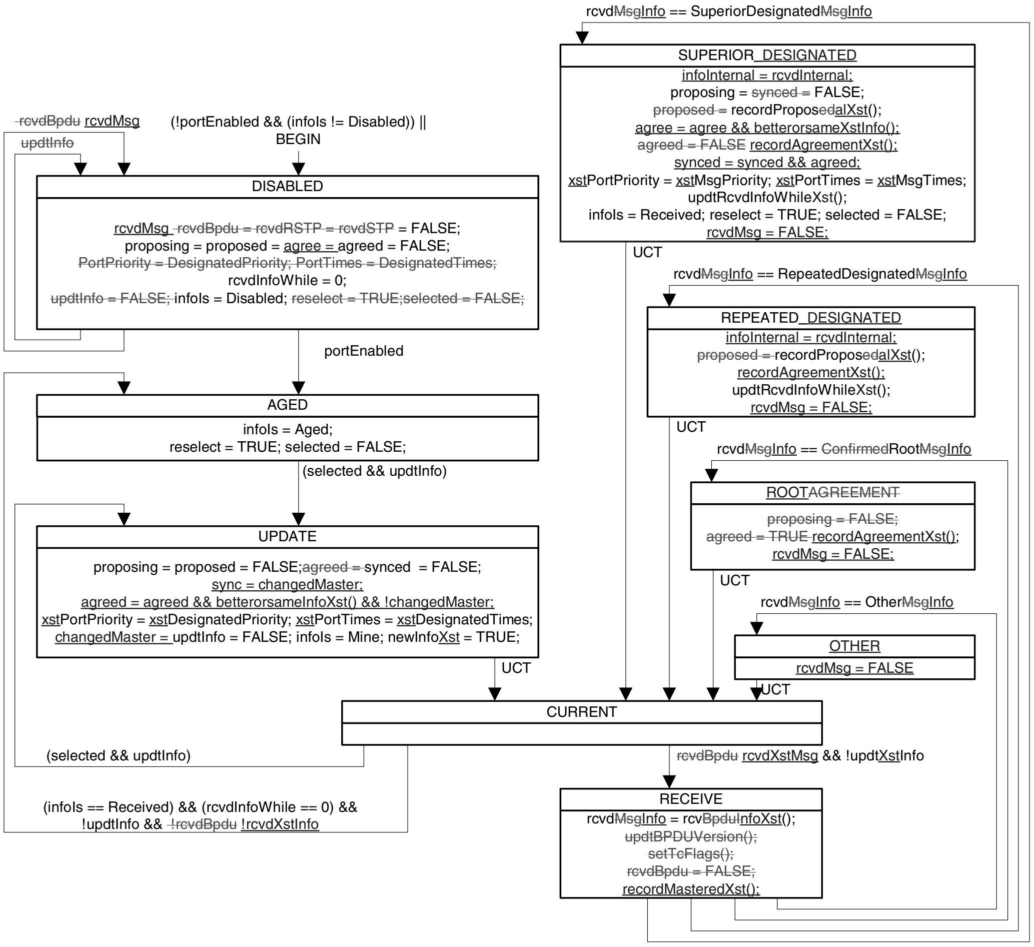

rcvd~~Msg~~Info == SuperiorDesignated~~Msg~~Info
SUPERIOR_DESIGNATED
infoInternal = rcvdInternal;
proposing = ~~synced =~~ FALSE;
~~proposed =~~ recordPropos~~ed~~alXst();
agree = agree && betterorsameXstInfo();
~~agreed = FALSE~~ recordAgreementXst();
synced = synced && agreed;
xstPortPriority = xstMsgPriority; xstPortTimes = xstMsgTimes;
updtRcvdInfoWhileXst();
infoIs = Received; reselect = TRUE; selected = FALSE;
rcvdMsg = FALSE;
UCT
~~rcvdBpdu~~ rcvdMsg
~~updtInfo~~
(!portEnabled && (infoIs != Disabled)) ||
BEGIN
DISABLED
rcvdMsg ~~rcvdBpdu = rcvdRSTP = rcvdSTP~~ = FALSE;
proposing = proposed = agree = agreed = FALSE;
~~PortPriority = DesignatedPriority; PortTimes = DesignatedTimes;~~
rcvdInfoWhile = 0;
~~updtInfo = FALSE;~~ infoIs = Disabled; ~~reselect = TRUE;selected = FALSE;~~
portEnabled
rcvd~~Msg~~Info == RepeatedDesignated~~Msg~~Info
REPEATED_DESIGNATED
infoInternal = rcvdInternal;
~~proposed =~~ recordPropos~~ed~~alXst();
recordAgreementXst();
updtRcvdInfoWhileXst();
rcvdMsg = FALSE;
UCT
AGED
infoIs = Aged;
reselect = TRUE; selected = FALSE;
(selected && updtInfo)
rcvd~~Msg~~Info == ~~Confirmed~~Root~~Msg~~Info
ROOT~~AGREEMENT~~
~~proposing = FALSE;~~
~~agreed = TRUE~~ recordAgreementXst();
rcvdMsg = FALSE;
UCT
UPDATE
proposing = proposed = FALSE;~~agreed =~~ synced = FALSE;
sync = changedMaster;
agreed = agreed && betterorsameInfoXst() && !changedMaster;
xstPortPriority = xstDesignatedPriority; xstPortTimes = xstDesignatedTimes;
changedMaster = updtInfo = FALSE; infoIs = Mine; newInfoXst = TRUE;
rcvd~~Msg~~Info == Other~~Msg~~Info
OTHER
rcvdMsg = FALSE
UCT
UCT
CURRENT
(selected && updtInfo)
~~rcvdBpdu~~ rcvdXstMsg && !updtXstInfo
RECEIVE
rcvd~~Msg~~Info = rcv~~Bpdu~~InfoXst();
~~updtBPDUVersion();~~
~~setTcFlags();~~
~~rcvdBpdu = FALSE;~~
recordMasteredXst();
(infoIs == Received) && (rcvdInfoWhile == 0) &&
!updtInfo && ~~!rcvdBpdu~~ !rcvdXstInfo

Figure G-5—Port Information state machine

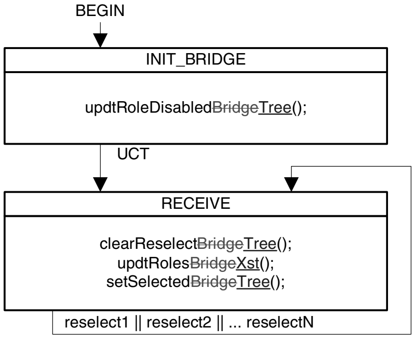

BEGIN
INIT_BRIDGE
updtRoleDisabled~~Bridge~~Tree();
UCT
RECEIVE
clearReselect~~Bridge~~Tree();
updtRoles~~Bridge~~Xst();
setSelected~~Bridge~~Tree();
reselect1 || reselect2 || ... reselectN

Figure G-6—Port Role Selection

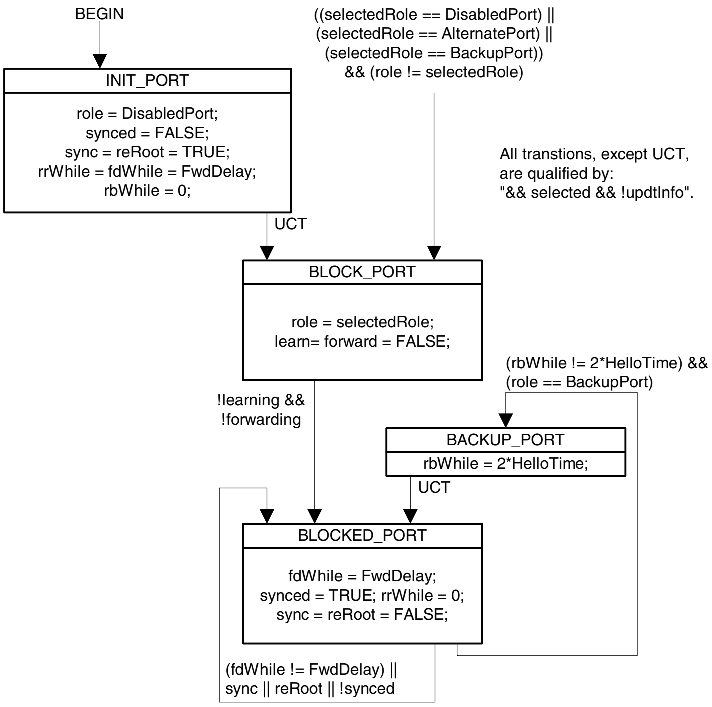
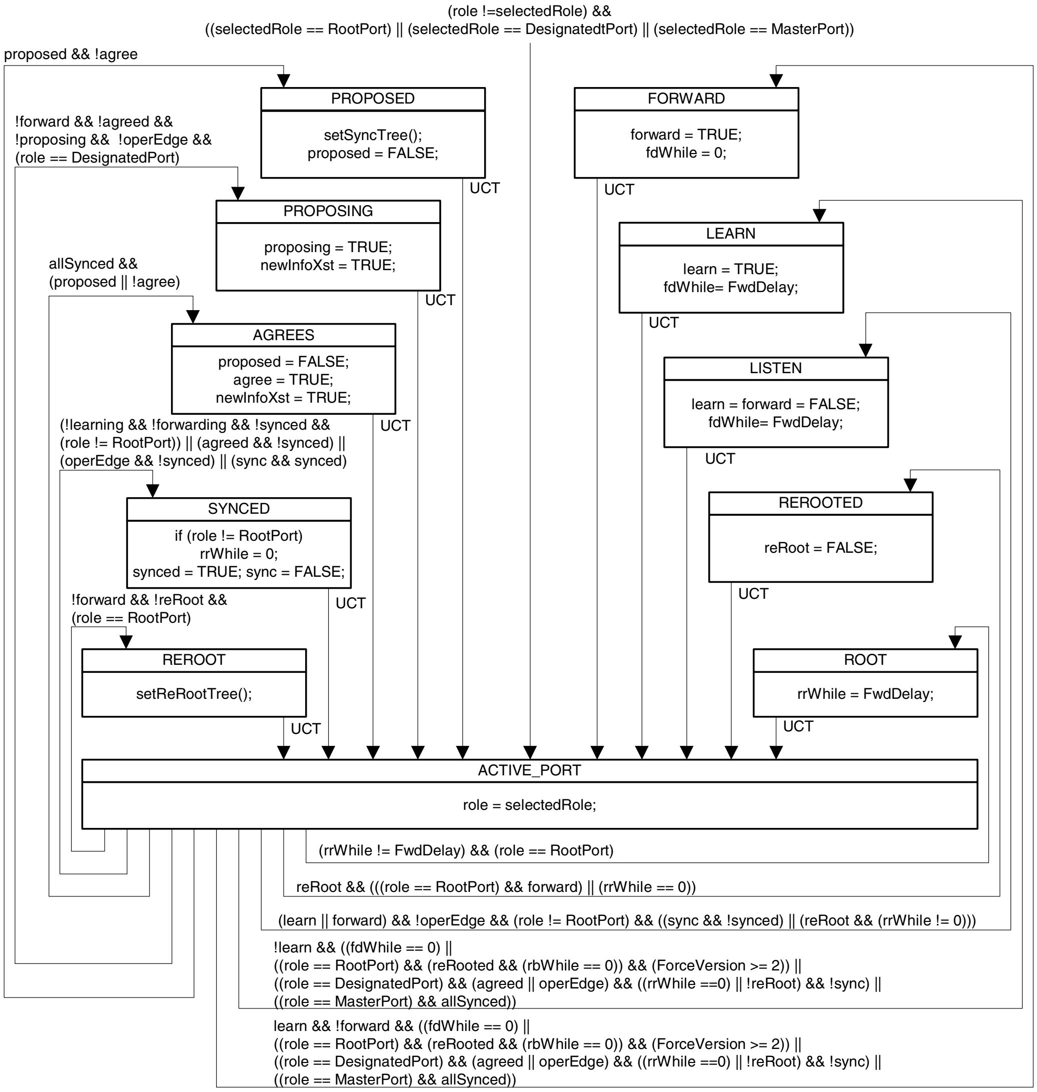


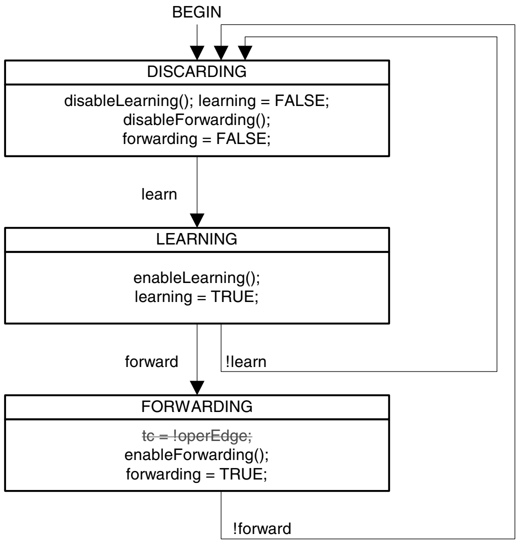
**Figure G-7—Port Role Transitions state machine—
Part 1: Disabled, Alternate, and Backup Roles**


(role !=selectedRole) &&
((selectedRole == RootPort) || (selectedRole == DesignatedtPort) || (selectedRole == MasterPort))
proposed && !agree
PROPOSED
setSyncTree();
proposed = FALSE;
UCT
!forward && !agreed &&
!proposing && !operEdge &&
(role == DesignatedPort)
PROPOSING
proposing = TRUE;
newInfoXst = TRUE;
UCT
allSynced &&
(proposed || !agree)
AGREES
proposed = FALSE;
agree = TRUE;
newInfoXst = TRUE;
UCT
(!learning && !forwarding && !synced &&
(role != RootPort)) || (agreed && !synced) ||
(operEdge && !synced) || (sync && synced)
SYNCED
if (role != RootPort)
rrWhile = 0;
synced = TRUE; sync = FALSE;
UCT
!forward && !reRoot &&
(role == RootPort)
REROOT
setReRootTree();
UCT
FORWARD
forward = TRUE;
fdWhile = 0;
UCT
LEARN
learn = TRUE;
fdWhile= FwdDelay;
UCT
LISTEN
learn = forward = FALSE;
fdWhile= FwdDelay;
UCT
REROOTED
reRoot = FALSE;
UCT
ROOT
rrWhile = FwdDelay;
UCT
ACTIVE_PORT
role = selectedRole;
(rrWhile != FwdDelay) && (role == RootPort)
reRoot && (((role == RootPort) && forward) || (rrWhile == 0))
(learn || forward) && !operEdge && (role != RootPort) && ((sync && !synced) || (reRoot && (rrWhile != 0)))
!learn && ((fdWhile == 0) ||
((role == RootPort) && (reRooted && (rbWhile == 0)) && (ForceVersion >= 2)) ||
((role == DesignatedPort) && (agreed || operEdge) && ((rrWhile ==0) || !reRoot) && !sync) ||
((role == MasterPort) && allSynced))
learn && !forward && ((fdWhile == 0) ||
((role == RootPort) && (reRooted && (rbWhile == 0)) && (ForceVersion >= 2)) ||
((role == DesignatedPort) && (agreed || operEdge) && ((rrWhile ==0) || !reRoot) && !sync) ||
((role == MasterPort) && allSynced))

All transtions, except UCT, are qualified by "&& selected && !updtInfo".

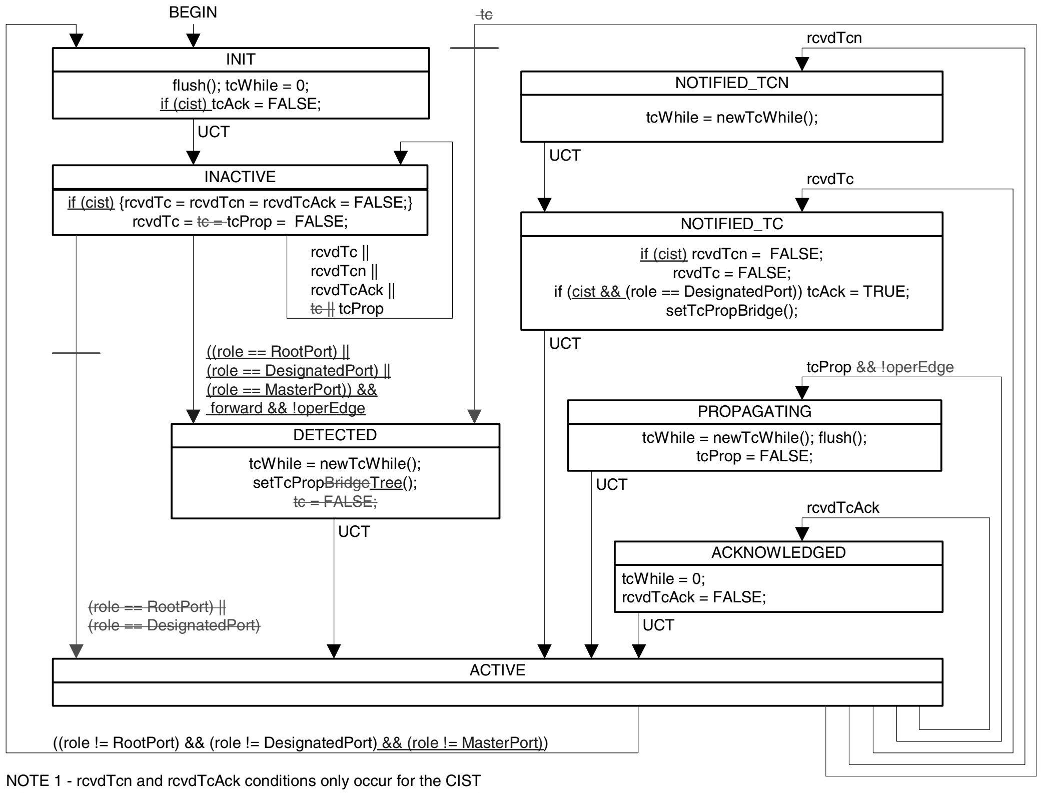
**Figure G-8—Port Role Transitions state machine—
Part 2: Root, Designated, and Master Roles**

NOTE—Figure G-8 combines the Root Port and Designated Port role figures from the RSTP definition in IEEE Std 802.1w-2001; the differences between this figure and the RSTP figures are therefore not shown.



NOTE: A small system dependent delay may occur on each of the transitions shown.

Figure G-9—Port State Transition state machine



NOTE 1 - rcvdTcn and rcvdTcAck conditions only occur for the CIST
NOTE 2 - The flush() procedure, when called on a transition from ACTIVE to INIT, should take account of any system dependent delay associated with port state transitions

Figure G-10—Topology Change state machine