Recognized as an

American National Standard (ANSI)

IEEE Std 802.1w-2001

[Amendment to IEEE Std 802.1D, 1998 Edition (ISO/IEC 15802-3:1998)
and IEEE Std 802.1t-2001]

IEEE Standard for Local and metropolitan area networks— Common specifications

Part 3: Media Access Control (MAC) Bridges— Amendment 2: Rapid Reconfiguration

Sponsor

LAN/MAN Standards Committee
of the
IEEE Computer Society

Approved 14 June 2001
IEEE-SA Standards Board

Approved 25 October 2001
American National Standards Institute

Abstract: This amendment to IEEE Std 802.1D, 1998 Edition (ISO/IEC 15802-3: 1998) and IEEE Std 802.1t-2001 defines the changes necessary to the operation of a MAC Bridge in order to provide rapid Spanning Tree reconfiguration capability.

Keywords: local area networks, MAC Bridge management, MAC Bridges, media access control (MAC) bridges, Rapid Spanning Tree Algorithm and Protocol (RSTP)

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

Print: ISBN 0-7381-2924-0 SH94939
PDF: ISBN 0-7381-2925-9 SS94939

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.

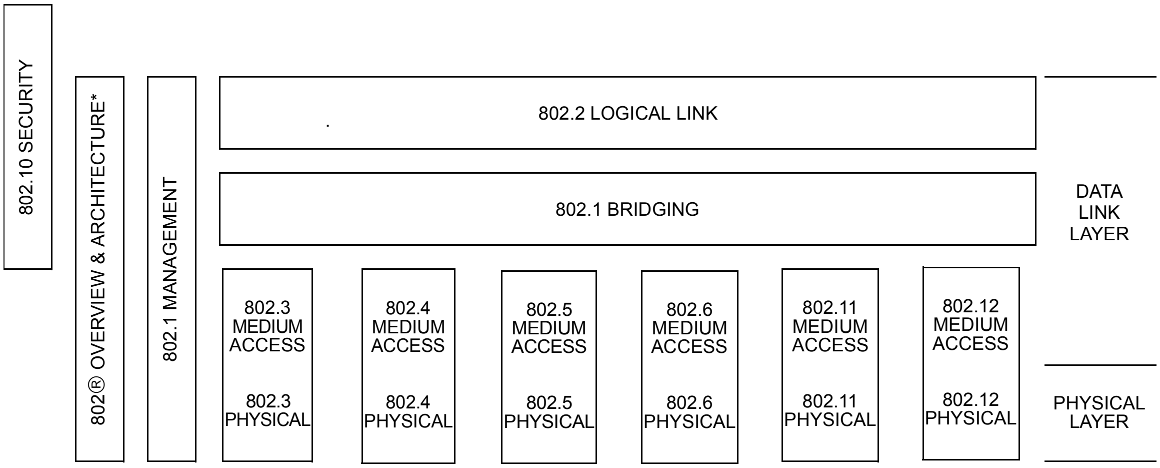
IEEE is the sole entity that may authorize the use of certification marks, trademarks, or other designations to indicate compliance with the materials set forth herein.

Introduction

[This introduction is not a part of IEEE Std 802.1w-2001, IEEE Standard for Local and metropolitan area networks—Common specifications—Part 3: Media Access Control (MAC) Bridges: Amendment 2—Rapid Reconfiguration.]

This amendment to IEEE Std 802.1D, 1998 Edition (ISO/IEC 15802-3:1998) defines the changes necessary to the operation of a MAC Bridge in order to provide rapid Spanning Tree reconfiguration capability. These changes are documented in the usual form for Amendments to IEEE 802® standards; i.e., as an explicit set of editing instructions that, if correctly applied to the text of IEEE Std 802.1D, 1998 Edition (ISO/IEC 15802-3:1998), will create an amended document.

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 numbers.)



* 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[1]: *Overview and Architecture*. This standard provides an overview to the family of IEEE 802 Standards. This document forms part of the 802.1 scope of work.

[1] The 802 Architecture and Overview 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.

- ANSI/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.

- ANSI/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.

- ANSI/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.

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

- ANSI/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.

- ANSI/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*.

- ANSI/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.

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

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

- ANSI/IEEE Std 802.4 [ISO/IEC 8802-4]: *Token Bus Access Method and Physical Layer Specifications*.

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

- ANSI/IEEE Std 802.6 [ISO/IEC 8802-6]: *Distributed Queue Dual Bus Access Method and Physical Layer Specifications*.

- ANSI/IEEE Std 802.10: *Interoperable LAN/MAN Security.* Currently approved: Secure Data Exchange (SDE).

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

- ANSI/IEEE Std 802.12: [ISO/IEC 8802-12] *Demand Priority Access Method, Physical Layer and Repeater Specification*.

- 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 Std 802.17: *Resilient Packet Ring Access Method and Physical Layer Specifications.*

In addition to the family of standards, the following is a recommended practice for a common physical layer technology:

- IEEE Std 802.7: *IEEE Recommended Practice for Broadband Local Area Networks*.

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 802 family of standards. Thus the conformance test documents for 802.3 are numbered 1802.3, the conformance test documents for 802.5 will be 1802.5, and so on. Similarly, ISO will use 18802 to number conformance test standards for 8802 standards.

Participants

When the IEEE 802.1 Working Group approved this standard, it had the following membership:

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

Les Bell
Alan Chambers
Marc Cochran
Paul Congdon
Hesham El Bakoury
Norman W. Finn
Sharam Hakimi
Bob Hott
Toyoyuki Kato
Hal Keen
Daniel Kelley
Keith Klamm
Joe Laurence
Bill Lidinsky
Yaron Nachman
LeRoy Nash
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 this standard. Balloters may have voted for approval, disapproval, or abstention.

Jacob Ben Ary
James T. Carlo
Linda T. Cheng
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
Joseph M. Gwinn
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
David Solomon
Fred J. Strauss
Jonathan R. Thatcher
Mark-Rene Uchida
Scott A. Valcourt
John Viaplana
Paul A. Willis
Oren Yuen

When the IEEE-SA Standards Board approved this standard on 14 June 2001, it had the following membership:

Donald N. Heirman, *Chair*
James T. Carlo, *Vice Chair*
Judith Gorman, *Secretary*

Chuck Adams
Mark D. Bowman
Clyde R. Camp
Richard DeBlasio
Harold E. Epstein
H. Landis Floyd
Jay Forster*
Howard M. Frazier
James H. Gurney
Raymond Hapeman
Richard J. Holleman
Richard H. Hulett
Lowell G. Johnson
Joseph L. Koepfinger*
Peter H. Lips
Paul J. Menchini
Daleep C. Mohla
Robert F. Munzner
Ronald C. Petersen
Malcolm V. Thaden
Geoffrey O. Thompson
Akio Tojo
Howard L. Wolfman

*Member Emeritus

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

Satish K. Aggarwal, *NRC Representative*
Alan H. Cookson, *NIST Representative*
Donald R. Volzka, *TAB Representative*

Jennifer McClain Longman
IEEE Standards Project Editor

Contents

1. Overview 1

1.1 Introduction 1
1.2 Scope 2

2. References 3

4. Abbreviations 3

5. Conformance 4

5.1 Static conformance requirements 4
5.2 Options 5

6. Support of the MAC Service 6

6.3 Quality of service maintenance 6
6.4 Internal Sublayer Service provided within the MAC Bridge 8
6.5 Support of the Internal Sublayer Service by specific MAC procedures 9

8. The Spanning Tree algorithm and protocol 10

9. Encoding of Bridge Protocol Data Units (BPDUs) 10

9.1 Structure 10
9.2 Encoding of parameter types 11
9.3 BPDU formats and parameters 13

12. Generic Attribute Registration Protocol (GARP) 18

12.7 Overview of GARP protocol operation 18

14. Bridge management 19

14.2 Managed objects 19
14.8 Bridge Protocol Entity 19

17. Rapid Spanning Tree Algorithm and Protocol (RSTP) 24

17.1 Requirements 24
17.2 Requirements of the MAC bridges 25
17.3 Overview 26
17.4 Computation of the active topology 26
17.5 Port States 29
17.6 Topology examples 31
17.7 Communicating Spanning Tree Information 35
17.8 Changing Spanning Tree Information 35
17.9 Changing Port States 36
17.10 Updating Learned Station Location Information 38
17.11 RSTP and point-to-point links 42
17.12 STP compatibility 42

17.13 Rapid Spanning Tree state machines 42
17.14 Notational conventions used in State Diagrams 45
17.15 State machine timers 46
17.16 State machine performance parameters 47
17.17 Per-Bridge Variables 48
17.18 Per-Port variables 49
17.19 State machine procedures 54
17.20 The Port Timers state machine 59
17.21 The Port Information state machine 59
17.22 The Port Role Selection state machine 61
17.23 The Port Role Transitions state machine 62
17.24 Port State Transition state machine 67
17.25 Topology Change state machine 68
17.26 Port Protocol Migration state machine 70
17.27 Port Transmit state machine 71
17.28 Performance 73

Annex A (normative) PICS Proforma 77

Annex B (informative) Calculating Spanning Tree parameters 93

Annex F (informative) Target topology, migration, and interoperability 101

Annex G (informative) Bibliography 108

IEEE Standard for
Local and metropolitan area networks—
Common specifications

Part 3: Media Access Control (MAC) Bridges— Amendment 2: Rapid Reconfiguration

Editorial notes

This amendment to IEEE Std 802.1D, 1998 Edition (ISO/IEC 15802-3:1998) and IEEE Std 802.1t-2001 defines the changes necessary to the operation of a MAC Bridge in order to provide rapid reconfiguration capability. These changes are defined as a series of additions to, and modifications of, the combined text that is generated by applying the editing instructions contained in IEEE Std 802.1t-2001 to the text of IEEE Std 802.1D, 1998 Edition; this amendment, therefore, assumes all material, including references, abbreviations, definitions, procedures, services, and protocols defined in IEEE Std 802.1D, 1998 Edition and IEEE Std 802.1t-2001. Text shown in ***bold italics*** in this amendment defines the editing instructions necessary in order to incorporate the modifications and additions into the base text. Three editing instructions are used: ***change***, ***delete***, and ***insert***. ***Change*** is used to make a change to existing material. The editing instruction specifies the location of the change and describes what is being changed either by using ~~strikethrough~~ (to remove old material) or underscore (to add new material). ***Delete*** removes existing material. ***Insert*** adds new material without changing the existing material. Insertions may require renumbering. If so, renumbering instructions are given in the editing instruction. Editorial notes will not be carried over into future editions of IEEE Std 802.1D, 1998 Edition.

1. Overview

Change the Introduction and Scope as indicated below:

1.1 Introduction

IEEE 802® Local Area Networks (or LANs; see 3.4) of all types can be connected together using MAC Bridges. Each individual LAN has its own independent MAC. The Bridged LAN created allows the interconnection of stations attached to separate LANs as if they were attached to a single LAN, although they are in fact attached to separate LANs each with its own MAC. A MAC Bridge operates below the MAC Service Boundary, and is transparent to protocols operating above this boundary, in the Logical Link Control (LLC) sublayer or Network Layer (ISO/IEC 7498-1: 1994[1]). The presence of one or more MAC Bridges can lead to differences in the Quality of Service provided by the MAC sublayer; it is only because of such differences that MAC Bridge operation is not fully transparent.

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

A Bridged LAN can provide for

a) The interconnection of stations attached to LANs of different MAC types;

b) An effective increase in the physical extent, the number of permissible attachments, or the total performance of a LAN;

c) Partitioning of the physical LAN for administrative or maintenance reasons;

d) Validation of access to the LAN~~.~~;

e) Increased availability of the MAC Service in the face of reconfiguration or failure of components of the Bridged LAN.

NOTE 1—Scope, definitions, references, and conformance requirements relating to the operation of Source Routing Transparent Bridge operation can be found in Annex C.1.

NOTE 2—Validation of access to the LAN is supported when this standard is used in conjunction with the Port-based access control mechanisms specified in IEEE Std 802.1X-2001.

1.2 Scope

For the purpose of compatible interconnection of data processing equipment using the IEEE 802 MAC Service supported by interconnected IEEE 802 LANs (see 3.4) using different or identical Media Access Control methods, this standard specifies a general method for the operation of MAC Bridges. To this end it

a) Positions the bridging function within an architectural description of the MAC Sublayer.

b) Defines the principles of operation of the MAC Bridge in terms of the support and preservation of the MAC Service, and the maintenance of Quality of Service.

c) Specifies the MAC Internal Sublayer Service provided by individual LANs to the Media Access Method Independent Functions that provide frame relay in the Bridge.

d) Identifies the functions to be performed by Bridges, and provides an architectural model of the internal operation of a Bridge in terms of Processes and Entities that provide those functions.

e) Establishes the requirements for a protocol between the Bridges in a Bridged LAN to configure the network, and specifies the distributed computation of a Spanning Tree active topology.

f) Specifies the encoding of the Bridge Protocol Data Units (BPDUs).

g) Establishes the requirements for Bridge Management in the Bridged LAN, identifying the managed objects and defining the management operations.

h) Establishes the requirements for a protocol between Bridges in a Bridged LAN to configure multicast filtering information, and specifies the means of registering and distributing multicast filtering information by means of the GARP Multicast Registration Protocol (GMRP).

i) Specifies the encoding of GMRP protocol data units.

j) Specifies performance requirements and recommends default values and applicable ranges for the operational parameters of a Bridge.

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

l) Specifies criteria for the use of MAC-specific bridging methods.

m) Specifies enhancements to the operation of Spanning Tree protocol, and to the other mechanisms that support reconfiguration of the physical and filtering connectivity, in order to allow support for rapid reconfiguration of Bridged LAN connectivity.

This standard specifies the operation of MAC Bridges that attach directly to IEEE 802 LANs, as specified in the relevant MAC standards for the MAC technology or technologies implemented.

The configuration protocol and associated algorithm required for the distributed computation of a Spanning Tree active topology referred to in item e) appears in two forms in this standard. Clause 8 specifies a version of the Spanning Tree algorithm and protocol that is consistent with the specification contained in IEEE Std 802.1D, 1998 Edition; Clause 17 specifies the Rapid Spanning Tree Algorithm and Protocol; this version offers a significant reduction in the time taken to reconfigure the active topology of the Bridged LAN in the face of changes to the physical topology or its configuration parameters. The two versions of the algorithm and protocol are capable of interoperating within the same Bridged LAN; hence, it is not necessary for implementations to support both versions of the Spanning Tree algorithm and protocol.

In view of the improved performance offered, it is recommended that the Rapid Spanning Tree algorithm and Protocol is supported in preference to the original version in new MAC Bridge implementations.

NOTE 1—The original version of the Spanning Tree algorithm and protocol has been retained in this standard to allow implementations that pre-date the availability of the Rapid Spanning Tree Algorithm and Protocol to continue to claim conformance to IEEE Std 802.1D.

The specification of Remote Bridges, which interconnect LANs using Wide Area Network (WAN) media for the transmission of frames between Bridges, is outside the scope of this standard.

NOTE 2—Remote MAC Bridging is specified in ~~ISO/IEC 15802-5: 1997 [ANSI/IEEE Std 802.1G, 1997 Edition]~~ ANSI/IEEE Std 802.1G, 1998 Edition (ISO/IEC 15802-5:1998) [B1][2].

2. References

Add the following references, in alphabetical sequence with the existing references:

IEEE Std 802.1t-2001, Supplement to ISO/IEC 15802-3 (IEEE Std 802.1D) Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Common specifications—Part 3: Media Access Control (MAC) Bridges: Technical and editorial Corrections.

IEEE Std 802.1X-2001, IEEE Standard for Local and Metropolitan Area Networks: Port based Network Access Control.

ISO/IEC 14882: 1998, Information Tecnology—Programming languages—C++.

4. Abbreviations

Insert the following abbreviations, in the correct collating sequence:

RSTP	Rapid Spanning Tree Algorithm and Protocol
STP	Spanning Tree Algorithm and Protocol
RST BPDU	Rapid Spanning Tree Bridge Protocol Data Unit
TCN	Topology Change Notification

[2] The numbers in brackets preceded by the letter B correspond to those of the bibliography in Annex G.

5. Conformance

5.1 Static conformance requirements

Replace items h) through j) as follows:

~~h) Implement the Spanning Tree Algorithm and Protocol, described in Clause 8, as specified in 8.7.~~

~~i) Not exceed the values given in 8.10.2 for the following parameters:~~

  ~~1) Maximum bridge transit delay~~

  ~~2) Maximum Message Age increment overestimate~~

  ~~3) Maximum BPDU transmission delay~~

~~j) Use the value given in Table 8-3 for the following parameter:~~

  ~~1) Hold Time~~

h) Implement one of:

  1) The Spanning Tree Algorithm and Protocol, described in Clause 8, as specified in 8.7, or

  2) The Rapid Spanning Tree Algorithm and Protocol, described in Clause 17, as specified in 17.13.

i) If the Spanning Tree Algorithm and Protocol is implemented:

  1) Not exceed the values given in 8.10.2 for the following parameters:

    i) Maximum bridge transit delay

    ii) Maximum Message Age increment overestimate

    iii) Maximum BPDU transmission delay

  2) Use the value given in Table 8-3 for the following parameter:

    i) Hold Time

j) If the Rapid Spanning Tree Algorithm and Protocol is implemented:

  1) Not exceed the values given in 17.28.2 for the following parameters:

    i) Maximum bridge transit delay

    ii) Maximum Message Age increment overestimate

    iii) Maximum BPDU transmission delay

  2) Use the value given in Table 17-5 for the following parameter:

    i) Hold Time

  3) Implement the adminEdgePort and operEdgePort parameters and the operation of the Bridge Detection state machine on all Ports of the Bridge, as defined in Clause 18 of IEEE Std 802.1t-2001 .

  4) Implement the adminPointToPointMAC and operPointToPointMAC parameters on all Ports of the Bridge, in accordance with the definition of those parameters in 6.4 and any relevant specific MAC procedures defined in 6.5.

5.2 Options

Change items e) through i) as follows:

e) Provide the capability to assign values to the following parameters to allow configuration of the Spanning Tree active topology:

   1) Bridge Priority

   2) Port Priority

   3) Path Cost for each Port

   A Bridge that provides this capability shall implement the range of values specified in 8.10.2 and Table 8-4 and Table 8-5 if the Spanning Tree Algorithm and Protocol is implemented, or shall implement the range of values specified in 17.28.2 and Table 17-6 and Table 17-7 if the Rapid Spanning Tree Algorithm and Protocol is implemented.

f) Provide the capability to set the values of the following parameters of the Spanning Tree Algorithm and Protocol:

   1) Bridge Max Age

   2) Bridge Hello Time

   3) Bridge Forward Delay

   A Bridge that provides this capability shall implement the range of values specified in 8.10.2 and Table 8-3 if the Spanning Tree Algorithm and Protocol is implemented, or shall implement the range of values specified in 17.28.2 and Table 17-5 if the Rapid Spanning Tree Algorithm and Protocol is implemented.

g) Support management of the Bridge. Bridges claiming to support management shall support all the management objects and operations defined in Clause 14.

h) Support the use of a remote management protocol. Bridges claiming to support remote management shall

   1) State which remote management protocol standard(s) or specification(s) are supported.

   2) State which standard(s) or specification(s) for managed object definitions and encodings are supported for use by the remote management protocol.

i) Support the ability to disable topology change detection, as described in 8.5.5.10, only if the Spanning Tree Algorithm and Protocol is implemented.

NOTE—The Rapid Spanning Tree Algorithm and Protocol does not support the ability to disable topology change detection; in RSTP, this function has been superseded by the Admin and Oper Edge Port parameters.

Add the following as items p and q):

p) Implement both the Spanning Tree Algorithm and Protocol, described in Clause 8, as specified in 8.7, and the Rapid Spanning Tree Algorithm and Protocol, described in Clause 17, as specified in 17.13. The implementation shall support only one of the two algorithms on all Bridge Ports at any given time.

q) Support the ability to transfer learned MAC Address information from a retiring Root Port to a new Root Port, as described in 17.10.

6. Support of the MAC Service

Change the following subclauses of Clause 6 as indicated:

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 Bridge can increase or lower the service availability.

The service availability can be increased by automatic reconfiguration of the 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 Bridge itself, through denial of service by the Bridge, or through frame filtering by the 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 Internal Sublayer Service (6.4 and 6.5);

b) Detection of component failure through the operation of the Spanning Tree algorithm and protocol;

c) Explicit signalling of reconfiguration events through the operation of the Spanning Tree algorithm and protocol.

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

A Bridge may deny service and discard frames (6.3.2) in order to preserve other aspects of the MAC Service (6.3.3 an 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. Bridges may filter frames in order to localize traffic in the 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 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). However, filtering information that is statically configured cannot be modified in this way.

A 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 Bridges, except as a consequence of a failure, removal, or insertion of a 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 Bridged LAN, and is independent of individual instances of service provision.

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

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 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 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 Bridged LAN necessitate frame discard for a limited period of time to maintain other aspects of Quality of Service (see 8.3.3 and 17.9).

   5) The device attached to the Port is not authorized for access to the network.

   6) The configuration of Static Filtering Entries in the Filtering Database (7.9.1) disallows the forwarding of frames with particular destination addresses on specific Ports.

   7) As Static Filtering Entries are associated with particular Ports or combinations of Ports, there is a possibility that mis-configuration of Static Filtering Entries will lead to unintended frame discard during or following automatic reconfiguration of the Bridged LAN.

6.3.3 Frame misordering

The MAC Service (9.2 of ISO/IEC 15802-1) ~~does not permit the~~ permits a negligible rate of reordering of frames with a given user priority for a given combination of destination address and source address. MA_UNITDATA.indication service primitives corresponding to MA_UNITDATA.request primitives, with the same requested priority and for the same combination of 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 Bridges (7.7) is such that the frame-ordering characteristics of the MAC Service are preserved.

Where Bridges in a 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), frame misordering cannot occur during normal operation. Where RSTP is in use (see Clause 17), there is an increased probability that frames that are in transit through the Bridged LAN will be misordered, due to the fact that a 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 RSTP Bridges to be used in environments where sensitive protocols are in use, the forceVersion parameter (17.16.1) can be used to force an RSTP Bridge to operate in an STP-compatible manner. A more detailed discussion of misordering in RSTP can be found in F.2.4.

6.3.4 Frame duplication

The MAC Service (9.2 of ISO/IEC 15802-1) ~~does not permit the~~ permits a negligible rate of duplication of frames. The operation of Bridges ~~does not introduce~~ introduces a negligible rate of duplication of user data frames.

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

~~A Bridge shall not duplicate user data frames.~~ Where Bridges in a 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), there is an increased probability that a Spanning Tree reconfiguration event can cause frames that are in transit through the Bridged LAN to be duplicated, due to the fact that a Bridge can buffer frames awaiting transmission through its Ports. As the probability of duplication occurring as a result of such an event is small, and the frequency of Spanning Tree reconfiguration events is also small, the degradation of the properties of the MAC service caused by this source of frame duplication is considered to be negligible. A more detailed discussion of frame duplication in RSTP can be found in F.2.4.

6.4 Internal Sublayer Service provided within the MAC Bridge

Add a new subclause 6.4.3 as follows:

6.4.3 Point to Point MAC parameters

In addition to the unit-data service primitives described, the Internal Sublayer Service provided by a MAC entity to the MAC Relay Entity within a Bridge makes available a pair of parameters that permit inspection of, and control over, the administrative and operational state of the point-to-point status of the MAC entity by the MAC Relay Entity. These parameters are defined as follows:

operPointToPointMAC: This parameter can take two values, as follows:

a) **True**. This value indicates that the MAC is connected to a point-to-point LAN segment; i.e., there is at most one other system attached to the LAN segment.

b) **False**. This value indicates that the MAC is connected to a non-point-to-point LAN segment; i.e., there can be more than one other system attached to the LAN segment.

adminPointToPointMAC: This parameter can take three values, as follows:

a) **ForceTrue**. This value indicates that the administrator requires the MAC to be treated as if it is connected to a point-to-point LAN segment, regardless of any indications to the contrary that are generated by the MAC entity.

b) **ForceFalse**. This value indicates that the administrator requires the MAC to be treated as if it is connected to a non-point-to-point LAN segment, regardless of any indications to the contrary that are generated by the MAC entity.

c) **Auto**. This value indicates that the administrator requires the point-to-point status of the MAC to be determined in accordance with the specific MAC procedures defined in 6.5.

If adminPointToPointMAC is set to ForceTrue, then operPointToPointMAC shall be set True. If adminPointToPointMAC is set to ForceFalse, then operPointToPointMAC shall be set False.

If adminPointToPointMAC is set to Auto, then the value of operPointToPointMAC is determined in accordance with the specific procedures defined for the MAC entity concerned, as defined in 6.5. If these procedures determine that the MAC entity is connected to a point-to-point LAN segment, then operPointToPointMAC is set TRUE; otherwise it is set FALSE. In the absence of a specific definition of how to determine whether the MAC is connected to a point-to-point LAN segment or not, the value of operPointToPointMAC shall be FALSE.

The value of operPointToPointMAC is determined dynamically; i.e., it is re-evaluated whenever the value of adminPointToPointMAC changes, and whenever the specific procedures defined for the MAC entity evaluate a change in its point-to-point status.

6.5 Support of the Internal Sublayer Service by specific MAC procedures

6.5.1 Support by IEEE Std 802.3 (CSMA/CD)

Add the following text at the end of this subclause:

From the point of view of determining the value of operPointToPointMAC (6.4.3), the MAC is considered to be connected to a point-to-point LAN segment if any of the following conditions are true:

a) The MAC entity concerned contains a Link Aggregation sublayer, and the set of physical MACs associated with the Aggregator are all aggregatable; or

b) The MAC entity concerned supports autonegotiation (Clause 28 of IEEE Std 802.3), and the autonegotiation function has determined that the LAN segment is to be operated in full duplex mode; or

c) The MAC entity has been configured by management means for full duplex operation.

Otherwise, the MAC is considered to be connected to a LAN segment that is not point-to-point.

8. The Spanning Tree algorithm and protocol

Change the definition of Hold Time in Table 8-3, as follows:

Table 8-3—Spanning Tree algorithm timer values

Parameter	Recommended or default value	Fixed value	Range
Bridge Hello Time	2.0	—	1.0–10.0
Bridge Max Age	20.0	—	6.0–40.0
Bridge Forward Delay	15.0	—	4.0–30.0
Hold Time	—	~~Not more than 3 BPDUs transmitted in any 2 second interval~~ Not more than TxHoldCount (17.16.6) BPDUs transmitted in any HelloTime (17.16.3) interval	—

All times are in seconds.
— Not applicable.
Subclause 8.10.2 constrains the relationship between Bridge Max Age and Bridge Forward Delay.

NOTE—Implementations that are conformant to the definition of Hold Time in IEEE Std 802.1D, 1998 Edition are also conformant to the revised definition of Hold Time stated in this table.

9. Encoding of Bridge Protocol Data Units (BPDUs)

Change the text of Clause 9 as shown:

This clause specifies the structure and encoding of the BPDUs for the Spanning Tree Protocol (Clause 8) and Rapid Spanning Tree Protocol (Clause 17), exchanged between Bridge Protocol Entities.

9.1 Structure

9.1.1 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). The bits in an octet are numbered from 1 to 8, where 1 is the low-order bit.

When consecutive bits within an octet are used to represent a binary number, the higher bit number has the most significant value.

When consecutive octets are used to represent a binary number, the lower octet number has the most significant value.

All Bridge Protocol Entities respect these bit and octet ordering conventions, thus allowing communications to take place.

9.1.2 Components

A Protocol Identifier is encoded in the initial octets of all BPDUs. This standard ~~reserves~~ specifies a single Protocol Identifier value for use in BPDUs. All other Protocol Identifier values are reserved for future standards use. This standard places no further restriction on the structure, encoding, or use of BPDUs with different values of the Protocol Identifier field, should these exist, by other standard protocols.

BPDUs used by Bridge Protocol Entities operating the Spanning Tree Algorithm and Protocol specified in Clause 8 and the Rapid Spanning Tree Algorithm and Protocol specified in Clause 17 use the ~~reserved~~specified Protocol Identifier value and have the following structure.

9.2 Encoding of parameter types

9.2.1 Encoding of protocol identifiers

A Protocol Identifier shall be encoded in two octets.

9.2.2 Encoding of protocol version identifiers

A Protocol Version Identifier shall be encoded in one octet. If two Protocol Version Identifiers are interpreted as unsigned binary numbers, then the greater number will be associated with the more recently defined Protocol Version.

9.2.3 Encoding of BPDU types

The type of the BPDU shall be encoded as a single octet. The bit pattern contained in the octet merely serves to distinguish the type; no ordering relationship between BPDUs of different types is implied.

9.2.4 Encoding of flags

A flag shall be encoded as a bit in a single octet. A number of flags may be thus encoded in a single octet. A flag is set if the corresponding bit in the octet takes the value 1. Bit positions in the octet that do not correspond to flags defined for a given type of BPDU are reset, i.e., shall take the value 0. No additional flags will be defined for a BPDU of given protocol version and type.

9.2.5 Encoding of Bridge Identifiers

A Bridge Identifier shall be encoded as eight octets, taken to represent an unsigned binary number. Two Bridge Identifiers may be numerically compared and the lesser number shall denote the Bridge with ~~higher~~ better priority.

NOTE 1—Use of the terms “higher” and “lower” to describe both the relative numerical values and the relative priority of Spanning Tree priority information can cause confusion, as lesser numbers convey better priorities. In this clause and in Clause 17 (Rapid Spanning Tree), relative numeric values are described as “least,” “lesser,” “equal,” and “greater,” and their comparisons as “less than,” “equal to,” or “greater than,” while relative Spanning Tree priorities are described as “best,” “better,” “the same,” “different,” and “worse” and their comparisons as “better than,” “the same as,” “different from,” and “worse than.” The terms “higher” and “lower” have been retained in Clause 8. Further, in Clause 17, the terms “superior” and “inferior” have been introduced for comparisons not simply based on strict ordered comparison of priority components. This distinction is not necessary in Clause 8.

The four most significant bits of the most significant octet of a Bridge Identifier comprises a settable priority component that permits the relative priority of Bridges to be managed (8.5.3.7 and Clause 14). The next most significant twelve bits of a Bridge Identifier (the four least significant bits of the most significant octet, plus the second most significant octet) comprise a locally assigned system ID extension. The six least significant octets ensure the uniqueness of the Bridge Identifier; they shall be derived from the globally unique Bridge Address (7.12.5) according to the following procedure.

NOTE 2—The number of bits that are considered to be part of the system ID (60 bits) differs in this version of the standard from the 1998 and prior versions (formerly, the priority component was 16 bits and the system ID component 48 bits). This change was made in order to allow implementations of Multiple Spanning Trees (P802.1s, a supplement to IEEE Std 802.1Q) to make use of the 12-bit system ID extension as a means of generating a distinct Bridge Identifier per

VLAN, rather than forcing such implementations to allocate up to 4094 MAC addresses for use as Bridge Identifiers. To maintain management compatibility with older implementations, the priority component is still considered, for management purposes, to be a 16-bit value, but the values that it can be set to are restricted to only those values where the least significant 12 bits are zero (i.e., only the most significant 4 bits are settable).

The third most significant octet is derived from the initial octet of the MAC Address; the least significant bit of the octet (Bit 1) is assigned the value of the first bit of the Bridge Address, the next most significant bit is assigned the value of the second bit of the Bridge Address, and so on. In a Bridged LAN utilizing 48-bit MAC Addresses, the fourth through eighth octets are similarly assigned the values of the second to the sixth octets of the Bridge Address.

9.2.6 Encoding of Root Path Cost

Root Path Cost shall be encoded as four octets, taken to represent an unsigned binary number, a multiple of arbitrary cost units. Subclause 8.10.2 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.

9.2.7 Encoding of Port Identifiers

A Port Identifier shall be encoded as two octets, taken to represent an unsigned binary number. If two Port Identifiers are numerically compared, the lesser number shall denote the Port with ~~higher~~ better priority. The more significant four bits of a Port Identifier is a settable priority component that permits the relative priority of Ports on the same Bridge to be managed (8.5.5 and Clause 14). The less significant twelve bits is the Port Number expressed as an unsigned binary number. The value 0 is not used as a Port Number.

NOTE—The number of bits that are considered to be part of the Port Number (12 bits) differs in this version of the standard from the 1998 and prior versions (formerly, the priority component was 8 bits and the Port Number component also 8 bits). This change was made in recognition of the fact that modern switched LAN infrastructures call for increasingly large numbers of Ports to be supported in a single Bridge. To maintain management compatibility with older implementations, the priority component is still considered, for management purposes, to be an 8-bit value, but the values that it can be set to are restricted to those values where the least significant 4 bits are zero (i.e., only the most significant 4 bits are settable).

9.2.8 Encoding of Timer Values

Timer Values shall be encoded in two octets, taken to represent an unsigned binary number multiplied by a unit of time of 1/256 of a second. This permits times in the range 0 to, but not including, 256 s to be represented.

9.2.9 Encoding of Port Role values

Port Role values shall be encoded in two consecutive flag bits, taken to represent an unsigned integer, as follows:

a) A value of 0 indicates Unknown.

b) A value of 1 indicates Alternate or Backup.

c) A value of 2 indicates Root.

d) A value of 3 indicates Designated.

The Unknown value of Port Role cannot be generated by a valid implementation; however, this value is accepted on receipt.

NOTE—Should the Unknown value of the Port Role parameter be received, the state machines will effectively treat the RST BPDU as if it were a Configuration BPDU.

9.2.10 Encoding of Length Values

Length Values shall be encoded in two octets, taken to represent an unsigned binary number.

9.3 BPDU formats and parameters

9.3.1 Configuration BPDUs

The format of the Configuration BPDUs is shown in Figure 9-1. Each transmitted Configuration BPDU shall contain the following parameters (8.5.1) and no others. Where a specific parameter value is indicated in this subclause, that parameter value shall be encoded in all transmitted Configuration BPDUs:

a) The Protocol Identifier is encoded in Octets 1 and 2 of the BPDU. It takes the value 0000 0000 0000 0000, which identifies the Spanning Tree Algorithm and Protocol as specified in Clause 8 and the Rapid Spanning Tree Algorithm and Protocol as specified in Clause 17.

b) The Protocol Version Identifier is encoded in Octet 3 of the BPDU. It takes the value 0000 0000.

c) The BPDU Type is encoded in Octet 4 of the BPDU. This field ~~shall take~~ takes the value 0000 0000. This denotes a Configuration BPDU.

d) The Topology Change Acknowledgment flag is encoded in Bit 8 of Octet 5 of the BPDU.

e) The Topology Change flag is encoded in Bit 1 of Octet 5 of the BPDU.

f) The remaining flags, Bits 2 through 7 of Octet 5, are unused and take the value 0.

g) The Root Identifier is encoded in Octets 6 through 13 of the BPDU.

h) The Root Path Cost is encoded in Octets 14 through 17 of the BPDU.

i) The Bridge Identifier is encoded in Octets 18 through 25 of the BPDU.

j) The Port Identifier is encoded in Octets 26 and 27 of the BPDU.

k) The Message Age timer value is encoded in Octets 28 and 29 of the BPDU.

l) The Max Age timer value is encoded in Octets 30 and 31 of the BPDU.

m) The Hello Time timer value is encoded in Octets 32 and 33 of the BPDU.

n) The Forward Delay timer value is encoded in Octets 34 and 35 of the BPDU.

The Message Age (Octets 28 and 29) shall be less than Max Age (Octets 30 and 31).

	Octet
Protocol Identifier	1
	2
Protocol Version Identifier	3
BPDU Type	4
Flags	5
Root Identifier	6
	7
	8
	9
	10
	11
	12
	13
Root Path Cost	14
	15
	16
	17
Bridge Identifier	18
	19
	20
	21
	22
	23
	24
	25
Port Identifier	26
	27
Message Age	28
	29
Max Age	30
	31
Hello Time	32
	33
Forward Delay	34
	35

Figure 9-1—Configuration BPDU parameters and format

9.3.2 Topology change notification BPDUs

The format of the Topology Change Notification BPDUs is shown in Figure 9-2. Each transmitted Topology Change Notification BPDU shall contain the following parameters (8.5.2) and no others. Where a specific parameter value is indicated in this subclause, that parameter value shall be encoded in all transmitted Topology Change Notification BPDUs:

a) The Protocol Identifier is encoded in Octets 1 and 2 of the BPDU. It takes the value 0000 0000 0000 0000, which identifies the Spanning Tree Algorithm and Protocol as specified in Clause 8 and the Rapid Spanning Tree Algorithm and Protocol as specified in Clause 17~~of this standard~~.

b) The Protocol Version Identifier is encoded in Octet 3 of the BPDU. It takes the value 0000 0000.

c) The BPDU Type is encoded in Octet 4 of the BPDU. This field ~~shall take~~ takes the value 1000 0000 (where bit 8 is shown at the left of the sequence). This denotes a Topology Change Notification BPDU.

	Octet
Protocol Identifier	1
	2
Protocol Version Identifier	3
BPDU Type	4

Figure 9-2—Topology change notification BPDU parameters and format

9.3.3 Rapid Spanning Tree BPDUs (RST BPDUs)

The format of the RST BPDUs is shown in Figure 9-3. Each transmitted RST BPDU shall contain the following parameters and no others. Where a specific parameter value is indicated in this subclause, that parameter value shall be encoded in all transmitted RST BPDUs:

a) The Protocol Identifier is encoded in Octets 1 and 2 of the BPDU. It takes the value 0000 0000 0000 0000, which identifies the Spanning Tree Algorithm and Protocol as specified in Clause 8 and the Rapid Spanning Tree Algorithm and Protocol as specified in Clause 17.

b) The Protocol Version Identifier is encoded in Octet 3 of the BPDU. It takes the value 0000 0010.

c) The BPDU Type is encoded in Octet 4 of the BPDU. This field takes the value 0000 0010. This denotes a Rapid Spanning Tree BPDU.

d) The Topology Change flag is encoded in Bit 1 of Octet 5 of the BPDU (see 17.19.16).

e) The Proposal flag is encoded in Bit 2 of Octet 5 of the BPDU (see 17.19.16).

f) The Port Role is encoded in Bits 3 and 4 of Octet 5 of the BPDU (see 17.19.16).

g) The Learning flag is encoded in Bit 5 of Octet 5 of the BPDU (see 17.19.16).

h) The Forwarding flag is encoded in Bit 6 of Octet 5 of the BPDU (see 17.19.16).

i) The Agreement flag is encoded in Bit 7 of Octet 5 of the BPDU (see 17.19.16).

j) The Topology Change Acknowledgment flag is encoded in Bit 8 of Octet 5 of the BPDU as zero (see 17.19.16).

k) The Root Identifier is encoded in Octets 6 through 13 of the BPDU (see 17.18.17, 17.19.16).

l) The Root Path Cost is encoded in Octets 14 through 17 of the BPDU (see 17.18.17, 17.19.16).

m) The Bridge Identifier is encoded in Octets 18 through 25 of the BPDU. (see 17.18.17, 17.19.16)

n) The Port Identifier is encoded in Octets 26 and 27 of the BPDU (see 17.18.17, 17.19.16).

o) The Message Age timer value is encoded in Octets 28 and 29 of the BPDU (see 17.18.18, 17.19.16).

p) The Max Age timer value is encoded in Octets 30 and 31 of the BPDU (see 17.18.18, 17.19.16).

q) The Hello Time timer value is encoded in Octets 32 and 33 of the BPDU (see 17.18.18, 17.19.16).

r) The Forward Delay timer value is encoded in Octets 34 and 35 of the BPDU (see 17.18.18, 17.19.16).

s) The Version 1 Length value is encoded in Octet 36 of the BPDU. It takes the value 0000 0000, which indicates that there is no Version 1 protocol information present.

NOTE—The presence of a Version 1 Length value of 0, indicating that no version 1 information is present, is required in Version 2 BPDUs in order to make it possible to define subsequent versions of the protocol that can carry additional parameters other than those defined for Version 1 of the protocol (defined in IEEE Std 802.1G).

The Message Age (Octets 28 and 29) shall be less than Max Age (Octets 30 and 31).

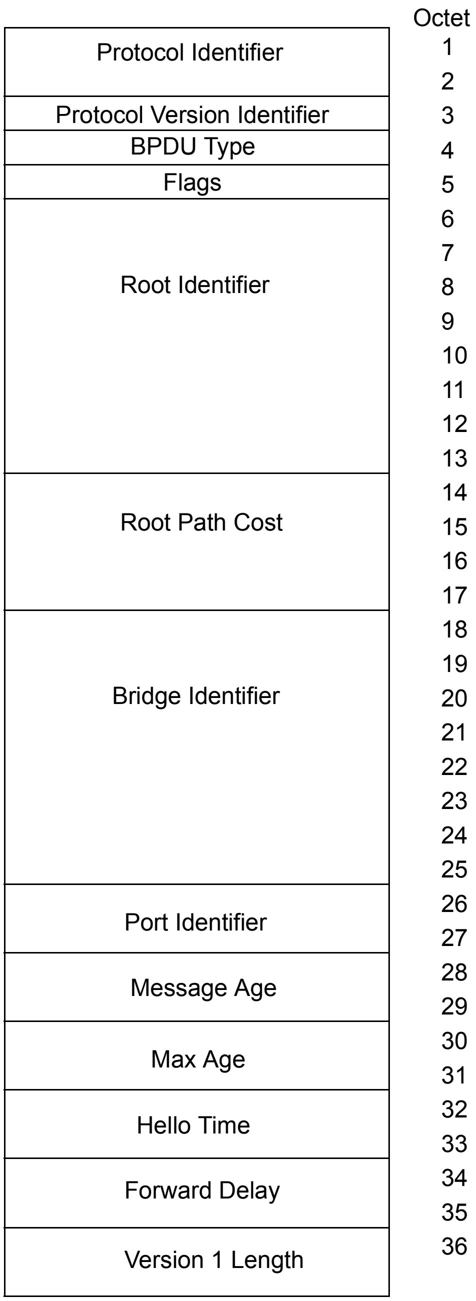


Figure 9-3—RST BPDU parameters and format

9.3.4 Validation of received BPDUs

A Bridge Protocol Entity shall process a received BPDU as specified in 8.7 and 17.13 if and only if the BPDU contains at least four octets and the Protocol Identifier has the value specified for BPDUs (9.3.2), and

a) The BPDU Type denotes a Configuration BPDU and the BPDU contains at least 35 octets, and the Bridge Identifier and Port Identifier parameters from the received BPDU do not match the values that would be transmitted in a BPDU from this Port; or

NOTE 1—If the Bridge Identifier and Port Identifier both match the values that would be transmitted in a BPDU from this Port, then the BPDU is discarded, in order to prevent processing of the Port's own BPDUs; for example, if they are received by the Port as a result of a loopback condition. If a loopback condition exists, then there may be other undesirable side effects with respect to Bridge operation caused by the looping back of data frames relayed through the Port.

b) The BPDU Type denotes a Topology Change Notification BPDU; or

c) The BPDU Type denotes a Rapid Spanning Tree BPDU and the BPDU contains at least 36 octets, and the Bridge Identifier and Port Identifier parameters from the received BPDU do not match the values that would be transmitted in a BPDU from this Port.

NOTE 2—The operation of the RSTP Port Information state machine (see 17.21) checks that the value of the BPDU's Message Age parameter is less than that of its Max Age parameter, and if not, will immediately age out the received information.

~~In case a), any octets that are present beyond Octet 35 are ignored, as far as processing according to this standard is concerned. Similarly, in case b), any octets beyond Octet 4 are ignored.~~

~~NOTE—The Protocol Version Identifier is not checked on receipt, in order to allow the possibility of future specification of extensions to the Spanning Tree Protocol, identified as new versions by different values of the Protocol Version Identifier.~~

The following rules apply to the validation and interpretation of BPDUs, in order to ensure that backwards compatibility is maintained between versions of this protocol.

For an implementation that supports version A of the protocol, a received BPDU of a given type that carries a protocol version number B is interpreted as follows:

d) Where B is greater than or equal to A, the BPDU shall be interpreted as if it carried the supported version number, A. Specifically:

   1) All BPDU types, parameters, and flags that are defined in version A shall be interpreted in the manner specified for version A of the protocol for the given BPDU type.

   2) All BPDU types, parameters, and flags that are undefined in version A for the given BPDU type shall be ignored.

   3) All octets that appear in the BPDU beyond the largest numbered octet defined for version A for the given BPDU type shall be ignored.

e) Where B is less than A, the BPDU shall be interpreted as specified for the version number, B, carried in the BPDU. Specifically:

   1) All BPDU parameters and flags shall be interpreted in the manner specified for version B of the protocol for the given BPDU type.

   2) All BPDU parameters and flags that are undefined in version B for the given BPDU type shall be ignored.

   3) All octets that appear in the BPDU beyond the largest numbered octet defined for version B for the given BPDU type shall be ignored.

NOTE 3—In other words, if the protocol version implemented differs from the protocol version number carried in the BPDU, then only those BPDU types, parameters, and flags that are specified within the lesser numbered protocol version are interpreted by the implementation (in accordance with the lesser numbered protocol version's specification), and no attempt is made to interpret any additional BPDU types, parameters, and flags that may be specified within the greater numbered protocol version. In the specific case of STP (version 0) and RSTP (version 2), as there is only a single RST BPDU type defined in version 2, and as the RST BPDU type is undefined in version 0, a version 0 implementation will ignore all RST BPDUs. Version 2 implementations, however, recognize and process both version 0 and version 2 BPDUs. As version 2 makes no changes to the BPDU types defined for version 0 (and always transmits such BPDU types with 0 as the version identifier), version 0 BPDUs are always interpreted by version 2 implementations according to their version 0 definition.

12. Generic Attribute Registration Protocol (GARP)

12.7 Overview of GARP protocol operation

Add new subclause 12.7.10:

12.7.10 Use of GARP in point-to-point LANs

The full GARP participant state machine was designed to operate correctly in shared media environments. The fact that in such environments there might be three or more active GARP participants was the motivation behind the Passive Member and Observer states; the result is reduced traffic on a shared media LAN where more than one participant is interested in registering a particular attribute value.

In LANs that are based on point-to-point connectivity between systems, i.e., where it is certain that there can at most be only two GARP participants on any given LAN, the added complexity of the full Applicant state machine is redundant. Therefore, it is recommended in such environments to implement the Simple Applicant state machine in preference to the full Applicant state machine. The Point to Point MAC parameters (see 6.4.3) provide a means of determining whether a given MAC supports a shared media LAN segment or a point to point LAN segment.

The method of determining the timing of transmission opportunities in the state machines (the use of a timer value randomized between 0 and JoinTime seconds) was also chosen to accommodate the use of GARP in shared media environments, to avoid the risk of multicast storms occurring during periods of configuration change. In point-to-point LANs, these considerations do not apply. More particularly, in LANs that support RSTP, it is desirable to allow transmission opportunities to occur without delay under circumstances where a Rapid Spanning Tree configuration change is occurring, in order to minimize the period during which denial of service might occur due to the delay in propagating registration changes. Therefore, in point-to-point LANs, it is recommended that the definition of when a transmission opportunity (transmitPDU!—see 12.8) can occur is redefined as follows:

transmitPDU! An opportunity to transmit a GARP PDU has occurred. A maximum transmission rate is imposed of no more than three such transmission opportunities in any period of 1.5*JoinTime seconds.

14. Bridge management

Change 14.2, Managed Objects, as follows:

14.2 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 7.3 and 12.2. Specifically

a) The Bridge Management Entity (14.4 and 7.11).

b) The individual MAC Entities associated with each Bridge Port (14.5, 7.2, 7.5, and 7.6).

c) The Forwarding Process of the MAC Relay Entity (14.6, 7.2, and 7.7).

d) The Filtering Database of the MAC Relay Entity (14.7 and 7.9).

e) The Bridge Protocol Entity (14.8, 7.10, ~~and~~ Clause 8, and Clause 17).

f) GARP Participants (Clause 12).

g) GMRP participants (14.10, IEEE Std 802.1D Clause 10).

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. This standard specifies one such protocol encoding in Clause 15 (for optional remote management).

Change 14.8, Bridge Protocol Entity, and its subclauses, as follows:

14.8 Bridge Protocol Entity

The Bridge Protocol Entity is described in 7.10, ~~and~~ Clause 8, and Clause 17. The objects that comprise this managed resource are

a) The Protocol Entity itself and

b) The Ports under its control.

14.8.1 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 Read Bridge Protocol Parameters and Set Bridge Protocol Parameters.

14.8.1.1 Read Bridge Protocol parameters

14.8.1.1.1 Purpose

To obtain information regarding the Bridge's Bridge Protocol Entity.

14.8.1.1.2 Inputs

None.

14.8.1.1.3 Outputs

a) Bridge Identifier—as defined in 8.5.3.7 and 17.17.3.

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) was last True, or in an RSTP Bridge, the count in seconds of the time since the tcWhile timer (17.15.7) 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 Bridge, the count of times that there has been at least one non-zero tcWhile timer (17.15.7).

d) Topology Change—in an STP Bridge, the value of the Topology Change parameter (8.5.3.12), or in an RSTP Bridge, asserted if the tcWhile timer (17.15.7) for any Port is non-zero.

e) Designated Root (8.5.3.1 and 17.18.17).

f) Root Path Cost (8.5.3.2 and 17.18.17).

g) Root Port (8.5.3.3 and 17.17.5).

h) Max Age (8.5.3.4 and 17.18.18).

i) Hello Time (8.5.3.5 and 17.16.3).

j) Forward Delay (8.5.3.6 and 17.16.2).

k) Bridge Max Age (8.5.3.8 and 17.17.4).

l) Bridge Hello Time (8.5.3.9 and 17.17.4).

m) Bridge Forward Delay (8.5.3.10 and 17.17.4).

n) Hold Time (8.5.3.14) or Transmission Limit (TxHoldCount in 17.16.6).

o) forceVersion—the value of the Force Protocol Version parameter for the Bridge (provided only by RSTP Bridges; see 17.16.1).

14.8.1.2 Set Bridge Protocol parameters

14.8.1.2.1 Purpose

To modify parameters in the Bridge's Bridge Protocol Entity in order to force a configuration of the Spanning Tree and/or tune the reconfiguration time to suit a specific topology. In RSTP implementations, this operation causes these values to be set for all Ports of the Bridge.

14.8.1.2.2 Inputs

a) Bridge Max Age—the new value (8.5.3.8 and 17.17.4).

b) Bridge Hello Time—the new value (8.5.3.9 and 17.17.4).

c) Bridge Forward Delay—the new value (8.5.3.14 and 17.17.4).

d) Bridge Priority—the new value of the priority part of the Bridge Identifier (8.5.3.7 and 17.17.3).

e) forceVersion (optional)—the new value of the Force Protocol Version parameter (provided only by RSTP Bridges; see 17.16.1).

14.8.1.2.3 Outputs

a) Operation status. This takes one of the following values:

   1) Operation rejected due to invalid Bridge Priority value (14.3); or

   2) Operation accepted.

14.8.1.2.4 Procedure

The input parameter values are checked for compliance with 8.10.2 (STP Bridges) or 17.28.2 (RSTP Bridges). 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 (STP Bridges) or Table 17-5 (RSTP 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 (STP Bridges) or Table 17-5 (RSTP Bridges), then the Bridge need not take action.

~~Otherwise, the Bridge's Bridge Max Age, Bridge Hello Time, and Bridge Forward Delay parameters are set to the supplied values. The Set Bridge Priority procedure (8.8.4) is used to set the priority part of the Bridge Identifier to the supplied value.~~

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) is used to set the priority part of the Bridge Identifier to the supplied value.

c) In RSTP Bridges, the priority component of the Bridge Identifier (17.17.3) is updated using the supplied value. For all Ports of the Bridge, the reselect parameter (17.18.29) is set TRUE, and the selected parameter (17.18.31) is set FALSE.

14.8.2 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 Read Port Parameters, Force Port State, ~~and~~ Set Port Parameters, and Force BPDU Migration Check.

14.8.2.1 Read Port Parameters

14.8.2.1.1 Purpose

To obtain information regarding a specific Port within the Bridge’s Bridge Protocol Entity.

14.8.2.1.2 Inputs

a) Port Number—the number of the Bridge Port.

14.8.2.1.3 Outputs

a) Uptime—count in seconds of the time elapsed since the Port was last reset or initialized.

b) State—the current state of the Port (i.e., Disabled, Listening, Learning, Forwarding, or Blocking) (8.4, ~~and~~ 8.5.5.2, and 17.5).

c) Port Identifier—the unique Port identifier comprising two parts, the Port Number and the Port Priority field (8.5.5.1 and 17.18.16).

d) Path Cost (8.5.5.3 and 17.16.5).

e) Designated Root (8.5.5.4 and 17.18.17).

f) Designated Cost (8.5.5.5 and 17.18.17).

g) Designated Bridge (8.5.5.6 and 17.18.17).

h) Designated Port (8.5.5.7 and 17.18.17).

i) Topology Change Acknowledge (8.5.5.8 and 17.18.37).

j) adminEdgePort (18.3.3). Present in implementations that support the identification of edge ports.

k) operEdgePort (18.3.4). Present in implementations that support the identification of edge ports.

l) MAC Enabled—the current state of the MAC Enabled parameter (6.4.2 of IEEE Std 802.1t-2001). Present if the implementation supports the MAC Enabled parameter.

m) MAC Operational—the current state of the MAC Operational parameter (6.4.2 of IEEE Std 802.1t-2001). Present if the implementation supports the MAC Operational parameter.

n) adminPointToPointMAC—the current state of the adminPointToPointMAC parameter (6.4.3). Present if the implementation supports the adminPointToPointMAC parameter.

o) operPointToPointMAC - the current state of the operPointToPointMAC parameter (6.4.3). Present if the implementation supports the operPointToPointMAC parameter.

14.8.2.2 Force port state

14.8.2.2.1 Purpose

To ~~force~~ set the Administrative Bridge Port state (see 17.5) for the specified Port ~~in~~to Disabled or ~~Blocking~~ Enabled.

14.8.2.2.2 Inputs

a) Port Number—the number of the Bridge Port.

b) State—either Disabled or ~~Blocking (8.4 and 8.5.5.2)~~ Enabled.

14.8.2.2.3 Outputs

None.

14.8.2.2.4 Procedure

In Bridges that support STP, if ~~If~~ the selected state is Disabled, the Disable Port procedure (8.8.3) is used for the specified Port. If the selected state is ~~Blocking~~ Enabled, the Enable Port procedure (8.8.2) is used.

In Bridges that support RSTP, the effect of changing this parameter is defined in 17.5.

14.8.2.3 Set Port Parameters

14.8.2.3.1 Purpose

To modify parameters for a Port in the Bridge's Bridge Protocol Entity in order to force a configuration of the Spanning Tree.

14.8.2.3.2 Inputs

a) Port Number—the number of the Bridge Port.

b) Path Cost—the new value (8.5.5.3 and 17.16.5).

c) Port Priority—the new value of the priority field for the Port Identifier (8.5.5.1 and 17.18.17).

d) adminEdgePort—the new value of the adminEdgePort parameter (18.3.3). 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 802.1t-2001). May be present if the implementation supports the MAC Enabled parameter.

f) adminPointToPointMAC—the new value of the adminPointToPointMAC parameter (6.4.3). May be present if the implementation supports the adminPointToPointMAC parameter.

14.8.2.3.3 Outputs

a) Operation status. This takes one of the following values:

   1) Operation rejected due to invalid Port Priority value (14.3); or

   2) Operation accepted.

14.8.2.3.4 Procedure

~~The~~ In STP Bridges, the Set Path Cost procedure (8.8.6) is used to set the Path Cost parameter for the specified Port. The Set Port Priority procedure (8.8.5) is used to set the priority part of the Port Identifier (8.5.5.1) to the supplied value.

In RSTP Bridges, the Path Cost (17.16.5) and Port Priority (17.18.17) parameters for the Port are updated using the supplied values. The reselect parameter for the Port (17.18.29) is set TRUE, and the selected parameter for the Port (17.18.31) is set FALSE.

Insert new subclause 14.8.2.4, Force BPDU Migration Check

14.8.2.4 Force BPDU Migration Check

This operation is available only in Bridges that support RSTP, as specified in Clause 17.

14.8.2.4.1 Purpose

To force the specified Port to transmit RST BPDUs (see 17.26).

14.8.2.4.2 Inputs

a) Port Number—the number of the Bridge Port.

14.8.2.4.3 Outputs

None.

14.8.2.4.4 Procedure

The mcheck variable (17.18.10) for the specified Port is set to the value TRUE if the value of the forceVersion variable (17.16.1) is greater than or equal to 2.

Insert new Clause 17 as follows:

17. Rapid Spanning Tree Algorithm and Protocol (RSTP)

The configuration algorithm and protocol described in this clause reduce the Bridged LAN topology to a single Spanning Tree. The configuration algorithm and protocol described here supersede the Spanning Tree Algorithm and Protocol described in Clause 8, and provide significantly faster reconfiguration. Clause 8 is retained in this standard in order to allow existing implementations of the Spanning Tree Algorithm and Protocol to remain conformant, and Bridges based on either specification can interoperate successfully. However, it is recommended that the Rapid Spanning Tree Algorithm and Protocol (RSTP) be adopted in future MAC Bridge implementations.

17.1 Requirements

The Rapid Spanning Tree Algorithm and Protocol operates to support, preserve, and maintain the quality of the MAC Service (6,1, 6.2, and 6.3), meeting the following requirements:

a) It selects a simply and fully connected active topology from a Bridged LAN of arbitrary physical topology, eliminating data loops (6.3.3 and 6.3.4).

b) It provides for fault tolerance by automatic reconfiguration of the Spanning Tree topology as a result of the failure of LAN components, and for the automatic accommodation of any Bridge or Bridge Port added to the Bridged LAN without the formation of transient data loops (6.1).

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 (6.1).

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 (6.1 and 6.3.10).

e) It operates 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 (6.2).

f) The communications bandwidth consumed by the Bridges in establishing and maintaining a Spanning Tree on any particular LAN is always a very small fraction of the total available bandwidth and is independent of the total traffic supported by the Bridged LAN regardless of the total number of Bridges or LANs (6.3.10).

Additionally, the algorithm and protocol meet the following goals, which limit the complexity of Bridges and their configuration:

g) The memory requirements associated with each Bridge Port are independent of the number of Bridges and LANs in the Bridged LAN.

h) 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.

i) In normal operation, the time taken to configure the active topology of a Bridge LAN is independent of the timer values of the protocol.

NOTE—The values of the timers used can therefore be considerably relaxed. This contrasts with the operation of the Spanning Tree Algorithm and Protocol specified in Clause 8, where the time taken to configure is sensitive to the values of timers.

17.2 Requirements of the MAC bridges

In order for the Bridge Protocol to operate, the following are required:

a) A unique MAC group address, recognized by all the Bridges within the Bridged LAN, that identifies the Bridge Protocol Entities of all Bridges attached to an individual LAN.

b) An identifier for each Bridge, unique within the Bridged LAN.

c) A distinct Port identifier for each Bridge Port, that can be assigned independently of the values used in other Bridges.

Values for each of these parameters, or a mechanism for assigning values to them, shall be provided by each Bridge. In the case of MAC Bridges that use 48-bit Universally Administered Addresses, the unique MAC Address that identifies the Bridge Protocol Entities is the Bridge Group Address (7.12.3).

In addition, to allow the configuration of the Spanning Tree active topology to be managed, the following are required:

d) A means of assigning the relative priority of each Bridge within the set of Bridges in the Bridged LAN.

e) A means of assigning the relative priority of each Port within the set of Ports of an individual Bridge.

f) A means of assigning a path cost component to each Port.

These parameters may be set by management when Bridge Management is supported.

17.3 Overview

The Rapid Spanning Tree Algorithm is a distributed algorithm that selects one Bridge to be the "root" of a fully ("spanning") and simply ("tree") connected active topology, and assigns Port Roles (17.4.1) to individual ports on each Bridge in the Bridged LAN. Port Roles are assigned according to whether the port is to be part of the active topology connecting the Bridge to the Root Bridge (a Root Port) or connecting a LAN through the Bridge to the Root Bridge (a Designated Port), or is an Alternate or Backup Port that may provide connectivity if other Bridges, Bridge Ports, or LANs fail or are removed.

State machines associated with the Port Roles maintain and change the Port States (7.4, 17.5) that control the processing and forwarding of frames by a MAC Relay Entity (7.3). A Port State (17.5) of Discarding, Learning, or Forwarding is assigned to support (6.1) and maintain the quality (6.3) of the MAC Service. In particular, to reduce the probability of data loops and the duplication and misordering of frames to a negligible level, a transition to the Forwarding Port State may be delayed until the assignment of Port Roles is known to be consistent between Bridges.

NOTE—The Rapid Spanning Tree Algorithm and Protocol cannot protect against temporary loops caused by the interconnection 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.

The operation of RSTP provides for rapid recovery of connectivity following the failure of a Bridge, Bridge Port, or a LAN. A new Root Port can transition rapidly to the Forwarding Port State, and the use of explicit acknowledgements between Bridges allow Designated Ports to transition rapidly to the Forwarding Port State. The timers used by RSTP define worst case delays, and are used only as a backup to the normal operation of the protocol.

RSTP 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 at the edge of the Bridged LAN, i.e., where no further Bridges are reachable via that LAN segment.

17.4 Computation of the active topology

17.4.1 Port Role Assignment

The Rapid Spanning Tree Algorithm assigns one of the following Port Roles to each Bridge Port: Root Port, Designated Port, Alternate Port, or Backup Port. A fifth role, Disabled Port, identifies a Port as having no role within the operation of Spanning Tree. Port Role assignments for ports throughout the Bridged Local Area Network are determined as follows by:

a) A unique Bridge Identifier associated with each Bridge

b) A Path Cost associated with each Bridge Port

c) A Port Identifier associated with each Bridge Port

The Bridge with the best Bridge Identifier is selected as the Root Bridge. The unique Bridge Identifier for each Bridge is derived, in part, from the Bridge Address (7.12.5) and, in part, from a manageable priority component (9.2.5). The relative priority of Bridges is determined by the numerical comparison of the unique identifiers, with the lower numerical value indicating the better identifier.

Every Bridge has a Root Path Cost associated with it. For the Root Bridge this is zero. For all other Bridges it is the sum of the Path Costs for each Bridge Port receiving frames on the least cost path from the Root Bridge to that Bridge. The Path Cost associated with each Port may be manageable. Additionally, 17.28.2

recommends default values for the Path Costs associated with Ports attached to LANs of specific MAC types and speeds.

The Bridge Port on each Bridge receiving the frames on the least cost path from the Root Bridge is assigned the role of Root Port for that Bridge (the Root Bridge does not have a Root Port). If a Bridge has two or more ports with the same least Path Cost sum from the Root, then the port with the best Port Identifier is selected as the Root Port. Part of the Port Identifier is fixed and is different for each Port on a Bridge, and part is a manageable priority component (9.2.7). The relative priority of Ports is determined by the numerical comparison of the unique identifiers, with the lower numerical value indicating the better identifier.

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

In a Bridged Local Area Network whose physical topology is stable, i.e., the Rapid Spanning Tree Algorithm has communicated consistent information throughout the network, every LAN has one and only one assigned Designated Port, and every Bridge with the exception of the Root Bridge has a Root Port connected to a LAN.

Any operational Bridge Port that is not assigned a Port Role of Root Port or Designated Port is a Backup Port if that Bridge is the Designated Bridge for the attached LAN, and an Alternate Port otherwise.

An Alternate Port offers an alternate path in the direction of the Root Bridge to that provided by the Bridge's own Root Port, whereas a Backup Port acts as a backup for the path provided by a Designated Port in the direction of the leaves of the Spanning Tree. Backup Ports exist only where there are two or more connections from a given Bridge to a given LAN; hence, they (and the Designated Ports that they back up) can only exist where two ports are connected together in loopback by a point to point link, or where the Bridge has two or more connections to a shared media LAN segment.

NOTE—The distinction between the Alternate and Backup Port Roles does not appear in the Spanning Tree Algorithm and Protocol described in Clause 8. This distinction is introduced in RSTP in order to make it possible to describe the possibility of rapidly transitioning an Alternate Port to Forwarding on failure of the Root Port.

If the Port's MAC_Operational parameter (see 6.4.2 in IEEE Std 802.1t-2001) is FALSE, either as a consequence of management action or because the Port is not operable, the Bridge Port is assigned a Port Role of Disabled Port. If the Port's MAC_Operational parameter becomes TRUE, the Bridge Port is initially assigned a Port Role of Designated Port, as it is not aware of any other Bridge attached to the same LAN.

17.4.2 Spanning tree priority vectors

17.4.2.1 Definition

RSTP Bridges send information to each other, in Configuration Messages (see 17.7), to assign Port roles. The information sent for this purpose is known as a *spanning tree priority vector*. Spanning tree priority vectors provide the basis for a concise specification of RSTP's computation of the active topology, both in distributed terms across an entire Bridged LAN and in terms of the operation of each individual Bridge in support of the overall distributed algorithm. Each spanning tree priority vector comprises the following components:

a) Bridge Identifier (17.4.1) of the originating Root Bridge

b) Root Path Cost (17.4.1) for the transmitting Bridge

c) Bridge Identifier of the transmitting Bridge

d) Port Identifier (17.4.1) of the Port through which the message was transmitted

e) Port Identifier of the Port through which the message was received (where this information is available and relevant)

The first two components of a spanning tree priority vector are significant throughout the Bridged LAN, being propagated and updated along each path in the active topology. The next two components are locally significant, assigned hop by hop for each LAN or Bridge for use as tie-breakers in decisions between spanning tree priority vectors that are equal in their first two components. The fifth component is never conveyed in Configuration Messages, but is assigned hop by hop for each LAN or Bridge for use as a tie-breaker in local decisions between spanning tree priority vectors.

The set of all spanning tree priority vectors is totally ordered. Decisions about a given Port's role are made by comparing spanning tree priority vectors. For all components, a lesser numerical value is better, and earlier components in the above list 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. This process of receiving information and adding to it and passing it on can be described in terms of the message priority vector received in a configuration message and a set of priority vectors used to facilitate the computation of priority vector information to be held for each Port, or to be transmitted in further Configuration Messages to other Bridges further from the Root.

17.4.2.2 Spanning tree priority vector type definitions

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

port priority vector = *{RootBridgeID : RootPathCost : DesignatedBridgeID : DesignatedPortID : BridgePortID}*

The *message priority vector* is the spanning tree priority vector conveyed in a received Configuration Message. For a Bridge *B* receiving a Configuration Message on Port P_B from a Designated Port P_D on Bridge *D* claiming a Root identifier of R_D and a Root Path Cost of RPC_D:

message priority vector = $\{R_D : RPC_D : D : P_D : P_B\}$

This 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 ID* and *Designated Port ID* 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 the following is true:

$$((R_D < RootBridgeID)) \,||$$
$$((R_D == RootBridgeID) \,\&\&\, (RPC_D < RootPathCost)) \,||$$
$$((R_D == RootBridgeID) \,\&\&\, (RPC_D == RootPathCost) \,\&\&\, (D < DesignatedBridgeID)) \,||$$
$$((R_D == RootBridgeID) \,\&\&\, ((RPC_D == RootPathCost)$$
$$\&\&\, (D == DesignatedBridgeID) \,\&\&\, ((P_D < DesignatedPortID)) \,||$$
$$((D == DesignatedBridgeID) \,\&\&\, (P_D == DesignatedPortID))$$

If the message priority vector received in a valid BPDU is superior it will replace the current port priority vector.

A *root path priority vector* for the Port can be calculated from a received port priority vector, by adding the receiving Port's path cost PPC_{PB} to the *Root Path Cost* component, and including the receiving Port's Port ID as the final component.

$$\textit{root path priority vector} = \{R_D : RPC_D + PPC_{PB} : D : P_D : P_B\}$$

The *bridge priority vector* for a Bridge *B* is the priority vector that would, with the *Designated Port ID* set equal to the transmitting *Port ID*, 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.

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

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 DesignatedBridgeID *D* is not equal to *B*. In the case that the bridge priority vector is the best of this set of priority vectors, Bridge *B* has been selected as the Root. Assuming the best priority root path priority vector of this set to be that of port P_B above, then:

$$\begin{aligned} \textit{root priority vector} &= \{B : 0 : B : 0 : 0\} && \text{if } B \text{ is better than } R_D\text{, or} \\ &= \{R_D : RPC_D + PPC_{PB} : D : P_D : P_B\} && \text{if } B \text{ is worse than } R_D \end{aligned}$$

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 *BridgePortID* components.

$$\begin{aligned} \textit{designated priority vector} &= \{B : 0 : B : Q_B : Q_B\} && \text{if } B \text{ is better than } R_D\text{, or} \\ &= \{R_D : RPC_D + PPC_{PB} : B : Q_B : Q_B\} && \text{if } B \text{ is worse than } R_D \end{aligned}$$

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 port priority vector will be updated. The message priority vector in RST BPDUs transmitted by a Port always comprises the first four components of the port priority vector of the Port, even if the Port is a Root Port.

17.5 Port States

The part that a Bridge Port plays, or may play, in the active topology of the Bridged LAN is summarized in the Port State (7.4). This reflects the operational state of the MAC entity associated with the Bridge Port, administrative control over the Bridge Port's participation in the active topology, the selection of the active topology, and the controlled transition of the Port to participating in this topology, avoiding temporary loops.

The operational state of the MAC entity is represented by the MAC Operational parameter (6.4.2 of IEEE Std 802.1t-2001).

The administrative control over the Bridge Port's participation is represented by the *Administrative Bridge Port state*. This control can be set to *Enabled* or *Disabled* by management (14.8.2.2).

Selection of the active topology is accomplished by the assignment of the Port role by RSTP. A Root or Designated Port Role includes the Port in the active topology. An Alternate or Backup Port Role excludes the Port from the active topology. If the administrative Bridge Port state is disabled, the Port will be excluded from the active topology and assigned the Disabled Port Role.

The *Port State* (7.4) of the Bridge Port controls the operation of the Forwarding and Learning processes (7.7 and 7.8), and, in RSTP, takes the values *Discarding, Learning,* or *Forwarding*. State machines associated with each Port Role assign one of the Port States, Discarding, Learning, or Forwarding, to each Bridge Port.

Table 17-1 shows the relationship between the Port State values recognized by STP (*Disabled, Blocking, Listening, Learning,* and *Forwarding*) and the RSTP Port State values.

Table 17-1—Relationship between Port State values in STP and RSTP

STP Port State	Administrative Bridge Port State	MAC Operational	RSTP Port State	Active Topology (Port Role)
DISABLED	Disabled	FALSE	Discarding	Excluded (Disabled)
DISABLED	Enabled	FALSE	Discarding	Excluded (Disabled)
BLOCKING	Enabled	TRUE	Discarding	Excluded (Alternate, Backup)
LISTENING	Enabled	TRUE	Discarding	Included (Root, Designated)
LEARNING	Enabled	TRUE	Learning	Included (Root, Designated)
FORWARDING	Enabled	TRUE	Forwarding	Included (Root, Designated)

NOTE—The condition represented in Table 17-1 by the Administrative Port State of Enabled with MAC Operational FALSE, and the consequent STP Port State of DISABLED is represented in the current IETF Bridge MIB (IETF RFC 1493) by the BROKEN state.

The Bridge Protocol Entity shall include a Port whose Administrative Bridge Port State is Enabled, and whose MAC is operational, in its computation of the active topology. BPDUs received on that Port shall be processed as required by the Rapid Spanning Tree Algorithm and Protocol. A Port whose Administrative Bridge Port State is Disabled, or whose MAC is not operational, is assigned a Port Role of Disabled Port. A Port that is assigned the Disabled Port role is not operable, is not under control of the Spanning Tree Algorithm, and does not receive or transmit BPDUs.

The active topology of the Bridged Local Area Network is formed by the interconnection of LANs and Bridges through Ports that are assigned the Forwarding Port State. Frames are forwarded in both directions through those Ports.

In a Bridged Local Area Network whose Spanning Tree information has been completely distributed and is stable, i.e., consistent Port Roles have been assigned throughout the network, the Rapid Spanning Tree Protocol’s state machines ensure that every Root Port and Designated Port transitions to the Forwarding Port State. Every Alternate Port and Backup Port is always in the Discarding Port State. The connectivity through any Bridge is thus between its Root Port and its Designated Ports, and since every LAN has one and only one assigned Designated Port and every Bridge with the exception of the Root Bridge has a Root Port connected to a LAN, this connectivity will connect all LANs (i.e., the connectivity is “spanning”) and will be loop free (i.e., the active topology is a “tree” structure).

When the configuration information being propagated changes, and assigned Port Roles are not known to be consistent throughout the network, the Rapid Spanning Tree Protocol’s state machines can select the Discarding Port State or the Learning Port State for Root Ports or Designated Ports, preventing temporary data loops until it is known that these Ports can participate in the active topology. In the Discarding Port

State, station location information is not added to the Filtering Database, since further changes in Port Role can result in the information acquired being incorrect when the active topology becomes stable. In the Learning Port State, information is added to the Filtering Database in the expectation that the Port Role selection will be confirmed. This allows station location information to be acquired prior to frame relay in order to reduce the number of frames unnecessarily flooded.

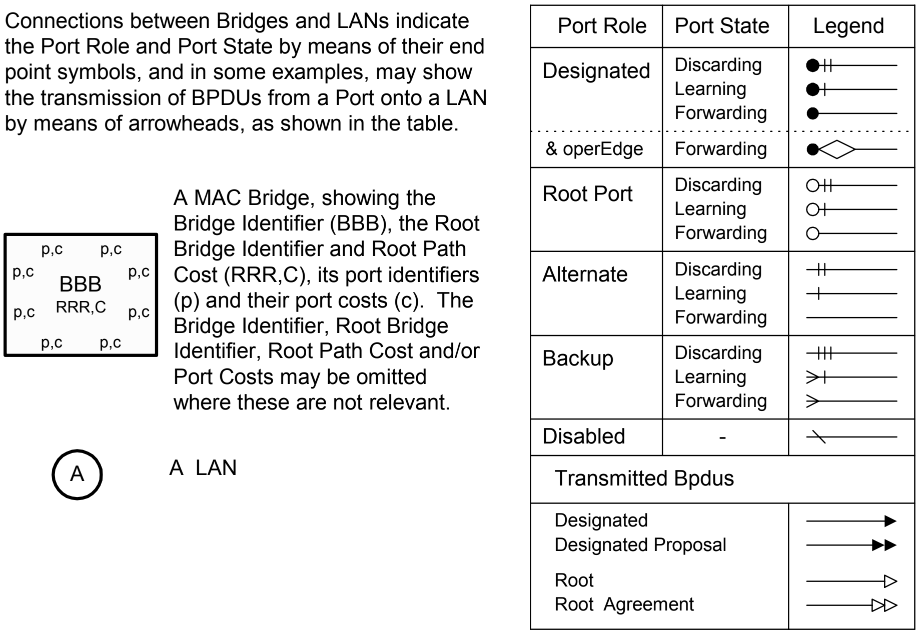
When a Port is in the Forwarding Port State, the Forwarding Process can forward frames received from that Port and can submit forwarded frames for transmission to that Port. When a Port is in the Discarding Port State or the Learning Port State, the Forwarding Process shall discard received frames and shall not submit forwarded frames for transmission.

When a Port is in the Forwarding Port State or the Learning Port State, the Learning Process shall incorporate station location information to the Filtering Database.

When a Port is in the Discarding Port State, the Learning Process shall not add station location information to the Filtering Database.

17.6 Topology examples

The examples shown in this subclause make use of the diagrammatic conventions shown in Figure 17-1.



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 table.

A MAC Bridge, showing the Bridge Identifier (BBB), the Root Bridge Identifier and Root Path Cost (RRR,C), its port identifiers (p) and their port costs (c). The Bridge Identifier, Root Bridge Identifier, Root Path Cost and/or Port Costs may be omitted where these are not relevant.

A LAN

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

Figure 17-1—Diagrammatic conventions

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 17-2 shows an example of the physical topology of a Bridged LAN. The example shows a simple, redundantly connected, structured wiring configuration, interconnecting the Bridges via point-to-point links that form point-to-point LANs A through N. For clarity, only Bridges and LANs are shown, with the unused Bridge Ports (Ports 3 and 4 of Bridges 555 through 888) available for connecting further devices to the network.

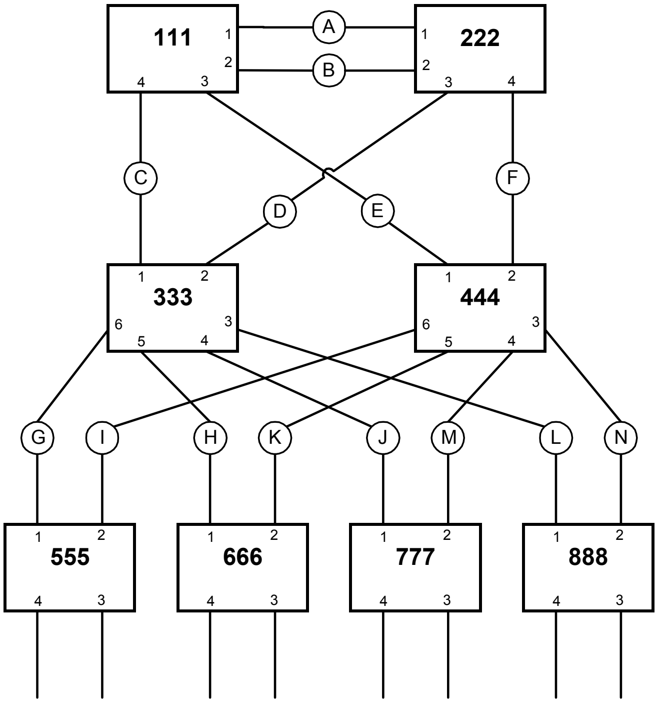


Figure 17-2—Physical topology example

Figure 17-3 shows the active topology, i.e. the logical connectivity, of the same Bridged LAN after enabling the Ports and configuration by the establishment of a Spanning Tree. Bridge 111 has been selected as the Root (though one cannot tell simply by looking at the active topology which Bridge is the Root).

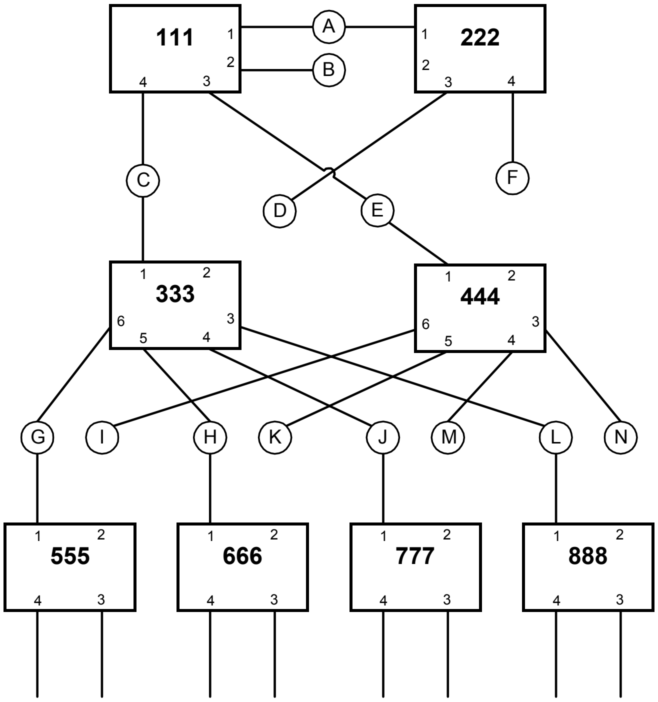


Figure 17-3—Active topology example

Figure 17-4 shows the Port Roles and Port States associated with each Bridge Port in this configuration of the Bridged LAN. From this figure, it can be seen that Bridge 111 is the Root, as its Ports are all Designated Ports, whereas the remaining Bridges have one Root Port.

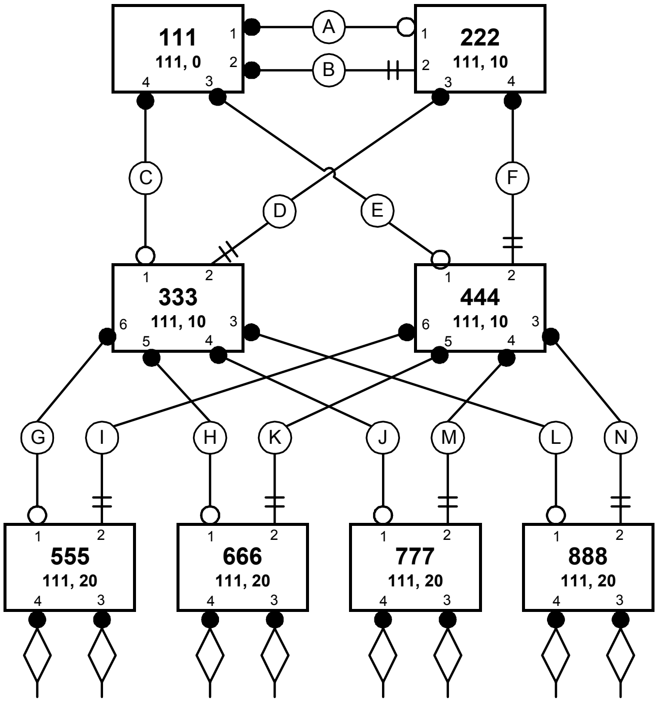


Figure 17-4—Port Roles and Port States

Figure 17-5 shows the result of connecting two of the Ports of Bridge 888 to the same LAN, for example, a coaxial 802.3 segment. As Port 4 of Bridge 888 has worse priority than Port 3 and both offer the same Root Path Cost, Port 4 will be assigned the Backup Port Role and will therefore be in the Discarding Port State. Should Port 3 or its connection to LAN O fail or become disabled, Port 4 will be assigned the Designated Port Role and will proceed to transition to the Forwarding Port State.

Figure 17-6 shows a “ring” topology constructed from point-to-point links, as can be employed in some resilient backbone configurations. Bridge 111 is the Root, as in previous examples.

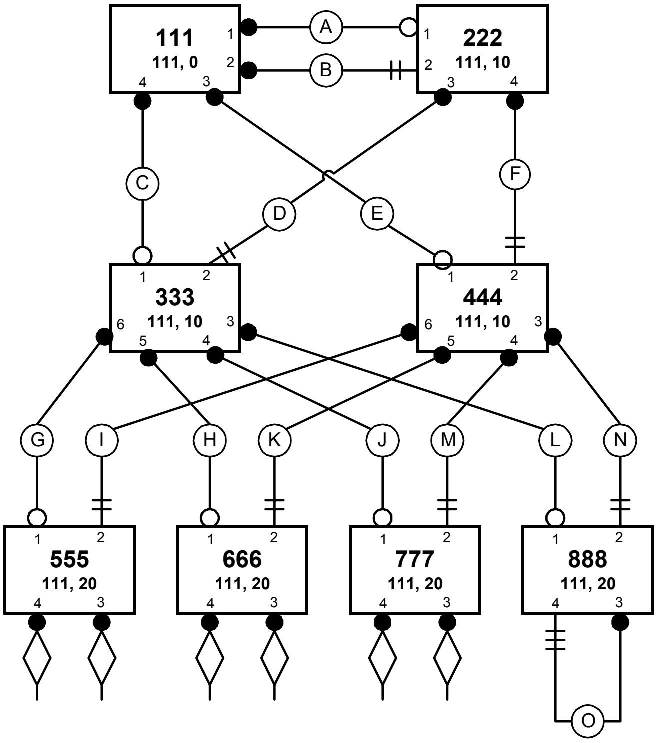


Figure 17-5—Backup Port example

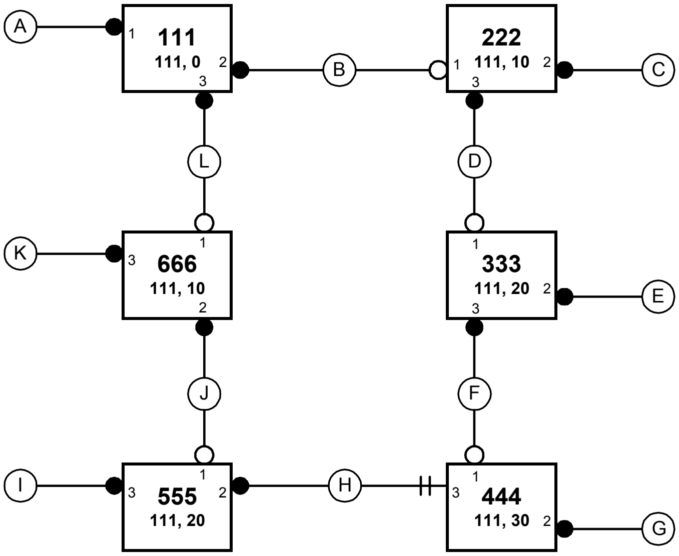


Figure 17-6—“Ring Backbone” example

17.7 Communicating Spanning Tree Information

Bridges transmit and receive MAC frames, each containing a Bridge Protocol Data Unit (BPDU—see Clause 9), 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 may be used by a Bridge in calculating its own BPDU to transmit, and may stimulate that transmission.

BPDUs are used to convey two types of Spanning Tree messages as follows:

a) Configuration Messages

b) Topology Change Notification (TCN) Messages

A Configuration Message can be encoded and transmitted as a Configuration BPDU (9.3.1), or as an RST BPDU (9.3.3). A TCN Message can be encoded as a TCN BPDU (9.3.2), or as an RST BPDU (9.3.3) with the TC flag set. The choice of BPDU format used to encode Spanning Tree messages is determined by the state of the Port Protocol Migration state machine (17.26).

Each Configuration Message contains, among other parameters, a message priority vector (see 17.4.2.2) consisting of the unique identifier of the Bridge that the transmitting Bridge believes to be the Root, the cost of the path to the Root from the transmitting Bridge, the identifier of the transmitting Bridge, and the identifier of the transmitting Port. This information is sufficient to allow a receiving Bridge to determine whether the transmitting Port has a better claim to be the Designated Port on the LAN on which the Configuration Message was received than any other Port currently believed to be the Designated Port, and to determine whether the receiving Port should become the Root Port for the Bridge if it is not already.

Configuration Messages are transmitted if the information to be transmitted by a Designated Port changes. 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 17.28.2).

Each Bridge Port receives information from the Designated Bridge on the LAN it is connected to, recording the message priority vector in the port priority vector (17.4.2.2), and replacing information recorded from any previous Designated Bridge if the received vector is better. The Port itself updates the port priority vector and becomes the Designated Port for the LAN initially when no information has yet been received from any other Bridge, and at any time that its designated priority vector is better than the port priority vector recorded from received BPDUs.

17.8 Changing Spanning Tree Information

Adding a new Bridge or LAN connectivity to the Bridged Area Network can result in the propagation of superior Spanning Tree information, changing Port Roles in all or part of the network. Information is considered to be superior if its message priority vector is better than the receiving Port's port priority vector, or if the Spanning Tree information has been propagated by the same Designated Bridge and Designated Port as are recorded in the Port's port priority vector, and the message priority vector or the timer information differ from those recorded for the Port. The new information will be propagated rapidly from Bridge to Bridge, superseding prior information and stimulating new Configuration Message transmissions until the leaves of the Spanning Tree defined by the new configuration are reached. The immediate stimulation and transmission of information will cease as these new Configuration Messages reach Designated Ports that have already received the new information through redundant paths in the network, or reach LANs that are not redundantly connected. Configuration Message transmissions will then once more occur at regular intervals from Ports selected as Designated Ports.

Removal or failure of Bridged Local Area Network components, or management of parameters determining the topology, can result in the propagation of worse information, each Bridge accepting new information from the prior Designated Bridge for the LAN connected to its Root Port. The MAC_Operational parameter (see 6.4.2 of IEEE Std 802.1t-2001) associated with each Bridge Port can signal failure conditions in some MACs; however, not all component failure conditions can be signalled in this way. To ensure that old information does not endlessly circulate through redundant paths in the network, preventing the effective propagation of the new information, a message age and a maximum age are associated with the Configuration Message information originated by the Root Bridge. On reception of a Configuration Message, the message age is increased by not less than a specified fraction of the maximum age. Received information is discarded and the Port made a Designated Port if the message age exceeds the maximum age. Thus the number of Bridges the information can traverse before being discarded is limited, and the loss of a network component will be detected by aging out of Spanning Tree information, if it was not detected by means of the MAC_Operational parameter.

If a Bridge's Root Port's MAC_Operational parameter becomes FALSE, the Port becomes a Disabled Port and received Spanning Tree information is immediately discarded. An Alternate Port, if one exists, will be selected as the new Root Port, or the Bridge itself will become the new Root Bridge. If the Bridge has Designated Ports, changed Spanning Tree information will be transmitted and propagated. This enables rapid reassignment of Port Roles in Bridges dependent for connectivity on the physical topology represented by the prior information.

17.9 Changing Port States

The Port State of each of a Bridge's Ports is controlled by a state machine, whose goal is to maximize connectivity, without introducing temporary data loops in the network. It attempts to transition Root Ports and Designated Ports 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. To transition a Port to the Forwarding Port State, this transition needs to be consistent with the Port Roles assigned to other Ports in the region of the network including this Port and bounded by Ports that are not in the Forwarding Port State or by LANs that are attached to only one Bridge Port.

A Bridge knows that the transition to the Forwarding Port State can be made if:

a) The Port Role has been Root Port or Designated Port for long enough for Spanning Tree information supporting this role assignment to have reached all Bridges in the network, and for contradictory information to be received from any Bridge following the change in Spanning Tree information that first caused this Port to be assigned the Root Port or Designated Port role, or

b) The Port is now a Root Port and any Ports on the Bridge that have been Root Port so recently that Spanning Tree information might not have reached all Bridges in the network, or have been contradicted if necessary, are not and will not be put in the Forwarding Port State until that time has elapsed (with the exception of c) below), or

c) The Port is a Designated Port and attaches to a LAN that has at most one other Bridge attached, and that Bridge's Port Role assignments are consistent with this Bridge and their Port States are known not to be Forwarding if they attach to LANs that connect to Bridges whose Port Roles are not consistent with that Bridge, or

d) The Port is a Designated Port and attaches to a LAN that is known to have no other Bridge Ports attached.

Condition a) above makes use of Forwarding Delay as the basis for establishing that enough time has elapsed to allow the transition to Forwarding be made.

NOTE—In STP (Clause 8), condition a) is the only condition used to determine whether a Port can transition to Forwarding. Conditions b) through d) apply only to RSTP.

Conditions b) and c) can be illustrated with reference to Figure 17-7. A topology change occurs as a result of management changing the Port priorities in Bridge 222; as a result, Bridge 222's old Alternate Port becomes its new Root Port, and its old Root Port becomes an Alternate Port. Assuming that the initial configuration shown in the figure had been stable for a significant time, and that Bridge B's old Root Port's Port State has been made Discarding, Bridge 222 can immediately make its new Root Port's Port State Forwarding, by applying condition b).

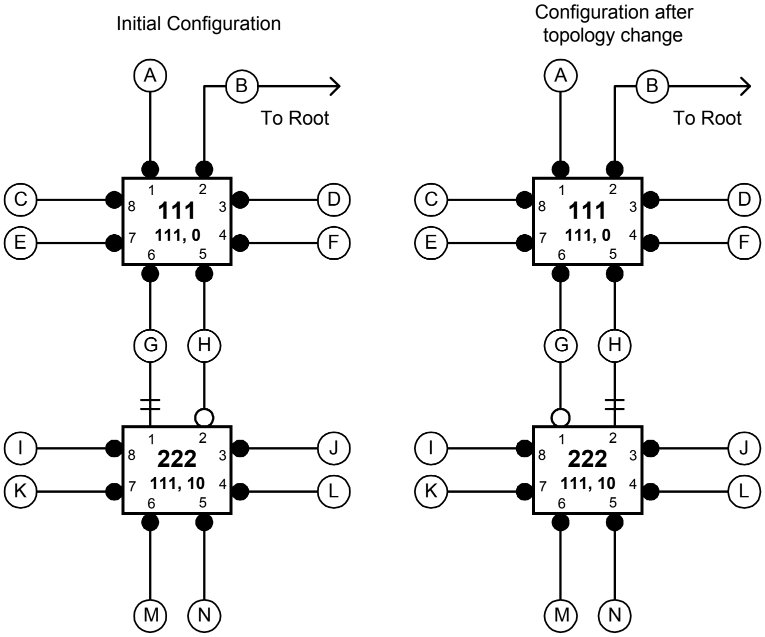


Figure 17-7—Root Port transition example

If the initial configuration had been stable for some time, Bridge 111's Designated Port attached to LAN G will be Forwarding, and therefore, no change is required in the state of Bridge 111's Designated Ports. However, if this Port is not Forwarding for some other reason (for example, a very recent transition from administratively disabled to enabled), Bridge 111 will take steps to make the Port State of this Designated Port Forwarding. In order to do so, it needs to know that Bridge B's port assignments, and the port assignments of all Bridges actively connected to Bridge 222, are consistent with Bridge 111 making the Port State of that Designated Port Forwarding, as stated in condition c). This is achieved by means of an explicit handshake between Bridges 111 and 222 on LAN G; if Bridge 111 receives a positive acknowledgement to the handshake, then the Port State of this Designated Port can be made Forwarding.

Condition d) applies to Ports that are known to be at the edge of the Bridged LAN; i.e., the value of the Port's operEdgePort parameter (17.18) is TRUE.

The handshake between Bridges that is involved in rapidly transitioning a designated Port to Forwarding involves the use by the Port Role Transitions state machine of the following state machine variables:

e) **proposing (17.18.20).** This variable is asserted by a Designated Port that is currently not Forwarding, and is conveyed to the Root Port of the neighboring Bridge in the Proposal flag of an RST BPDU (see 9.3.3).

f) **proposed (17.18.19).** This variable is asserted when an RST BPDU with a Designated Port role is received on a point to point link, and the Proposal flag in the BPDU is asserted. The proposed variable indicates to the Root Port that the Designated Port attached to the same LAN wishes to progress rapidly to Forwarding.

g) **sync (17.18.34).** When proposed is asserted, the Root Port will in turn assert sync for all other Ports of the Bridge; this has the effect of requesting those of the Bridge's Designated Ports that are not edge Ports to revert to Discarding.

h) **synced (17.18.35).** Once the Designated Port has reverted to Discarding, it asserts its synced variable. Alternate and Backup Ports, and Designated Ports that are edge Ports, assert their synced variable immediately. The Root Port monitors the synced variables of all of the other Ports in the Bridge; once all of the other Ports have asserted synced, the Root Port asserts synced and transmits an RST BPDU back to the designated Port, with the Agreement flag in the BPDU asserted.

i) **agreed (17.18.1).** This variable is asserted when an RST BPDU is received, and the BPDU carries an Agreement flag and a Port Role of Root Port. When agreed is asserted, the Designated Port knows that its neighbouring Bridge has confirmed that it can proceed to the Forwarding state without further delay.

The Designated Port(s) of the downstream Bridge can, in turn, request permission of their neighboring Bridge(s) to rapidly transition to Forwarding. The effect of this handshake is that a “cut” in the active topology is propagated from the original Designated Port through all Bridges on the subtree below it (in the direction away from the Root), until the cut reaches the edge of the Bridged LAN.

17.10 Updating Learned Station Location Information

In normal stable operation, learned station location information held in the Filtering Database need only change as a consequence of the physical relocation of stations. It may, therefore, be desirable to employ a long aging time for Dynamic Filtering Entries in the Filtering Database (7.9.2), especially as many end stations transmit frames following power-up after relocation, which would cause station location information to be relearned.

However, when the active topology of a Bridged LAN reconfigures, end stations can appear to move from the point of view of a Bridge in the network. This is true even if the states of the Ports on that Bridge have not changed. It is necessary for station location to be relearned following a change in the active topology, even if only part of the Bridged LAN has reconfigured.

For all Ports other than edge Ports (i.e., for all Ports whose operEdgePort parameter is FALSE; see 17.18), if a Root Port or a Designated Port becomes an Alternate, Backup, or Disabled Port, then stations are no longer reachable through that Port. Therefore, Dynamic Filtering Entries for that Port are removed from the Filtering Database. Conversely, if an Alternate Port becomes a Root Port or a Designated Port, stations that were formerly reachable through other ports on that Bridge might be reachable through that port. Therefore, Dynamic Filtering Entries for other Ports are removed from the Filtering Database.

In addition, for all Ports other than edge Ports, the Rapid Spanning Tree Protocol state machines ensure the transmission of TCN Messages to convey the above information. The Rapid Spanning Tree Protocol state machines ensure the transmission of TCN Messages to notify other Bridges that stations previously accessible through other Bridge Ports might now be accessible through the transmitting Bridge Port. Accordingly, Bridges receiving a TCN Message on one Bridge Port remove Dynamic Filtering Entries for their other Ports from their Filtering Databases.

A change of state of an edge Port (i.e., a Port whose operEdgePort parameter is TRUE; see 17.18) does not result in topology change notifications to be sent to other Bridges, as a change of state of an edge Port does not affect the connectivity or station location information for the rest of the Bridged LAN. Similarly, a Bridge that has one or more edge Ports does not remove Dynamic Filtering Entries from its Filtering Database for edge Ports as a result of a change of state of an edge Port, or as a result of receiving TCN Messages from another Bridge.

NOTE—The flushing rules as described define the minimum set of Ports that are required to be flushed in order to ensure that learned information that is no longer valid is removed from the Filtering Databases of the set of Bridges in the Bridged LAN. However, it is valid for a Bridge to flush more Ports than are strictly necessary, if this proves to be desirable for implementation reasons; for example, a Bridge might choose to flush all learned addresses from its Filtering Database rather than to selectively flush only addresses learned on the specified set of Ports. This does not result in incorrect operation of the Bridge, as it is simply returning the Filtering Database to its initial state before any station location information was learned; however, it can result in more flooding of frames with unknown destination addresses than is strictly necessary for correct operation.

A further, optional, optimization to the flushing algorithm can be achieved when a topology change involves a rapid Root Port transition; i.e., a topology change where the existing Root Port is made Discarding, and an existing Alternate Port can immediately be made Forwarding. Changing the state of the retiring Root Port to Discarding has the (temporary) effect of partitioning the Spanning Tree into two trees as follows:

a) A main tree that contains the Root and all Bridges and LANs hitherto reachable through the retiring Root Port

b) A subtree that contains the Bridge that contains the retiring Root Port and all other LANs and Bridges that are downstream of that Root Port (i.e., in the direction of the edge of the Bridged LAN)

In Figure 17-8, placing Port 1 of Bridge 888 into Discarding would (temporarily) partition the Bridged LAN into a subtree consisting of Bridge 888 plus any LANs and systems attached to its Ports 3 and 4, and a main tree comprising all other Bridges and their attached LANs and systems. All addresses that had been learned on the retiring Root Port (Port 1 of Bridge 888) are, by definition, addresses that reside somewhere on the main tree (and conversely, none of these addresses can reside on the subtree). Therefore, when the Alternate Port (Port 2 of Bridge 888) becomes the new Root Port and transitions to Forwarding, it provides the new path from the subtree to the Root, and hence, by definition, the path whereby all addresses on the main tree can be reached. All MAC addresses in the Filtering Database that had been learned on the retiring Root Port can therefore be moved to the new Root Port; i.e., the Filtering Database is modified as if that set of MAC addresses had been learned on the new Root Port. The use of this optimization can reduce the need to flood frames to Ports on the subtree in cases where the frames are destined for addresses known to be located on the main tree.

Figure 17-9 and Figure 17-10 illustrate the need to flush addresses following a topology change, and the sequence of events involved. .

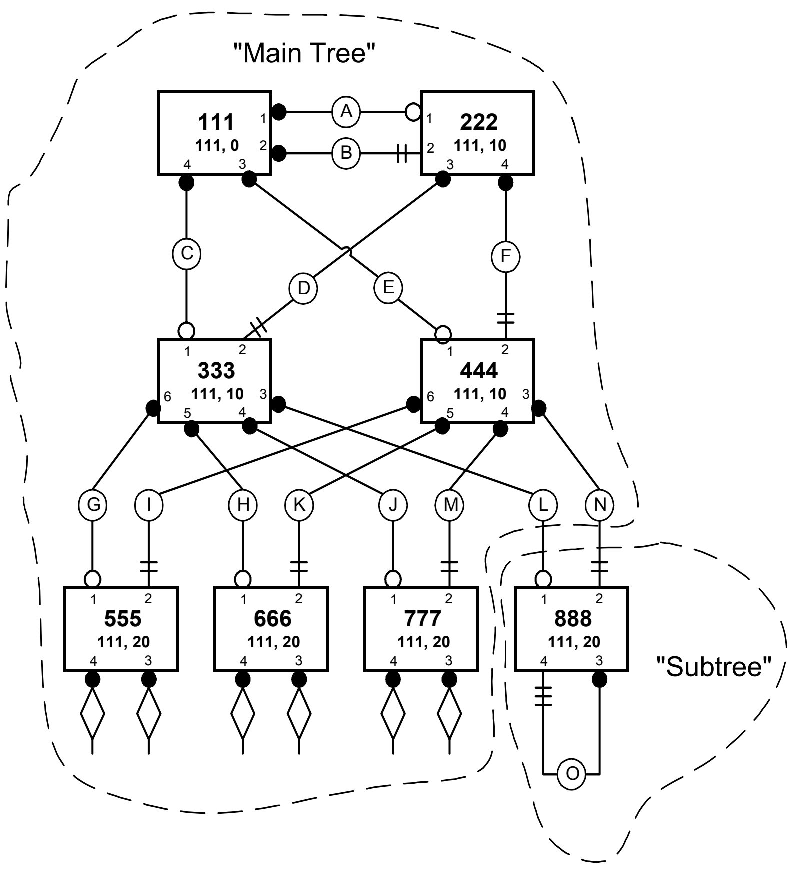
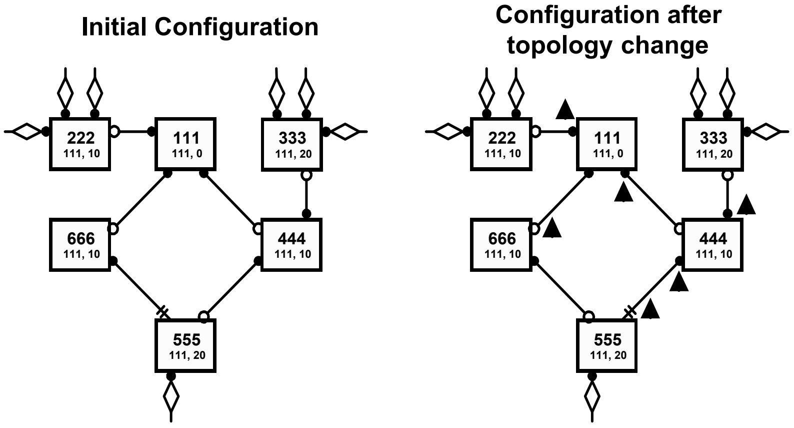


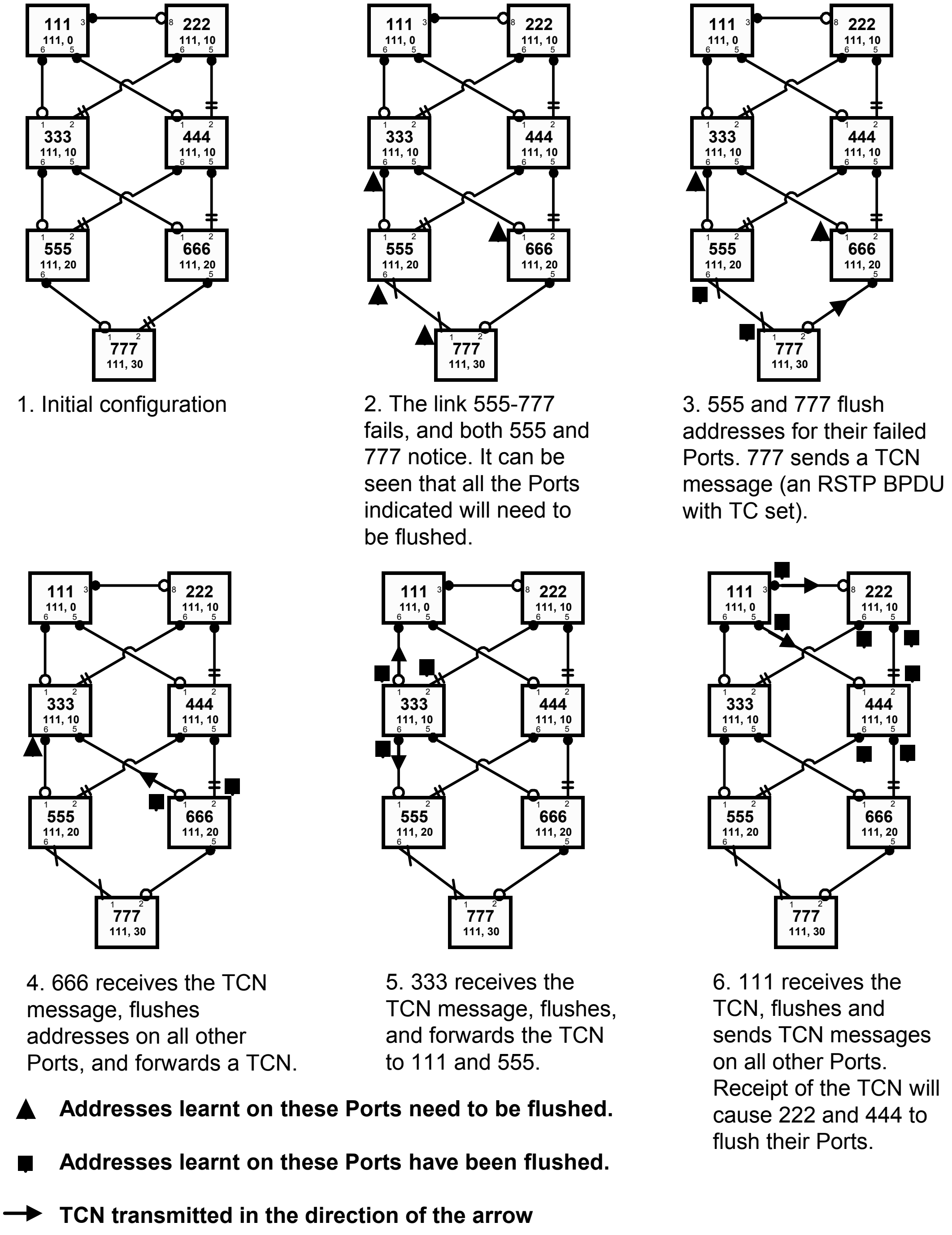
Figure 17-8—Root Port transition – tree partitioning


Initial Configuration

Configuration after topology change

▲ **Addresses learnt on these Ports need to be flushed.**
NOTE--Designated Ports where operEdge ==TRUE
do not need to be flushed.

Figure 17-9—Address flushing example



1. Initial configuration

2. The link 555-777 fails, and both 555 and 777 notice. It can be seen that all the Ports indicated will need to be flushed.

3. 555 and 777 flush addresses for their failed Ports. 777 sends a TCN message (an RSTP BPDU with TC set).

4. 666 receives the TCN message, flushes addresses on all other Ports, and forwards a TCN.

5. 333 receives the TCN message, flushes, and forwards the TCN to 111 and 555.

6. 111 receives the TCN, flushes and sends TCN messages on all other Ports. Receipt of the TCN will cause 222 and 444 to flush their Ports.

▲ **Addresses learnt on these Ports need to be flushed.**

■ **Addresses learnt on these Ports have been flushed.**

→ **TCN transmitted in the direction of the arrow**

Figure 17-10—Address flushing – worked example

In Figure 17-9, Bridge 555 transitions one of its Ports from the Alternate Port Role to the Root Port Role, causing it to transition from Discarding to Forwarding. The former Root Port becomes an Alternate Port, and consequently becomes Discarding. Any addresses that had been learnt on the former Root Port have to be flushed. Any of the Ports of the other Bridges that could have learnt addresses from frames forwarded by Bridge 555 through its former Root Port have to flush their learnt addresses. For a given Bridge receiving a TCN Message, the only Ports that it can be certain it does not need to flush are as follows:

a) Any Ports whose operEdgePort parameter is TRUE

b) The Port through which the TCN Message was received

Hence, any Dynamic Filtering Entries in the Bridge's Filtering Database that contain information learned on all Ports for which these two conditions do not hold are removed. As the Designated Ports of Bridges 222 and 333 in Figure 17-9 are all edge Ports, there is no need to remove any filtering database entries from either of these Bridges.

In Figure 17-10, a structured wiring configuration is used to show the sequence of events following the loss of a link towards the periphery of the network. This example assumes that all Bridges are RSTP Bridges, and therefore all TCN messages are transmitted as RST BPDUs with the TC flag set (see 17.7)

17.11 RSTP and point-to-point links

Some of the rapid state transitions that are possible within RSTP are dependent upon whether the Port concerned can only be connected to exactly one other Bridge (i.e., it is served by a point-to-point LAN segment), or can be connected to two or more Bridges (i.e., it is served by a shared medium LAN segment). The adminPointToPointMAC and operPointToPointMAC parameters (6.4.3) allow the point-to-point status of the link to be manipulated administratively, and the operational state to be signalled to the RSTP state machines.

Rapid transition from an Alternate Port Role to a Root Port Role is unaffected by the value of operPointToPointMAC for the Port; however, rapid transition of a Designated Port to Forwarding is only possible if operPointToPointMAC is TRUE (i.e., the LAN segment associated with the Port is point-to-point), unless the port has been defined to be an edge Port (i.e., operEdgePort is TRUE). Hence, for Designated Ports, if operPointToPointMAC and operEdgePort are both FALSE, transitions from Discarding to Learning and from learning to Forwarding can only take place after a delay of Forward Delay.

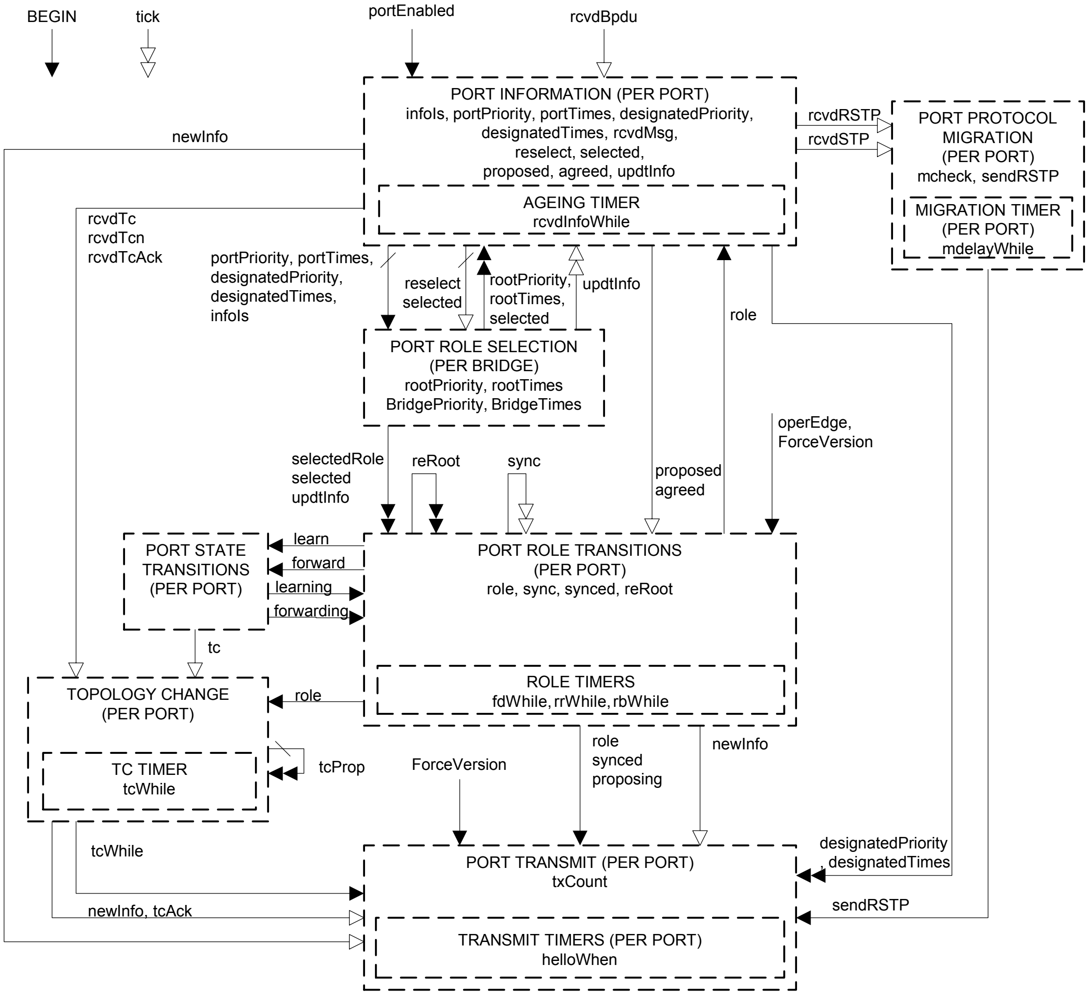
17.12 STP compatibility

Under some circumstances, it is possible for the rapid state transitions employed by RSTP to result in an increase in the rates of frame duplication and misordering in the Bridged LAN, as discussed in F.2.4. In order to allow RSTP Bridges to support applications and protocols that may be sensitive to frame duplication and misordering, a Force Protocol Version parameter, controlled by management, allows RSTP to be operated with the rapid transitions disabled (see 17.16.1). The value of this parameter applies to all Ports of the Bridge.

17.13 Rapid Spanning Tree state machines

The operation of each Bridge Port is represented by a set of state machines. Figure 17-11 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 Rapid Spanning Tree state machines and the variables local to each machine.

With the exception of the Port Role Selection state machine, an instance of each Rapid Spanning Tree state machine shall be implemented per Bridge Port. A single Port Role Selection state machine shall be implemented per Bridge.



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
PST:	Port State Transitions Machine
TCM:	Topology Change Machine
PPM:	Port Protocol Migration Machine
PTX:	Port Transmit Machine
PTI:	Port TImers Machine

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 17-11—RSTP state machines - overview and interrelationships

In addition to the state machines and associated variables and procedures defined in this clause, the operation of RSTP makes use of the operEdgePort parameter (14.8.2) in order to allow rapid state transitions to occur at the edge of the Bridged LAN. Therefore, implementations that support RSTP shall also support the following:

a) The use of the adminEdgePort and operEdgePort parameters

b) Modification of the adminEdgePort parameter by management (14.8.2)

c) The operation of the Bridge Detection state machine, as defined in Clause 18 of IEEE Std 802.1t-2001

The process of BPDU transmission (represented by the Port Transmit state machine) and BPDU reception (represented by the Port Information state machine) interact with the other state machines by setting flag variables (in the case of reception) or resetting them (on transmission of an appropriate BPDU). The transmission process is responsible for its own rate limiting to enforce a minimum of one transmission and a maximum of three transmissions per hello time period.

NOTE—The operation of the Bridge as a whole can be represented by the interaction between Bridge Ports specified, and by parameters of the Bridge stored in “Port 0.” This removes the need for any “per Bridge” specification elements, and helps ensure the minimum dependencies between Bridge Ports. This in turn supports the development of implementations that scale well with increasing numbers of Bridge Ports. This shift of focus to “per Port operation” for the RSTP is supported by underlying technical changes from the Spanning Tree Algorithm and Protocol (Clause 8).

Transmission of BPDUs is prompted by the following:

d) Changes to the information that a Designated Port derives from the current Root Port

e) A port based Hello Timer

The Port Protocol Migration state machine determines whether the BPDUs transmitted will use the RST BPDU format (to communicate on LANs where only RSTP Bridges are present) or the Configuration BPDU and TCN BPDU formats (to communicate on LANs where one or more STP Bridges are present).

The generation and propagation of topology change notifications is performed by the Topology Change state machine. Topology Changes and Topology Change Notifications are propagated by setting a “tcWhile” timer on each port through which the change or notification is to be propagated. This in turn causes regular TCN Messages to be sent through a Root Port (for the duration of the tcWhile timer, or until a topology change acknowledgment is received).

Port timers are simple down counters, decremented on a per second “tick” until they reach zero. The Port Timers state machine provides this functionality for the set of timers defined in 17.15, and uses the tick variable defined in 17.18.

The Port State Transitions state machine moves the Root Port and Designated Ports to the Forwarding Port State, and Alternate and Backup Ports to Discarding.

The need to transition to a new Port Role is signalled to the Port Transition Machines for all Ports of the Bridge by a per-Bridge Port Role Selection state machine. Whenever new information is received on any Port, or the current information for the Port is aged out as a result of Message Age exceeding Max Age, the Port Role Selection state machine computes any role changes that may be required. As the selection computation may not complete before further new information is received, a signal is used between the Port Role Selection state machine and the Port Transition state machines to allow the signalling of new information to the Port Role Selection state machine, and to signal when the computation results produced by the Port Role Selection state machine are to be considered valid. This interlock allows the selection process to be restarted on receipt of new information without the Port Transition state machines acting prematurely on transitory role re-assignments.

17.14 Notational conventions used in State Diagrams

State diagrams are used to represent the operation of a function as a group of connected, mutually exclusive states. Only one state of a function can be active at any given time.

Each state is represented in the state diagram as a rectangular box, divided into two parts by a horizontal line. The upper part contains the state identifier, written in upper case letters. The lower part contains any procedures that are executed on entry to the state.

All permissible transitions between states are represented by arrows, the arrowhead denoting the direction of the possible transition. Labels attached to arrows denote the condition(s) that must be met in order for the transition to take place. A transition that is global in nature (i.e., a transition that occurs from any of the possible states if the condition attached to the arrow is met) is denoted by an open arrow; i.e., no specific state is identified as the origin of the transition.

On entry to a state, the procedures defined for the state (if any) are executed exactly once, in the order that they appear on the page. Each action is deemed to be atomic; i.e., execution of a procedure completes before the next sequential procedure starts to execute. No procedures execute outside of a state block. On completion of all of the procedures within a state, all exit conditions for the state (including all conditions associated with global transitions) are evaluated continuously until such a time as one of the conditions is met. All exit conditions are regarded as Boolean expressions that evaluate to TRUE or FALSE; if a condition evaluates to TRUE, then the condition is met. When the condition associated with a global transition is met, it supersedes all other exit conditions including UCT. The label UCT denotes an unconditional transition (i.e., UCT always evaluates to TRUE). The label ELSE denotes a transition that occurs if none of the other conditions for transitions from the state are met (i.e., ELSE evaluates to TRUE if all other possible exit conditions from the state evaluate to FALSE).

Where two or more exit conditions with the same level of precedence become TRUE simultaneously, the choice as to which exit condition causes the state transition to take place is arbitrary.

A variable that is set to a particular value in a state block retains this value until a subsequent state block executes a procedure that modifies the value.

Where it is necessary to segment a state machine description across more than one diagram, a transition between two states that appear on different diagrams is represented by an exit arrow drawn with dashed lines, plus a reference to the diagram that contains the destination state. Similarly, dashed arrows and a dashed state box are used on the destination diagram to show the transition to the destination state. In a state machine that has been segmented in this way, any global transitions that can cause entry to states defined in one of the diagrams are deemed to be potential exit conditions for all of the states of the state machine, regardless of which diagram the state boxes appear in.

Should a conflict exist between the interpretation of a state diagram and either the corresponding global transition tables or the textual description associated with the state machine, the state diagram takes precedence.

The interpretation of the special symbols and operators used in the state diagrams is as defined in Table 17-2; these symbols and operators are derived from the notation of the “C++” programming language, ISO/IEC 14882.

Table 17-2—State machine symbols

Symbol	Interpretation
( )	Used to force the precedence of operators in Boolean expressions and to delimit the argument(s) of actions within state boxes.
;	Used as a terminating delimiter for actions within state boxes. Where a state box contains multiple actions, the order of execution follows the normal English language conventions for reading text.
=	Assignment action. The value of the expression to the right of the operator is assigned to the variable to the left of the operator. Where this operator is used to define multiple assignments, e.g., a = b = X the action causes the value of the expression following the right-most assignment operator to be assigned to all of the variables that appear to the left of the right-most assignment operator.
!	Logical NOT operator.
&&	Logical AND operator.
||	Logical OR operator.
if...then...	Conditional action. If the Boolean expression following the if evaluates to TRUE, then the action following the then is executed.
!=	Inequality. Evaluates to TRUE if the expression to the left of the operator is not equal in value to the expression to the right.
==	Equality. Evaluates to TRUE if the expression to the left of the operator is equal in value to the expression to the right.
*	Arithmetic multiplication operator.
-	Arithmetic subtraction operator.

17.15 State machine timers

The timers defined for the operation of RSTP are defined per-Port.

17.15.1 fdWhile

The Forward Delay timer, with an initial value of Forward Delay (abbreviated to FwdDelay in the state diagrams).

17.15.2 helloWhen

This variable gives rise to periodic transmission of BPDUs. Where periodic transmissions are required, it ensures that at least one transmission occurs in each HelloTime period.

17.15.3 mdelayWhile

The “migration delay” timer. This timer enforces a minimum time for which RST BPDUs and Configuration BPDUs are sent in states of the Port Protocol Migration state machine. This allows time for another RSTP Bridge on the same LAN to synchronize its migration state with this Port before the receipt of a BPDU can cause this Port to change the type of BPDUs that it transmits. The timer is initialized to the value of the constant MigrateTime.

17.15.4 rbWhile

This “recent backup while” timer is nonzero if this port is, or has recently been, a Backup Port. The initial value for this timer is twice HelloTime. The timer is set to its initial value when the Port becomes a Backup Port, and this value is maintained while the Port continues to be a Backup Port.

17.15.5 rcvdInfoWhile

The time remaining before the information held for this Port expires; i.e., before Message Age equals or exceeds Max Age for received information on this Port.

17.15.6 rrWhile

This “recent root while” timer is nonzero if this Port is, or has recently been, a Root Port. The initial value for this timer is Forward Delay, as communicated by the Root Bridge. The timer is set to its initial value when the Port becomes a Root Port, and this value is maintained while the Port continues to be a Root Port. The timer is set to zero if the Port becomes Discarding.

17.15.7 tcWhile

The interval for which TCN Messages are sent through the Root Port and for which Configuration Messages are sent with the Topology Change flag set. The newTcWhile procedure (17.19.7) determines the starting value of this timer.

17.16 State machine performance parameters

These parameters are treated as constants by the state machines; their values can be modified only by management action.

17.16.1 ForceVersion

ForceVersion is equal to the value of the “Force Protocol Version” variable for the Bridge (see 17.12). In an RSTP Bridge, this variable can take the value 0 (indicating that the RSTP state machines operate in “STP Compatibility” mode) or 2 (the default value, indicating that the RSTP state machines operate normally).

In “STP Compatibility” mode, rapid transitions of Alternate Ports to Root Ports, and rapid transitions of Designated Ports to Forwarding, are disabled, and the RSTP algorithm transmits Configuration BPDUs and TCN BPDUs only. Any received RST BPDUs are discarded.

In normal RSTP operation (ForceVersion == 2), RST BPDUs are transmitted, unless a legacy system is detected by a Port and indicated by the operation of the Port Protocol Migration state machine, in which case version 0 Configuration BPDUs and TCN BPDUs are transmitted on that Port.

This value can be modified only by management.

17.16.2 FwdDelay

FwdDelay is equal to the Bridge Forward Delay component of BridgeTimes (17.17.4).

17.16.3 HelloTime

HelloTime is equal to the Bridge Hello Time component of BridgeTimes (17.17.4).

17.16.4 MigrateTime

MigrateTime is the constant that is used as the initial value of the timer mdelayWhile. The value of MigrateTime is 3 s.

17.16.5 PortPathCost

The contribution of the path through this Port, when the Port is the Root Port, to the total cost of the path to the Root for this Bridge.

This parameter is used, added to the value of the Designated Cost parameter for the Root Port, as the value of the Root Path Cost parameter offered in all Configuration Messages transmitted by the Bridge, when it is not the Root.

When Bridge Management is supported, this parameter may be updated by management action.

17.16.6 TxHoldCount

TxHoldCount is the value used by the Port Transmit state machine to limit the maximum transmission rate. The default value of this constant is as defined in Table 8-3.

17.17 Per-Bridge Variables

17.17.1 BEGIN

This variable is controlled by the system initialization process. A value of TRUE causes all 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.

17.17.2 BridgeIdentifier

BridgeIdentifier is the unique Bridge Identifier assigned to this Bridge.

The Priority component of the Bridge Identifier may be modified by management (see 9.2.5 and 14.8.1.2).

17.17.3 BridgePriority

BridgePriority is the value of the bridge priority vector, as defined in 17.4.2.2. The first (RootBridgeID) and third (DesignatedBridgeID) components are both equal to the value of the Bridge Identifier (17.17.2). The remaining elements of this variable are set to zero.

BridgePriority is used by updtRolesBridge() in determining the value of the rootPriority variable (see 17.19.21).

17.17.4 BridgeTimes

BridgeTimes has four components as follows:

a) The current values of Bridge Forward Delay (see Table 8-3), Bridge Hello Time (see Table 8-3), and Bridge Max Age (see Table 8-3). These parameter values are determined only by management.

b) A Message Age value of zero.

BridgeTimes is used by updtRolesBridge() in determining the value of the rootTimes variable (see 17.19.21).

17.17.5 rootPortId

This is the Port Identifier of the Root Port, and forms the fifth component of the root priority vector, as defined in 17.4.2.2.

17.17.6 rootPriority

The rootPriority variable comprises the first four components of the Bridge's root priority vector, as defined in 17.4.2.2.

17.17.7 rootTimes

The rootTimes variable comprises the Bridge's timer parameter values (Message Age, Max Age, Forward Delay, and Hello Time). The values of these timers are derived from the values stored in portTimes (17.18.18) for the Root Port. Max Age, Forward Delay, and Hello Time are set equal to the values held by the Root Port, and 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).

17.18 Per-Port variables

Per-Port variables are of the following types:

a) Priority vector and associated timer information concerned with the computation of the Spanning Tree

b) Variables derived from the parameters of incoming BPDUs

c) Externally generated signals that affect the operation of one or more state machines

d) Variables used to communicate between state machines

e) Variables used for miscellaneous housekeeping tasks within each state machine

17.18.1 agreed

This variable is set TRUE if the value of the ForceVersion parameter is greater than or equal to 2, and the value of the operPointToPointMAC parameter (6.4.3) associated with the Port is TRUE, and an RST BPDU has been received, and the Port Role field in the BPDU indicates the value Root Port, and the Agreement flag is set.

The variable is set FALSE by the Port Role Transitions and Port Information state machines.

17.18.2 designatedPriority

The designatedPriority variable comprises the first four components of the Port's designated priority vector value, as defined in 17.4.2.2. The fifth component of the designated priority vector value is portId (17.18.16).

17.18.3 designatedTimes

The designatedTimes variable comprises the set of timer parameter values (Message Age, Max Age, Forward Delay, and Hello Time) that are used to update Port Times when updtInfo is set. The value of designatedTimes is copied from the rootTimes Parameter (17.17.7) by the operation of the updtRolesBridge() procedure.

17.18.4 forward

This is the administrative state for the packet forwarding function for this port provided by the Bridge Relay Entity. It is set TRUE by the Port Role Transitions state machine to instruct the Port State Transitions state machine to enable packet forwarding. It is set FALSE by the Port Role Transitions state machine to instruct the Port State Transitions state machine to disable packet forwarding.

17.18.5 forwarding

This is the operational state for the packet forwarding function. It is set TRUE by the Port State Transitions state machine when packet forwarding is enabled. It is set FALSE by the Port State Transitions state machine when packet forwarding is disabled.

17.18.6 infoIs

This is a variable taking the values Mine, Aged, Received, or Disabled, to indicate the origin/state of the Port's Spanning Tree information (portInfo) held for the Port, as follows:

a) If infoIs is **Received**, the port has received current (not aged out) information from the Designated Bridge for the attached LAN (a point-to-point bridge link being a special case of a LAN).

b) If infoIs is **Mine**, information for the port has been derived from the Root Port for the Bridge (with the addition of root port cost information). This includes the possibility that the Root Port is "Port 0," i.e., the bridge is the Root Bridge for the Bridged Local Area Network.

c) If infoIs is **Aged**, information from the Root Bridge has been aged out. Just as for "reselect" (see 17.18.29), the state machine does not formally allow the "Aged" state to persist. However, if there is a delay in recomputing the new root port, correct processing of a received BPDU is specified.

d) Finally if the port is disabled, infoIs is **Disabled**.

17.18.7 initPm

This is a variable used by the Port Protocol Migration state machine to prevent repeated re-entry into the INIT state when the Port is disabled.

17.18.8 learn

This is the administrative state for the source address learning function for this port provided by the Bridge Relay Entity. It is set TRUE by the Port Role Transitions state machine to instruct the Port State Transitions state machine to enable source address learning. It is set FALSE by the Port Role Transitions state machine to instruct the Port State Transitions state machine to disable source address learning.

17.18.9 learning

This is the operational state for the source address learning function. It is set TRUE by the Port State Transitions state machine when source address learning is enabled. It is set FALSE by the Port State Transitions state machine when source address learning is disabled.

17.18.10 mcheck

This is a Boolean value that may be set to TRUE by the operation of management in order to force the Port Protocol Migration state machine into the SEND_RSTP state. Its value is set to FALSE on entry to the SEND_RSTP state. Forcing the state machine to send RST BPDUs in this manner can be used to test whether all legacy Bridges on a given LAN have been removed. The value of mcheck cannot be set TRUE if the value of ForceVersion (17.16.1) is less than 2; i.e., the mcheck facility is ineffective if the Bridge is set to operate in "STP Compatibility" mode.

17.18.11 msgPriority

The msgPriority variable comprises the first four components of the message priority vector value conveyed in a received BPDU, as defined in 17.4.2.2.

17.18.12 msgTimes

The msgTimes variable comprises the timer parameter values (Message Age, Max Age, Forward Delay, and Hello Time) conveyed in a received BPDU.

17.18.13 newInfo

This is a Boolean variable that is set TRUE if a BPDU is to be transmitted. Its value is set FALSE by the Port Transmit state machine.

17.18.14 operEdge

This reflects the value of the operEdgePort parameter, as defined by the operation of the Bridge Detection state machine (Clause 18 of IEEE Std 802.1t-2001). The value of this parameter is used by a Designated Port in order to determine how rapidly it may transition to the Forwarding Port State (see 17.23, Port Role Transitions state machine). The Bridge Detection state machine sets this parameter to the value of adminEdgePort (18.1.3 of IEEE Std 802.1t-2001) on initialization, and forces its value to FALSE if any BPDUs are received on the Port.

17.18.15 portEnabled

This variable reflects the operational state of the MAC service supporting the Bridge Port. Its value is TRUE if the MAC_Operational parameter (see 6.4.2 of IEEE Std 802.1t-2001) for the Port is TRUE, and is otherwise FALSE.

17.18.16 portId

This is the Port Identifier for this Port. This variable forms the fifth component of the port priority and designated priority vectors defined in 17.4.2.2.

17.18.17 portPriority

The portPriority variable comprises the first four components of the Port's port priority vector value, as defined in 17.4.2.2.

17.18.18 portTimes

The portTimes variable comprises the Port's timer parameter values (Message Age, Max Age, Forward Delay, and Hello Time). These timer values are used in BPDUs transmitted from the Port.

17.18.19 proposed

This is set TRUE when a Configuration Message indicating a Designated Port's desire to receive a confirmation of its role and permission to rapidly transition to forwarding has been received on a point-to-point link; i.e., the value of the operPointToPointMAC parameter (6.4.3) associated with the Port is TRUE and the message carries a Proposal flag. Set FALSE by the Port Role Transitions state machine.

17.18.20 proposing

This is set TRUE by the Port Role Transitions state machine if a Configuration Message indicating a Designated Port's desire to receive a confirmation of its role and permission to rapidly transition to forwarding is to be transmitted; i.e., set TRUE when the Port Role is Designated, the Port State is not Forwarding and the proposing flag is FALSE. The message is to be transmitted at the next transmission opportunity (i.e., transmit the message as soon as possible, subject to the restriction on transmission rate imposed by Transmission Limit; see 17.28.2). This variable is used by the Port Transmit state machine to set the value of the Proposal flag in transmitted RST BPDUs. This variable is set FALSE by the operation of the Port Information state machine.

17.18.21 rcvdBPDU

This is a variable that is set to the value TRUE when:

a) A Config BPDU or a TCN BPDU is received on this Port; or

b) The value of the ForceVersion variable is greater than or equal to 2, and an RST BPDU is received on this Port.

It is set to FALSE by the Port Information state machine (17.21).

17.18.22 rcvdMsg

This is set to the result of the rcvBPDU procedure. It can take the values SuperiorDesignatedMsg, RepeatedDesignatedMsg, ConfirmedRootMsg, or OtherMsg.

17.18.23 rcvdRSTP

This is a Boolean variable that is set to the value TRUE when an RST BPDU is received on the Port. Its value is set FALSE by the Port Protocol Migration state machine.

17.18.24 rcvdSTP

This is a variable that is set to the value TRUE when a Configuration BPDU or a Topology Change Notification BPDU is received on the Port. Its value is set to FALSE by the Port Protocol Migration state machine.

17.18.25 rcvdTc

This is a Boolean variable that is set TRUE when a Configuration Message with a Topology Change flag is received. Its value is set FALSE by the Topology Change state machine.

17.18.26 rcvdTcAck

This is a Boolean variable that is set TRUE when a Configuration Message with a Topology Change Acknowledge flag is received. Its value is set FALSE by the Topology Change state machine.

17.18.27 rcvdTcn

This is a Boolean variable that is set TRUE when a TCN BPDU is received. Its value is set FALSE by the Topology Change state machine.

17.18.28 reRoot

This is a signal controlled by the Root Port. If set TRUE, it instructs any other Ports with the rrWhile timer still running (recent roots) to revert to the Discarding state. For such Ports, the rrWhile timer is stopped once it has reverted to the Discarding state. The reRoot variable is set FALSE when no Port other than the Root Port has rrWhile running.

17.18.29 reselect

This variable is set TRUE by the Port Information state machine if Port Roles are to be re-computed by the Port Role Selection state machine. The variable is set FALSE by the Port Role Selection state machine at the commencement of its computation. If the variable is set TRUE during computation by the Port Role Selection state machine, then computation is repeated.

17.18.30 role

The role is the assigned Port Role. The port is either a DisabledPort, a RootPort, a DesignatedPort, an AlternatePort, or a BackupPort.

17.18.31 selected

This variable is set FALSE by the Port Information state machine at the same time as reselect is set TRUE. The variable is set TRUE by the Port Role Selection state machine at the completion of its computation if reselect is FALSE.

17.18.32 selectedRole

The selectedRole is a newly computed role for the Port.

17.18.33 sendRSTP

This is set to the value TRUE if RST BPDUs are to be sent, FALSE if Configuration BPDUs and/or TCN BPDUs are to be sent.

17.18.34 sync

The sync is a signal controlled by the root port. If set TRUE, it instructs any designated port whose operEdge parameter is FALSE, and that is not in agreement with the current Spanning Tree information, to revert to the Discarding state, thereby establishing agreement with current Spanning Tree information. Its value is set FALSE by the operation of the Port Role Transitions state machine.

17.18.35 synced

This is TRUE if the Port is in agreement with the current Spanning Tree information; otherwise FALSE. This variable is used by the Port Transmit state machine to set the value of the Agreement flag in transmitted RST BPDUs.

17.18.36 tc

This is a Boolean variable that can be set TRUE by the Port State Transition state machine, to indicate that a topology change has occurred. Its value is set FALSE by the Topology Change state machine.

17.18.37 tcAck

This is a Boolean variable that is set TRUE if a Configuration Message with a topology change acknowledge flag set is to be transmitted. The message is to be transmitted at the next transmission opportunity. Its value is set FALSE by the Port Transmit state machine.

17.18.38 tcProp

This is a Boolean variable that can be set TRUE by the Topology Change state machine of any other Port, to indicate that a topology change should be propagated through this Port. Its value is set FALSE by the Topology Change state machine.

17.18.39 tick

This variable is set to TRUE at one second intervals, by the operation of a system clock external to the definition of the state machines. The variable is set FALSE by the Port Timers state machine.

The provision of this clock function is implementation specific.

17.18.40 txCount

This is a counter used by the Port Transmit state machine in order to limit the maximum BPDU transmission rate.

17.18.41 updtInfo

This variable is set TRUE by the Port Role Selection state machine to indicate to the Port Information state machine that it should copy designatedPriority to portPriority and designatedTimes to portTimes. Its value is set FALSE by the Port Information state machine.

NOTE—The operation of the state machines does not invoke reselection if a received BPDU carries the same information as has been previously received.

17.19 State machine procedures

The following naming convention is used for the names of procedures that modify multiple variables (either multiple variables of a single Port or variables of multiple Ports):

a) ***set***: The procedure sets the value of the variables to TRUE.

b) ***clear:*** The procedure sets the value of the variables to FALSE.

c) ***updt:*** The procedure sets the values of the variables in some other way.

The suffix “Bridge” is used for procedures that can modify a variable in all Ports of the Bridge. For example, ***setSyncBridge()*** is the name of a procedure that sets a variable TRUE for all Bridge Ports.

Where procedures are used to determine the value of a single variable, the procedure’s returned value is explicitly assigned to the variable in the state machine concerned.

17.19.1 clearReselectBridge()

This procedure sets reselect = FALSE for all Ports of the Bridge.

17.19.2 disableForwarding()

This procedure takes the steps necessary to disable the forwarding function for this Port. The procedure does not complete until forwarding has been disabled.

17.19.3 disableLearning()

This procedure takes the steps necessary to disable the learning function for this Port. The procedure does not complete until learning has been disabled.

17.19.4 enableForwarding()

This procedure takes the steps necessary to enable the forwarding function for this Port. The procedure does not complete until forwarding has been enabled.

17.19.5 enableLearning()

This procedure takes the steps necessary to enable the learning function for this Port. The procedure does not complete until learning has been enabled.

17.19.6 flush()

This procedure flushes (i.e., removes) all Dynamic Filtering Entries in the Filtering Database that contain information learned on this port, unless this Port is an edge Port (i.e., operEdge is TRUE).

17.19.7 newTcWhile()

This procedure sets the value of tcWhile 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).

17.19.8 rcvBpdu()

This procedure returns SuperiorDesignatedMsg if the received BPDU is an RST BPDU with a Designated Port Role, or a Config BPDU, and either the received message priority vector (see 17.4.2.2) is strictly better than the Port's port priority vector, or the Designated Bridge and Designated Port components of the received message priority vector are the same as those of the port priority vector and the message priority vector as a whole or any of the received timer parameter values (msgTimes - see 17.18.12) differ from those already held in portTimes (17.18.18).

This procedure returns RepeatedDesignatedMsg if the received BPDU is an RST BPDU with a Designated Port Role, or a Config BPDU, and both the message priority vector (msgPriority; see 17.18.11) and the received timer parameter values (msgTimes; see 17.18.12) are the same as those already held in portPriority (17.18.17) and portTimes (17.18.18).

NOTE 1—The message priority vector carried in a repeated designated message is superior to the port priority vector, according to the definition in 17.4.2.2. However, the only action to be taken on receipt of a repeated designated message is to update the Port’s received information timeout timer (rcvdInfoWhile; see 17.15.5); hence the distinction made between superior and repeated designated messages.

This procedure returns ConfirmedRootMsg if the received BPDU was received on a point to point link, and the BPDU is an RST BPDU with a Root Port Role with the first four components of the received message priority vector the same as those of the port priority vector (see 17.4.2.2), and the Agreement flag in the BPDU is set.

NOTE 2—As the definition of rcvdBpdu takes account of the value of ForceVersion, and prevents RST BPDUs from being processed by the Port Information state machine if the value of ForceVersion is greater than or equal to 2, RST BPDUs are not processed by this procedure unless ForceVersion is greater than or equal to 2.

The procedure returns OtherMsg otherwise (the received BPDU contains inferior information, or is a TCN BPDU).

17.19.9 recordProposed()

The recordProposed() returns TRUE if the BPDU is an RST BPDU with a Designated Port Role, and the Proposal flag is set in the BPDU, and the attached LAN is a point to point link. Otherwise, returns FALSE.

NOTE—As the definition of rcvdBpdu takes account of the value of ForceVersion, and prevents RST BPDUs from being processed by the Port Information state machine if the value of ForceVersion is greater than or equal to 2, RST BPDUs are not processed by this procedure unless ForceVersion is greater than or equal to 2.

17.19.10 setSyncBridge()

This procedure sets sync TRUE for all Ports of the Bridge.

17.19.11 setReRootBridge()

This procedure sets reRoot TRUE for all Ports of the Bridge.

17.19.12 setSelectedBridge()

This procedure sets the selected variable TRUE for all Ports of the Bridge if reselect is FALSE for all Ports. If reselect is TRUE for any Port, this procedure takes no action.

17.19.13 setTcFlags()

This procedure sets rcvdTc or rcvdTcAck TRUE if the Topology Change or Topology Change Acknowledgment flags, respectively, are set in a ConfigBPDU or RST BPDU. Sets rcvdTcn TRUE if the BPDU is a TCN BPDU.

17.19.14 setTcPropBridge()

This procedure sets tcprop TRUE for all Ports except the Port that called the procedure.

17.19.15 txConfig()

This procedure transmits a Configuration BPDU. The first four components of the message priority vector (17.4.2.2) conveyed in the BPDU are set to the value of portPriority (17.18.17) 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 value of the Message Age, Max Age, Fwd Delay, and Hello Time parameters conveyed in the BPDU are set to the values held in portTimes (17.18.18) for the Port.

17.19.16 txRstp()

This procedure transmits an RST BPDU. The first four components of the message priority vector (17.4.2.2) conveyed in the BPDU are set to the value of portPriority (17.18.17) for this Port. The Port Role in the BPDU (9.3.3) is set to the current value of the role variable for the transmitting port (17.18.30). The Agreement and Proposal flags in the BPDU are set to the values of the synced (17.18.35) and proposing (17.18.20) variables for the transmitting Port, respectively. The 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 value of the Message Age, Max Age, Fwd Delay, and Hello Time parameters conveyed in the BPDU are set to the values held in portTimes (17.18.18) for the Port.

17.19.17 txTcn()

This procedure transmits a TCN BPDU.

17.19.18 updtBPDUVersion()

This procedure sets rcvdSTP TRUE if the BPDU received is a version 0 or version 1 PDU, either a TCN or a Config BPDU. It sets rcvdRSTP TRUE if the received BPDU is an RST BPDU and (ForceVersion >= 2).

17.19.19 updtRcvdInfoWhile()

This procedure sets rcvdInfoWhile to the number of seconds that Spanning Tree information received on a Port will be held before it is either refreshed by receipt of a further configuration message or aged out.

The effective age of the port information (portPriority and portTimes) is taken as the value of the Message Age parameter carried in a received BPDU, incremented by the greater of (1/16th Max Age) and 1 s, and rounded to the nearest whole second. The value of Message Age and Max Age used in this calculation are taken from the portTimes variable (17.18.18).

If this effective age does not exceed Max Age, the value assigned to rcvdInfoWhile is the lower of the following:

a) Max Age minus this effective age

b) Three times the Hello Time

If this effective age exceeds Max Age, the value assigned to rcvdInfoWhile is zero.

17.19.20 updtRoleDisabledBridge()

This procedure sets selectedRole to DisabledPort for all Ports of the Bridge.

17.19.21 updtRolesBridge()

This procedure calculates the following Spanning Tree priority vectors (17.4.2.2) and Spanning Tree timer values as follows:

a) The *root path priority vector* for each Bridge Port that is not Disabled and has a *port priority vector* (portPriority plus portId; see 17.18.16 and 17.18.17) that has been recorded from a received message and has not yet aged out

b) The *root path times* (the value of portTimes; see 17.18.18) associated with each root path priority vector

c) The Bridge's *root priority vector*, chosen as the best of the set of Spanning Tree priority vectors comprising the Bridge's own *bridge priority vector* (BridgePriority; see 17.17.3) 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 (see 17.4.2.2)

d) The Bridge's *root times*, determined as follows:

   1) If the chosen root priority vector is the bridge priority vector, *root times* is equal to BridgeTimes (see 17.17.4).

   2) If the chosen root priority vector is not the bridge priority vector, *root times* is equal to the value of *root path times* associated with the chosen root priority vector, with the Message Age component incremented by the greater of (1/16 of the Max Age component) and 1, rounded to the nearest whole second.

e) The *designated priority vector* for each port.

f) The *designated times* for each Port (equal to the value of *root times*).

The first four components of the root priority vector are recorded in the rootPriority variable (17.17.6) for the Bridge. The fifth component of the root priority vector is recorded in the rootPortId variable (17.17.5) for the Bridge.

The root times are recorded in the rootTimes variable (17.17.7) for the Bridge.

The first four components of the designated priority vector for each Port are recorded in the designatedPriority variable (17.18.2) for that Port.

The designated times for each Port are recorded in the designatedTimes variable (17.18.3) for that Port.

The port role for each Port is assigned, and its port priority vector and Spanning Tree timer information are updated as follows:

g) If the Port is Disabled (infoIs = Disabled), selectedRole is set to DisabledPort. Otherwise:

h) If the port priority vector information was aged (infoIs = Aged), updtInfo is set and selectedRole is set to DesignatedPort.

i) 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.

j) 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.

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 higher 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.

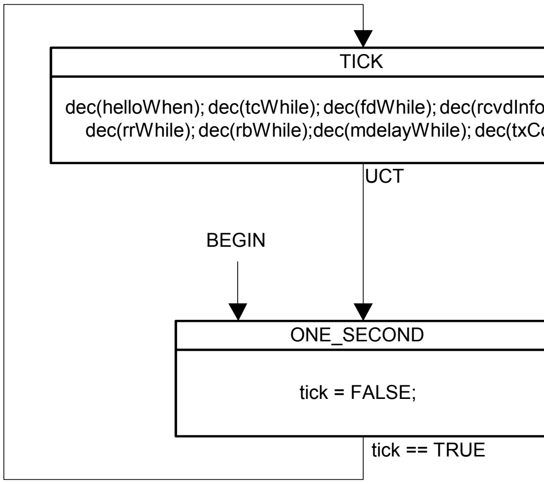
l) If the port priority vector was received in a Configuration Message and is not aged (infoIs = Received), the root priority vector is not now derived from it, the designated priority vector is not higher than the port priority vector, and the designated bridge and designated port components of the port priority vector reflect another port on this bridge, selectedRole is set to BackupPort and updtInfo is reset.

17.20 The Port Timers state machine

The Port Timers state machine for a given Port is responsible for decrementing the timer variables for that Port each second. The state machine enters the ONE_SECOND state on initialization, and the tick variable is set FALSE. A regular one second tick signal, generated by an external system clock function, causes a transition to take place to the TICK state; within that state, all nonzero counters are decremented by one. The state machine then exits to the ONE_SECOND state to clear the tick variable and await the next tick.

The state machine that makes use of a given timer variable is responsible for setting the variable to its initial value.

The Port Timers state machine shall implement the function specified by the state diagram contained in Figure 17-12 and the attendant definitions contained in 17.15.


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;
}
```

Figure 17-12—Port Timers state machine

17.21 The Port Information state machine

The Port Information state machine shall implement the function specified by the state diagram contained in Figure 17-13 and the attendant definitions contained in 17.15 through 17.19.

This state machine is responsible for recording the Spanning Tree information currently in use by this Port, aging that information out if it was derived from an incoming BPDU, and recording the origin of the information in the infoIs variable. This variable, plus the reselect variable, the selected variable, the portPriority variable, and the portTimes variable, are exported by the state machine for use by the Port Role Selection state machine to determine when port roles need to be re-computed.

The DISABLED state is entered on initialization, or if the Port is inoperable (portEnabled is FALSE) and infoIs is not equal to Disabled. In addition to the variable initialization functions performed in this state, infoIs is set to Disabled, selected is set FALSE, and reselect is set TRUE, to cause the Port Role Selection state machine to re-compute the Port roles for the Bridge. Any BPDUs received while in this state, or requests to update the Port's Spanning Tree information, are discarded. The machine transitions to the AGED state once the Port is operable and initialization is complete.

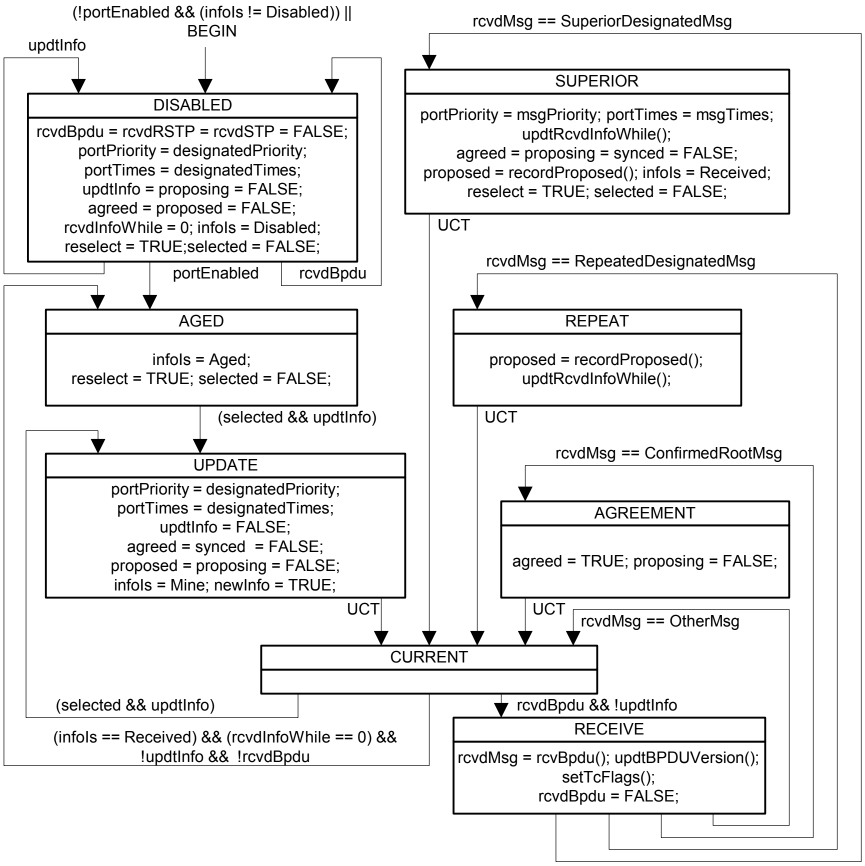

(!portEnabled && (infoIs != Disabled)) ||
BEGIN
updtInfo
DISABLED
rcvdBpdu = rcvdRSTP = rcvdSTP = FALSE;
portPriority = designatedPriority;
portTimes = designatedTimes;
updtInfo = proposing = FALSE;
agreed = proposed = FALSE;
rcvdInfoWhile = 0; infoIs = Disabled;
reselect = TRUE;selected = FALSE;
portEnabled
rcvdBpdu
AGED
infoIs = Aged;
reselect = TRUE; selected = FALSE;
(selected && updtInfo)
UPDATE
portPriority = designatedPriority;
portTimes = designatedTimes;
updtInfo = FALSE;
agreed = synced = FALSE;
proposed = proposing = FALSE;
infoIs = Mine; newInfo = TRUE;
UCT
rcvdMsg == SuperiorDesignatedMsg
SUPERIOR
portPriority = msgPriority; portTimes = msgTimes;
updtRcvdInfoWhile();
agreed = proposing = synced = FALSE;
proposed = recordProposed(); infoIs = Received;
reselect = TRUE; selected = FALSE;
UCT
rcvdMsg == RepeatedDesignatedMsg
REPEAT
proposed = recordProposed();
updtRcvdInfoWhile();
UCT
rcvdMsg == ConfirmedRootMsg
AGREEMENT
agreed = TRUE; proposing = FALSE;
UCT
rcvdMsg == OtherMsg
CURRENT
(selected && updtInfo)
(infoIs == Received) && (rcvdInfoWhile == 0) &&
!updtInfo && !rcvdBpdu
rcvdBpdu && !updtInfo
RECEIVE
rcvdMsg = rcvBpdu(); updtBPDUVersion();
setTcFlags();
rcvdBpdu = FALSE;

Figure 17-13—Port Information state machine

The AGED state can also be entered from the CURRENT state if the current Spanning Tree information for the Port (portPriority and portTimes) originated from a received BPDU, and rcvdInfoWhile has expired; i.e., the information has not been refreshed with information from the same source before the ageing timer associated with the information has expired. In the AGED state, the infoIs variable is set to Aged, and reselect is set TRUE to indicate the need to re-compute Spanning Tree roles. The state machine exits to the UPDATE state once the Port Role Selection state machine has completed its re-computation of roles.

The UPDATE state can also be entered from the CURRENT state when port role re-computation has completed, and the Port Role Selection state machine requires the Port Information state machine to update the Port's portPriority and portTimes from the designatedPriority and designatedTimes. The infoIs variable is set to Mine, and newInfo is set TRUE. The state machine exits from this state to CURRENT.

In the CURRENT state, the Spanning Tree information held for the Port is up to date. The state machine is waiting to process three types of event: incoming BPDU information, timing out of old BPDU information, or requests to update the Port's portPriority and portTimes. If a BPDU is received, the state machine transitions to the RECEIVE state.

In the RECEIVE state, the BPDU is analyzed to determine its type, and the rcvdMsg variable is set accordingly. The updtBPDUVersion function determines whether the BPDU is a Config or TCN BPDU, or an RST BPDU. This information is used by the Port Protocol Migration state machine to determine the type of BPDU that the Port Transmit state machine will generate. Any topology change information contained in the BPDU is recorded. The state machine exits to the SUPERIOR state if the received BPDU contains a message priority vector from a new Designated Bridge, or a changed message priority vector from the current Designated Bridge. The state machine exits to the REPEAT state if the BPDU carries unchanged information from the current Designated Bridge. The state machine exits to the AGREEMENT state if the received message represents an agreement that this Port can transition to Forwarding. Otherwise, the state machine transitions back to the CURRENT state.

In the SUPERIOR state, the message priority vector from the BPDU (msgPriority) is recorded in portPriority, the timer information conveyed in the BPDU (msgTimes) is recorded in portTimes, the rcvdInfoWhile timer is updated using the Message Age carried in the BPDU, and infoIs is set to Received. The reselect variable is set TRUE to signal the need to re-compute Port Roles. If the BPDU carries a Proposal flag, proposed is set TRUE. The state machine transitions back to the CURRENT state.

In the REPEAT state, the message age associated with the information is recorded, and if the BPDU carries a Proposal flag, proposed is set TRUE. The rcvdInfoWhile timer is updated using the Message Age carried in the BPDU. The state machine transitions back to the CURRENT state.

In the AGREEMENT state, agreed is set TRUE, proposing is set FALSE, and the state machine transitions back to the CURRENT state.

17.22 The Port Role Selection state machine

The Port Role Selection state machine shall implement the function specified by the state diagram contained in Figure 17-14 and the attendant definitions contained in 17.15 through 17.19.

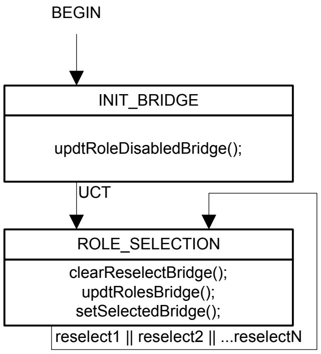


Figure 17-14—Port Role Selection state machine

This state machine is responsible for computing the Port roles for all Ports of the Bridge.

On initialization, the INIT_BRIDGE state is entered, the Port role is set to Disabled Port for all Ports of the Bridge, and the state machine transitions to the ROLE_SELECTION state.

The ROLE_SELECTION state is re-entered whenever any of the reselect variables for any of the Bridge Ports becomes TRUE, indicating that re-computation of the Port roles is necessary. The reselect variables for all Ports of the Bridge are set FALSE on entry to the state. The updtRolesBridge function (see 17.19.21) then computes the Port roles, by setting the value of the selectedRole variable for each Port to the appropriate value (DisabledPort, RootPort, DesignatedPort, BackupPort or AlternatePort). Once computation is complete, the selected variable for each Port is set TRUE.

Clearing the reselect variables before the start of computation ensures that if new information becomes available during computation, the ROLE_SELECTION state will be immediately re-entered on completion of the computation.

17.23 The Port Role Transitions state machine

The Port Role Transitions state machine shall implement the function specified by the state diagrams contained in Figure 17-15, Figure 17-16, and Figure 17-17, and the attendant definitions contained in 17.15 through 17.19.

As Figure 17-15, Figure 17-16, and Figure 17-17 are component parts of the same state machine, the global transitions associated with all three diagrams are possible exit transitions from the states shown in any of the three diagrams.

Two variables, learn and forward (see 17.18.4 and 17.18.8), are used by this state machine to signal to the Port State Transitions state machine the need to change the Port State. Two further variables, learning and forwarding (see 17.18.5 and 17.18.9), are used to signal back to this state machine when the Port State transition has actually occurred. Hence, in this state machine, state transitions that depend upon the actual Port State are qualified by the current value of the learning or forwarding variables. State transitions that are concerned with requesting changes in Port State are qualified by the current value of the learn or forward variables, in order to avoid making repeated requests to change to the same Port State when previous requests are being actioned by the Port State Transitions state machine.

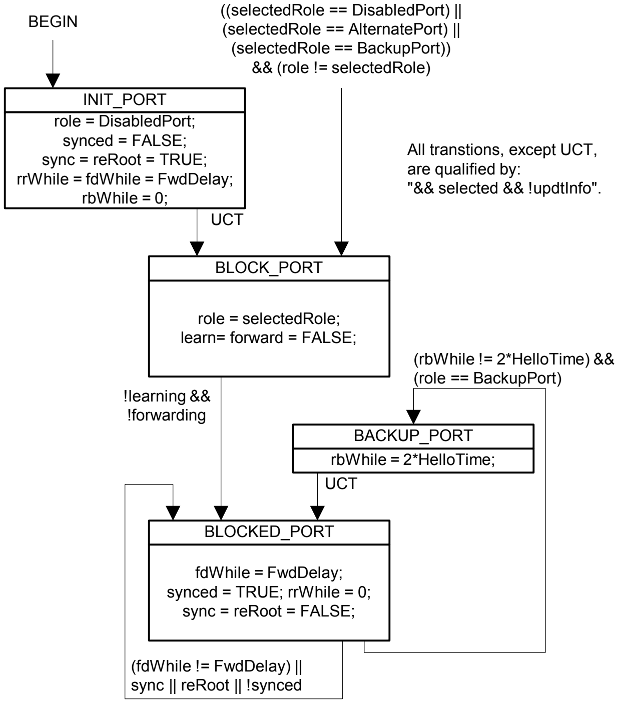
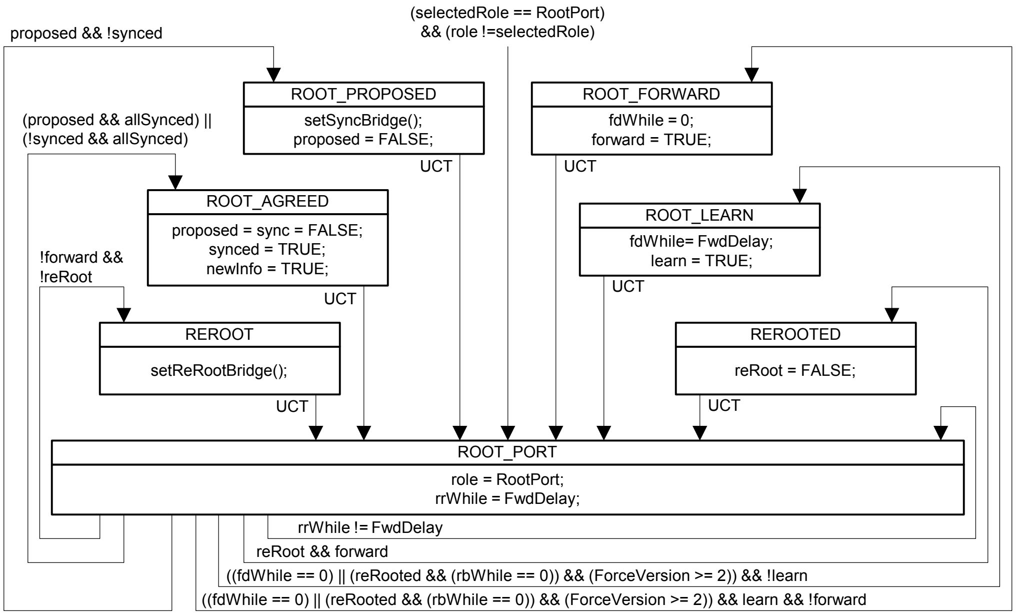
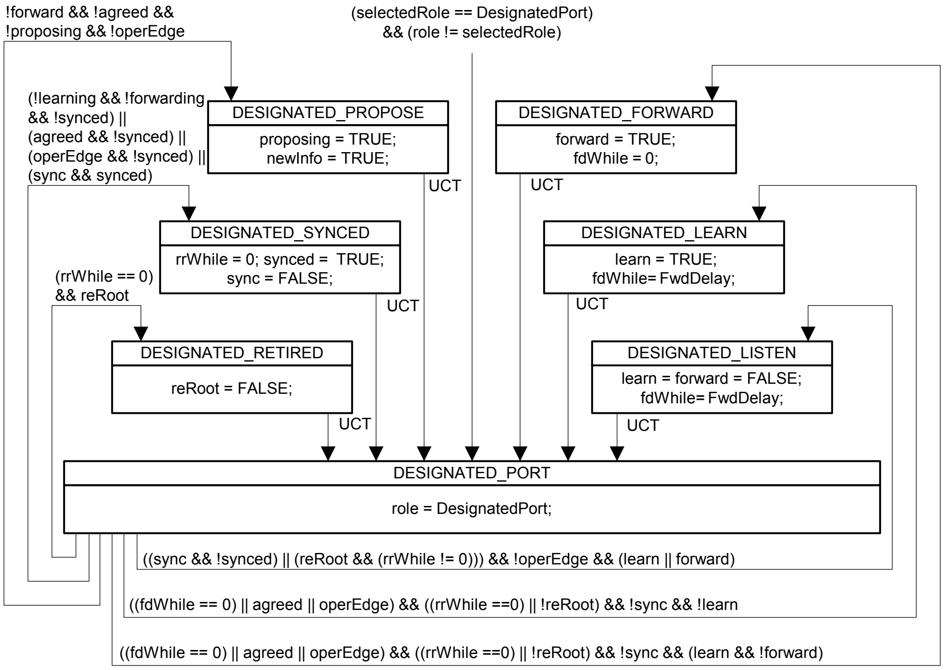


Figure 17-15—Port role transitions state machine—Part 1: Disabled, alternate, and backup role



All transtions, except UCT, are qualified by "&& selected && !updtInfo".
The following abbreviations are used in this diagram:
allSynced: : (synced1 && synced2 && ... syncedN) for all Ports other than this Root Port.
reRooted: ((rrWhile1 == 0) && (rrWhile2 == 0) && ... (rrWhileN == 0)) for all ports except this Root Port.

Figure 17-16—Port role transitions state machine—Part 2: Root port role



All transtions, except UCT, are qualified by "&& selected && !updtInfo".

Figure 17-17—Port role transitions state machine—Part 3: Designated port role

17.23.1 Disabled, alternate, and backup port role transitions

Figure 17-15 shows initialization of the port role transitions state machine and the Disabled, Alternate, and Backup roles and their associated states.

The INIT_PORT state is entered on initialization. The role variable (the current role of the Port) is set to DisabledPort. Once it has set the synced, sync, reRoot, rrWhile, fdWhile, and rbWhile variables to their initial states, the state machine transitions to the BLOCK_PORT state.

NOTE—The initialization action will also cause the Port Information state machine to set infoIs to Disabled, which in turn will result in the Port Role Selection state machine setting the selectedRole for the Port to DisabledPort.

The BLOCK_PORT state is entered if there is a discrepancy between the current role understood by the state machine (role) and the role selected by the Port Role Selection state machine (selectedRole), and if the selectedRole is DisabledPort, AlternatePort or BackupPort; i.e., any state for which the Port should be placed permanently in the Discarding Port State. The role is set to the value of selectedRole, and learn and forward are set FALSE in order to disable learning and forwarding. Once learning and forwarding are both disabled, the state machine transitions to the BLOCKED_PORT state.

The BLOCKED_PORT state sets sync and reRoot to FALSE, and synced to TRUE. The rrWhile timer is cleared, and fdWhile is set to FwdDelay. Re-entry to this state occurs if sync or reRoot become TRUE, or if synced becomes FALSE. The state is also re-entered if fdWhile is not equal to FwdDelay; this ensures that the value of fdWhile is equal to FwdDelay should a subsequent role change occur to make this a Root Port or Designated Port.

A transition from BLOCKED_PORT to BACKUP_PORT occurs if the value of rbWhile is not equal to twice HelloTime and the Port role is BackupPort; this ensures that the value of rbWhile will be at its maximum if a role change to RootPort or DesignatedPort occurs.

17.23.2 Root Port role transitions

Figure 17-16 shows the states associated with the Root Port role.

The ROOT_PORT state is entered from any of the states in Figure 17-15 and Figure 17-17 if a recomputation of Port roles by the Port Role Selection state machine causes the selectedRole to be changed to RootPort. This state is also entered via a UCT from any of the other states shown in Figure 17-16. If the previous role was DisabledPort, AlternatePort or BackupPort, the value of fdWhile will be equal to FwdDelay on entry to this state. If the previous role was DesignatedPort, fdWhile can have any value between zero and FwdDelay on entry to this state, depending upon the progress that had been made towards the Forwarding Port State while the Port was a DesignatedPort. On entry to the state, the role variable is set to RootPort, and rrWhile is set to FwdDelay. Re-entry to the ROOT_PORT state occurs if rrWhile is not equal to FwdDelay.

Entry to the REROOT state from the ROOT_PORT state occurs if a change in Port State to Forwarding has not been requested and any recent Root Ports of the Bridge have not yet been instructed to revert to the Discarding Port State. The reRoot variable is set TRUE for all Bridge Ports.

Entry to the REROOTED state from the ROOT_PORT state occurs if a change in Port State to Forwarding has been requested but a reRoot request is still outstanding (reRoot is TRUE). The reRoot variable is set FALSE.

Entry to the ROOT_LEARN state from the ROOT_PORT state occurs if learn is FALSE, and either:

a) The fdWhile timer has expired; or
b) The rbWhile timer for this Port is zero, rrWhile is zero for all Ports except this Root Port, and the protocol version selected by ForceVersion is version 2 or greater.

These conditions mean that the Port State can transition from Discarding to Learning; either one Forwarding Delay time has expired since the Port State became Discarding, or there is no historical state associated with any of the Bridge Ports that would prevent the state transition from taking place (no other Port in the Bridge has recently been a RootPort, and this Port has not recently been a BackupPort). The learn variable is set TRUE to indicate to the Port State Transition state machine that the Port State should be set to Learning, and the fdWhile timer is restarted with the value FwdDelay.

Entry to the ROOT_FORWARD state from the ROOT_PORT state occurs if learn is TRUE, and forward is FALSE, and either:

c) The fdWhile timer has expired; or
d) The rbWhile timer for this Port is zero, reRooted is TRUE, and the protocol version selected by ForceVersion is version 2 or greater.

These conditions mean that the Port State can transition from Learning to Forwarding; either one Forwarding Delay time has expired since the Port became Learning, or there is no historical state associated with any of the Bridge Ports that would prevent the state transition from taking place (no other Port in the Bridge has recently been a RootPort, and this Port has not recently been a BackupPort). The forward variable is set TRUE to indicate to the Port State Transition state machine that the Port State should be set to Forwarding, and the fdWhile timer is cleared.

Entry to the ROOT_PROPOSED state from the ROOT_PORT state occurs if a Proposal flag is received from the designated Bridge on the LAN, and the Port is not in agreement with current Spanning Tree information. The sync variable is set TRUE for all ports of the Bridge, in order to force any designated Ports that are not edge Ports to revert to Discarding, and the proposed variable is cleared.

Entry to the ROOT_AGREED state from the ROOT_PORT state occurs either if all other Ports are in agreement with current Spanning Tree information (synced is TRUE for all other Ports), but this Port still has synced set FALSE; or if a Proposal flag is received from the designated Bridge on the LAN and the Port is already in agreement with current Spanning Tree information. The proposed and sync variables are set FALSE, synced is set TRUE, and the newInfo variable is set TRUE to force the Port Transmit state machine to send the necessary confirmation to the designated Bridge.

17.23.3 Designated Port role transitions

Figure 17-17 shows the states associated with the Designated Port role.

Entry to the DESIGNATED_PORT state from any of the states in Figure 17-15 and Figure 17-16 if a re-computation of Port roles by the Port Role Selection state machine causes the selectedRole to be changed to DesignatedPort. This state is also entered via a UCT from any of the other states shown in Figure 17-17. If the previous role was DisabledPort, AlternatePort, or BackupPort, the value of fdWhile will be equal to FwdDelay on entry to the state. If the previous role was RootPort, fdWhile can have any value between zero and FwdDelay on entry to this state, depending upon the progress that had been made towards the Forwarding Port State while the Port was a RootPort. On entry to the state, the role variable is set to DesignatedPort.

Entry to the DESIGNATED_RETIRED state from the DESIGNATED_PORT state occurs if the reRoot variable is set TRUE and the rrWhile timer is not running; i.e., the current Root Port has requested recent Root Ports to be retired, but this Port is not a recent Root Port. The reRoot variable is set FALSE.

Entry to the DESIGNATED_LISTEN state from the DESIGNATED_PORT state occurs if either learn or forward is TRUE, and the Port is not an edge Port, and either:

a) The rrWhile timer is running, and the current Root Port has requested that recent Root Ports revert to the Discarding Port State; or

b) The Port is not in agreement with current Spanning Tree information, and the current Root Port has instructed Designated Ports that are not in agreement with current Spanning Tree information to revert to the Discarding Port State.

The learn and forward variables are set FALSE to indicate to the Port State Transition state machine that the Port State should be set to Discarding, and the fdWhile timer is set equal to FwdDelay.

Entry to the DESIGNATED_LEARN state from the DESIGNATED_PORT state occurs if learn and sync are both FALSE, and either rrWhile is not running or there is no outstanding request from the Root Port to retire recent Root Ports, and either:

c) The forwarding delay has expired; or

d) The Port is an edge Port; or

e) An agreement has been received in a BPDU from the Root Port of the Bridge attached to the LAN (agreed == TRUE).

These conditions mean that the Port State can transition from Discarding to Learning; there is no state associated with the Root Port that would prevent the state transition from taking place (the Root Port is not requesting this Port to revert to Discarding for any reason), and either one Forwarding Delay time has expired since the Port became Discarding, or the Port is already in agreement with current Spanning Tree information, or the other Bridge has synchronized its Port states with the current Spanning Tree information. The learn variable is set TRUE to indicate to the Port State Transition state machine that the Port State should be set to Learning, and the fdWhile timer is restarted with the value FwdDelay.

Entry to the DESIGNATED_FORWARD state from the DESIGNATED_PORT state occurs if forward and sync are both FALSE, and learn is TRUE, and either rrWhile is not running or there is no outstanding request from the Root Port to retire recent Root Ports, and either:

f) The forwarding delay has expired; or

g) The Port is an edge Port; or

h) An agreement has been received in a BPDU from the Root Port of the Bridge attached to the LAN (agreed == TRUE).

These conditions mean that the Port State can transition from Learning to Forwarding; there is no state associated with the Root Port that would prevent the state transition from taking place (the Root Port is not requesting this Port to revert to Discarding for any reason), and either one Forwarding Delay time has expired since the Port became Discarding, or the Port is already in agreement with current Spanning Tree information, or the other Bridge has synchronized its Port states with the current Spanning Tree information. The forward variable is set TRUE to indicate to the Port State Transition state machine that the Port State should be set to Forwarding, and the fdWhile timer is cleared.

Entry to the DESIGNATED_PROPOSE state from the DESIGNATED_PORT state occurs if the Port is not an edge Port, and forward is FALSE, and the Port is not in agreement with current Spanning Tree information, and the Port has not already sent a Proposal flag to the Bridge on the LAN to which it is connected. The proposing variable is set TRUE to indicate that a Proposal flag is to be sent (and to prevent multiple Proposals being generated), and the newInfo variable is set TRUE to indicate to the Port Transmit state machine that there is new information to be transmitted.

Entry to the DESIGNATED_SYNCED state from the DESIGNATED_PORT state occurs if any of the following are true:

i) The Port is neither Learning nor Forwarding, but the Port is not indicating that it is in agreement with current Spanning Tree information (the synced variable is FALSE).

j) A response has been received from the Bridge on the LAN to which the Port is connected, indicating that the Port may proceed to the Forwarding Port State, but the Port is not indicating that it is in agreement with current Spanning Tree information.

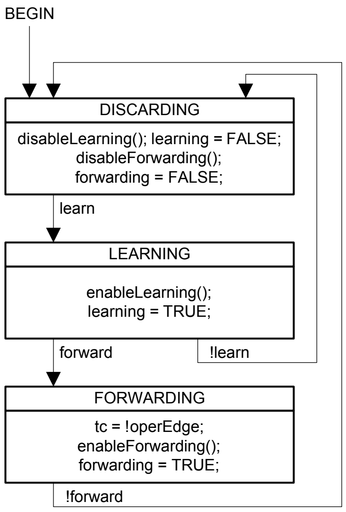
k) The Port is an edge Port, but the Port is not indicating that it is in agreement with current Spanning Tree information.

l) The Root Port has requested this Port to establish agreement with current Spanning Tree information, and the Port is already in agreement with current Spanning Tree information.

In other words, the Port is now in agreement with current Spanning Tree information, but one or more of the variables related to agreement need to be updated to reflect that fact. The synced variable is set TRUE, sync is set FALSE, and the rrWhile timer is stopped.

17.24 Port State Transition state machine

The Port State Transition state machine shall implement the function specified by the state diagram contained in Figure 17-18 and the attendant definitions contained in 17.15 through 17.19.



NOTE: A small system dependent delay may occur on each of the transitions shown.

Figure 17-18—Port State Transition state machine

This state machine models the changes in the Port State (17.5) between Discarding, Learning and Forwarding. These state changes are initiated by the Port Role Transitions state machine, by means of the learn and forward variables; once the state machine has made the state transition demanded by the state of these variables, this is signalled back via the learning and forwarding variables. This allows modelling of situations in which there may be a system-dependent delay imposed between requesting that a Port start to learn or forward and the state change actually taking place. The functions disableLearning, disableForwarding, enableLearning, and enableForwarding model the system-dependent actions that need to take place in order to perform these state changes; these functions do not complete their operation until the desired change of learning or forwarding behavior has been achieved.

The state machine is initialized in the DISCARDING state; both learning and forwarding are disabled and the learning and forwarding variables are set FALSE. Exit from this state to LEARNING takes place if the learn variable is set TRUE.

On entry to the LEARNING state, learning is enabled and then the learning variable is set TRUE. The state machine transitions back to DISCARDING if learn becomes FALSE, and transitions to FORWARDING if forward becomes TRUE.

In the FORWARDING state, the tc variable is set TRUE if the Port is not an edge Port; this signals to the Topology Change state machine that topology change information should be propagated, as this Port has been added to the active topology. Forwarding is enabled and then the forwarding variable is set TRUE. The state machine will revert to the DISCARDING state if forward becomes FALSE.

NOTE—The tc variable is set TRUE only on an increase in connectivity (i.e., a Port going into FORWARDING); this is a deliberate change from the operation of STP, where Topology Change notifications are also generated on a loss of connectivity (i.e., a Port transitioning from FORWARDING to DISCARDING).

17.25 Topology Change state machine

The Topology Change state machine shall implement the function specified by the state diagram contained in Figure 17-19 and the attendant definitions contained in 17.15 through 17.19.

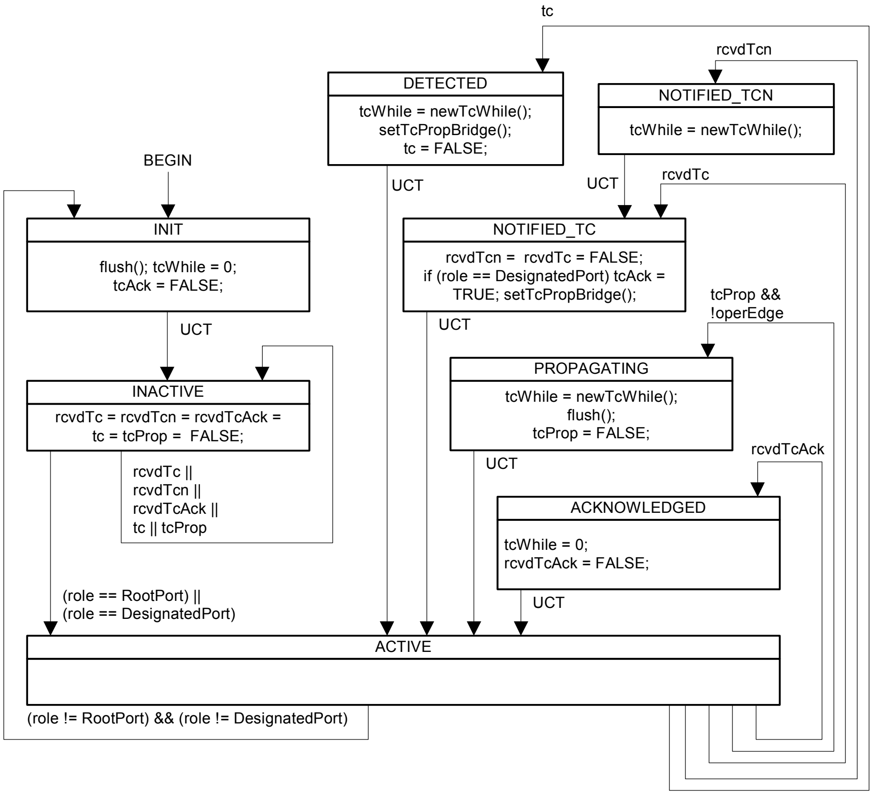


Figure 17-19—Topology Change state machine

This state machine is responsible for topology change detection, notification, and propagation, and for filtering database flushing.

The state machine enters the INIT state on initialization. The Filtering Database is flushed to remove information that was learned on this Port. The tcAck variable is set FALSE. The state machine transitions to the INACTIVE state.

In the INACTIVE state, the tc, tcProp, rcvdTc, rcvdTcn and rcvdTcAck variables are set FALSE; if any of these variables becomes TRUE, the state is re-entered in order to force them FALSE again. This ensures that no attempt is made to propagate TC information on Ports that are in the Backup or Alternate roles. The state machine will remain in the INACTIVE state until the Port role becomes either DesignatedPort or RootPort, which causes a transition to ACTIVE.

The ACTIVE state can also be entered via a UCT from the DETECTED, NOTIFIED, PROPAGATING, and ACKNOWLEDGED states. The state machine remains in the ACTIVE state until the following:

a) The Port role reverts to Alternate or Backup, causing a transition to INIT; or

b) The tc variable becomes TRUE, causing a transition to DETECTED; or

c) A TCN BPDU is received, causing a transition to NOTIFIED_TCN; or

d) A BPDU with the TC flag set is received, causing a transition to NOTIFIED_TC; or

e) The Port is not an edge Port and the tcProp variable is set TRUE, causing a transition to PROPAGATING; or

f) A BPDU is received with the TC acknowledgement flag set, causing a transition to ACKNOWLEDGED.

The DETECTED state is entered when this Port has detected a topology change. The tcWhile variable is set (see 17.19.7), and tcProp is set TRUE for all other Ports of the Bridge, causing them to propagate TC information if they are part of the active topology and are not edge Ports. The tc variable is cleared before the transition back to ACTIVE.

The NOTIFIED_TCN state is entered if the Port receives incoming TCN information (i.e., the originating Bridge is running STP). The tcWhile timer is set for the Port to ensure that the topology change is propagated to all Bridges that are downstream from this Port. The state machine then transitions to NOTIFIED_TC.

The NOTIFIED_TC state is entered if the Port receives incoming TC information, or as an unconditional transition from NOTIFIED_TCN. If the Port role is DesignatedPort, the tcAck variable is set TRUE. The rcvdTC and rcvdTCN variables are cleared, and tcProp is set TRUE for all other Ports of the Bridge, causing them to propagate TC information if they are part of the active topology and are not edge Ports.

The PROPAGATING state is entered if the Port is not an edge Port and its tcProp variable is set by some other Port of the Bridge, indicating the need to propagate a topology change. The tcWhile timer is set (see 17.19.7), tcProp is set FALSE, and the Filtering Database is flushed to remove information that was learned on this Port. While the tcWhile timer is nonzero, the Port Transmit state machine will propagate TCN messages towards the Root at Hello Time intervals if the Port is a Root Port.

The ACKNOWLEDGED state is entered if a BPDU is received in which the TC acknowledge flag is set. The tcWhile variable is cleared, and rcvdTcAck is set FALSE.

17.26 Port Protocol Migration state machine

The Port Protocol Migration state machine shall implement the function specified by the state diagram contained in Figure 17-20 and the attendant definitions contained in 17.15 through 17.19.

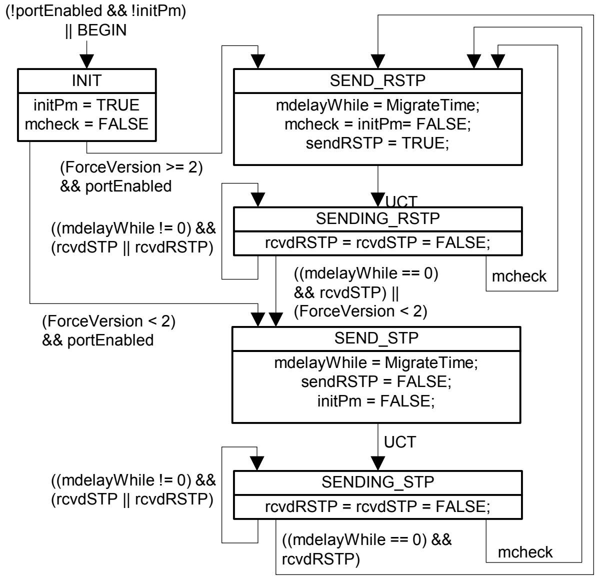


Figure 17-20—Port protocol migration state machine

Clause 9 specifies three BPDU types as follows:

a) Configuration BPDUs

b) Topology Change Notification BPDUs

c) Rapid Spanning Tree BPDUs

Configuration and Topology Change Notification BPDUs are the only BPDU types recognized and transmitted by the Spanning Tree Algorithm and Protocol defined in Clause 8. The Rapid Spanning Tree Algorithm and Protocol is able to recognize and transmit all three types of BPDU. RSTP uses these different BPDU types as follows:

d) On Ports where no Configuration or Topology Change Notification BPDUs have been received (i.e., edge Ports or Ports that are connected only to Bridges that support RSTP), and if the ForceVersion variable is set to 2 or more, only Rapid Spanning Tree BPDUs are used, making use of the Topology Change Notification flag to signal topology changes as necessary.

e) On Ports where Configuration or Topology Change Notification BPDUs have been received (i.e., Ports that are connected to one or more Bridges that do not support RSTP), or if the ForceVersion variable is set to less than 2, Configuration BPDUs and Topology Change Notification BPDUs are used, and RST BPDUs are not used.

In order to achieve this end, it is necessary for systems that support RSTP to be sensitive to the presence of Bridges that support only the Spanning Tree Algorithm and Protocol (see Clause 8) on any of the LANs to which they are connected, and to take notice of the ForceVersion parameter value, and to adjust their use of BPDUs accordingly. The Port Protocol Migration state machine in Figure 17-20 defines how the three types of BPDU are to be used on a given Port. When this state machine is in either the SEND_RSTP or

SENDING_RSTP state, only RST BPDUs are transmitted, as required by the operation of RSTP; when this state machine is in either the SEND_STP or SENDING_STP state, only Configuration BPDUs and Topology Change Notification BPDUs are transmitted. The state machine communicates the transmission format to the Port Transmit state machine via the value of the sendRSTP parameter. The mcheck variable provides a means whereby the state machine can be forced to re-check the appropriate BPDU format to send (but only if ForceVersion is 2 or more); this variable can be set TRUE by management.

The INIT state is entered upon initialization, or when the Port is disabled. The initPm variable is set TRUE to prevent repeated re-entry to this state if the reason for entry was that the Port is disabled. When initialization is complete (BEGIN is FALSE) and the Port is operable, transition to the SEND_STP state occurs if ForceVersion is set to less than 2 (“STP Compatibility” mode); otherwise, transition to the SEND_RSTP state occurs.

The SEND_RSTP state can also be entered from the SENDING_RSTP or SENDING_STP states. The sendRSTP variable is set TRUE, causing the Port Transmit state machine to start transmitting RST BPDUs. The mcheck variable is set FALSE, and mdelayWhile is started using the constant MigrateTime. The initPm variable is set FALSE, to allow the state machine to transition to INIT should the Port become disabled.

The SENDING_RSTP state is entered via a UCT from SEND_RSTP. While mdelayWhile is running (nonzero), the state machine ignores all received BPDUs; if either rcvdRSTP or rcvdSTP become TRUE, the state is re-entered and the variables are set FALSE. The use of the mdelayWhile timer provides a hysteresis period that avoids undue oscillation between the use of the two types of PDU. If mcheck becomes TRUE, indicating that management has requested re-checking of the appropriate BPDU type to send, the state machine returns to SEND_RSTP. Once the mdelayWhile timer has expired, receipt of a Config or TCN BPDU on the Port (rcvdSTP is TRUE) causes a transition to SEND_STP. If ForceVersion is changed to a value of less than 2, transition to SEND_STP occurs.

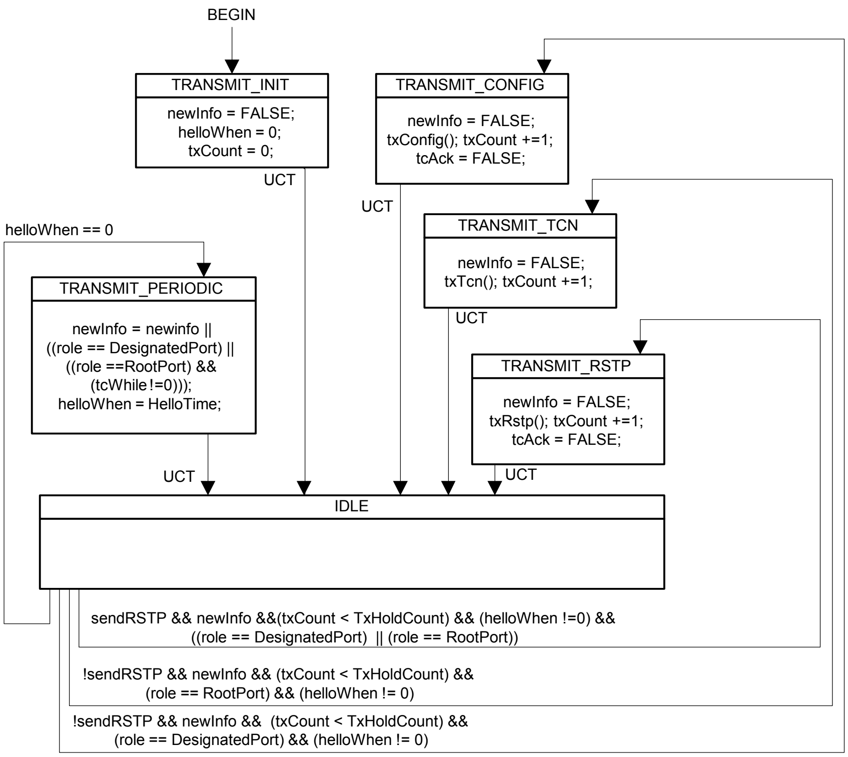
In the SEND_STP state, mdelayWhile is restarted using MigrateTime, and sendRSTP is set FALSE, causing the Port Transmit state machine to start transmitting Config and TCN BPDUs. The initPm variable is set FALSE, to allow the state machine to transition to INIT should the Port become disabled.

The SENDING_STP state is entered via a UCT from SEND_STP. While mdelayWhile is running (nonzero), the state machine ignores all received BPDUs; if either rcvdRSTP or rcvdSTP become TRUE, the state is re-entered and the variables are set FALSE. The use of the mdelayWhile timer provides a hysteresis period that avoids undue oscillation between the use of the two types of PDU. If mcheck becomes TRUE, indicating that management has requested re-checking of the appropriate BPDU type to send, the state machine returns to SEND_RSTP. Once the mdelayWhile timer has expired, receipt of an RST BPDU on the Port (rcvdRSTP is TRUE) causes a transition to SEND_RSTP.

NOTE—The variables mcheck and rcvdRSTP cannot be set TRUE while ForceVersion is set to less than 2. Hence, if ForceVersion is less than 2, this state machine cannot transition out of the SENDING_STP state, regardless of the type of BPDU received, and management attempts to set mcheck TRUE will fail.

17.27 Port Transmit state machine

The Port Transmit state machine shall implement the function specified by the state diagram contained in Figure 17-21 and the attendant definitions contained in 17.15 through 17.19.



All transtions, except UCT, are qualified by "&& selected &&!updtInfo".

Figure 17-21—Port Transmit state machine

The Port Transmit state machine is responsible for transmitting BPDUs, in the format indicated by the sendRSTP variable, when the newInfo variable indicates that there is new information. The state machine imposes a rate limit on the transmission of BPDUs; the variable txCount is incremented every time a transmission occurs, and if txCount reaches TxHoldCount, transmissions can no longer take place. The value of txCount is decremented every second by the operation of the Port Timers state machine; hence, a “credit” of one further transmission is made every second, up to a maximum of TxHoldCount. The consequence of this is that at least one, and not more than TxHoldCount, BPDUs can be transmitted during every HelloTime period.

This state machine makes use of the sendRSTP variable (managed by the Port Protocol Migration state machine) to determine whether RST BPDUs, or Config and TCN BPDUs, will be transmitted. The actual information transmitted in the BPDUs differs, depending upon the value of sendRSTP, as indicated in the description of the txConfig and txRstp functions (see 17.19.15 and 17.19.16).

The TRANSMIT_INIT state is entered on initialization. The helloWhen timer and txCount (the count of transmissions within the HelloTime period) are cleared, and the newInfo flag is set FALSE.

The state machine enters the IDLE state from any other state via a UCT. Exit from this state occurs if any of the following conditions occur:

a) The state machine transitions to TRANSMIT_PERIODIC if helloWhen expires.

b) The state machine transitions to TRANSMIT_RSTP if sendRSTP is TRUE, and newInfo is TRUE, and txCount is less than TxHoldCount, and helloWhen is nonzero, and the Port is either a Designated Port or a Root Port.

c) The state machine transitions to TRANSMIT_TCN if sendRSTP is FALSE, and newInfo is TRUE, and txCount is less than TxHoldCount, and the Port is a Root Port, and helloWhen is nonzero.

d) The state machine transitions to TRANSMIT_CONFIG if sendRSTP is FALSE, and newInfo is TRUE, and txCount is less than TxHoldCount, and the Port is a Designated Port, and newInfo is TRUE, and helloWhen is nonzero.

The TRANSMIT_PERIODIC state sets helloWhen to helloTime, and sets newInfo TRUE if:

e) The Port is a Designated Port; or

f) The Port is a Root Port and tcWhile is running.

The TRANSMIT_RSTP state causes an RST BPDU to be transmitted. The txCount variable is incremented, newInfo is cleared, and tcAck is set FALSE.

The TRANSMIT_TCN state causes a TCN BPDU to be transmitted. The txCount variable is incremented, and newInfo is cleared.

The TRANSMIT_CONFIG state causes a Configuration BPDU to be transmitted. The txCount variable is incremented, newInfo is cleared, and tcAck is set FALSE.

17.28 Performance

This subclause places requirements on the performance of the Bridges in a Bridged LAN and on the setting of the parameters of RSTP. These are necessary to ensure that the algorithm and protocol operate correctly.

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.

It specifies absolute maximum values for performance parameters. The ranges of applicable values are specified to assist in the choice of operational values and to provide guidance to implementors.

The values shown here are based on those originally specified for STP. As the operation of RSTP is relatively insensitive to the values of these timers, and as in many cases, these values now represent backup timers that cope with exception conditions, there is no reason to deviate from the recommended values of these timers. It is therefore recommended that the values of these timers are not changed in normal operation.

NOTE—In Bridged LANs containing only RSTP-capable Bridges, there is no need to vary the parameter values from their recommended or default values. However, in order for RSTP Bridges to integrate into LANs that contain legacy Bridges that do not implement RSTP, the values shown include the range of permissible values that were originally defined for the Spanning Tree Algorithm and Protocol defined in Clause 8.

17.28.1 Requirements

For correct operation, the parameters and configuration of Bridges in the Bridged LAN ensure the following:

— Bridges do not initiate reconfiguration if none is needed. This means that a Bridge Protocol Message is not timed out before its successor arrives, unless a failure has occurred.

— Following reconfiguration, frames are not forwarded on the new active topology, while frames that were initially forwarded on the previous active topology are still in the Bridged LAN. This ensures that frames are not duplicated.

These requirements are met through placing restrictions on the following:

— The **maximum bridge diameter** of the Bridge LAN: The maximum number of Bridges between any two points of attachment of end stations.

— The **maximum bridge transit delay**: The maximum time elapsing between reception and transmission by a Bridge of a forwarded frame, frames that would otherwise exceed this limit being discarded.

— The **maximum BPDU transmission delay**: The maximum delay prior to the transmission of a Bridge Protocol Data Unit following the need to transmit such a BPDU arising, as specified in 17.7.

— The **maximum Message Age increment overestimate** that may be made to the value of the Message Age parameter in transmitted BPDUs or to the age of stored Bridge Protocol Message information.

— The values of the **Bridge Hello Time**, **Bridge Max Age**, **Bridge Forward Delay**, and **Transmission Limit** parameters.

Additionally, a Bridge shall not

— Underestimate the increment to the Message Age parameter in transmitted BPDUs

— Underestimate Forward Delay

— Overestimate the Hello Time interval

17.28.2 Parameter values

Recommended default, absolute maximum, and ranges of parameters are specified in Tables 17-3 through 17-7.

Table 17-3—Maximum bridge diameter

Parameter	Recommended value
Maximum bridge diameter	7

Table 17-4—Transit and transmission delays

Parameter	Recommended value	Absolute maximum
Maximum bridge transit delay	1.0	4.0
Maximum BPDU transmission delay	1.0	4.0
Maximum Message Age increment overestimate	1.0	4.0

All times are in seconds.

Table 17-5—Rapid Spanning Tree algorithm timer values

Parameter	Recommended or default value	Fixed value	Range
Bridge Hello Time	2.0	—	1.0–10.0
Bridge Max Age	20.0	—	6.0–40.0
Bridge Forward Delay	15.0	—	4.0–30.0
Transmission Limit (TxHoldCount - see 17.16.6)	3	—	—

All times are in seconds.
— Not applicable.
Subclause 17.28.2 constrains the relationship between Bridge Max Age and Bridge Forward Delay.

Table 17-6—Bridge and port priority parameter values

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

Table 17-7—Path cost parameter values

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

*Bridges conformant to IEEE Std 802.1D, 1998 Edition, i.e., that support only 16 bit values for Path Cost, should use 65 535 as the Path Cost for these link speeds when used in conjunction with Bridges that support 32 bit Path Cost values.

If the values of **Bridge Hello Time**, **Bridge Max Age**, and **Bridge Forward Delay** can be set by management, the Bridge shall have the capability to use the full range of values in the parameter ranges specified in Table 8-3, with a resolution of *r* seconds, where $0 < r <= 1$.

A Bridge shall use the value of **Transmission Limit** shown in Table 8-3.

A Bridge shall enforce the following relationships:

$2 \times (\mathit{Bridge\_Forward\_Delay} - 1.0\ \mathit{seconds}) >= \mathit{Bridge\_Max\_Age}$

$\mathit{Bridge\_Max\_Age} >= 2 \times (\mathit{Bridge\_Hello\_Time} + 1.0\ \mathit{seconds})$

It is recommended that default values of the **Path Cost** parameter for each Bridge Port be based on the values shown in Table 17-7, the values being chosen according to the speed of the LAN segment to which each Port is attached.

Where intermediate link speeds are created as a result of the aggregation of two or more links of the same speed (see IEEE Std 802.3, 2000 Edition), it may be appropriate to modify the recommended values shown in Table 17-7 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 shall not automatically cause changes in Path Cost for the Port concerned.

NOTE 1—The values shown in Table 17-7 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.

If the values of the **Bridge Priority** and the **Port Priority** for each of the Ports can be set by management, the Bridge shall have the capability to use the full range of values in the parameter ranges specified in Table 17-6, with a granularity of 4096 for Bridge Priority and a granularity of 16 for Port Priority.

NOTE 2—The stated ranges and granularities for Bridge Priority and Port Priority differ from the equivalent text and table in IEEE Std 802.1D-1998 and earlier versions of this standard. The rationale for these changes can be found in 9.2.5 and 9.2.7 of IEEE Std 802.1t-2001. Expressing these value ranges in steps of 4096 and 16, respectively, (rather than as 4-bit values with a range of 0 to 15) allows these parameters to be managed consistently across old and new implementations of this standard; the step values chosen ensure that the low-order bits that have been re-assigned cannot be modified, but that the magnitude of the priority values can be directly compared with those based on previous versions of the standard.

If the value of **Path Cost** can be set by management, the Bridge shall have the capability to use the full range of values in the parameter ranges specified in Table 17-7, with a granularity of 1.

NOTE 3—BPDUs are capable of carrying 32 bits of Path Cost information; however, IEEE Std. 802.1D, 1998 Edition and earlier revisions of this standard limited the range of the Path Cost parameter to a 16 bit unsigned integer value. The recommended values shown in Table 17-7 make use of the full 32 bit range available in BPDUs in order to extend the range of link speeds supported by the protocol. The recommended values for any intermediate link speed can be calculated as 20 000 000 000/(Link Speed in Kb/s). Limiting the range of the Path Cost parameter to 1–200 000 000 ensures that the accumulated Path Cost cannot exceed 32 bits over a concatenation of 20 hops. In LANs where Bridges that use the recommended values defined in IEEE Std. 802.1D, 1998 Edition and Bridges that use the recommended values shown in this table are required to interwork, either the older Bridges will need to be re-configured in order to make use of the Path Cost values shown, or the new Bridges will need to be re-configured to make use of Path Cost values compatible with the values used by the older Bridges. The range of Path Costs that can be configured in an older Bridge is insufficient to accommodate the range of data rates available.

Annex A

(normative)

PICS Proforma[3]

A.3.4.2 Predicates

Add the following as item c) of A.3.4.2, renumbering subsequent items:

c) A predicate-name, for a predicate defined as a boolean expression constructed by combining item-references using the boolean operator AND: the value of the predicate is true if all of the items are marked as supported;

Change the tables of the PICS Proforma, and add new tables, as shown:

A.4 PICS proforma for ~~ISO/IEC 15802-3~~ IEEE Std 802.1D

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).	

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

A.4.2 Protocol summary, ~~ISO/IEC 15802-3~~ IEEE Std 802.1D

Identification of protocol specification	~~ISO/IEC 15802-3, Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Common specifications—Part 3:~~ IEEE Std 802.1D, Standards for Local and Metropolitan Area Networks: Media Access Control (MAC) Bridges
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 ~~ISO/IEC 15802-3~~ IEEE Std 802.1D.)	No [ ] Yes [ ]

Date of Statement	

A.5 Major capabilities and options

Item	Feature	Status	References	Support
(1a)	Communications Support Which Media Access Control types are supported on Bridge Ports, implemented in conformance with the relevant MAC standards?		6.5	
(1a.1)*	CSMA/CD, IEEE Std 802.3	O.1		Yes [ ] No [ ]
(1a.2)*	Token Bus, ~~ISO/IEC 8802-4~~ IEEE Std 802.4	O.1		Yes [ ] No [ ]
(1a.3)*	Token Ring, ~~ISO/IEC 8802-5~~ IEEE Std 802.5	O.1		Yes [ ] No [ ]
(1a.4)*	FDDI, ISO 9314-2	O.1		Yes [ ] No [ ]
(1a.5)*	DQDB, ~~ISO/IEC 8802-6~~ IEEE Std 802.6	O.1		Yes [ ] No [ ]
(1a.6)*	ISLAN, ~~ISO/IEC 8802-9~~ 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, ~~ISO/IEC 8802-12~~ IEEE Std 802.12 (IEEE Std 802.3 format)	O.1		Yes [ ] No [ ]
(1a.9)*	Demand Priority, ~~ISO/IEC 8802-12~~ IEEE Std 802.12 (ISO/IEC 8802-5 format)	O.1		Yes [ ] No [ ]
(1a.11)*	Wireless LAN, ~~ISO/IEC DIS 8802-11~~ IEEE Std 802.11	O.1		Yes [ ] No [ ]
(1b)	Is LLC Type 1 supported on all Bridge Ports in conformance with ~~ISO/IEC 8802-2~~ IEEE Std 802.2?	M	7.2, 7.3, 7.12, ~~ISO/IEC 8802-2~~ 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 Annex D shall also be completed).	O	Annex C	Yes [ ] No [ ]
(1d)*	Does the implementation support the use of the adminEdgePort and operEdgePort parameters on any Ports?	O	5.2, 14.8.2	Yes [ ] No [ ]
	State which Bridge Ports support the adminEdgePort and operEdgePort parameters			Ports__________

A.5 Major capabilities and options (*Continued*)

Item	Feature	Status	References	Support
(1e)	Does the implementation support the operation of the Bridge Detection State Machine on any Ports?	1d:M	5.2, 14.8.2, 18.2	Yes [ ] No [ ]
	State which Bridge Ports support the operation of the Bridge Detection State Machine			Ports__________
(2)	Relay and filtering of frames (A.6)	M	7.5, 7.6, 7.7	Yes [ ]
(2a)	Does the Bridge support Basic Filtering Services?	M	6.6.5, 7.7.2	Yes [ ]
(2b)*	Does the Bridge support Extended Filtering Services?	O	6.6.5, 7.7.2	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	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	6.6.5	Yes [ ] No [ ]
(2e)	Does the Bridge support expedited traffic classes on any of its Ports?	O	7.1.2, 7.7.3	Yes [ ] No [ ]
(4)*	Does the Bridge support management of the priority of relayed frames?	O	6.5, 7.5.1, 7.7.3, 7.7.5, Table 7-1, Table 7-2, Table 7-3	Yes [ ] No [ ]
(5)	Maintenance of filtering information (A.7)	M	7.8, 7.9	Yes [ ]
(7a)	Can the Filtering Database be read by management?	O	7.9	Yes [ ] No [ ]
(7c)*	Can Static Filtering Entries be created and deleted?	O	7.9.1	Yes [ ] No [ ]
(7g)	Can Static Filtering Entries be created and deleted in the Permanent Database?	O	7.9.6	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	7.9.1	Yes [ ] No [ ]
(7i)	Can Group Registration Entries be dynamically created, updated and deleted by GMRP?	2c:M	7.9.3, 10	Yes [ ] N/A [ ]
(10)	Addressing (A.8)	M	7.12	Yes [ ]
(9a)*	Can the Bridge be configured to use 48-bit Universal Addresses?	O.~~3~~2	7.12	Yes [ ] No [ ]
(9b)*	Can the Bridge be configured to use 48-bit Local Addresses?	O.~~3~~2	7.12	Yes [ ] No [ ]
(13)*	Spanning Tree algorithm and protocol (A.9)	~~M~~O.3	8, 9	Yes [ ] No [ ]
(rst)*	Rapid Spanning Tree algorithm and protocol (A.10)	O.3	9, 17	Yes [ ] No [ ]

A.5 Major capabilities and options (*Continued*)

Item	Feature	Status	References	Support
(both)	Support only one of Spanning Tree algorithm and Rapid Spanning Tree algorithm on all Ports of the Bridge at any given time	BOTH: M	5.2	Yes [ ] N/A [ ]
(edge)	Implement adminEdge and operEdge and the Bridge Detection state machine	rst:M	5.1, 17.13, IEEE Std 802.1t-2001 Clause 18	Yes [ ] N/A [ ]
(ptp)	Implement adminPointToPointMAC and operPointToPointMAC and associated MAC procedures	rst:M	6.4, 6.5	Yes [ ] N/A [ ]
(16)*	Does the Bridge support management of the Spanning Tree topology?	O	8.2, 17.2	Yes [ ] No [ ]
(17)*	Does the Bridge support management of the protocol timers?	O	8.10, 17.28	Yes [ ] No [ ]
(19)*	Bridge Management Operations	O	14	Yes [ ] No [ ]
(20a)*	Are the Bridge Management Operations supported via a Remote Management Protocol?	19:O.4	5	Yes [ ] No [ ] N/A [ ]
(20b)*	Are the Bridge Management Operations supported via a local management interface?	19:O.4	5	Yes [ ] No [ ] N/A [ ]

Predicates:
BOTH = 13 AND rst

A.6 Relay and filtering of frames

Item	Feature	Status	References	Support
(2f)	Are received frames with media access method errors discarded?	M	6.4, 7.5	Yes [ ]
(2g)	Are correctly received frames submitted to the Learning Process?	M	7.5	Yes [ ]
(2h)	Are user data frames the only type of frame relayed?	M	7.5	Yes [ ]
(2i)	Are request with no response frames the only frames relayed?	M	7.5	Yes [ ]
(2j)	Are all frames addressed to the Bridge Protocol Entity submitted to it?	M	7.5	Yes [ ]
(2k)	Are user data frames the only type of frame transmitted?	M	7.6	Yes [ ]
(2l)	Are request with no response frames the only frames transmitted?	M	7.6	Yes [ ]
(2m)	Are relayed frames queued for transmission only under the conditions in 7.7.3?	M	7.7.3, 8.4	Yes [ ]
(2n)	Is the order of relayed frames preserved in accordance with the requirements of the forwarding process?	M	7.7.3, 7.1.1	Yes [ ]
(2o)	Is a relayed frame submitted to a MAC Entity for transmission only once?	M	7.7.4, 6.3.4	Yes [ ]

A.6 Relay and filtering of frames (*Continued*)

Item	Feature	Status	References	Support
(2p)	Is a maximum bridge transit delay enforced for relayed frames?	M	7.7.3	Yes [ ]
(2q)	Are queued frames discarded if a Port leaves the Forwarding State?	M	7.7.3	Yes [ ]
(2r)	Is the user priority of relayed frames preserved where possible?	M	6.4	Yes [ ]
(2s)	Is the user priority set to the Default User Priority for the reception Port otherwise?	M	6.4	Yes [ ]
(2t)	Is the user priority regenerated by means of the User Priority Regeneration Table?	M	7.5.1, Table 7-1	Yes [ ]
(2u)	Is mapping of Regenerated User Priority to Traffic Class performed by means of the Traffic Class Table?	M	7.7.3, Table 7-2	Yes [ ]
(2v)	Is the access priority derived from the Regenerated User Priority as defined by the values in Table 7-3 for each outbound media access method supported by the Bridge?	M	7.7.5, Table 7-3	Yes [ ]
(2w)	Does the implementation introduce an undetected frame error rate greater than that achievable by preserving the FCS?	X	7.7.6, 6.3.7	No [ ]
(2x)	Is the FCS of frames relayed between Ports of the same MAC type preserved?	O	7.7.6	Yes [ ] No [ ]
(2y)	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	6.5.4, ISO 9314-2	No [ ] N/A [ ]
(2z)	Is only Asynchronous service used?	MS1:M	ISO 9314-2 Clause 8.1.4	Yes [ ] N/A [ ]
(2aa)	On receiving a frame from an FDDI ring for forwarding, does the bridge set the C indicator?	MS1:O	6.5.4, ISO 9314-2 Clause 7.3.8	Yes [ ] No [ ] N/A [ ]
(2ab)	On receiving a frame from an FDDI ring for forwarding, does the bridge leave the C indicator unaltered?	MS1:O	6.5.4, ISO 9314-2 Clause 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.5	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 7-1?	4:O.5	7.5.1, Table 7-1	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 7-2?	MS2:O	7.7.3, Table 7-2	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	6.4	Yes [ ]

A.6 Relay and filtering of frames (*Continued*)

Item	Feature	Status	References	Support
(4e)	Does the Bridge support the recommended default mappings between received user priority and Regenerated User Priority for each Port as defined in Table 7-1?	¬ 4:M	7.5.1, Table 7-1	Yes [ ]
(4f)	Does the Bridge support the recommended default user_priority to traffic class mappings shown in Table 7-2 for each Port?	MS3:M	7.7.3, Table 7-2	Yes [ ] N/A [ ]
(4g)	Is the Bridge able to use any values other than those shown in Table 7-3 when determining the access priority for the media access methods shown?	X	7.7.5, Table 7-3	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 OR 1a.9)
MS2 = 2d AND 4
MS3 = 2d 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	7.8, 7.9.2, 8.4	Yes [ ]
(5b)	Are Dynamic Filtering Entries created on receipt of frames with a group source address?	X	7.8, 7.9.2	No [ ]
(5c)	Does the Filtering Database support Static Filtering Entries?	M	7.9.1	Yes [ ]
(5d)	Can a Dynamic Filtering Entry be created that conflicts with an existing Static Filtering Entry?	X	7.8, 7.9, 7.9.1, 7.9.2	No [ ]
(5e)	Does the Filtering Database support Dynamic Filtering Entries?	M	7.9.2	Yes [ ]
(5f)	Does the creation of a Static Filtering Entry remove any conflicting information in a Dynamic Filtering Entry for the same address?	M	7.9.1, 7.9.2	Yes [ ]
(5g)	Does each Static Filtering Entry specify a MAC Address specification and a Port Map?	M	7.9.1	Yes [ ]
(5h)	Are Dynamic Filtering Entries removed from the Filtering Database if not updated for the Ageing Time period?	M	7.9.2	Yes [ ]
(5i)	Does each Dynamic Filtering Entry specify a MAC Address specification and a Port Map?	M	7.9.2	Yes [ ]
(5j)	Is the Filtering Database initialized with the entries contained in the Permanent Database?	M	7.9.6	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	7.9.3	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	7.9.3	Yes [ ]
(5m)	Are Group Registration Entries created, updated and removed from the Filtering Database in accordance with the specification of GMRP?	2c:M	7.9.3, 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	7.9.3, 10	No [ ]
(6a)	State the Filtering Database Size.	M	7.9	____ entries
(6b)	State the Permanent Database Size.	M	7.9	____ 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	7.9.1	Yes [ ]
(7e)	Can Static Filtering Entries be made for group MAC Addresses?	**7c:**M	7.9.1	Yes [ ]
(7f)	Can a Static Filtering Entry be made for the broadcast MAC Address?	**7c:**M	7.9.1	Yes [ ]
(8a)	Can the Bridge be configured to use the default value of Ageing Time recommended in Table 7-4?	O	7.9.2, Table 7-4	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	7.9.2, Table 7-4	Yes [ ] No [ ]

A.8 Addressing

Item	Feature	Status	References	Support
(10a)	Does each Port have a separate MAC Address?	M	7.12.2	Yes [ ]
(10b)	Are all BPDUs transmitted to the same group address?	M	7.12.3, 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	7.12.3, 8.2	Yes [ ]
(10d)	Is the source address of BPDUs the address of the transmitting Port?	**9a:**M	7.12.3	Yes [ ]
(10d1)	Is the LLC address of BPDUs the standard LLC address identified for the Spanning Tree Protocol?	M	7.12.3, Table 7-8	Yes [ ]
(10e)	Is the Bridge Address a Universal Address?	M	7.12.5, 8.2	Yes [ ]
(10f)	Are frames addressed to any of the Reserved Addresses relayed by the Bridge?	X	7.12.6	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	7.12.4	Yes [ ] No [ ]
(11b)	Is Bridge Management accessible through all Ports using the All LANs Bridge Management Group Address?	13:O	7.12.4	Yes [ ] No [ ]
(11c)	Is the Bridge Address the Address of Port 1?	**9a:**O	7.12.5	Yes [ ] No [ ] N/A [ ]
(11d)	Are Group Addresses additional to the Reserved Addresses preconfigured in the Permanent Database?	O	7.12.6	Yes [ ] No [ ]
	If item (11d) is not supported, mark N/A and continue at item (12a)			N/A [ ]
(11e)	Can the additional preconfigured entries in the Filtering Database be deleted?	**11d:**O	7.12.6	Yes [ ] No [ ]
(12a)	Can a group MAC Address be assigned to identify the Bridge Protocol Entity?	**9b:**M	8.2	Yes [ ] N/A [ ]
(12c)	Does each Port of the Bridge have a distinct identifier?	M	8.2, 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?	M	8.5.3	Yes [ ]
	Designated Root		8.5.3.1	
	Root Cost		8.5.3.2	
	Root Port		8.5.33	
	Max Age		8.5.3.4	
	Hello Time		8.5.3.5	
	Forward Delay		8.5.3.6	
	Bridge Identifier		8.5.3.7	
	Bridge Max Age		8.5.3.8	
	Bridge Hello Time		8.5.3.9	
	Bridge Forward Delay		8.5.3.10	
	Topology Change Detected		8.5.3.11	
	Topology Change		8.5.3.12	
	Topology Change Time		8.5.3.13	
	Hold Time		8.5.3.14	
(13b)	Are all the following Bridge Timers maintained?	M	8.5.4	Yes [ ]
	Hello Timer		8.5.4.1	
	Topology Change Notification Timer		8.5.4.2	
	Topology Change Timer		8.5.4.3	
(13c)	Are all the following Port Parameters maintained for each Port?	M	8.5.5	Yes [ ]
	Port Identifier		8.5.5.1	
	State		8.5.5.2, 8.4	
	Path Cost		8.5.5.3	
	Designated Root		8.5.5.4	
	Designated Cost		8.5.5.5	
	Designated Bridge		8.5.5.6	
	Designated Port		8.5.5.7	
	Topology Change Acknowledge		8.5.5.8	
	Configuration Pending		8.5.5.9	
	Change Detection Enabled		8.5.5.10	
(13d)	Are all the following Timers maintained for each Port?	M	8.5.6	Yes [ ]
	Message Age Timer		8.5.6.1	
	Forward Delay Timer		8.5.6.2	
	Hold Timer		8.5.6.3	
(13e)	Are Protocol Parameters and Timers maintained, and BPDUs transmitted, as required on each of the following events?	M	8.7, 8.9, 8.5.3, 8.5.4, 8.5.5, 8.5.6	Yes [ ]
	Received Configuration BPDU		8.7.1	

A.9 Spanning Tree algorithm (*Continued*)

Item	Feature	Status	References	Support
	Received Topology Change Notification BPDU		8.7.2	
	Hello Timer Expiry		8.7.3	
	Message Age Timer Expiry		8.7.4	
	Forward Delay Timer Expiry		8.7.5	
	Topology Change Notification Timer Expiry		8.7.6	
	Topology Change Timer Expiry		8.7.7	
	Hold Timer Expiry		8.7.8	
(13f)	Do the following operations modify Protocol Parameters and Timers, and transmit BPDUs as required?	M	8.8, 8.9, 8.5.3, 8.5.4, 8.5.5, 8.5.6	Yes [ ]
	Initialization		8.8.1	
	Enable Port		8.8.2	
	Disable Port		8.8.3	
	Set Bridge Priority		8.8.4	
	Set Port Priority		8.8.5	
	Set Path Cost		8.8.6	
(13g)	Does the implementation support the ability to set the value of the Change Detection Enabled parameter to Disabled?	O	8.5.5.10	Yes [ ] No [ ]
(14a)	Does the Bridge underestimate the increment to the Message Age parameter in transmitted BPDUs?	X	8.10.1	No [ ]
(14b)	Does the Bridge underestimate Forward Delay?	X	8.10.1	No [ ]
(14c)	Does the Bridge overestimate the Hello Time interval?	X	8.10.1	No [ ]
(15a)	Does the Bridge use the specified value for Hold Time?	M	8.10.2, Table 8-3	Yes [ ]
(15b)	As a default behavior, is the Path Cost for a Port unaffected by any dynamic changes in the Port's data rate?	M	8.10.2	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?	**16:**M	8.2, 8.5.3.7, 8.8.4	Yes [ ]
(16b)	Can the relative priority of the Ports be set?	**16:**M	8.2, 8.5.5.1, 8.8.5	Yes [ ]
(16c)	Can the path cost for each Port be set?	**16:**M	8.2, 8.5.5.3, 8.8.6	Yes [ ]
	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?	**17:**M	8.10.2, 8.5.3.8, Table 8-3	Yes [ ]
(17b)	Can Bridge Hello Time be set to any of the range of values specified?	**17:**M	8.10.2, 8.5.3.9, Table 8-3	Yes [ ]
(17c)	Can Bridge Forward Delay be set to any of the range of values specified?	**17:**M	8.10.2, 8.5.3.10, Table 8-3	Yes [ ]

A.9 Spanning Tree algorithm (*Continued*)

Item	Feature	Status	References	Support
(18a)	Do all BPDUs contain an integral number of octets?	M	9.1.1	Yes [ ]
(18b)	Are all the following BPDU parameter types encoded as specified?	M	9.1.1, 9.2	Yes [ ]
	Protocol Identifiers		9.2.1	
	Protocol Version Identifiers		9.2.2	
	BPDU Types		9.2.3	
	Flags		9.2.4	
	Bridge Identifiers		9.2.5	
	Root Path Cost		9.2.6	
	Port Identifiers		9.2.7	
	Timer Values		9.2.8	
(18c)	Do Configuration BPDUs have the format, parameters and parameter values specified?	M	9.3.1	Yes [ ]
(18d)	Do Topology Change Notification BPDUs have the format, parameters and parameter values specified?	M	9.3.2	Yes [ ]
(18e)	Are received BPDUs validated as specified?	M	9.3.4	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	M	17.2	Yes [ ]
(par1)	Not exceed the values in 17.28.2 for max Bridge transit delay, max message age increment overestimate and max BPDU transmission delay	M	5.1, 17.28.2	Yes [ ]
(par2)	Use the value given in Table 17-5 for Transmission Limit	M	5.1, Table 17-5	Yes [ ]
(inc)	Inclusion of active Ports in computation of the active topology	M	17.5	Yes [ ]
(pro)	Processing of BPDUs received on Ports included in the computation of the active topology	M	17.5	Yes [ ]
(dis)	Discarding received frames in the Discarding state	M	17.5	Yes [ ]
(lrn)	Incorporating station location information to the Filtering Database in the Learning and Forwarding states	M	17.5	Yes [ ]
(nlrn)	Not incorporating station location information to the Filtering Database in the Discarding state	M	17.5	Yes [ ]
(rlrn)	Transfer learned MAC addresses from a retiring Root Port to a new Root Port	O	17.10	Yes [ ] No [ ]

A.10 Rapid Spanning Tree algorithm (*Continued*)

Item	Feature	Status	References	Support
(sm)	A single instance of the Port Role Selection state machine per Bridge and an instance of all other state machines per Port	M	17.13	Yes [ ]
(ptmr)	Port Timers state machine support	M	17.15, 17.20	Yes [ ]
(pism)	Port Information state machine support	M	17.15, 17.21	Yes [ ]
(prssm)	Port Role Selection state machine support	M	17.15, 17.22	Yes [ ]
(prtsm)	Port Role Transitions state machine support	M	17.15, 17.23	Yes [ ]
(pstsm)	Port State Transition state machine support	M	17.15, 17.24	Yes [ ]
(tcsm)	Topology Change state machine support	M	17.15, 17.25	Yes [ ]
(ppmsm)	Port Protocol Migration state machine support	M	17.15, 17.26	Yes [ ]
(ptsm)	Port Transmit state machine support	M	17.15, 17.27	Yes [ ]
(cde)	Not support Change Detection Enabled parameter	M	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.	M	17.28.1	Yes [ ]
(htim)	Use of Transmission Limit	M	Table 17-5	Yes [ ]
(prel)	Enforcement of parameter relationships	M	17.28.2	Yes [ ]
(pcst)	No defaulting to use of automatic path cost changes	M	17.28.2	Yes [ ]
(prv)	Range and granularity of priority values	M	17.28.2	Yes [ ]
(pcv)	Range and granularity of path cost values	M	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?	**16:**M	17.2, 17.4, 17.13	Yes [ ]
(mgt2)	Can the relative priority of the Ports be set?	**16:**M	17.2, 17.4, 17.13	Yes [ ]
(mgt3)	Can the path cost for each Port be set?	**16:**M	17.2, 17.4, 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?	**17:**M	17.2, 17.13, Table 17-5	Yes [ ]
(tmr2)	Can Bridge Hello Time be set to any of the range of values specified?	**17:**M	17.2, 17.13, Table 17-5	Yes [ ]
(tmr3)	Can Bridge Forward Delay be set to any of the range of values specified?	**17:**M	17.2, 17.13, Table 17-5	Yes [ ]
*(pdu1)	Do all BPDUs contain an integral number of octets?	M	9.1.1	Yes [ ]
(pdu2)	Are all the following BPDU parameter types encoded as specified?	M	9.1.1, 9.2	Yes [ ]
	Protocol Identifiers		9.2.1	
	Protocol Version Identifiers		9.2.2	
	BPDU Types		9.2.3	

A.10 Rapid Spanning Tree algorithm (*Continued*)

Item	Feature	Status	References	Support
	Flags		9.2.4	
	Bridge Identifiers		9.2.5	
	Root Path Cost		9.2.6	
	Port Identifiers		9.2.7	
	Timer Values		9.2.8	
(pdu3)	Do Configuration BPDUs have the format, parameters, and parameter values specified?	M	9.3.1	Yes [ ]
(pdu4)	Do Topology Change Notification BPDUs have the format, parameters, and parameter values specified?	M	9.3.2	Yes [ ]
(pdu5)	Do Rapid Spanning Tree BPDUs have the format, parameters, and parameter values specified?	M	9.3.3	Yes [ ]
(pdu6)	Are received BPDUs validated as specified?	M	9.3.4	Yes [ ]

A.11 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	14.4.1.1	Yes [ ]
(19b)	Read Bridge	**19:**M	14.4.1.2	Yes [ ]
(19c)	Set Bridge Name	**19:**M	14.4.1.3	Yes [ ]
(19d)	Reset Bridge	**19:**M	14.4.1.4	Yes [ ]
(19e)	Read Port	**19:**M	14.4.2.1	Yes [ ]
(19f)	Set Port Name	**19:**M	14.4.2.2	Yes [ ]
(19g)	Read Forwarding Port Counters	**19:**M	14.6.1.1	Yes [ ]
(19h)	Read Filtering Database	**19:**M	14.7.1.1	Yes [ ]
(19i)	Set Filtering Database Ageing Time	**19:**M	14.7.1.2	Yes [ ]
(19j)	Read Permanent Database	**19:**M	14.7.5.1	Yes [ ]
(19k)	Create Filtering Entry	**19:**M	14.7.6.1	Yes [ ]
(19l)	Delete Filtering Entry	**19:**M	14.7.6.2	Yes [ ]
(19m)	Read Filtering Entry	**19:**M	14.7.6.3	Yes [ ]
(19n)	Read Filtering Entry Range	**19:**M	14.7.6.4	Yes [ ]
(19o)	Read Bridge Protocol Parameters	**19:**M	14.8.1.1	Yes [ ]
(19p)	Set Bridge Protocol Parameters	**19:**M	14.8.1.2	Yes [ ]
(19q)	Read Port Parameters	**19:**M	14.8.2.1	Yes [ ]
(19r)	Force Port State	**19:**M	14.8.2.2	Yes [ ]
(19s)	Set Port Parameters	**19:**M	14.8.2.3	Yes [ ]
(19t)	Read Port Default User Priority	MS4:M	14.6.2.1	Yes [ ] N/A [ ]
(19u)	Set Port Default User Priority	MS4:M	14.6.2.2	Yes [ ] N/A [ ]
(19v)	Read Port User Priority Regeneration Table	MS5:M	14.6.2.3	Yes [ ] N/A [ ]

A.11 Bridge management (*Continued*)

Item	Feature	Status	References	Support
(19w)	Set Port User Priority Regeneration Table	MS5:M	14.6.2.4	Yes [ ] N/A [ ]
(19x)	Read Port Traffic Class Table	MS7:M	14.6.3.1	Yes [ ] N/A [ ]
(19y)	Set Port Traffic Class Table	MS7:M	14.6.3.2	Yes [ ] N/A [ ]
(19z)	Read Outbound Access Priority Table	MS6:M	14.6.3.3	Yes [ ] N/A [ ]
(19aa)	Read GARP Timers	MS8:M	14.9.1.1	Yes [ ] N/A [ ]
(19ab)	Set GARP Timers	MS8:M	14.9.1.2	Yes [ ] N/A [ ]
(19ac)	Read GARP Applicant Controls	MS8:M	14.9.2.1	Yes [ ] N/A [ ]
(19ad)	Set GARP Applicant Controls	MS8:M	14.9.2.2	Yes [ ] N/A [ ]
(19ae)	Read GARP State	MS8:M	14.9.3.1	Yes [ ] N/A [ ]
(19af)	Force BPDU Migration Check	MS9:M	14.8.2.4	Yes [ ] N/A [ ]
	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	5.	
(20d)	What standard(s) or specifications for Managed Objects and Encodings are supported?	**20a:**M	5.	
	If item (20b) is not supported, mark N/A and continue at A.12			N/A [ ]
(20e)	What specification of the local management interface is supported?	**20b:**M	5.	

Predicates:
MS4=19 AND 4a
MS5=19 AND 4b
MS6=19 AND 4
MS7=19 AND 4c
MS8=19 AND 2b
MS9=RST AND 19

A.12 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	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	16.2	

Guaranteed Bridge Relaying Rate	TR
_ _ _ _ _ _ _ _ _ _ _ frames per second	_ _ _ _ _ _ second(s)

A.12 Performance *(Continued)*

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.13 GARP and GMRP

Item	Feature	Status	References	Support
	If Item 2b is not supported, mark N/A and continue at item (22j).			N/A [ ]
(22a)	Is the GMRP Application address used as the destination MAC Address in all GMRP protocol exchanges?	2b:M	10.4.1, 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	12.4, 12.5, Table 7-8	Yes [ ]
(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	10.3.1, 12.4, 12.5, 12.11	Yes [ ]
(22d)	Does the implementation support the operation of the Applicant, Registrar, and Leave All state machines?	2b:M	12.8, 13	Yes [ ]
(22e)	Does the Bridge propagate GMRP registration information only on Ports that are part of the active topology for the Base Spanning Tree Context?	2b:M	12.3.3, 12.3.4, 13	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	7.12.3, 12.5	Yes [ ]
(22g)	Does the GMRP application operate as defined in Clause 10?	2b:M	10, 10.3	Yes [ ]
(22h)	Are received GARP PDUs that are not well formed for the GARP Applications supported, discarded?	2b:M	10.3.1, 12.4, 12.5, 12.10, 12.11	Yes [ ]

A.13 GARP and GMRP (*Continued*)

Item	Feature	Status	References	Support
(22i)	Does the implementation support the use of the Restricted Group Registration parameter for each Port?	2b:O	5.2, 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	7.12.3, 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	7.12.3, 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	7.12.3, 12.5	No [ ]
(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	7.12.3, 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	7.12.3, 12.5	Yes [ ]
22o	Are all GARP PDUs received on disabled Ports discarded?	M	12.2	Yes [ ]

Annex B

(informative)

Calculating Spanning Tree parameters

Change the contents of Annex B as shown.

This annex describes the method and rationale for calculating the recommended values and operational ranges for the essential Spanning Tree Algorithm performance parameters.

This material was originally developed in order to provide a rationale for determining the parameter values needed for the correct operation of the Spanning Tree Algorithm and Protocol (STP in Clause 8). As the operation of RSTP (Clause 17) is largely insensitive to the choice of timer values, these values are used by RSTP as a “backstop” to the normal operation of the protocol; i.e., to ensure correct protocol operation in the face of exception conditions caused by lost messages or failure to detect hardware state changes, and to allow seamless integration of STP and RSTP Bridges in the same Bridged LAN.

B.1 Overview

The calculation is described in a number of steps. Each of these steps establishes values for a number of the parameters that are then used as the basis for the following steps.

The description and equations given are pertinent to a homogeneous Bridged LAN, i.e., one in which all the individual LANs and Bridges are of the same type and speed. It is easy to extend this for a heterogeneous Bridged LAN.

The explanation is illustrated by recommended values for IEEE 802 LANs. All times are given in seconds.

B.2 Abbreviations and special symbols

dia	**maximum bridge diameter**
life	maximum **frame lifetime**
t_d	average frame **transit delay**
ma_d	average **medium access delay**
mma_d	**maximum medium access delay**
bt_d	**maximum bridge transit delay**
time_unit	the resolution of **Message Age**
msg_aio	**maximum Message Age increment overestimate**
msg_ao	**maximum Message Age overestimate**
pdu_d	**maximum BPDU transmission delay**
lost_msgs	maximum number of lost Bridge Protocol Messages to be tolerated prior to reconfiguration
msg_prop	**maximum Bridge Protocol Message propagation time**
hello_t	**Hello Time**
hold_t	**Hold Time**
max_age	**Max Age**
fwd_delay	**Forward Delay**

B.3 Calculation

B.3.1 Lifetime, diameter, and transit delay

B.3.1.1 Step

Choose the maximum bridge diameter for the Bridged LAN and the maximum bridge transit delay. Note that, where the individual LANs support a range of transmission priorities, the bridge transit delay may vary according to priority.

B.3.1.2 Basis of choice

The frame lifetime is equal to the maximum bridge diameter times the maximum bridge transit delay plus the maximum medium access delay for the initial transmission, i.e.,

$$life = (dia \times bt\_d) + mma\_d \tag{1}$$

The average **frame transit delay** between end systems in a Bridged LAN is greater than that experienced in a single LAN by the sum of the average **forwarding delays** and **frame transmission delays** of Bridges in the path between the end systems. These will be of the order of the **medium access delays** for lightly loaded LANs. So for systems at the extremities of the Bridged LAN there will be the following:

$$t\_d >= (dia \times ma\_d) + ma\_d \tag{2}$$

This bounds any enthusiasm for insisting on low **maximum bridge transit delays** and **high maximum bridge diameters**.

B.3.1.3 Recommended values for IEEE 802 Bridged LANs

$$\begin{aligned} mma\_d &>= 0.5 \\ life &<= 7.5 \\ dia &= 7 \\ bt\_d &= 1.0 \end{aligned}$$

B.3.2 Transmission of BPDUs

B.3.2.1 Step

Select the transmission priority for BPDUs and a value for the **maximum BPDU transmission delay**.

B.3.2.2 Basis of choice

In general, a high transmission priority will be chosen, since the continued operation of the Bridged LAN depends on the successful transmission and reception of BPDUs. In some cases, other traffic native to an individual LAN may be more important.

The lowest value that could be chosen for the **maximum BPDU transmission delay** then is the **maximum medium access delay** for frames of that priority. In recognition of implementation difficulties that may arise in trying to achieve this figure, it seems more reasonable to choose the value to be equal to the **maximum bridge transit delay** for frames transmitted with that priority.

$$pdu\_d = bt\_d \tag{3}$$

B.3.2.3 Recommended values for IEEE 802 Bridged LANs

Priority transmission is not available for all IEEE 802 media access methods. Therefore, the following has been selected:

$$\begin{aligned} pdu\_d &= bt\_d \\ &= 1.0 \end{aligned}$$

Where priority transmission is available in the media access method concerned, it is recommended that the highest available transmission priority is used.

B.3.3 Accuracy of message age

B.3.3.1 Step

Select an appropriate value for the **maximum Message Age increment overestimate**.

This is the maximum overestimate of the increment made to the value of the Message Age parameter in transmitted Bridge Protocol Data Units. This parameter allows a Bridge receiving a Protocol Message to discard the information in it when it becomes too old. The transmitting Bridge should not be allowed to underestimate the value of this field.

Calculate the value of the **maximum Message Age overestimate**, which is the maximum overestimate any Bridge can make of the age of received Bridge Protocol Message information.

B.3.3.2 Basis of choice

The choice of **maximum Message Age increment overestimate** is governed by the following:

a) *time_unit*—the resolution with which the Message Age parameter is carried in Configuration Messages.

b) The granularity and accuracy of timers in the Bridge.

c) The **maximum BPDU transmission delay**.

Assuming the Bridge timers are not necessarily synchronized with received BPDUs, that they are accurate, and that they have a granularity of *time_unit*, there will be, as a best effort, the following:

$$msg\_aio = pdu\_d + time\_unit \tag{4}$$

NOTE—As time_unit is small, this term in Equation (4) has been approximated to zero in the recommended values of msg_aio and msg_ao shown in B.3.3.3.

This value should be rounded up to the nearest multiple of *time_unit*. It is worth noting here that any Bridge will always increment the value by at least one unit.

Making the same allowance for the timers in a Bridge receiving and storing Bridge Protocol Message information, the **maximum Message Age overestimate** will be equal to the **maximum Message Age increment overestimate** times the **maximum bridge diameter** minus one:

$$msg\_ao = msg\_aio \times (dia - 1) \tag{5}$$

B.3.3.3 Recommended values for IEEE 802 Bridged LANs

$msg\_aio = 1.0$
$msg\_ao = 6.0$

B.3.4 Hello time

B.3.4.1 Step

Provisionally select a value for the Hello Time.

B.3.4.2 Basis of choice

The choice of Hello Time is made with regard to its contribution to the maximum Bridge Protocol Message propagation time (see next step).

There is no point in transmitting Bridge Protocol Messages at intervals more frequent than the **maximum BPDU transmission delay**. In the worst case, where there is an attempt to guarantee correct operation, these messages would just run into one another.

A provisional value of twice the **maximum BPDU transmission delay** is suggested.

$$hello\_t = 2 \times pdu\_d \tag{6}$$

B.3.4.3 Recommended values for IEEE 802 Bridged LANs

$hello\_t = 2.0$

B.3.5 Bridge protocol message propagation time

B.3.5.1 Step

Calculate the **maximum Bridge Protocol Message propagation time**.

B.3.5.2 Basis of choice

The **maximum Bridge Protocol Message propagation time** is the maximum time taken for a Bridge Protocol Message information to cross the Bridged LAN, from Bridge to Bridge. This is composed of the following components:

a) The maximum propagation time for a single Bridge Protocol Message to cross the Bridged LAN, i.e., **maximum BPDU transmission delay** times the **maximum bridge diameter** minus one.

b) An allowance of **Hello Time** times the maximum number of consecutive lost Bridge Protocol Messages to be tolerated (note that losing even a single message should be a rare occurrence).

c) A further allowance of **Hello Time**, since we should not assume synchronization with the Root Bridge, and we may have to wait that long for it to transmit the next BPDU.

$$msg\_prop =((lost\_msgs + 1) \times hello\_t) + pdu\_d \times (\mathrm{dia}- 1) \tag{7}$$

B.3.5.3 Recommended values for IEEE 802 Bridged LANs

Assuming *lost_msgs* = 3,

msg_prop = 14.0

B.3.6 Hold time

B.3.6.1 Step

Select a value for **Hold Time**.

B.3.6.2 Basis of choice

If **Hold Time** is greater than the **maximum BPDU transmission delay**, then the **Maximum Bridge Protocol Message propagation time** will be set, in the worst scenario, by a delay of **Hold Time** at each Bridge rather than by a delay of **maximum BPDU transmission delay**. This would invalidate the conclusion in B.3.5, above. Therefore, the following has been chosen:

$$hold\_t \leq pdu\_d \tag{8}$$

B.3.6.3 Recommended values for IEEE 802 Bridged LANs

~~*Not more than 3 BPDUs transmitted in any Hello Time interval*~~ *Not more than TxHoldCount (17.16.6) BPDUs transmitted in any HelloTime (17.16.3) interval*

B.3.7 Max age

B.3.7.1 Step

Calculate the lower limit for **Max Age** for the Bridged LAN.

B.3.7.2 Basis of choice

Under stable conditions (i.e., no failure, removal or insertion of Bridges and other LAN components), Bridges on the periphery of Bridged LAN must not time out the Root. To do so would result in temporary local denial of service.

This means that **Max Age** must be adequate to cope with the worst-case propagation delays and Protocol Message Age inaccuracies as follows.

If at any time a Bridge is depending on Protocol Message information whose age has been maximally overestimated, then the sum of

a) The interval between the transmission of the next Protocol Message that it receives from the Root and the original transmission of the Protocol Message information it is currently using,

b) The overestimate of the Age of the current information, and

c) The propagation time of the next Protocol Message to be received

must be less than Max Age, or the Bridge will timeout the Protocol Message information and attempt to become the Root itself.

$$
\begin{aligned}
max\_age &= ((lost\_msgs + 1) \times hello\_t) + msg\_ao + (pdu\_d \times (\mathrm{dia}-1)) \\
&= msg\_ao + msg\_prop
\end{aligned}
$$

B.3.7.3 Recommended values for IEEE 802 Bridged LANs

$max\_age = 20.0$

B.3.8 Forward delay

B.3.8.1 Step

Calculate the **Forward Delay**.

B.3.8.2 Basis of choice

When the Forward Delay Timer for a Port expires and the Bridge starts forwarding received frames on that Port, it must be determined that there are no longer any frames in the system that were being forwarded on the previous active topology. If there are, then there is a risk of duplicating frames or, if remnants of the old active topology still exist while the new topology is being established, of creating data loops.

So the Listening and Learning periods during which the Forward Delay Timer runs must cover the following consecutive periods:

a) From the first Bridge Port entering the Listening State (and staying there through the subsequent reconfiguration) to the last Bridge in the Bridged LAN hearing of the change in **active topology**.

b) For the last Bridge to stop the forwarding of frames received on the previous topology and for the last frame so forwarded to disappear.

In a), above, there may be a difference of up to **maximum Message Age overestimate** in the times at which Bridges timeout old Root information and are prepared to become or listen to a new Root. Following this, it can take **maximum Bridge Protocol Message propagation time** for the news of the new topology to propagate from the new Root to all Bridges.

For b), above, the time to stop forwarding will be the **maximum transmission halt delay**, which is bounded by the maximum bridge transit delay (for all priorities); subsequently, the frame will disappear within the frame lifetime.

So there is the following:

$$2 \times fwd\_d >= msg\_ao + msg\_prop + bt\_d + life \qquad (9)$$

B.3.8.3 Recommended values for IEEE 802 Bridged LANs

$fwd\_d = 15.0$

B.4 Selection of parameter ranges

B.4.1 Absolute maximum values

It might be desirable to configure a LAN or Bridge with a greater **maximum medium access delay** than assumed in the calculations for recommended values above. This could be a consequence of the type of traffic carried by the LAN or particular aspects of a Bridge implementation, designed to maximize the throughput, for example. However, it is highly desirable that absolute maximum values of **maximum bridge transit delay**, **maximum BPDU transmission delay**, and **maximum Message Age increment overestimate** be mandated by this standard in order to provide for interoperability.

A Bridge operating with absolute maximum values of these parameters should be configurable with Bridges employing recommended values in a Bridged LAN of a **bridge diameter** of at least 3. This criterion is met by the following:

$$\mathit{bt\_d} <= 2.0$$
$$\mathit{pdu\_d} <= 2.0$$
$$\mathit{msg\_aio} <= 2.0$$

These limits are believed to encompass the requirement for parameter values greater than those recommended in B.3.

B.4.2 Hold time

There is no benefit in reducing **Hold Time** below the recommended value of **maximum BPDU transmission delay**. Nor would any purpose be served, in terms of reduced use of bandwidth or processing capability in a Bridge, by increasing **Hold Time**. It is, therefore, appropriate to fix the value of this parameter as a constant:

~~*hold_t = 1.0*~~ *Not more than TxHoldCount (17.16.6) BPDUs transmitted in any HelloTime (17.16.3) interval* *(10)*

B.4.3 Range of hello time

There is no requirement for **HelloTime** to be less than **Hold Time**. Similarly, no purpose would be served by setting **Hello Time** to more than twice the absolute maximum value for **maximum BPDU transmission delay**. Therefore the following have been chosen:

$$1.0 <= \mathit{hello\_t} <= 4.0$$

B.4.4 Maximum required values of max age and forward delay

The maximum required values for **Max Age** and **Forward Delay** are calculated using the equations of B.3 with the following parameter values:

$$\mathit{dia} = 7$$
$$\mathit{mma\_d} <= 2.0$$
$$\mathit{bt\_d} = 2.0$$
$$\mathit{pdu\_d} = 2.0$$
$$\mathit{msg\_aio} = 2.0$$
$$\mathit{hello\_t} = 4.0$$
$$\mathit{lost\_msgs} = 3$$

which gives the following:

$$\mathit{max\_age} = 40.0$$
$$\mathit{fwd\_delay} = 30.0$$

Although these are believed to be the maximum values required, there is no desire to prevent greater values being used.

B.4.5 Minimum values for max age and forward delay

The minimum realistic values for **Max Age** and **Forward Delay** are calculated using the equations of B.3 with the following parameter values:

dia= 2
mma_d<= 0.5
bt_d= 0.5
pdu_d= 0.5
~~*hold_t = 1.0*~~ *Not more than TxHoldCount (17.16.6) BPDUs transmitted in any HelloTime (17.16.3) interval*
msg_aio= 1.0
hello_t= 1.0
lost_msgs= 3

which gives the following:

max_age<= 6.0
fwd_delay= 4.0

It is suggested that Bridge implementations do not permit lower values of **Max Age** and **Forward Delay** to be used in order to guard against absurd settings.

B.4.6 Relationship between max age and forward delay

In order to further guard against bad settings of parameters that affect the correctness of operation of the Spanning Tree Algorithm and Protocol, it is suggested that Bridges enforce the relationship between **Max Age** and **Forward Delay** given in B.3.8 by ensuring that

$$2 \times (\mathit{fwd\_delay} - 1.0) >= \mathit{max\_age}$$

Annex F

(informative)

Target topology, migration, and interoperability

Insert a new top-level subclause heading "F.1 GARP, GMRP and Extended Filtering Services" to encompass all existing material in this annex, renumbering all existing subclauses as "F.1.XXX". For example, "F.1 Target Topology" becomes "F.1.1 Target Topology", etc.

Insert a second top-level subclause, as follows:

F.2 RSTP considerations

F.2.1 Overview of protocol changes

The specification of RSTP as it appears in Clause 17 is explicitly designed to be compatible with the Spanning Tree Algorithm and Protocol (STP), as specified in Clause 8 of IEEE Std 802.1D, 1998 Edition and prior revisions of the standard. Computation of the Spanning Tree is identical between STP and RSTP. Protocol changes are in the following areas:

a) Definition of a new Protocol Version number (version 2) for use with RSTP.

b) Definition of a new BPDU type (BPDU type 2) to distinguish RST BPDUs from Configuration and Topology Change BPDUs.

c) Inclusion of the Port Roles (Root Port, Designated Port, and Backup Port) in the computation of Port State (Discarding, Learning, and Forwarding). In particular, a new Root Port transitions rapidly to Forwarding.

d) Signalling to neighboring Bridges of a Bridge Port's desire to be Designated and Forwarding, and explicit acknowledgement by the neighboring Bridge on a point-to-point link. This allows the Port State to transition to Forwarding without waiting for a timer expiry.

e) Acceptance of messages from a prior Designated Bridge even if they conveyed "inferior" information. Additionally, a minimum increment to the Message Age is specified so that messages propagating in this way cannot "go round in circles" for long.

f) Improvements in the propagation of topology change information so that the information does not have to be propagated all the way to the Root Bridge and back before unwanted learnt source address information is flushed from the Filtering Databases.

g) Origination of BPDUs on a Port by Port basis, instead of transmission on Designated Ports following reception of information from the Root.

In addition to the changes to the state machines described in Clause 17, the following are required in order to support these changes:

h) Revised specification of timer values to accommodate changed behavior in the cases where neighboring Bridges do not support RSTP, and the forward delay timers do actually run to completion. The default timer values are chosen to work well; however, some care may be needed in environments where timers have been tuned to their minimum values.

i) Detection of point-to-point links (see 6.4.3) to allow selection of the procedures to indicate “Designated wanting to become Forwarding” (referred to as “propose” and “proposed” in the state machines) and “Yes, go ahead” (referred to as “agree” and “agreed” in the state machines). The adminPointToPointMAC and operPointToPointMAC parameters (6.4.3) allow point-to-point links to be identified.

j) Specification of BPDU message formats that include the information necessary to signal designated indications and confirmations.

F.2.2 BPDU formats

The original BPDU formats used in IEEE Std 802.1D, 1998 Edition and prior versions of the standard had been designed with the intent that they would permit easy extensions to the protocol. The basis for the intended backwards compatibility is that, for an implementation of version X of the protocol, an implementation should interpret version >X BPDUs as if they were version X, ignoring any parameters and/or flags added by the more recent version, and interpret version <=X BPDUs exactly as specified for the version concerned. In order for this to work, new versions of the protocol are allowed to add new parameters and/or flags, but not to redefine the semantics or encoding of any of the parameters and flags that existed in previous versions. Adoption of this approach would lead a correctly implemented version 0 device to ignore the protocol version field altogether, and also to ignore any parameters and/or flags that were not part of the version 0 protocol specification.

Unfortunately, while the 1998 and prior revisions of IEEE Std 802.1D are correctly specified in this regard, the interpretation of the words in the standard has not been consistent; consequently, there are implementations of IEEE Std 802.1D in the field that will discard BPDUs that do not carry protocol version 0, or that carry additional flags over and above those specified for version 0. The wording in Clause 9 has also been made much more explicit with regard to the requirements for protocol version handling, in order to ensure that this problem is not repeated in future implementations.

In order to ensure correct interworking between version 2 (RSTP) Bridges and version 0 Bridges, it has therefore been necessary not simply to define a new protocol version number and additional parameters to go with it, but also to allow version 2 Bridges to detect the presence of version 0 Bridges on its Ports and to use version 0 BPDUs if such a Bridge is detected (see the BPDU Migration State Machine in 17.26). If a version 0 device is detected, the information necessary to allow requests for, and confirmations of, Designated Port transitions to Forwarding cannot be exchanged; however, the key element of rapidly transitioning new Root Ports to Forwarding is retained.

The protocol version chosen for RST BPDUs is version 2; version 1 BPDUs (defined for use in Remote Bridging - IEEE 802.1G) are accommodated in the version 2 format by means of a place-holder of zero length for the version 1 protocol information.

The operation of the BPDU Migration State Machine is as follows. If a new Bridge is added to a LAN, it will start by transmitting a version 2 (RSTP) BPDU. For the initial 3 s period it will accept and process any BPDU format, but reception of version 0 BPDUs will not cause it to change the BPDU format it will use (when a transmission is required by the algorithm). If all other Bridges attached to that LAN are version 2 Bridges, then they will see the version 2 BPDU and will send version 2 BPDUs themselves (if they need to transmit anything). However, if a version 0 Bridge is present, it may persist in sending version 0 BPDUs after the 3 s have elapsed. Any version 0 BPDU received after the initial 3 s period in the SEND_NEW state causes the machine to transition to the SEND_OLD state. In this state, any BPDU transmissions that are required by the algorithm make use of version 0 BPDUs, and the state is not changed for three seconds. If after 3 s, a version 2 BPDU is seen, then the state machine reverts to the SEND_NEW state. It also reverts to this state on initialization, and on explicit management request.

A likely scenario is that remaining legacy Bridge Ports will be Root Ports or Alternate Ports. In this case, when a new style Designated Port checks to see if the legacy Bridges have been removed from the LAN (by starting to send version 2 BPDUs), the legacy Bridges will be silent until they time out the existing Root and attempt to become Designated. However, this will drive the version 2 Bridge to send version 0 BPDUs, and the legacy Bridge Port(s) will be forced back to Blocking well before they can enter the Learning or Forwarding states.

One subtlety of the approach chosen is that, in the case of a legacy system that discards BPDUs based only on the analysis of the Flags field, the static "Agreed" case in the RST BPDU sets a new flag (one undefined in version 0). This would cause such a legacy Bridge to attempt to become Designated, as in the earlier scenario, allowing the version 2 Bridge to detect its presence and to revert to sending version 0 BPDUs at an early stage. This is rather preferable to discovering that version 2 BPDUs are discarded only in times of significant network change.

It should be noted that the approach chosen allows the determination of BPDU type to be made on a per-Port basis, allowing any given Bridge to use version 0 BPDUs on some Ports, and others using version 2.

F.2.3 RSTP performance

The reconfiguration time of the STP (Clause 8) is radically improved in RSTP (Clause 17) by using Port roles ("Root Port," "Designated Port," "Alternate," or "Backup") to select forwarding state transitions. In STP, these transitions are solely determined by the current and desired forwarding states (Blocking or Forwarding).

A newly selected "Root Port" can be moved directly from the Blocking to the Forwarding state provided that the previously selected Root Port is made Blocking. Loop-free topology is maintained.

The RSTP transitions accommodate arbitrary reconfiguration during periods of network device and link changes. However, an important application is auto configuration of redundant connections as backup "resilient links." Assuming that the "next best" Root Port (an Alternate Port) has been precomputed, and the Designated Port to which that Alternate Port is connected is in Forwarding, physical connectivity can be restored within the time taken for the physical media to signal link failure (through "link beat" or "loss of light" indication). This could be as little as 10 ms, and does not require the exchange of any BPDUs between the Bridges concerned.

Where a link is lost but one of the Bridges involved has no precomputed backup Port, connectivity can be restored after link failure detection within the time necessary to exchange two BPDUs (one in each direction) on one of the remaining active Ports. This is needed in order to perform the "handshake" between two Bridges, where one requests to put its Designated Port into Forwarding, and the other (a Root or Alternate Port) responds indicating whether its Port states are consistent with the requested state change. A Root Port will give a positive response (i.e., giving the Designated Port permission to transition to Forwarding) if the other Ports of its Bridge are in a state that is consistent with the state change (i.e., they are all "agreed"); otherwise, and also if the Port is an Alternate Port, the response is negative.

The "handshake" between Designated and Root or Alternate Ports mentioned above has the effect of moving a cut in the network one Bridge nearer the edge of the Bridged LAN (i.e., one Bridge further away from the Root), as a Root Port can signal "agreed" once all of its Bridge's Designated Ports have been made Discarding. These Designated Ports, in their turn, request the downstream Root Ports for permission to change state to Forwarding, and so on, until the edge of the Bridged LAN is reached.

A worst-case reconfiguration (based on a maximum diameter of 7 for the Bridged LAN) would involve the time taken for six such handshakes to take place in sequence before the reconfiguration had been propagated from the point of the original cut to the edge of the LAN (i.e., for the entire network to reconfigure).

Such rapid recovery is in dramatic contrast to the slow (about 50 s) STP reconfiguration times. It provides a basis for highly available continuous network operation based on redundant low cost network devices, with recovery times commensurate with fault tolerant telecommunications equipment. Using the campus data network for voice applications with the proposed improvement, failure and recovery might result in the loss of only a few speech samples—a "click" on the line rather than loss of a call.

Although RSTP changes the dynamic effects of STP (Bridge Ports becoming forwarding or blocking at certain times), there is no change in the BPDUs sent between an RSTP Bridge and an STP Bridge. Nor is it necessary for all Bridges in the network to change their behavior. Those that do benefit from the much reduced reconfiguration delay, and can be introduced arbitrarily into an existing network.

NOTE—RSTP Bridges revert to using STP BPDUs and TCNs on a given Port if they detect the presence of an STP Bridge on that Port. The new RST BPDU format is only used if they detect that only RSTP Bridges are present.

F.2.4 Misordering and Duplication in RSTP Bridges

F.2.4.1 Background

In IEEE Std 802.1D, 1998 Edition (and earlier) Bridges using STP, the inherent delays imposed by STP before a Port is made Forwarding ensure that any frames of a conversation that were "in transit" prior to a reconfiguration event will have been delivered or discarded before any frames are transmitted on a Port that is made Forwarding as a result of the reconfiguration event. In a network configured according to the limitations stated in IEEE Std 802.1D, 1998 Edition, the only source of misordered or duplicated frames is a "magically healed" connection between two Bridges, for example, as a result of accidental interconnections between shared media hubs. This failure mechanism is common to both STP and RSTP; however, RSTP introduces a new means whereby misordering and duplication can occur.

RSTP by its nature and intent can reduce the delay before a Port is made Forwarding to very small values; in the case of Root Port transitions, the delay can be as short as the hardware can manage it, or in the case of Designated Port transitions, as short as two frame transmission times on a single segment. Hence, it is conceivable that a reconfiguration event can take place, and a Port be made Forwarding as a result, while prior frames of a conversation are still in flight.

This paper attempts to examine these issues and evaluate the circumstances in which a problem might occur.

F.2.4.2 Frame Duplication

For a unicast frame, where the destination address of the frame has been learnt by the Bridges that may be required to forward it, that frame can only be buffered for transmission on one Port of one Bridge at any one time. Hence, it would appear to be impossible for frame duplication to occur in this situation.

If the unicast address has not been learnt, then the frame will be flooded (and therefore buffered) on all outbound Ports; there is therefore a possibility that a reconfiguration event could cause duplication of a flooded frame. Figure F-1 illustrates this scenario. The figure shows a fragment of a Bridged LAN, consisting of three three-Port Bridges. The Root Bridge is assumed to be somewhere beyond Port 1 of Bridge A; the path costs associated with the various Ports of the three Bridges result in the configuration shown, with Bridge B Port 3 in Blocking and the remaining Ports all in Forwarding.

If the configuration is stable, then a unicast frame arriving at Port A1 and destined for an unlearnt unicast destination reachable through Port B2 will be flooded by Bridge A to its Ports 2 and 3, and by Bridge C to its Ports 2 and 3. However, as Port B3 is blocked, only the frame reaching Port B1 will be transmitted onwards through B2. Hence, a station reachable through B2 sees only a single copy of the frame.

If the configuration is not stable, then it is possible for the following sequence of events to occur:

a) Bridge A receives the frame through A1 and queues the frame on A2 and A3.

b) Bridge A transmits the frame through A2 and A3.

c) Bridges B and C receive the frame through B1 and C1, and queue the frame for transmission on B2, C2, and C3.

d) Bridge B transmits the frame through B2.

e) Bridge B detects that link A2-B1 has failed, and immediately places B3 in Forwarding as its new Root Port.

f) Bridge C transmits the frame through C2.

g) Bridge B receives a second copy of the frame through B3, and queues it for transmission through B2.

h) The second copy of the frame is transmitted through B2.

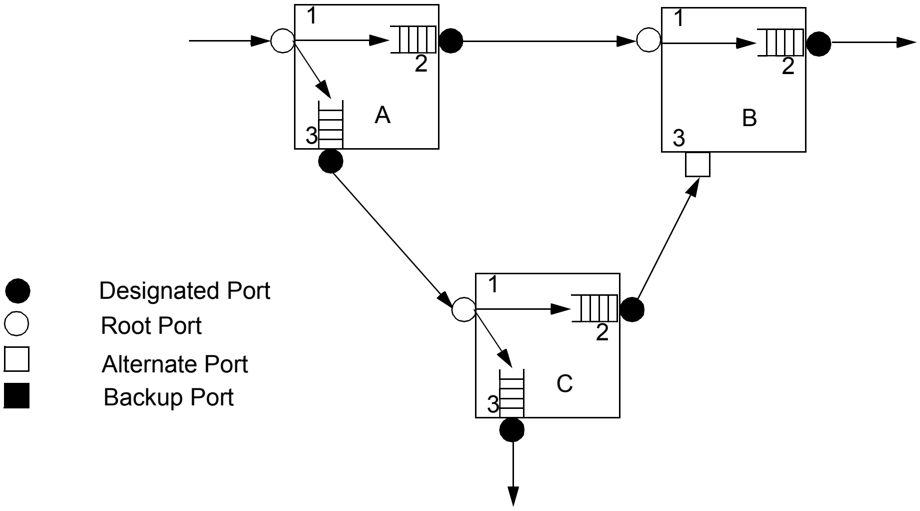


Figure F-1—Frame duplication scenario

As can be seen from the above, any station downstream of B2 has seen the same frame twice. However, as any subsequent response from the destination station would cause the address to be learned, the worst case, for those LAN protocols known to the reviewers of this standard, arising from this scenario would be two connect requests being seen by the destination machine, as all subsequent components of the conversation would be based on learned addresses.

There may be some increase in the probability that this effect will cause duplication of unicast frames, due to the fact that the reconfiguration event will itself cause addresses to be flushed from the filtering databases of Bridges in the LAN, and the flushing will be repeated over the duration of the reconfiguration event. However, the effect of repeated flushing would be most marked if the frequency of TCN transmission is comparable to the typical time period for which a frame is buffered in a Bridge.

Multicast frames are all potentially subject to duplication on reconfiguration events, as these are in general buffered at multiple outbound Ports. For those multicast LAN protocols known to the reviewers of this standard, frame duplication does not cause any problems.

F.2.4.3 Duplication – Conclusions

RSTP introduces an additional risk of duplication of unlearned unicast frames, and of multicast frames, on reconfiguration events. The risk will depend upon the configuration of the LAN, and the opportunity that the configuration gives for frames to be “in transit” in buffer storage on alternate paths to the destination(s) when a reconfiguration event takes place. The probability of frame duplication occurring as a result of a reconfiguration event will depend upon the frequency of reconfiguration events in the network, the chosen network topology, and the implementation details of the equipment used. Hence, the probability of duplication occurring is impossible to quantify without reference to the parameters of a particular Bridged LAN.

As far as the reviewers of this standard are aware, there are no LAN protocols that are sensitive to frame duplication. The existing possibility of duplication arising as a result of “magically healed” connections is a well-known problem, so this assessment is not based simply on a lack of information.

F.2.4.4 Frame Misordering

It is possible to conceive of a situation where, prior to a reconfiguration event, an end-to-end conversation was required to transit the maximum diameter of the Bridged LAN, and following reconfiguration, the same conversation only transits a single Bridge. This could happen, for example, with a Bridge that has a Root Port connected to one end of a LAN “diameter” and a Alternate Port connected to the other end of the “diameter,” as in the stable configuration shown in Figure F-2. In this configuration, a conversation is taking place between a station connected to A1 and a station connected to B2; as A2 is a blocked Alternate Port, all traffic between these two stations is relayed via Bridge C. Block the Root Port and make the Alternate Port Forwarding, and an almost instantaneous switch of location occurs for end stations downstream of that Bridge, from one “end” of the LAN to the other. As there could be frames in transit before the reconfiguration, frames transmitted immediately following the reconfiguration could arrive at their destination before the ones in transit, resulting in frame misordering as seen by the recipient. The following sequence of events illustrates the point:

a) Frame 1 of a conversation is received through A1, buffered and transmitted through A3, received through C1 and buffered awaiting transmission through C2.

b) Link A3-C1 fails; Bridge A immediately places A2 in Forwarding as its new Root Port.

c) Frame 2 of the same conversation is received through A1, buffered and transmitted through A2, received through B1 and buffered awaiting transmission through B2.

d) Bridge C transmits Frame 1 through C2 and Bridge B transmits Frame 2 through B2.

e) Frame 1 is received through B3, buffered for transmission through B2, and transmitted through B2.

Clearly, the receiving station connected to B2 sees the frames in the reverse of the order in which the originating station transmitted them.

F.2.4.5 Misordering – Conclusions

As with frame duplication, there is an additional risk of misordering of unicast and multicast frames on reconfiguration events. The risk will depend upon the configuration of the LAN, and the opportunity that the configuration gives for frames to be “in transit” in buffer storage on alternate paths to the destination(s) when a reconfiguration event takes place. The probability of frame misordering occurring as a result of a reconfiguration event will depend upon the frequency of reconfiguration events in the network, the chosen network topology, and the implementation details of the equipment used. Hence, the probability of misordering occurring is impossible to quantify without reference to the parameters of a particular Bridged LAN.

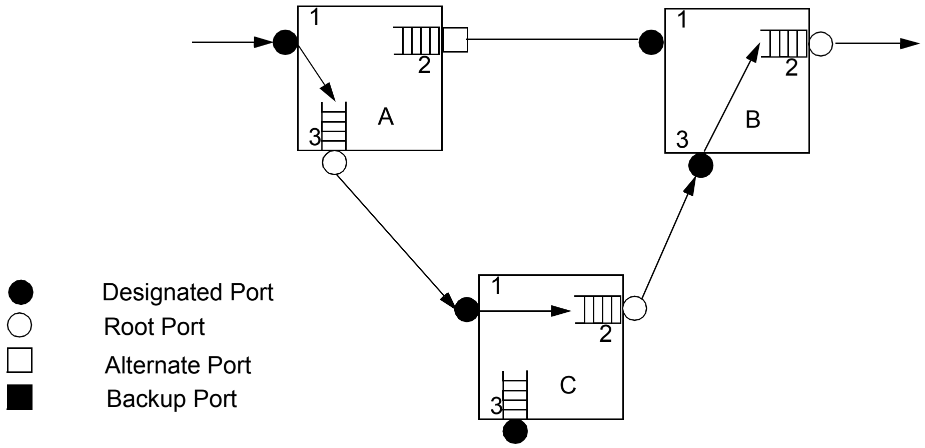


Figure F-2—Frame misordering scenario

Some LAN protocols, for example LAT, LLC2, and NETBEUI, are sensitive to misordering, so even a low incidence of misordering could result in perceived problems in networks that support these protocols. There is no obvious solution within the operation of Rapid Spanning Tree that would remove this residual level of misordering. Clause 17 defines an “STP compatibility mode” (see 17.16.1) in which the rapid transitions to Forwarding are disabled; this mode is recommended to support the use of such protocols.

F.2.4.6 Other considerations

The possibility that frames may be stored “in transit” on old routes after a reconfiguration has completed means that there is also a possibility that addresses will be mis-learnt, leading to a risk of denial of service for the addresses concerned. Hence, it is necessary to run short aging timers for a period after a reconfiguration to allow such frames to be delivered or discarded.

The Bridge model described in IEEE Std 802.1D/IEEE Std 802.1w does not include the concept of queueing on input to a Port; however, practical Bridge designs generally include some input queueing. While this is not a solution to the effects described above, a Bridge should flush any input queue associated with a Port that becomes Disabled. This behavior is consistent with the Bridge model; if a frame is in an input queue, it has not been received as far as the Bridge Port is concerned, and therefore should not be received after the Port becomes Disabled.

Insert a new Annex G.

Annex G

(informative)

Bibliography

[B1] IEEE Std 802.1G, 1998 Edition (ISO/IEC 15802-5:1998) IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Common Specifications—Part 5: Remote Media Access Control
(MAC) Bridging.