INTERNATIONAL STANDARD

ISO/IEC 15802-5

ANSI/IEEE Std 802.1G

First edition
1998-05-15

Information technology — Telecommunications and information exchange between systems — Local and metropolitan area networks — Common specifications —

Part 5:
Remote Media Access Control (MAC) bridging

Technologies de l'information — Télécommunications et échange d'information entre systèmes — Réseaux locaux et métropolitains — Spécifications communes —

Partie 5: Pontage pour le controle d'accès aux supports (MAC) à distance

Reference number
ISO/IEC 15802-5:1998(E)
ANSI/IEEE
Std 802.1G, 1998 edition

International Standard ISO/IEC 15802-5:1998(E)

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

International Standard ISO/IEC 15802-5 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 6, *Telecommunications and information exchange between systems.*

ISO/IEC 15802 consists of the following parts, under the general title *Information technology — Telecommunications and information exchange between systems — Local and metropolitan area networks — Common specifications*:

— *Part 1: Medium Access Control (MAC) service definition*

— *Part 2: LAN/MAN management*

— *Part 3: Media Access Control (MAC) bridges*

— *Part 4: System load protocol*

— *Part 5: Remote Media Access Control (MAC) bridging*

Annexes A and B form an integral part of this part of ISO/IEC 15802. Annexes C and D are for information only.

International Organization for Standardization/International Electrotechnical Commission
Case postale 56 • CH-1211 Genève 20 • Switzerland

International Standard ISO/IEC 15802-5: 1998(E)
ANSI/IEEE Std 802.1G, 1998 Edition

Information technology— Telecommunications and information exchange between systems— Local and metropolitan area networks— Common specifications—

Part 5: Remote Media Access Control (MAC) bridging

Sponsor

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

Adopted as an International Standard by the
International Organization for Standardization
and by the
International Electrotechnical Commission

Published by
The Institute of Electrical and Electronics Engineers, Inc.

International Standard ISO/IEC 15802-5:1998(E)

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

International Standard ISO/IEC 15802-5 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 6, *Telecommunications and information exchange between systems.*

ISO/IEC 15802 consists of the following parts, under the general title *Information technology — Telecommunications and information exchange between systems — Local and metropolitan area networks — Common specifications*:

— *Part 1: Medium Access Control (MAC) service definition*

— *Part 2: LAN/MAN management*

— *Part 3: Media Access Control (MAC) bridges*

— *Part 4: System load protocol*

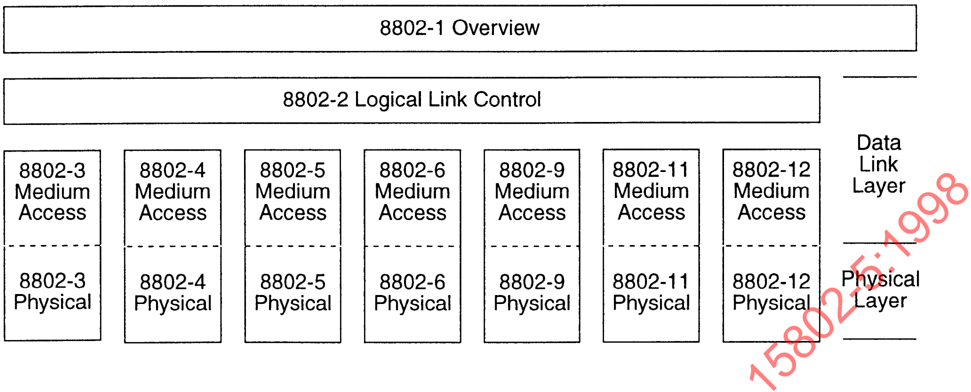
— *Part 5: Remote Media Access Control (MAC) bridging*

Annexes A and B form an integral part of this part of ISO/IEC 15802. Annexes C and D are for information only.

International Organization for Standardization/International Electrotechnical Commission
Case postale 56 • CH-1211 Genève 20 • Switzerland

Foreword to International Standard ISO/IEC 15802-5: 1998

This International Standard is part of a family of International Standards for Local and Metropolitan Area Networks. The relationship between this International Standard, which provides extensions to the behavior of ISO/IEC 10038, and the other members of the family is shown below. (The numbers in the figure refer to ISO/IEC Standard numbers.)



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

The International Standards defining the access technologies are as follows:

a) ISO/IEC 8802-3, utilizing carrier sense multiple access with collision detection (CSMA/CD) as the access method.
b) ISO/IEC 8802-4, utilizing token passing bus as the access method.
c) ISO/IEC 8802-5, utilizing token passing ring as the access method.
d) ISO/IEC 8802-6, utilizing distributed queuing dual bus as the access method.
e) ISO/IEC 8802-9, a unified access method offering integrated services for backbone networks.
f) ISO/IEC DIS 8802-11, a wireless LAN utilizing carrier sense multiple access with collision avoidance (CSMA/CA) as the access method.
g) ISO/IEC DIS 8802-12, utilizing Demand Priority as the access method.

ISO/IEC TR 8802-1, *Overview of Local Area Network Standards,* provides an overview of the series of ISO/IEC 8802 standards.

ISO/IEC 8802-2, *Logical Link Control,* is used in conjunction with the medium access standards to provide the data link layer service to network layer protocols.

ISO/IEC 15802-1, *Medium Access Control (MAC) service definition,* specifies the characteristics of the common MAC Service provided by all IEEE 802 LAN MACs. The service is defined in terms of primitives that can be passed between peer service users, their parameters, their interrelationship and valid sequences, and the associated events of the service.

ISO/IEC 15802-2, *LAN/MAN Management*, defines an OSI management-compatible architecture, and services and protocol elements for use in a LAN/MAN environment for performing remote management.

ISO/IEC 10038, *Media Access Control (MAC) bridges*, specifies an architecture and protocol for the interconnection of IEEE 802 LANs below the level of the logical link control protocol (to be renumbered 15802-3).

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.

ISO/IEC 15802-5, *Remote Media Access Control (MAC) bridging*, specifies extensions for the interconnection, using non-LAN communication technologies, of geographically separated IEEE 802 LANs below the level of the logical link control protocol.

IEEE Standards documents are developed within the Technical Committees of the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Board. Members of the committees serve voluntarily and without compensation. They are not necessarily members of the Institute. The standards developed within IEEE represent a consensus of the broad expertise on the subject within the Institute as well as those activities outside of IEEE that have expressed an interest in participating in the development of the standard.

Use of an IEEE Standard is wholly voluntary. The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. 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.

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.

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 all 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 technical 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 on standards and requests for interpretations should be addressed to:

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

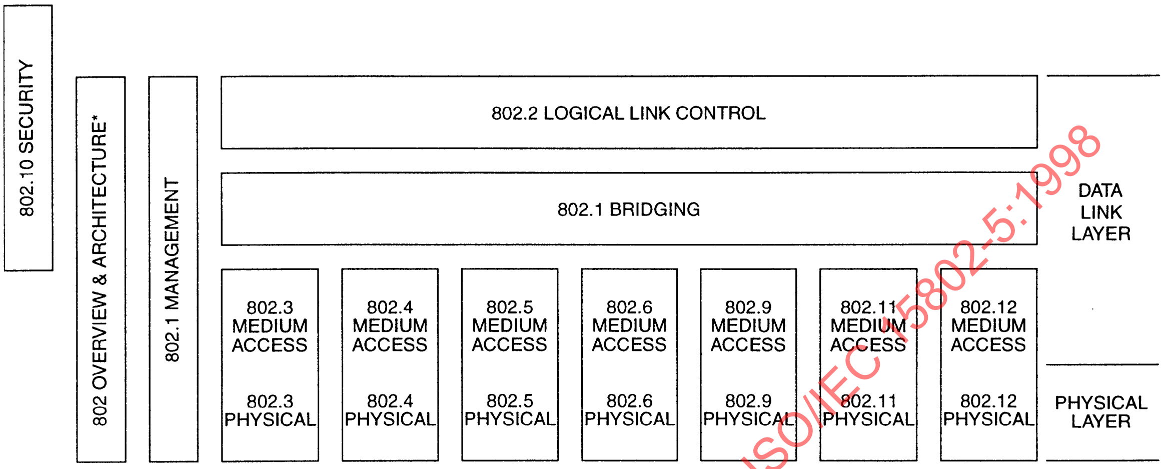
Note: Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. The IEEE shall not 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.

Authorization to photocopy portions of any individual standard for internal or personal use is granted by the Institute of Electrical and Electronics Engineers, Inc., provided that the appropriate fee is paid to Copyright Clearance Center. To arrange for payment of licensing fee, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; (508) 750-8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Introduction to ANSI/IEEE Std 802.1G, 1998 Edition

(This introduction is not a part of ANSI/IEEE Std 802.1G, 1998 Edition or of ISO/IEC 15802-5: 1998.)

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 (OSI) Basic Reference Model (ISO/IEC 7498-1: 1994). The access standards define seven 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 access technologies are as follows:

- IEEE Std 802 — *Overview and Architecture.* This standard provides an overview to the family of IEEE 802 Standards.
- ANSI/IEEE Std 802.1B and 802.1k [ISO/IEC 15802-2] — *LAN/MAN Management.* Defines an 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 [ISO/IEC 10038] — *Media Access Control (MAC) Bridges.* Specifies an architecture and protocol for the 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 802.1G [ISO/IEC 15802-5] — *Remote Media Access Control (MAC) Bridging*. Specifies extensions for the interconnection, using non-LAN communication technologies, of geographically separated IEEE 802 LANs below the level of the logical link control protocol.
- 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 Passing 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 802.9 [ISO/IEC 8802-9] *Integrated Services (IS) LAN Interface at the Medium Access Control (MAC) and Physical (PHY) Layers*
- ANSI/IEEE Std 802.10 *Interoperable LAN/MAN Security*
- IEEE Std 802.11 [ISO/IEC DIS 8802-11] *Wireless LAN Medium Access Control (MAC) and Physical Layer Specifications*
- ANSI/IEEE Std 802.12 [ISO/IEC DIS 8802-12] *Demand Priority Access Method, Physical Layer and Repeater 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 following additional working group has authorized standards projects under development:

- IEEE 802.14 *Standard Protocol for Cable-TV Based Broadband Communication Network*

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.

ANSI/IEEE Std 802.1G, 1998 Edition [ISO/IEC 15802-5: 1998]

This standard specifies extensions to the architecture and protocol of ISO/IEC 10038 that provide for the interconnection, using non-LAN communication technologies, of geographically separated IEEE 802 LANs below the level of the logical link control protocol.

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

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

IEEE 802 committee working documents are available from

IEEE Document Distribution Service
AlphaGraphics #35 Attn: P. Thrush
10201 N. 35th Avenue
Phoenix, AZ 85051
USA

Participants

The following is a list of participants in the Interworking effort of the IEEE Project 802.1 Working Group. Voting members at the time of Working Group approval are marked with an asterisk (*).

William P. Lidinsky,* *802.1 Working Group Chair*
Mick Seaman,* *Interworking Task Group Chair*
Alan M. Chambers,* *Technical Editor*

Floyd Backes*
Robert Barrett
Amatzia Ben-Artzi
Anthony Berent
Orna Berry
Laura Bridge*
James Carlo
Hon Wah Chin*
George Clapp
Stephen W. Cooper*
Paul Cowell*
Robert S. Crowder
Andy Davis*
Mike Dickerson
David Dyer-Bennett
Paul Eastman
Lionel Geretz
Richard Gilbert
Sharam Hakimi
John H. Hart*
Tony Jeffree
Jan-Olof Jemnemo*
Hal Keen*
Hans Lackner
Eugene Latham
Wen-Pai Lu
Andy Luque
John Messenger
John Montague
Jorg Ottensmeyer*
Richard Patti
Yonadav Perry*
Brian Phillips*
John Pickens
Paul Rosenblum
Venkat Prasad
Ronald Presti
Paul Ruocchio
Rich Seifert
Lee Sendelbach*
Steve Senum
Himanshu Shah*
Rosemary Slager*
W. Earl Smith
Elysia Tan
Robin Tasker*
Patricia Thaler
Surya Varanasi
Peter Videcrantz
Trevor Warwick
Bernd Widmann
Michele Wright
Michael D. Wright*
Amnon Yacoby
Carolyn Zimmer

The following persons were on the balloting committee:

Hassan M. Ahmed
Bernhard Albert
Jon M. Allingham
Kit Athul
Peter K. Campbell
James T. Carlo
David E. Carlson
Alan M. Chambers
Ian Crayford
Robert S. Crowder
Sourav K. Dutta
Paul S. Eastman
John E. Emrich
Philip H. Enslow
Changxin Fan
John W. Fendrich
Michael A. Fischer
Harvey A. Freeman
Robert J. Gagliano
Harry Gold
Patrick S. Gonia
J. Scott Haugdahl
Eli Herscovitz
Russell D. Housley
Jacob J. Hsu
Henry D. Keen
Peter M. Kelly
Gary C. Kessler
Yongbum Kim
William G. Lane
Lanse M. Leach
Randolph S. Little
Donald C. Loughry
Robert D. Love
Sam M. Madani
Peter Martini
Milan Merhar
Bennett Meyer
Richard H. Miller
David S. Millman
James F. Mollenauer
John E. Montague
Kinji Mori
David J. Morris
Wayne D. Moyers
Shimon Muller
Paul Nikolich
Ellis S. Nolley
Robert O'Hara
Donal O'Mahony
Joerg Ottensmeyer
Roger Pandanda
Lucy W. Person
Thomas L. Phinney
John R. Pickens
Vikram Prabhu
Kirk Preiss
David L. Propp
Vikram Punj
Andris Putnins
Brian M. Ramelson
Fernando Ramos
Edouard Y. Rocher

James W. Romlein
Floyd E. Ross
Christoph Ruland
Deepika Saxena
Mick Seaman
Lee A. Sendelbach
Adarshpal S. Sethi

Donald A. Sheppard
Michael A. Smith
Efstathios D. Sykas
Ahmed N. Tantawy
Patricia Thaler
Geoffrey O. Thompson
Robert C. Tripi
Mark-Rene Uchida

Barry M. Vornbrock
Yun-Che Wang
Donald F. Weir
Frank J. Weisser
Raymond P. Wenig
Mingcheng Xu
Oren Yuen

When the IEEE Standards Board approved this standard on 9 October 1996, it had the following membership:

Donald C. Loughry, *Chair* **Richard J. Holleman,** *Vice Chair*
Andrew G. Salem, *Secretary*

Gilles A. Baril
Clyde R. Camp
Joseph A. Cannatelli
Stephen L. Diamond
Harold E. Epstein
Donald C. Fleckenstein
Jay Forster*
Donald N. Heirman
Ben C. Johnson

E. G. "Al" Kiener
Joseph L. Koepfinger*
Stephen R. Lambert
Lawrence V. McCall
L. Bruce McClung
Marco W. Migliaro
Mary Lou Padgett
John W. Pope

Jose R. Ramos
Arthur K. Reilly
Ronald H. Reimer
Gary S. Robinson
Ingo Rüsch
John S. Ryan
Chee Kiow Tan
Leonard L. Tripp
Howard L. Wolfman

*Member Emeritus

Also included are the following nonvoting IEEE Standards Board liaisons:

Satish K. Aggarwal
Alan H. Cookson
Chester C. Taylor

Valerie E. Zelenty
IEEE Standards Project Editor

ISO/IEC 15802-5: 1998 [ANSI/IEEE Std 802.1G, 1998 Edition] was approved by the American National Standards Institute (ANSI) on 12 March 1997.

Contents

CLAUSE PAGE

This page intentionally left blank

Information technology—
Telecommunications and information exchange between systems—
Local and metropolitan area networks—Common specifications—

Part 5: Remote Media Access Control (MAC) bridging

1. Scope

ISO/IEC 8802 local area networks (LANs) 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 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 largely transparent to protocols operating above this boundary, in the Logical Link Control (LLC) sublayer or Network layer. 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.

A bridged LAN can provide for

— The interconnection of stations attached to LANs of different MAC types.
— An effective increase in the physical extent, the number of permissible attachments, or the total performance of a LAN.
— Partitioning of the physical LAN for administrative or maintenance reasons.

Local MAC bridges interconnect separate LANs, at geographically close points of attachment. Remote MAC bridges, which can be geographically separated, interconnect LANs sited locally to the individual remote MAC bridges, using non-LAN technologies for communication among the remote bridges. A given piece of equipment can be a local MAC bridge or remote MAC bridge, only, or it can combine the functions of a local MAC bridge and a remote MAC bridge.

This International Standard extends ISO/IEC 10038: 1993,[1] which specifies the operation of local MAC bridges, by specifying

— The aspects of operation of remote MAC bridges that are observable on the interconnected LANs, and
— A protocol for (optional) use between remote MAC bridges, across the non-LAN communications equipment that interconnects them, to configure the remote bridges within the bridged LAN in accordance with the Spanning Tree Algorithm of ISO/IEC 10038: 1993.

To this end, it

a) Extends the ISO/IEC 10038: 1993 architectural description of the MAC sublayer bridging function to cover interconnection of LANs by remote bridges (5.1).

b) States the principles of operation of remote bridges in terms of the support and preservation of the MAC service, and of maintenance of quality of service (5.1, 5.2, 5.3).

c) Extends the ISO/IEC 10038: 1993 definition of the MAC Internal Sublayer Service to apply to provision of the service by the non-LAN communications equipment that interconnects remote bridges (in addition to provision of the service by LANs, as covered in ISO/IEC 10038: 1993) (5.4).

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

d) Extends the ISO/IEC 10038: 1993 model of the internal operation of a bridge to apply to remote bridges, and provides a related model for their interconnection (Clause 6).

e) Specifies the behavior of remote bridges as observed on the LANs to which they are attached, both in relaying MAC frames between LANs and in supporting the Spanning Tree Protocol (Clauses 7 and 8).

f) Specifies the requirements for provision of the MAC Internal Sublayer Service by the non-LAN communications equipment (5.6, Clause 8).

g) Identifies managed objects for remote bridges, and defines the management operations on these objects (Clauses 9 and 10).

h) Specifies performance requirements for remote bridges, and recommends default values and applicable ranges for operational parameters (7.10, Clause 11).

i) Specifies an extension of the Spanning Tree Protocol for optional use over the non-LAN communications equipment that interconnects remote bridges (Clauses 12 and 13).

j) Specifies requirements on the non-LAN communications equipment relating to support of the optional Extended Spanning Tree Protocol, in terms of an abstract subnetwork service that is to be provided by the non-LAN communications (12.2).

k) Specifies the requirements to be satisfied by equipment claiming conformance to this International Standard (Clause 4).

l) Provides the protocol implementation conformance statement (PICS) proforma in compliance with the relevant requirements, and in accordance with the relevant guidance, given in ISO/IEC 9646-7: 1995 (Annex A).

Detailed specification of protocols and procedures to support either the MAC Internal Sublayer Service or the subnetwork service for support of the Extended Spanning Tree Protocol, over particular non-LAN communication technologies, is outside the scope of this International Standard.

Annex B summarizes the object identifier values allocated by this International Standard.

Annex C contains additional explanatory and tutorial material, intended to aid the reader in understanding a number of aspects of remote MAC bridging that were highlighted during the development of this International Standard.

Annex D summarizes the relationship of the material in this International Standard to ISO/IEC 10038: 1993.

NOTE—This International Standard is closely based upon the specification for local MAC bridging in ISO/IEC 10038: 1993, and the text largely follows the organization of that parent standard. However, dealing with remote bridging introduces the need for a number of new concepts, with associated protocol parameters and mechanisms, etc. Annex D identifies—at clause and subclause level, and occasionally still more precisely—the correspondences and differences between this International Standard and ISO/IEC 10038: 1993. In particular, it lists subclauses that contain significant new or modified requirements specific to remote MAC bridging.

2. References

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

IEEE Std 802-1990, IEEE Standards for Local and Metropolitan Area Networks: Overview and Architecture (ANSI).[2]

IEEE Std 802.1j-1996 (Supplement to IEEE Std 802.1D-1990 [ISO/IEC 10038: 1993]), Supplement to Information technology—Telecommunications and information exchange between systems—Local area networks—Media access control (MAC) bridges: Managed objects for MAC bridges (ANSI).

ISO/IEC 7498-1: 1994, Information technology—Open Systems Interconnection—Basic Reference Model: The Basic Model.[3]

ISO/IEC 8802-2: 1994 [ANSI/IEEE Std 802.2, 1994 Edition], Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 2: Logical link control.

ISO/IEC 9596-1: 1991, Information technology—Open Systems Interconnection—Common management information protocol—Part 1: Specification.

ISO/IEC 9646-7: 1995, Information technology—Open Systems Interconnection—Conformance testing methodology and framework—Part 7: Implementation Conformance Statements.

ISO/IEC 10038: 1993 (ANSI/IEEE Std 802.1D, 1993 Edition), Information technology—Telecommunications and information exchange between systems—Local area networks—Media access control (MAC) bridges.

ISO/IEC 10165-2: 1992, Information technology—Open Systems Interconnection—Structure of management information: Definition of management information.

ISO/IEC 10165-4: 1992, Information technology—Open Systems Interconnection—Structure of management information: Guidelines for the definition of managed objects.

ISO/IEC 10731: 1994, Information technology—Open Systems Interconnection—Basic Reference Model—Conventions for the definition of OSI services.

ISO/IEC 10742: 1994, Information technology—Telecommunications and information exchange between systems—Elements of management information related to OSI Data Link Layer standards.

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

ISO/IEC 15802-2: 1995 (ANSI/IEEE Std 802.1B, 1995 Edition), Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Common specifications—Part 2: LAN/MAN management.

[2] IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, USA.

[3] ISO/IEC publications are available from the ISO Central Secretariat, Case Postale 56, 1 Rue de Varembé, CH-1211, Genève 20, Switzerland/Suisse. ISO/IEC publications are also available in the United States from the Sales Department, American National Standards Institute, 11 West 42nd Street, 13th Floor, New York, NY 10036, USA.

3. Definitions

3.1 Definitions related to MAC bridging

The following definition from ISO/IEC 10038: 1993 applies, unchanged, to this International Standard.

3.1.1 bridged local area network: A concatenation of individual local area networks interconnected by MAC bridges.

The following terms specific to MAC bridging are defined in the body of ISO/IEC 10038: 1993. This International Standard extends the application of these terms to remote MAC bridging.

3.1.2 active topology: At any time, the set of communication paths in a bridged local area network that can be used in transferring data between end stations on the LANs.

3.1.3 Spanning Tree Algorithm: The abstract distributed algorithm that determines the active topology of an ISO/IEC 10038 bridged local area network.

3.1.4 Spanning Tree Protocol: The protocol that MAC bridges use in exchanging information across local area networks, in order to compute the active topology of a bridged LAN in accordance with the Spanning Tree Algorithm.

3.2 Definitions specific to remote MAC bridging

The following definitions are specific to this International Standard among the family of IEEE 802 standards. Other terms specific to remote MAC bridging are defined in the body of this International Standard.

3.2.1 adjacent bridges: Two local or remote bridges that are both attached to the same LAN or remote bridge group.

3.2.2 attach: The term used consistently to express the relationship between an end station, a bridge, or a bridge port and the physical or logical communications elements by which it is interconnected with the other bridges in a bridged LAN.

NOTE—In particular: an end station, a local bridge, or a LAN port is said to attach to a LAN; a remote bridge is said to attach to a LAN, a group, or a cluster; a virtual port is said to attach to a group, a subgroup, or a cluster. Where other uses are encountered, they are to be interpreted in a similar sense. The definitions following contain a number of examples.

3.2.3 bridge port: A LAN port or virtual port.

3.2.4 cluster: *See:* **remote bridge cluster.**

3.2.5 connect: The term used consistently to express the capability for communication, across a group or LAN, between adjacent bridges, or between the bridge ports of adjacent bridges. *Contrast:* **attach.**

3.2.6 Extended Spanning Tree Protocol: The protocol specified (in Clauses 12 and 13 of this International Standard) for optional use among the remote bridges of a group in determining how the virtual ports attaching them to the group are to participate in the active topology.

3.2.7 frame: *See:* **MAC frame.**

3.2.8 group: *See:* **remote bridge group.**

3.2.9 individual virtual port: A subgroup port that represents the capability for communication with one other remote bridge in the remote bridge group to which the port attaches.

NOTE—An individual virtual port always has another individual virtual port as its peer port; the two ports connect their respective remote bridges in a two-member subgroup.

3.2.10 LAN port: A MAC-sublayer point of attachment of a bridge to a local area network.

NOTE—This definition is equivalent to that of bridge port in ISO/IEC 10038: 1993, as applied to local MAC bridges.

3.2.11 local MAC bridge, local bridge: A MAC bridge that interconnects only local area networks to which it attaches directly as specified in the relevant MAC standards.

3.2.12 locally bridged local area network: A bridged local area network interconnected by local bridges only; a single LAN is also considered as (a degenerate case of) a locally bridged LAN.

3.2.13 MAC frame: Except where explicitly stated otherwise (in 5.4), a MAC frame conveying MAC user data across a LAN, or a representation of such a MAC frame within a MAC bridge or on the non-LAN communications equipment of a remote bridge group.

3.2.14 MAC service: The unconfirmed connectionless-mode MAC service defined in ISO/IEC 15802-1: 1995, as an abstraction of the features common to a number of specific MAC services for local area networks.

3.2.15 mixed-configuration group: A group to which the member remote bridges attach by a mixture of individual virtual ports and multipeer virtual ports.

NOTE—It is possible for some members of a mixed-configuration group to attach only by individual virtual ports.

3.2.16 multipeer virtual port: A subgroup port that represents the capability for communication with more than one other remote bridge in the remote bridge group to which the port attaches.

NOTE—A multipeer virtual port always has multipeer virtual ports as its peer ports.

3.2.17 peer port (of a virtual port): A virtual port is a peer (virtual) port of a given virtual port when both virtual ports attach to the same group and both represent the capability for communication between the remote bridges to which they belong.

3.2.18 port: *See:* **bridge port.**

3.2.19 remote bridge: *See:* **remote MAC bridge.**

3.2.20 remote bridge cluster: A subset of the remote bridges in a single group, all of which are providing, or preparing to provide, MAC-sublayer interconnection of the attached locally bridged local area networks and other groups. A remote bridge cluster is fully connected, i.e., it supports communication between any pair of the remote bridges that belong to it. Membership of a remote bridge cluster is determined dynamically through protocols operating in support of the Spanning Tree Algorithm. (See 6.13.4, 7.3.1.1, 7.3.2.4, 7.3.4.5, and 7.3.8 for fuller specification; 6.13.6 and C.7.3 for examples.)

NOTE—A cluster can—and often will—consist of all the remote bridges in the relevant group.

3.2.21 remote bridge group: A set of remote bridges, capable of communicating with each other over non-LAN communications equipment, which cooperate in providing actual or potential MAC-sublayer interconnection among all the attached locally bridged local area networks and other attached groups. Membership of a remote bridge group is determined statically, as an aspect of the configuration of the remotely bridged LAN. (See 6.13.3 and 6.13.6 for fuller specification and examples.)

NOTE—A remotely bridged LAN can contain more than one remote bridge group.

3.2.22 remote bridge subgroup: A set of remote bridges belonging to one group, such that each remote bridge in the subgroup has a single virtual port representing its communication with all the other remote bridges in the subgroup, and with no others (see 6.13.3, 6.13.6). Membership of a subgroup is determined statically, as an aspect of the configuration of the group.

3.2.23 remote MAC bridge: A MAC bridge interconnecting a locally bridged local area network and the non-LAN communications equipment of a remotely bridged LAN.

3.2.24 remotely bridged local area network: A bridged local area network of two or more locally bridged LANs interconnected using non-LAN communication technologies, and providing MAC-sublayer interworking between end stations attached to any of the LANs.

3.2.25 specific MAC service: The service provided by the MAC protocol and procedures of a specific local area network technology (which can contain features not present in other specific MAC services or in the ISO/IEC 15802-1 MAC service).

3.2.26 subgroup: *See:* **remote bridge subgroup.**

3.2.27 subgroup port: A virtual port by which a remote bridge attaches to a group consisting of two or more subgroups.

NOTE—Every virtual port is either a virtual LAN port or a subgroup port. In a group that is not a virtual LAN, every bridge attaches to the group by at least two subgroup ports.

3.2.28 virtual LAN: A group to which each member remote bridge attaches by a single virtual port (see 6.13.6.1).

NOTE—Each virtual LAN comprises a single subgroup.

3.2.29 virtual LAN port: A virtual port by which a remote bridge attaches to a virtual LAN.

3.2.30 virtual mesh: A group to which each member remote bridge attaches by a set of individual virtual ports, one for each other remote bridge in the group (see 6.13.6.1).

NOTE—A virtual mesh comprises $n \times (n-1)/2$ two-member subgroups, where n is the number of bridges in the group.

3.2.31 virtual port: An abstraction of a remote bridge's point(s) of attachment to the non-LAN communications equipment of a single group. A virtual port represents the capability for bidirectional communication with one, some, or all of the other remote bridges in that group.

NOTE—A bridge can attach by two or more virtual ports to a single group. (See 6.13.3 and 6.13.6 for fuller specification and examples.)

3.3 Acronyms and abbreviations

BPDU	bridge protocol data unit
Conf	confirm (primitive)
CSMA/CD	carrier sense multiple access with collision detection
DLE	Data Link entity
FCS	frame check sequence
Ind	indication (primitive)
LAN	local area network
LB-LAN	locally bridged local area network
LLC	logical link control
LMMS	LAN/MAN management service
MAC	media access control
MSDU	MAC service data unit
PICS	protocol implementation conformance statement
QOS	quality of service
RB	remote bridge
RB-LAN	remotely bridged local area network
Req	request (primitive)
Rsp	response (primitive)
SDU	service data unit

4. Conformance

4.1 Static conformance requirements

A remote MAC bridge for which conformance to this International Standard is claimed shall

a) Conform to the relevant MAC standards for the MAC technology or technologies implemented at its LAN ports, as specified (by reference to ISO/IEC 10038: 1993) in 5.5.

b) Conform to ISO/IEC 8802-2: 1994 for the implementation of a class of LLC supporting Type 1 operation as required by 6.3.

c) As specified in 6.7.2, support either

- — Filtering of all frames with equal source and destination MAC addresses, or
- — Forwarding (subject to the forwarding conditions) of all frames with equal source and destination MAC addresses.

d) Maintain the information required to make frame filtering decisions as specified in 6.7.1, 6.8, and 6.9.

e) Use stated values of the following parameters of the filtering database (6.9):

- — Filtering Database Size (the maximum number of entries that can be held in the filtering database).
- — Permanent Database Size (the maximum number of entries that can be held in the permanent database).

f) Support 48-bit universally administered MAC addresses, or 48-bit locally administered MAC addresses, or both.

g) Conform to the provisions for addressing specified (by reference to ISO/IEC 10038: 1993) in 6.12.

h) Support operation within at least one of the following classes of configuration specified in 6.13.3.2:

   — Virtual LAN
   — Virtual mesh
   — Mixed-configuration group

i) Provide

   — A means of assigning a group MAC address to identify the bridge protocol entity if 48-bit locally administered MAC addresses are used, and
   — A distinct port identifier for each LAN port and virtual port of the remote bridge, as specified in 7.5.4.1.1,

   as required for support of the Spanning Tree Algorithm (by reference to ISO/IEC 10038: 1993) by 7.2.

j) Implement the Spanning Tree Algorithm and operation of the associated Spanning Tree Protocol over its LAN ports as specified in 7.5 through 7.8.

k) Encode transmitted BPDUs and validate received BPDUs, at its LAN ports, as specified (by reference to ISO/IEC 10038: 1993) in 7.9.

l) Not exceed the values specified (by reference to ISO/IEC 10038: 1993) in 7.10 for the following parameters:

   — Maximum bridge transit delay
   — Maximum Message Age increment overestimate
   — Maximum BPDU transmission delay

m) Use the value specified (by reference to ISO/IEC 10038: 1993) in 7.10 for the Hold Time parameter.

n) Not exceed the values specified in 7.10.4 for the following parameters:

   — Group maximum bridge transit delay
   — Group maximum message propagation delay
   — Group maximum Message Age increment overestimate

o) Support relaying of frames (subject to the forwarding conditions in 6.7.1) to and from any other active remote bridges in the groups to which the remote MAC bridge belongs, in accordance with the requirements specified in Clause 8.

p) Use stated values of the time to detect a failure of the relaying connectivity requirement (8.1) and the time to detect a failure of the group connectivity requirement (12.2.2).

q) — Specify a Guaranteed Port Filtering Rate for each LAN port, a Guaranteed Bridge Relaying Rate for the bridge, and the related time intervals T_F and T_R as specified (by reference to ISO/IEC 10038: 1993) in Clause 11.
   — Not violate any of the other conformance provisions of this International Standard when operating within the parameters specified.

4.2 Options

A remote MAC bridge for which conformance to this International Standard is claimed may

a) Provide the capability to control the mapping of the priority of forwarded frames as specified in 6.7.4.

b) Provide the capability to read and update the filtering databases as specified in 6.9.

c) Support the exchange of filtering information with other remote bridges in one or more of the groups to which the remote MAC bridge belongs, as permitted in 6.9.

d) Implement the optional provisions for addressing bridge management, for associating the bridge address with a LAN port, and for configuring additional addresses in the permanent database, as specified (by reference to ISO/IEC 10038: 1993) in 6.12.

e) Provide the capability to assign values, in accordance with the ranges of values specified (by reference to ISO/IEC 10038: 1993) in 7.10, to the following parameters, to allow configuring of the spanning tree active topology as described in 7.2:

- — Bridge Priority
- — Port Priority
- — Path Cost

f) Provide the capability to set the values, in accordance with the ranges of values specified (by reference to ISO/IEC 10038: 1993) in 7.10, of the following parameters of the Spanning Tree Algorithm and Protocol:

- — Bridge Max Age
- — Bridge Hello Time
- — Bridge Forward Delay

g) Provide the capability to set the value of the Primary Reclustering Delay parameter for each non-virtual-LAN group to which the bridge belongs, in accordance with the range of values specified in 7.10.5.

h) Support use of a priority mechanism, with up to eight levels of priority, applicable to relaying of frames to or from other active remote bridges in one or more of the groups to which the remote MAC bridge belongs, as specified in 8.2.6.

i) Support management of the remote bridge: a remote bridge claiming to support management shall support all the managed objects and definitions specified in Clause 9.

j) Support IEEE 802.1 remote management: a bridge claiming to support such remote management shall

- — Conform to ISO/IEC 15802-2: 1995, and
- — Support remote management by use of the network management operations and encodings specified in Clause 10.

k) Support the Extended Spanning Tree Protocol in one or more of the groups to which the bridge belongs, as specified in Clauses 12 and 13.

4.3 Protocol implementation conformance statement (PICS)

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

5. Support of the MAC service

MAC bridges interconnect the separate LANs that comprise a bridged LAN by relaying frames between the separate MACs of the bridged LANs. A single local MAC bridge interconnects two LANs directly for any instance of relaying, as described in ISO/IEC 10038: 1993. A remote MAC bridge interconnects two LANs indirectly, via non-LAN communications equipment and one or more further remote bridges, for any instance of relaying (see 5.1, Figures 5-3 and 5-4).

This clause discusses

- Provision of the MAC service to end stations.
- Support of the MAC service by remote bridges and the non-LAN communications equipment that interconnects them.
- Preservation of the MAC service offered by the bridged LAN.
- Maintenance of quality of service in the bridged LAN.
- The Internal Sublayer Service offered to the remote bridges themselves to enable them to interconnect individual LANs.

5.1 Provision and support of the MAC service

The MAC service provided to end stations attached to a bridged LAN is the (unconfirmed) connectionless-mode MAC service defined in ISO/IEC 15802-1: 1995. The MAC service is defined as an abstraction of the features common to a number of specific MAC services; it describes the transfer of user data between source and destination end stations, via MA-UNITDATA request primitives and corresponding MA-UNITDATA indication primitives issued at MAC service access points. Each MA-UNITDATA request and indication primitive has four parameters: Destination Address, Source Address, MAC service data unit (MSDU), and Priority.

The model for provision of the MAC service is the simple service-provider / service-user model shown in Figure 5-1. For a pair of end stations attached to a single LAN, the MAC service is supported by implementations of the specific MAC service for the LAN in the two end stations, and by the LAN communications media and any Physical-layer relaying equipment. This support relates to the service-provider model in a straightforward way, with the specific MAC procedures being associated with the MAC service access points (see Figure 5-2).

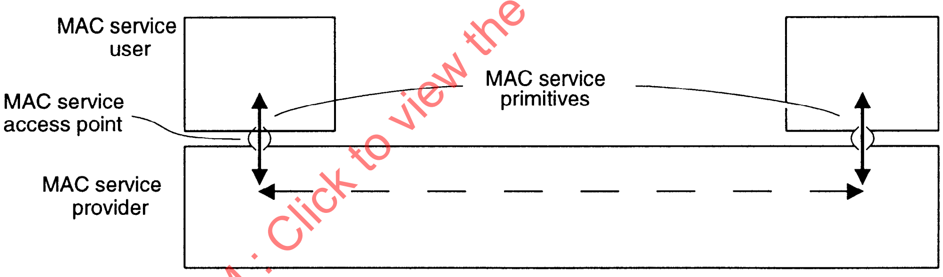


Figure 5-1—Model for provision of the MAC service

When MAC-sublayer communication is relayed through remote bridges, a more complicated representation for support of the MAC service throughout the MAC sublayer is needed. First, this has to take account of the presence of a specific MAC entity associated with each LAN port; second, it has to represent the conveying of the MAC service semantics across the non-LAN communications equipment; and finally, it has to represent the relaying of the MAC service semantics by bridges.

NOTE 1—An important distinction is made between *provision* of a service, according to the “black box” service provider model such as is shown in Figure 5-1, and *support* of the service, dealing with structure and functions within the interior of the box. This same distinction applies, below, to provision and support of the Internal Sublayer Service.

Figures 5-3 and 5-4 show the structure of the MAC sublayer support for remote bridging, in relation to the service-provider model. Figure 5-3 shows the simplest case, where relaying is via two remote bridges connected across a single remote bridge cluster. Figure 5-4 shows the case of relaying between two clusters by an additional remote bridge (the two clusters belong to different groups—see 6.13.3 and 6.7.1). The parts of Figures 5-3 and 5-4 labeled "MAC" represent the specific MAC entities for the LANs. The parts marked "MAC service support" represent the functions in the remote bridges that perform the equivalent communication of the MAC service semantics across the non-LAN equipment.

NOTE 2—Figures 5-1 through 5-4 illustrate MAC sublayer support as it applies to the communication path between a single given pair of LAN end stations. In general, a remote bridge is attached to one or more LANs. It offers multiple end stations the capability of communicating via remote bridge groups with end stations attached to other LANs. Instances of communication between different pairs of end stations can involve different sets of remote bridges and remote bridge groups. The central remote bridge in Figure 5-4 would attach to at least one LAN, which is not shown since it is not on the communication path between the two end stations being considered.

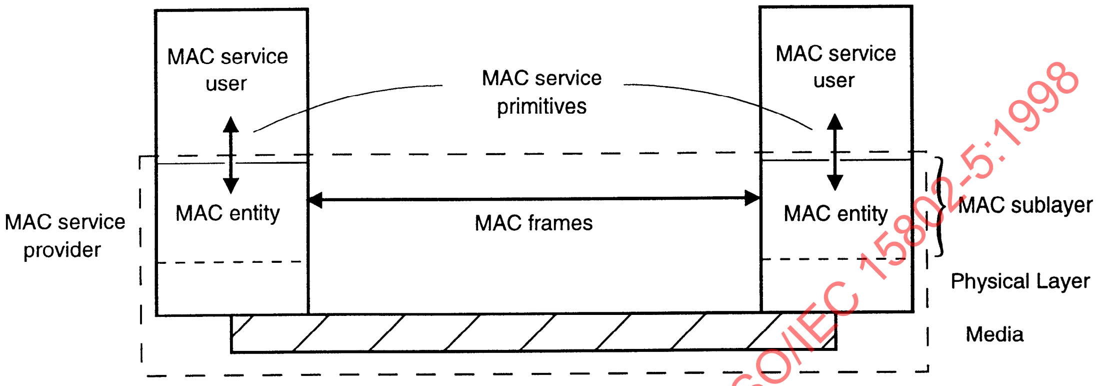


Figure 5-2—Support of the MAC service on a single LAN

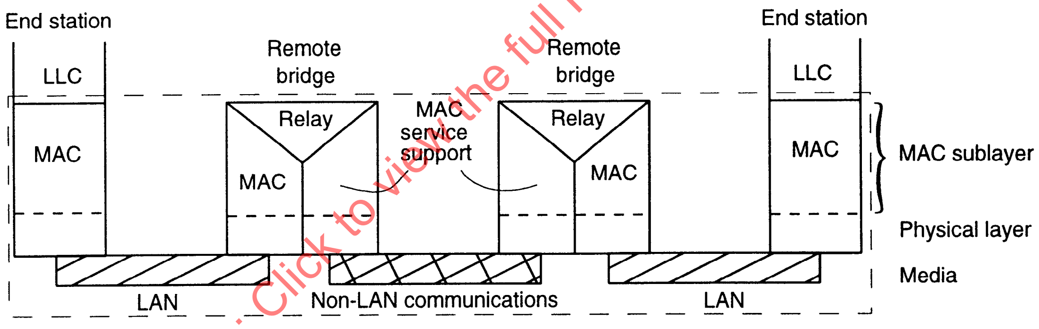


Figure 5-3—MAC service with remote bridging

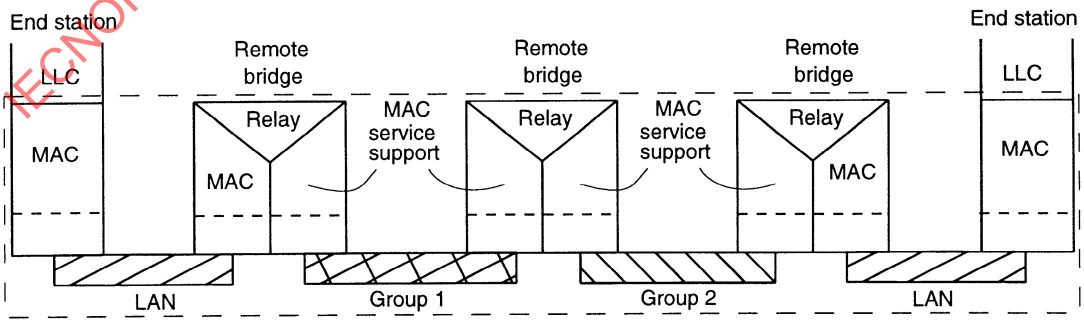


Figure 5-4—MAC service with remote bridging: multiple groups

5.2 Preservation of the MAC service

The MAC service offered by a remotely bridged LAN is similar (see 5.3) to that offered by a single LAN. In consequence

a) A bridge is not directly addressed by communicating end stations, except as an end station for management purposes: frames transmitted between end stations carry the MAC address of the peer end station in their Destination Address fields, not the MAC address, if any, of a bridge.

b) All MAC addresses are unique and addressable within the bridged LAN.

c) The MAC addresses of end stations are not restricted by the topology and configuration of the bridged LAN.

5.3 Quality of service (QOS) maintenance

As with local MAC bridges (ISO/IEC 10038), remote bridges should not significantly degrade the QOS relative to that provided on a single LAN. However, remote bridges often use intermediate-link technologies that offer lower data rates and longer delays than LANs; hence the degradation of service from a remote bridge can be greater than that introduced by a local bridge. A remote bridge does not violate MAC service invariants such as frame ordering and non-duplication; however, performance characteristics can be degraded from those provided by either a single LAN or a locally bridged LAN.

The QOS parameters to be considered are those relating to

a) Service availability
b) Frame loss
c) Frame misordering
d) Frame duplication
e) The transit delay experienced by frames
f) Frame lifetime
g) The undetected frame error rate
h) Maximum service data unit size supported
i) User priority
j) Throughput

A discussion of how the presence of remote bridges can affect the QOS provided to LAN end stations is presented in C.5.3. The following requirements apply to the operation of bridges in general; other requirements relating to maintenance of QOS are specified in 6.7, 7.10, and 8.2. (See C.5.3 for more detailed references.)

5.3.1 Undetected error rate

For frames relayed between LANs of the same MAC type, remote bridges shall not introduce an undetected frame error rate greater than that which would be achieved by preserving the frame check sequence (FCS). For frames relayed between LANs of different MAC types, remote bridges shall not increase the undetected frame error rate above 5×10^{-14} per octet of MSDU length.

5.4 Internal Sublayer Service provided within the remote MAC bridge

The Internal Sublayer Service provided by a MAC entity to the MAC relay entity within a remote bridge is that provided by the individual MAC for the LAN port. This observes the appropriate MAC procedures and protocol for the LAN to which it attaches. No control frames (i.e., MAC frames that do not convey MAC user data) are forwarded on any LAN other than that on which they originated.

In a remote bridge group, the same Internal Sublayer Service is provided to the MAC relay entities by the non-LAN access functions of the virtual ports. This International Standard does not specify any mechanisms (protocols, procedures, communications technologies, etc.) by which the Internal Sublayer Service is supported in a group. The operation of such mechanisms—whatever they may be—is modeled as the MAC service support function, which maps the Internal Sublayer Service onto the associated non-LAN communications.

The Internal Sublayer Service excludes specific features and procedures of individual LAN MAC methods, or of particular non-LAN communications technologies.

The primitives and parameters that describe the Internal Sublayer Service are as follows:

```
M-UNITDATA request      (
                        frame_type,
                        mac_action,
                        destination_address,
                        source_address,
                        mac_service_data_unit,
                        user_priority,
                        access_priority,
                        frame_check_sequence,
                        cluster_id
                        )

M-UNITDATA indication   (
                        frame_type,
                        mac_action,
                        destination_address,
                        source_address,
                        mac_service_data_unit,
                        user_priority,
                        frame_check_sequence,
                        cluster_id
                        )
```

For provision of the service by LANs, invocation of the primitives and assignment of values to their parameters are exactly as defined in ISO/IEC 10038: 1993, with the cluster_id parameter considered always to take the null value. The identification of the LAN to which a frame is to be transmitted is a local matter and is not expressed as a parameter of the service primitive.

For provision of the service within a remote bridge group,

a) Each M-UNITDATA indication primitive invoked corresponds to receipt, from the non-LAN communications equipment, of the complete information corresponding to a single relayed MAC frame.

b) An M-UNITDATA request primitive is invoked to transmit a frame across the non-LAN communications equipment.

c) Every primitive has the frame_type parameter value user_data_frame.

d) Every primitive has the mac_action parameter value request_with_no_response.

e) The access_priority parameter in a request primitive is the priority requested of the local MAC service support functions (those functions can include a priority mechanism for use across the non-LAN communications media).

f) The destination_address, source_address, mac_service_data_unit, and user_priority parameters are those of the relayed MAC frame.

g) The frame_check_sequence is either that of the relayed MAC frame together with an indication of the MAC type of the relevant LAN, or is absent.

h) The cluster_id parameter of an M-UNITDATA request identifies the remote bridge cluster to which the virtual port at which the primitive is issued is attached; the cluster_id parameter of an M-UNITDATA indication has the same value as in the corresponding M-UNITDATA request (see 6.5, 8.3).

Refer to ISO/IEC 10038: 1993 for full definitions of items a) through f) above.

5.5 Support of the Internal Sublayer Service by specific MAC procedures

The mapping of the Internal Sublayer Service to the MAC protocol and procedures of each individual LAN type, and the encoding of the parameters of the service in MAC frames, at each LAN port of a remote bridge shall be as specified in ISO/IEC 10038: 1993.

5.6 Support of the Internal Sublayer Service in remote bridge groups

Detailed mechanisms for support of the Internal Sublayer Service by specific non-LAN communications technologies are outside the scope of this International Standard. Any such mechanisms shall meet the requirements specified in Clause 8. (See also 6.2.4, and related subclauses of Annex C.)

6. Principles of operation

This clause establishes the principles of operation of a remote bridge, by reference to a model of that operation. It complements, and is compatible with, the specification for local bridges in ISO/IEC 10038: 1993. In addition, this clause establishes a model for the interconnection of remote bridges (6.13).

6.1 Remote bridge operation

The principal elements of remote bridge operation, as for local bridges, are

— Relay and filtering of frames.
— Maintenance of the information required to make frame filtering and relaying decisions.
— Management of the above.

6.1.1 Relay

A remote bridge relays MAC user data frames between the MACs of the LB-LANs attached to its LAN ports and the MAC service support functions associated with its virtual ports. A remote bridge that belongs to two or more remote bridge groups also relays MAC user data frames between the MAC service support functions associated with two or more of its virtual ports attaching to different groups. A remote bridge that attaches to two or more LANs relays MAC user data frames between the MACs of the LB-LANs attached to its LAN ports.

The functions that support the relaying of frames and maintain the QOS supported by the remote bridge are

a) Frame reception, on LAN ports and virtual ports.

b) Discard on received frame in error (see 5.3, C.5.3.2).

c) Frame discard if the frame_type is not user_data_frame, or if the mac_action is not request_with_no_response (see 5.4).

d) Frame discard following the application of filtering information.

e) Frame discard on transmittable MSDU size exceeded (see 5.3, C.5.3.8).

f) Forwarding of received frames to other LAN ports or virtual ports.

g) Frame discard to ensure that a maximum bridge transit delay or maximum cluster transit delay is not exceeded (see 5.3, C.5.3.6).

h) Selection of access priority for forwarded frames (see 5.3, C.5.3.9).

i) Recalculation of FCS, if necessary (see 5.3, C.5.3.7).

j) For frames transferred through virtual ports, mapping between M-UNITDATA primitives and the underlying MAC service support functions, including any cluster-specific error detection and correction mechanisms (see 5.3, C.5.3.7).

k) Frame transmission, onto LANs and/or clusters.

6.1.2 Filtering Information

A remote bridge filters frames to prevent duplication (see 5.3, C.5.3.4) and to localize traffic (see 5.3, C.5.3.10). The filtering functions that support the use and maintenance of filtering information in a remote bridge are

a) Permanent configuration of reserved addresses.

b) Explicit configuration of static filtering information.

c) Automatic learning of dynamic filtering information through observation of bridged LAN traffic.

d) Ageing out of filtering information that has been automatically learned.

e) Calculation and configuration of bridged LAN topology.

f) Optionally, exchange of dynamic filtering information with other remote bridges.

NOTE—Item f) is specific to remote bridge operation; the others are common to local bridges and remote bridges.

6.1.3 Bridge management

Relay and filtering of frames, and the maintenance of filtering information, are controlled and monitored using functions that support management of a remote bridge: these management functions are specified in Clause 9.

6.2 Remote bridge architecture

6.2.1 Architectural model of a remote bridge

Figure 6-1 illustrates a remote bridge with one LAN port and one virtual port, and Figure 6-2 illustrates the architecture of such a remote bridge. A remote bridge is modeled as consisting of

- A MAC relay entity that interconnects the bridge's ports
- At least one LAN port
- At least one virtual port
- One or more group communications entities, associated with the virtual ports
- Higher layer entities, including at least a bridge protocol entity

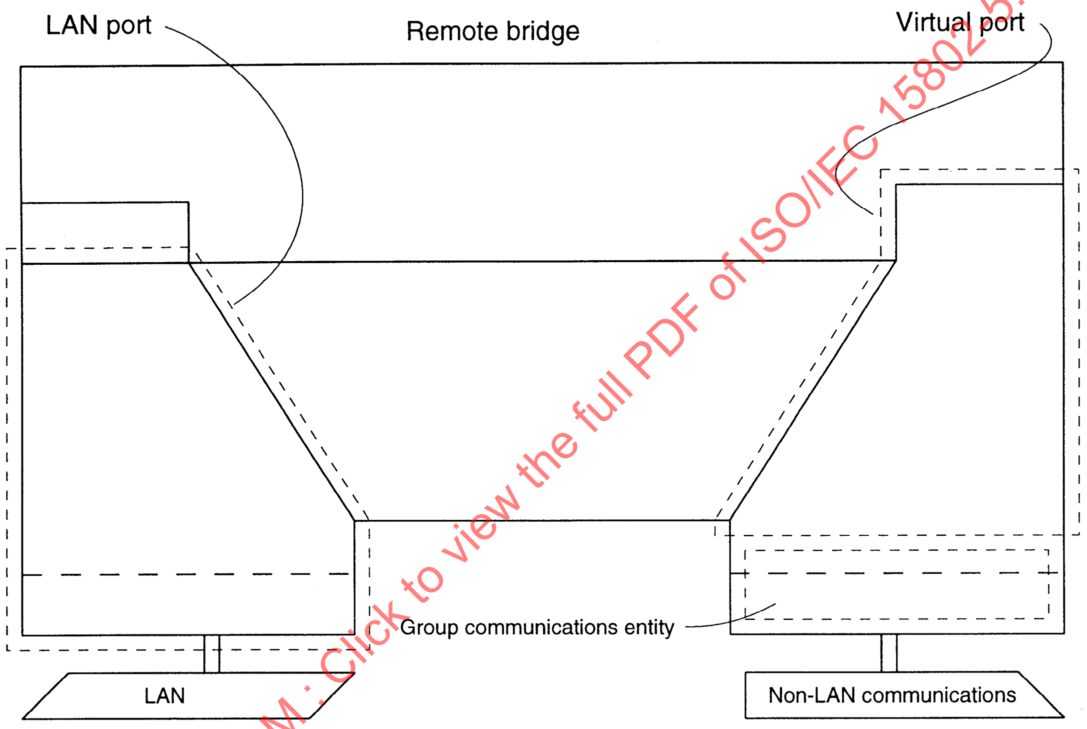


Figure 6-1—Remote bridge ports

6.2.2 MAC relay entity

The MAC relay entity handles the Media Access method independent functions of relaying frames between ports, filtering frames, and learning filtering information. It uses the Internal Sublayer Service provided by the separate MAC entities for each LAN port and by the MAC service support functions for each virtual port. (The Internal Sublayer Service and its support are described in 5.4 and 5.5.)

Frames are relayed between LAN ports and virtual ports (i.e., LAN to cluster or cluster to LAN), or between virtual ports attached to different clusters (inter-group relaying), or between LAN ports attached to different LANs.

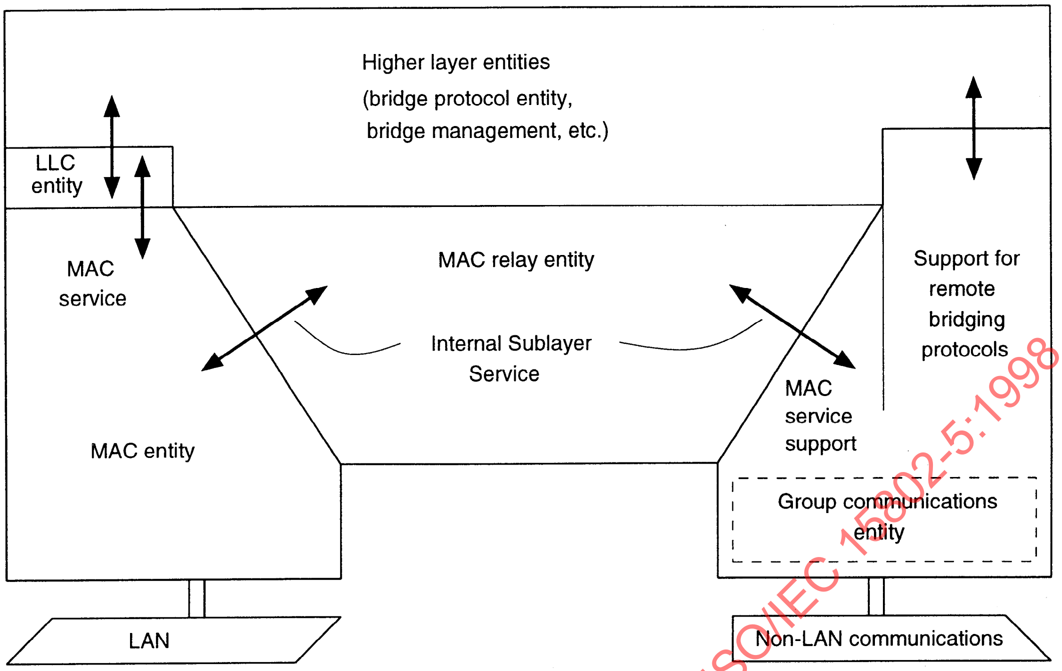


Figure 6-2—Remote bridge architecture

6.2.3 Ports

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

Each virtual port transmits and receives relayed MAC frames to and from the remote bridge cluster to which it attaches. Appropriate MAC service support functions map the Internal Sublayer Service to the non-LAN communications equipment. Transmission and reception of information exchanged between the remote bridges of a group are also modeled as occurring through virtual ports (see 6.13.3.1).

6.2.4 Group communications entity

The group communications entity provides an explicit recognition of the following two aspects of communication underlying the virtual ports:

a) The abstract communication functionality represented by virtual ports is supported by real communications equipment, and by protocols and procedures (outside the scope of this International Standard) that map the virtual-port-level communications to that real equipment.

b) The operation of some groups can require a remote bridge to communicate with others in the group, in ways that do not directly support communication represented at the virtual port level (e.g., in support of the requirements of 8.1 and 12.2.2).

(See C.6.13 for further discussion, and examples, of the group communications entity's role.)

6.2.5 Higher layer entities

The bridge protocol entity handles calculation and configuration of bridged LAN topology.

For communication over a LAN, via a LAN port, the bridge protocol entity and other higher layer protocol users, such as bridge management, make use of LLC procedures. These procedures are provided separately for each LAN port, and use the MAC service provided by the individual MAC entities.

For communication within a remote bridge group, via a virtual port, the bridge protocol entity and other higher layer protocol users make use of the group's communications through protocol support mechanisms whose details are outside the scope of this International Standard. (See 12.2.1 for a definition of the service to be provided by such mechanisms for use by the bridge protocol entity, if that operates the optional Extended Spanning Tree Protocol).

NOTE—These mechanisms could be the same, or partly the same, as those for the MAC service support function, or they could involve quite separate communications support. For the purposes of modeling the functions present in a remote bridge, such details are unimportant: virtual ports represent the abstraction, independent of those details, of the fact that communication occurs between remote bridges, relating to support both for relaying MAC frames and for use of higher-layer protocols.

6.3 Model of operation of a single remote bridge

The MAC relay entity's use of the Internal Sublayer Service is specified in 6.5 and 6.6. State information associated with each LAN port and virtual port, and cluster identification information associated with virtual ports and with the bridge's membership of groups, govern the ports' participation in the RB-LAN. (Port states are specified in detail in 7.4.)

Frames are accepted for transmission and delivered on reception to and from processes and entities which model the operation of the MAC relay entity in a remote bridge. These are

a) The forwarding process (6.7), which forwards received frames that are to be relayed through other ports, filtering frames on the basis of information contained in the filtering database (6.9) and on the states of the ports.

b) The learning process (6.8), which by observing the source MAC addresses of frames received on each port updates the filtering database (6.9), conditionally upon the port state.

c) The filtering database (6.9), which holds filtering information and supports queries by the forwarding process as to whether frames with given values of the destination MAC address field should be forwarded to a given port.

Each LAN port also functions as an end station providing the MAC service to LLC, which in turn supports operation of the bridge protocol entity (6.10) and of other possible users of LLC such as protocols providing bridge management (6.11). Each virtual port provides similar functions that support communication between bridge protocol entities in remote bridges belonging to the same remote bridge group (see also 6.2.1 and 6.2.3).

Figures 6-3 and 6-4 each illustrate a single instance of relaying a frame between two ports of a remote bridge, for frame reception on a LAN port and a virtual port, respectively (the cases of relaying between two virtual ports, and of relaying between two LAN ports, are similar). The forwarding process uses port state information and filtering database information to determine to which ports, if any, the received frame is to be forwarded.

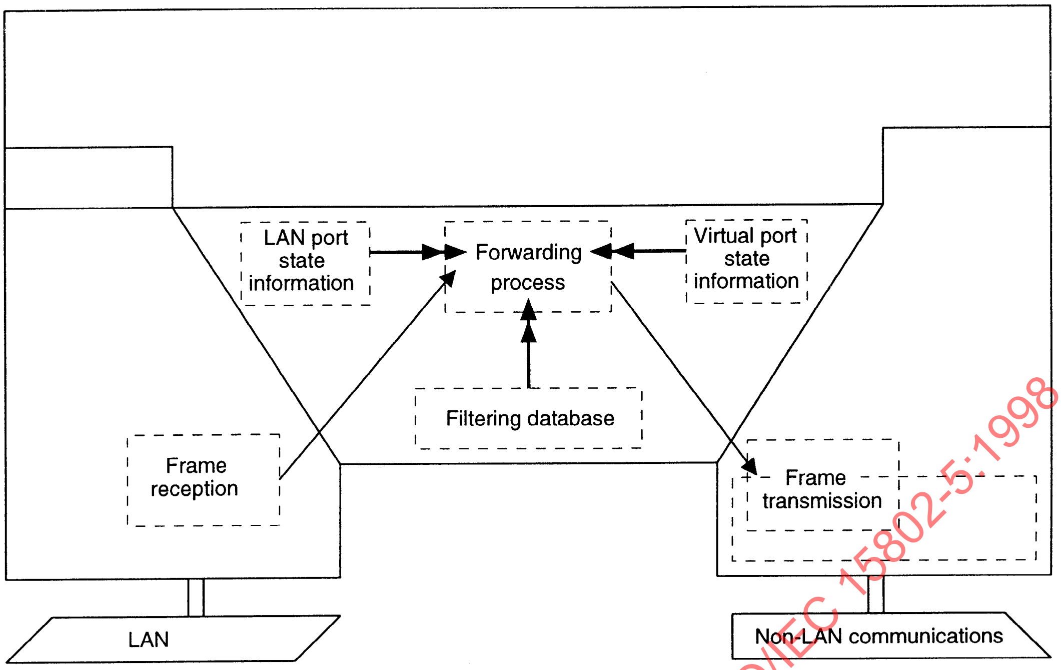


Figure 6-3—Remote bridge relaying, LAN port to virtual port

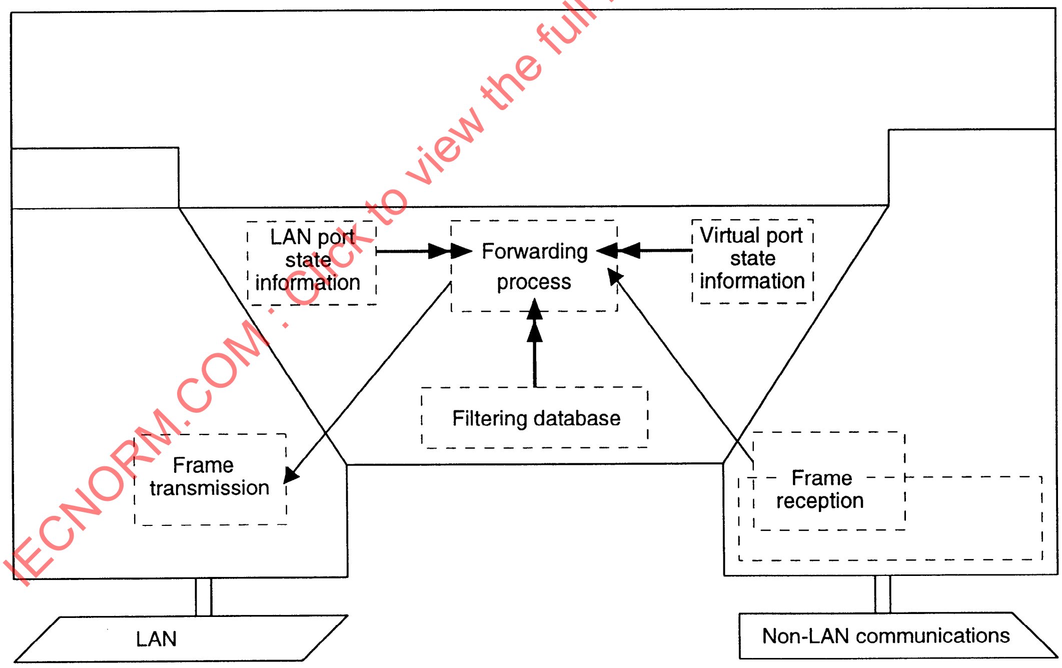


Figure 6-4—Remote bridge relaying, virtual port to LAN port

Figures 6-5 and 6-6 illustrate, similarly, the incorporation into the filtering database of station location information carried by a single received frame; port state information for the receiving port is used to determine whether or not the information is to be incorporated.

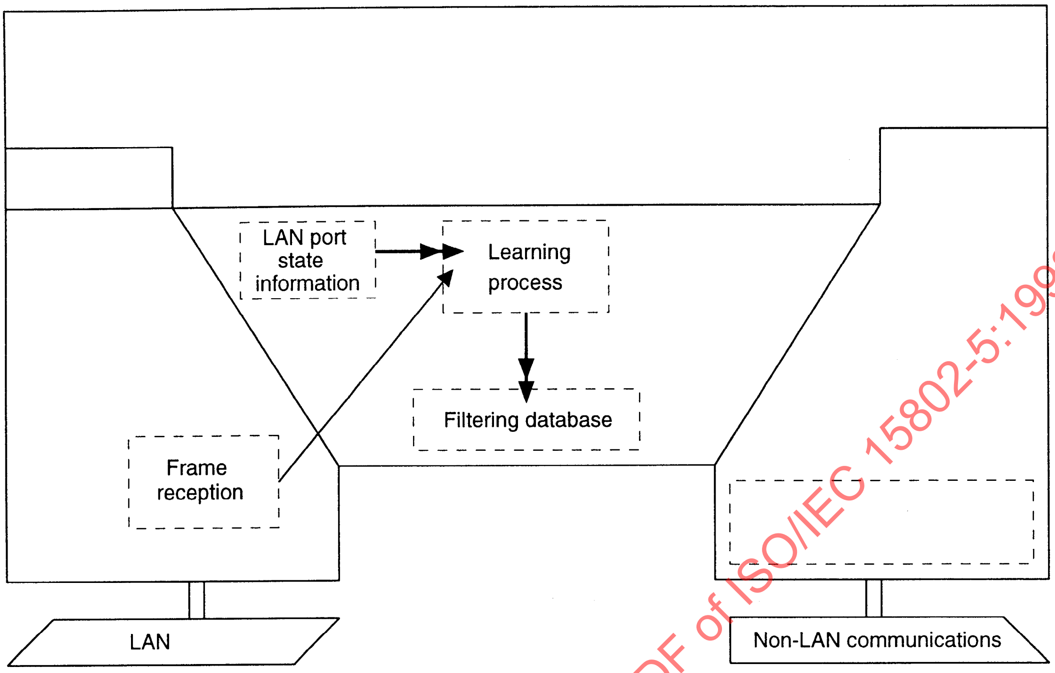


Figure 6-5—Observation of network traffic received on a LAN port

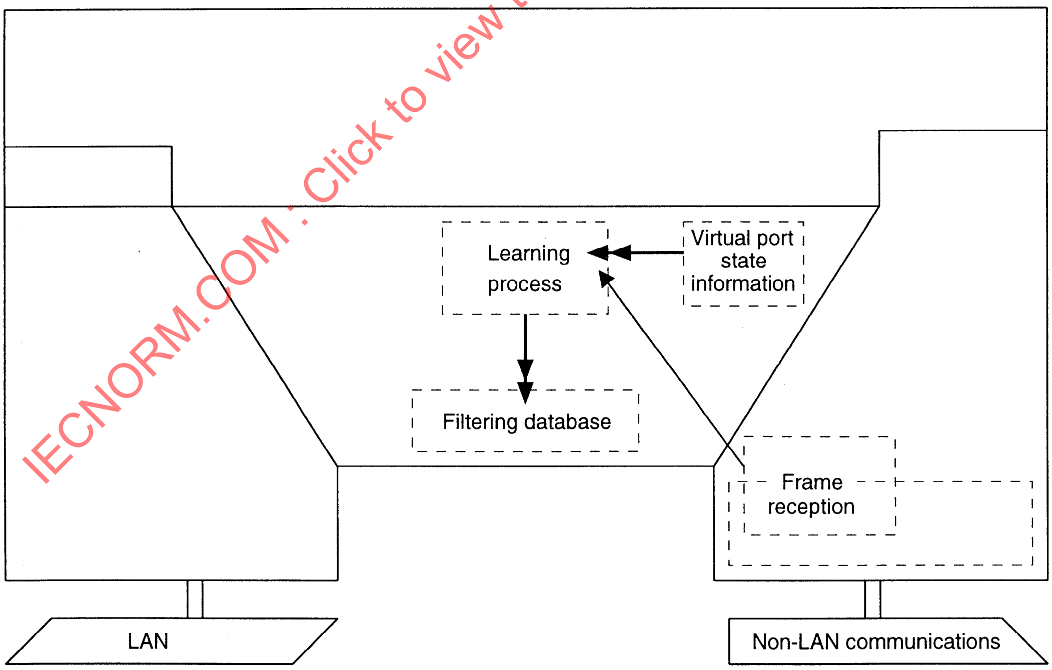


Figure 6-6—Observation of network traffic received on a virtual port

Figure 6-7 illustrates the bridge protocol entity operating the Spanning Tree Protocol in a remote bridge: this includes its reception and transmission of bridge protocol data units via a LAN port, and of the equivalent information via a virtual port; and modification of port state information to select the active topology of the RB-LAN. The bridge protocol entity is shown divided into three parts corresponding to these functions. If the optional Extended Spanning Tree Protocol is used within the group, the interface between the bridge protocol entity and the functions that support remote bridging protocols is the RB-Protocol Subnetwork Service specified in 12.2.1.

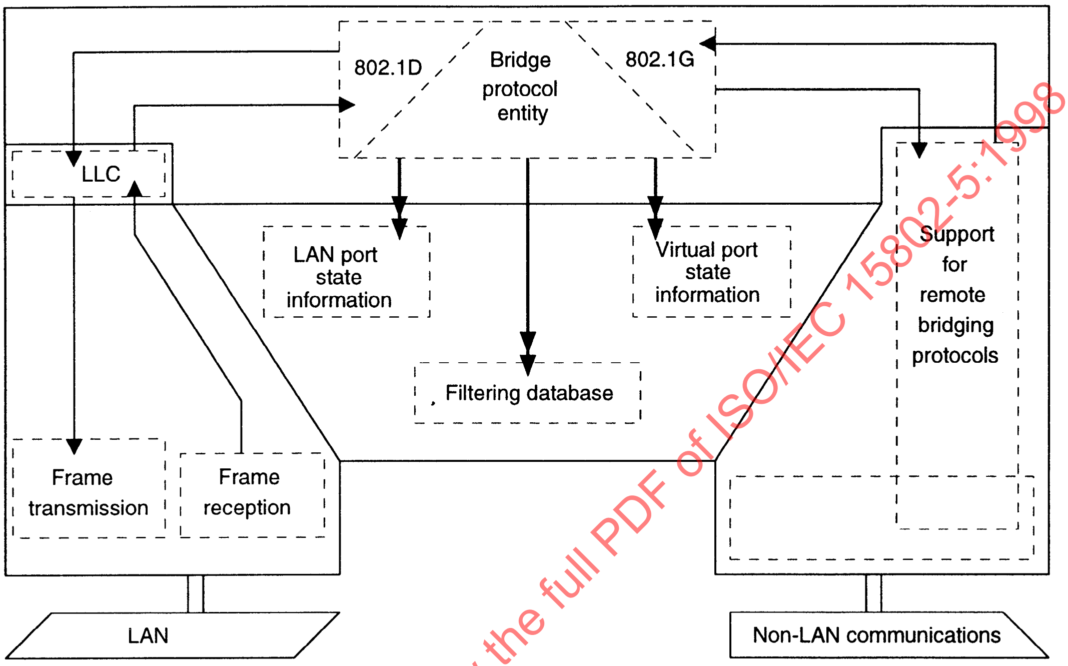


Figure 6-7—Operation of bridge-to-bridge protocols in a remote bridge

Figures 6-8 through 6-10 illustrate reception and transmission of frames by the bridge management entity via the LLC entity. Figure 6-8 shows the straightforward cases, of frames received and transmitted at a LAN port with which the bridge management entity is associated.

Figure 6-9 shows reception of a frame via a virtual port and the forwarding process, and Figure 6-10 shows the complementary transmission of a frame by the bridge management entity back through the forwarding process and the virtual port. In each of these scenarios, transmission to the LAN may be suppressed, as an optimization, if the frame has an individual destination MAC address (see 6.7.5).

NOTES

1—It is emphasized that these figures illustrate only a modeling of information flows, not the structure of an implementation. The model in Figures 6-8 through 6-10 applies to bridge management information carried over LLC, in MAC frames addressed to the MAC addresses of LAN ports. Considering the implementation of the flow illustrated in Figure 6-9 within a real remote bridge, the fact that a frame is addressed to the MAC address of the LAN port does not mean that it has to go to the port's adapter card, or equivalent, before re-entering the bridge.

2—Communications support for bridge management may be provided by non-LLC mechanisms; such support falls outside the scope of these illustrations.

3—The information flows shown in Figures 6-9 and 6-10 never apply to BPDUs of the Extended Spanning Tree Protocol, since these are not conveyed in MAC frames.

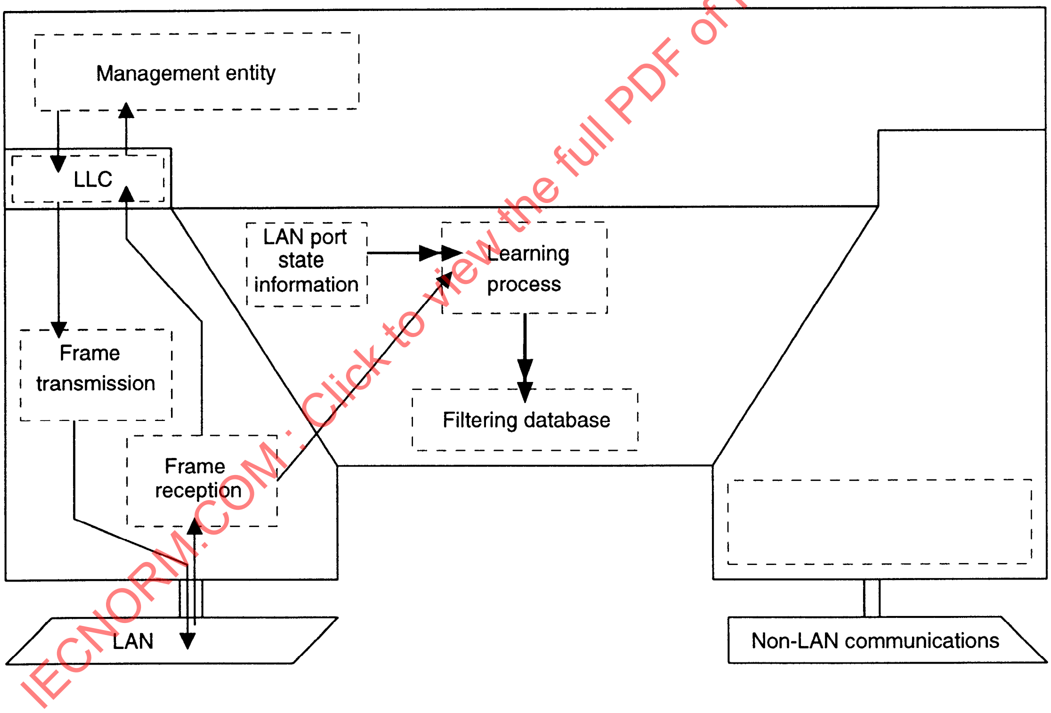


Figure 6-8—Reception and transmission of bridge management information at a LAN port of a remote bridge

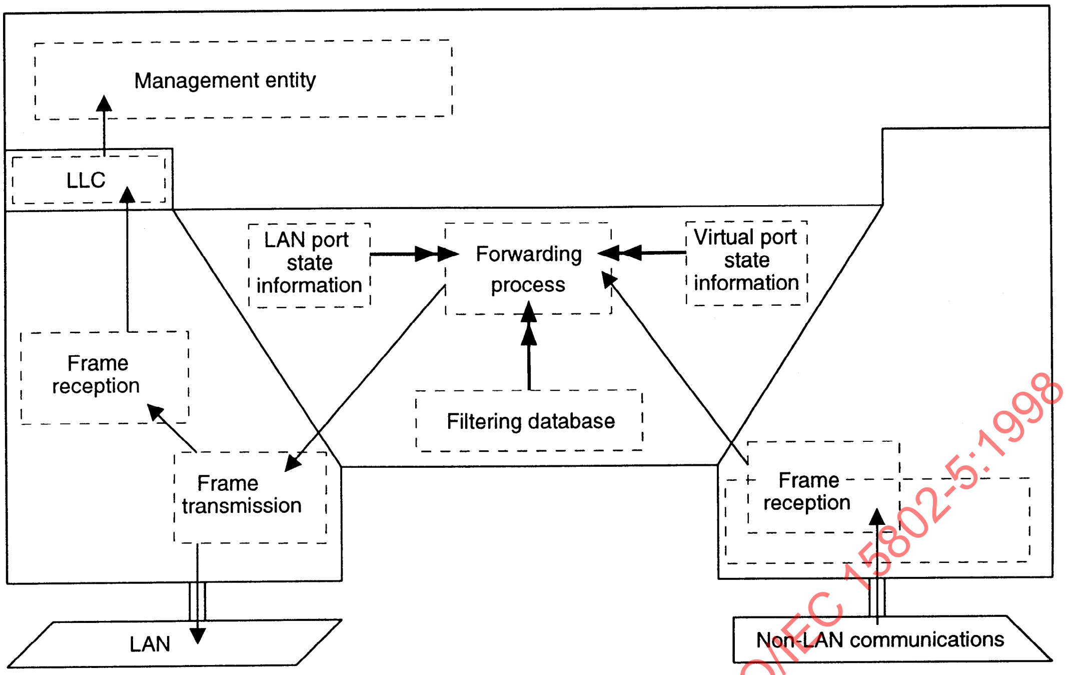


Figure 6-9—Reception of bridge management information at a virtual port of a remote bridge

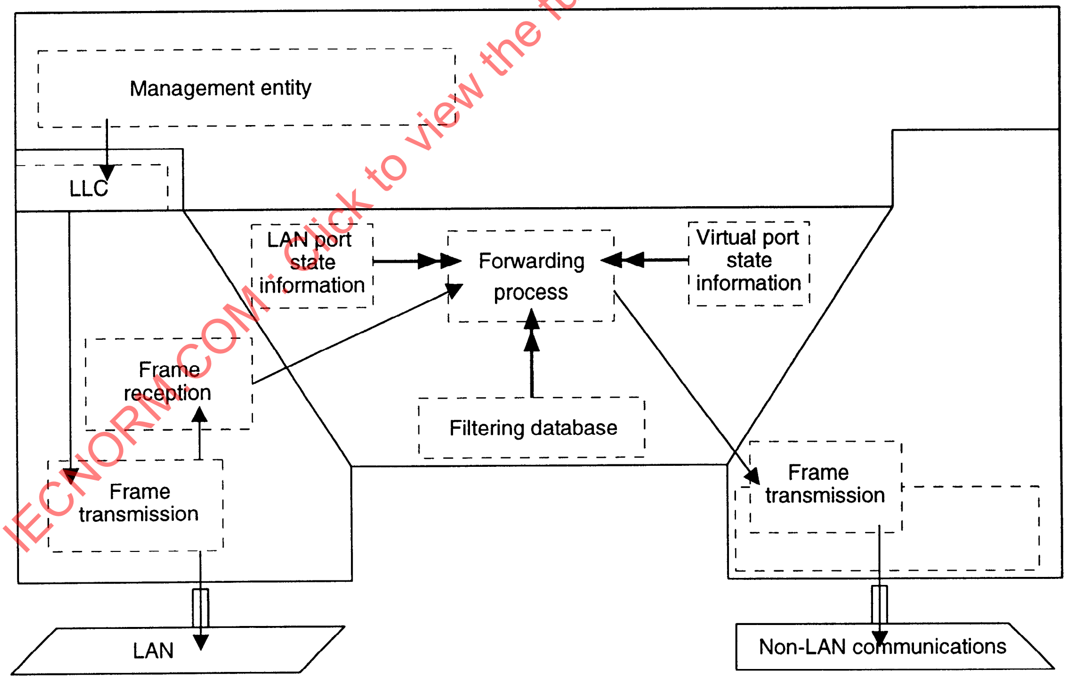


Figure 6-10—Transmission of bridge management Information through a virtual port of a remote bridge

6.4 Port states, active ports, and the active topology

State information associated with each bridge port governs whether or not it participates in relaying MAC frames. A port can be disabled by management, in which case it plays no part in the operation of the RB-LAN; a port that is not disabled can be dynamically excluded from participation in frame relaying by operation of the Spanning Tree Algorithm. If neither of these applies to a port, it is described as *forwarding*.

The *active topology* of an RB-LAN at any time is the set of communication paths formed by interconnecting the LANs, bridges, and group communications equipment by the forwarding ports. The function of the distributed Spanning Tree Algorithm is to construct an active topology that is simply connected relative to communication between any given pair of MAC addresses used to address end stations on the LANs.

Figure 6-7 illustrates the operation of the bridge protocol entity, which operates the Spanning Tree Algorithm and its related protocols, and its modification of port state information as part of determining the active topology of the RB-LAN. The port states associated with that determination of the active topology are specified in detail in 7.3.7 and 7.4.

Figure 6-4 illustrates the forwarding process's use of port state information: first, for a port receiving a frame, in order to determine whether the received frame is to be relayed through any other ports; and second, for another port in order to determine whether the frame is to be forwarded through that particular port.

The incorporation of end-station location information in the filtering database by the learning process also depends upon port states assigned as part of determining the active topology. If information associated with frames received on a port is to be incorporated in the filtering database, the port is described as being in a learning state; otherwise, it is in a non-learning state. Figures 6-5 and 6-6 illustrate the learning process's use of the port state information for a LAN port and a virtual port, respectively, in order to determine whether the station location information in received frames is to be incorporated in the filtering database.

6.5 Frame reception

Frames received on a LAN port of a remote bridge shall be discarded, or submitted to the forwarding process and/or learning process and/or LLC, as specified in ISO/IEC 10038: 1993.

Frames received, as M-UNITDATA indication primitives, on a virtual port of a remote bridge

a) Shall be discarded if their cluster_id parameters do not identify the remote bridge cluster to which the receiving port attaches; or

b) May be discarded if it is determined (from the parameters of the M-UNITDATA primitives) that they have been damaged so that they cannot be successfully delivered; or

c) If not discarded as in a) or b), shall be submitted to the forwarding process and the learning process.

NOTE—Discarding of frames from other clusters as in a) is necessary in order to preserve the simply connected active topology (6.4, 6.13.1, 6.13.4). Whether such frames can ever actually be received at a given real remote bridge is, in general, implementation-dependent; the statement in a), which is about the abstract model for any implementation, is needed in order to cover the full range of possible implementations. See also 8.3, C.7.3.4, and C.7.3.6.

6.6 Frame transmission

Frames are transmitted onto a LAN via a LAN port as specified in ISO/IEC 10038: 1993, by the individual MAC entity associated with the LAN port.

Similarly, the MAC service support function associated with each virtual port transmits the information corresponding to frames submitted to it, as M-UNITDATA request primitives, by the MAC relay entity.

6.7 Frame forwarding

The forwarding process forwards received frames to other ports, filtering them on the basis both of information contained in the filtering database, and of the states of the ports.

6.7.1 Forwarding conditions

A frame received on a LAN port or a virtual port and submitted to the forwarding process shall not be forwarded to any other port unless the receiving port is in Forwarding state (7.4).

Each frame received on a port that is in Forwarding state shall be submitted to the forwarding process, and queued for transmission on each port, and on no others, for which

a) The transmission port is in Forwarding state (7.4); and

b) The receiving and transmission ports are not both virtual ports attached to the same remote bridge group; and

c) either

   1) The filtering database indicates that frames with this value of the destination MAC address are to be forwarded through the transmission port; or

   2) The values of the source and destination MAC addresses are the same, and the bridge is configured to forward such frames;

except that

d) A frame shall not be queued for transmission on a LAN port if it conveys an MSDU of size greater than the maximum MSDU size supported by the LAN; and

e) A frame is not required to be queued for transmission on a virtual port if it conveys an MSDU of size greater than the maximum MSDU size supported by any of the destination remote bridges in the remote bridge cluster to which the virtual port is attached.

NOTES

1—The maximum MSDU size supported by a remote bridge can be greater than the maximum MSDU size supported by any of the LANs to which its LAN ports attach, if the bridge supports relaying between virtual ports belonging to two or more different groups.

2—One case of condition c)1) arises when a frame's destination MAC address is not in the filtering database (i.e., the destination MAC address is not known to the bridge). Such a frame is queued for transmission on all forwarding ports, apart from the receiving port and any subgroup ports attaching to the same group as the receiving port.

6.7.2 LLC duplicate address check

Like a local bridge, a remote bridge shall either

a) Filter frames whose source and destination MAC addresses have the same value, in order to localize traffic; or

b) Forward such frames, in order to support the optional LLC duplicate address check function.

The choice shall apply consistently to all LAN ports and virtual ports of a given remote bridge.

6.7.3 Queued frames

The forwarding process provides storage for queued frames, awaiting an opportunity to submit them for transmission to the MAC entity or MAC service support functions associated with each port. The order of queued frames shall be maintained for each user priority and each pair of source and destination MAC address values.

A frame queued by the forwarding process for transmission on a port shall be removed from that queue on submission for transmission on that port; no further attempt shall be made by the forwarding process to transmit the frame on that port even if the transmission is known to have failed. (Note that this does not preclude the use of retransmission for error recovery as part of the MAC service support functions over non-LAN communications equipment.)

A frame queued by the forwarding process for transmission on a port may be removed from that queue, and not subsequently transmitted, if the time for which buffering is guaranteed has been exceeded for that frame.

A frame queued by the forwarding process for transmission on a port shall be removed from that queue, and not subsequently submitted for transmission on that port, if that is necessary to ensure that the maximum bridge transit delay (see 5.3, C.5.3.6) will not be exceeded at the time at which the frame would be subsequently transmitted.

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

A frame queued for transmission on a port may be removed from that queue if it becomes known that the destination is not reachable.

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

6.7.4 Priority mapping

The forwarding process performs the mapping of the priority of forwarded frames (see 5.3, C.5.3.9) by determining the values of the user_priority and access_priority parameters in the M-UNITDATA request primitives used to relay frames.

The user_priority parameter in an M-UNITDATA request primitive (5.4) shall be either

— Equal to the value of the user_priority of the corresponding M-UNITDATA indication primitive, if that value was specified (e.g., if the frame was received on a virtual port, or on a LAN port using the token-passing bus or a token-passing ring access method); or

— Set to the value of the Outbound User Priority parameter for the transmission port, if the value in the corresponding M-UNITDATA indication primitive was unspecified (e.g., if the frame was received on a LAN port for a LAN using the CSMA/CD access method).

The access_priority parameter in an M-UNITDATA request primitive (5.4) shall be either

— Equal to the value of the user_priority parameter in the corresponding M-UNITDATA indication primitive; or

— Set to the value of the Outbound Access Priority parameter for the transmission port.

If bridge management is supported, the values of the Outbound User Priority and Outbound Access Priority parameters shall be capable of being set independently for each transmission port. The remote bridge shall have the capability to use the full range of values specified in Tables 3-1 and 3-2 of ISO/IEC 10038: 1993 for LAN ports, and all values in the range 0–7 for virtual ports. If bridge management is not supported, the remote bridge shall use the default values specified in Tables 3-1 and 3-2 of ISO/IEC 10038: 1993 for LAN ports, and the default value zero for each parameter of a virtual port.

NOTE—Some LAN access methods treat all values of access_priority equally and do not convey user_priority; virtual ports (see 5.6) can similarly limit support for one or both of these parameters. The Outbound User Priority and Outbound Access Priority parameters are still specified for these cases, to allow a consistent approach to the management both of virtual ports and of LAN ports of different MAC types.

6.7.5 Frames addressed to the remote bridge

When the MAC relay entity forwards a frame to a LAN port, and the frame's destination MAC address matches an individual MAC address associated with the port, the frame may be transmitted on the LAN. Alternatively, the physical transmission of the frame on the LAN may be suppressed. In either case, an MA_UNITDATA indication primitive shall be issued to the LLC entity associated with the LAN port, as though the frame had been received on the port.

When a frame is forwarded to a LAN port, and the frame's destination MAC address matches a group MAC address associated with the LAN port, the frame shall be transmitted on the LAN in the normal way. An MA_UNITDATA indication primitive shall be issued to the LLC entity associated with the LAN port, as though the frame had been received on the port.

NOTES

1—In each case (individual MAC address or group MAC address), the normal behavior of the MAC entity for the port will be to ensure that the LLC entity receives the frame, as required above. This is because the MAC service, ISO/IEC 15802-1: 1995, specifies that an MA_UNITDATA request containing a MAC address designating the MAC service access point at which the request is invoked shall result in an MA_UNITDATA indication at that MAC service access point.

2—The above never applies to MAC frames containing BPDUs. Such frames are never forwarded by a bridge's MAC relay entity.

6.8 The learning process

The learning process observes the source MAC addresses of frames received on each port and updates the filtering database conditionally on the state of the receiving port.

Frames received on ports are submitted to the learning process as specified in 6.5. When a frame is submitted, the learning process shall update a dynamic entry if one exists in the filtering database, and otherwise may create a dynamic entry, associating the port on which the frame was received with the source MAC address of the frame, if and only if

a) The port on which the frame was received is in a state that allows learning (7.4).

b) The frame's source MAC address is an individual MAC address.

c) No static entry for the MAC address already exists.

d) The resulting number of entries would not exceed the capacity of the filtering database.

If the filtering database is already filled up to capacity but a newly created entry would otherwise be made, an existing entry may be removed to make room for the new entry.

Figure 6-6 illustrates the operation of the learning process in including the station location information carried by a single frame, received on a virtual port, into the filtering database.

6.9 The filtering database

The filtering database holds filtering information which is either explicitly configured by management action or automatically entered. It supports queries by the forwarding process as to whether frames with given values of destination MAC address are to be forwarded to a given port. The specification given here is equivalent to that for local bridges in ISO/IEC 10038: 1993, extended to include all virtual ports within the scope of the filtering function.

Information is entered automatically into the filtering database by the learning process, and may also be entered automatically to reflect location information acquired by the learning processes in other remote bridges belonging to the same remote bridge cluster.

NOTE—The latter possibility can be viewed as allowing a cluster to implement a partially distributed filtering database. This can result in a reduction in the traffic conveyed across clusters (e.g., by avoiding flooding by one remote bridge which has timed out information, when other bridges retain current information because of traffic among them). This International Standard does not specify methods for performing such remote learning and updates.

The filtering database shall contain static entries (6.9.1), and shall be capable of containing dynamic entries (6.9.2). At most one type of entry shall exist for a given MAC address: a dynamic entry shall not be created if a static entry already exists for the same MAC address; and creation of a static entry shall cause the deletion of any dynamic entry for the same address.

When management of the remote bridge is supported, the filtering database shall be capable of being interrogated and updated by management using the operations specified in Clause 9. Alternatively, or additionally, interrogation and updating of the filtering database may be supported by local or private mechanisms.

6.9.1 Static entries

Static entries may be added to and removed from the filtering database under explicit management control. They are not automatically removed by any timeout mechanism.

A static entry specifies

a) The MAC address for which filtering is specified.

b) For each inbound LAN port and virtual port on which frames are received, a port map that specifies whether frames are to be filtered or forwarded to each outbound port.

The MAC addresses that can be specified shall include group addresses and the broadcast address.

6.9.2 Dynamic entries

Dynamic entries are created and updated by the learning process as described in 6.8. They may also be created and updated as a result of operations by the learning processes in other remote bridges, as described in 6.9.

A dynamic entry shall be removed after a specified time—the *ageing time*—has elapsed since the entry was created or last updated. A remote bridge's support for timeout values shall be as specified for local bridges in ISO/IEC 10038: 1993.

NOTE—Use of a short timeout value is specified in Clause 7, when the Spanning Tree Protocol is used to notify all bridges in an RB-LAN of topology changes. This allows the normal ageing time timeout, applicable when the topology does not change, to be long enough to cope with periods when end stations do not generate frames themselves, while not sacrificing the ability of the RB-LAN to continue to provide a service after automatic reconfiguration.

A dynamic entry specifies

a) The MAC address for which filtering is specified.

b) A port number.

Frames with the specified destination MAC address shall be forwarded only to the specified port. Only individual MAC addresses shall be present in dynamic entries.

6.9.3 Permanent database

The filtering database shall contain a permanent database, providing fixed storage for static entries; the filtering database shall be initialized with these static entries.

Static entries may be added to and removed from the permanent database by explicit management action.

6.10 Bridge protocol entity

The bridge protocol entity operates the Spanning Tree Protocol on the LANs to which a remote bridge's LAN ports are attached, and performs equivalent interchanges of information with other remote bridges in the same remote bridge group.

The bridge protocol entities of bridges attached to a given individual LAN in an RB-LAN communicate by exchanging bridge protocol data units (BPDUs), as specified in ISO/IEC 10038: 1993.

The bridge protocol entities of bridges belonging to the same group may communicate by exchanging BPDUs in accordance with the optional Extended Spanning Tree Protocol specified in Clauses 12 and 13; or they may use other means of achieving equivalent support of the Spanning Tree Algorithm across the group.

Figure 6-7 illustrates the operation of the bridge protocol entity including the reception and transmission of frames containing BPDUs, the reception and transmission of corresponding information (in BPDUs, or otherwise) over the non-LAN communications equipment of the group, the modification of the state information for individual ports, and notification of changes in active topology to the filtering database.

6.11 Bridge management

The remote bridge may provide facilities for management; these facilities and the operations that support them are specified in Clause 9.

Invocation of the management operations may be carried out by local or private means, or by use of standard management protocols. When ISO/IEC 15802-2: 1995 LAN/MAN Management is used, the protocol operations, identifiers, and values to be used in realizing the management operations shall be as specified in Clause 10 of this International Standard.

Bridge management is modeled as being performed by means of the bridge management entity. Figures 6-8 through 6-10 illustrate use of the MAC service in support of (standard or private) protocols for communicating with the bridge management entity. Communication both via a LAN port and via a virtual port is illustrated.

6.12 Addressing

Use of MAC addresses in frames transmitted and received on the LANs to which the LAN ports of a remote bridge are attached shall be as specified in ISO/IEC 10038: 1993. The reserved LLC Address (Standard LSAP) assigned to the Bridge Spanning Tree Protocol shall be used, as specified in ISO/IEC 10038: 1993, in the LLC PDUs used to convey BPDUs transmitted over a LAN by the bridge protocol entity of a remote bridge.

Where a remote bridge's bridge management entity is addressed by one or more specific MAC addresses associated with LAN ports of the remote bridge, as specified in ISO/IEC 10038: 1993, frames received on a forwarding virtual port and addressed to the bridge management entity shall be forwarded to the addressed LAN port in accordance with the forwarding conditions in 6.7. When such forwarding occurs, the frame is submitted to the LLC entity that supports the bridge management entity (the frame is not necessarily transmitted on the LAN to which the port attaches; see 6.7.5).

Similarly, a management request conveyed in a frame received on a forwarding virtual port and carrying the All LANs Bridge Management Group Address specified in ISO/IEC 10038: 1993 shall be forwarded to the remote bridge's LAN ports in accordance with the forwarding conditions in 6.7. One instance of the frame is submitted to the bridge management entity for each LAN port to which it is forwarded and which has a specific MAC address by which the bridge management entity is reachable.

Use of addresses for protocol functions within a remote bridge group is outside the scope of this International Standard.

6.13 Model of remote bridge interconnection

6.13.1 Roles and objectives of the model

The interconnection model has three roles:

- As an aid to describing and understanding the operation of a set of intercommunicating remote bridges.
- As a basis for establishing a structure for management of remote bridges and RB-LANs.
- As a framework for operation of the optional Extended Spanning Tree Protocol.

The objectives of the model are to

a) Accommodate the full variance in, and to allow the best use of, applicable non-LAN communications technologies.

b) Allow a range of sizes and complexities of remote bridges, and in particular to allow simple remote bridges to be interconnected with more complex remote bridges.

c) Provide for manageability of real RB-LAN implementations, by compatible extension of the corresponding features of ISO/IEC 10038: 1993; in particular, to

1) Provide for control of the spanning tree's configuration, across a complete RB-LAN; and

2) Provide for control of the filtering of user traffic within the RB-LAN.

6.13.2 Elements of the model

Interconnection of remote bridges is modeled in terms of *virtual ports*, *remote bridge groups*, and *remote bridge clusters*. Remote bridge groups and remote bridge clusters express, respectively, static and dynamic aspects of the configuration of a set of remote bridges cooperating in support of the MAC service. Virtual ports are abstractions of the communications functions between remote bridges, and are used in expressing the peer-to-peer communication between remote bridges in a group or cluster. The logical interconnection of a group's remote bridges by their virtual ports is described in terms of *subgroups*.

NOTE—It is important to note that virtual ports and subgroups are logical, abstract representations of the capability for communication among remote bridges. Virtual ports do not represent points of attachment to particular physical communications links, nor do subgroups represent discrete sets of such links. (See 6.13.5, 6.13.6, and Annex C.)

The modeling principles for the static and dynamic aspects of group configuration are described in 6.13.3 and 6.13.4, respectively. The functions of virtual ports in the overall model of remote bridge interconnection are described in 6.13.5, and basic examples of the model are provided in 6.13.6.

6.13.3 Static configuration: remote bridge groups, virtual ports, and subgroups

6.13.3.1 Virtual ports and subgroups

A remote bridge attaches to a remote bridge group through one or more virtual ports. A remote bridge may be attached to more than one group; in this case, each virtual port attaches to one and only one of the groups.

Each virtual port represents a remote bridge's ability to communicate with a particular, configured set of the other remote bridges in the group to which the port attaches. Such a set of remote bridges is termed a *subgroup*. The interconnection structure of the group is constrained so that each remote bridge in a subgroup has a single virtual port representing its ability to communicate with all the other bridges in the subgroup, and with no others.

NOTE—The above constraint means that a subgroup is fully connected, and simply connected, via the attaching virtual ports. It is fully connected because each member of the subgroup can communicate with every other member; and it is simply connected because there is a unique pair of virtual ports representing communication between any given pair of members.

A subgroup does not provide a means for a bridge belonging to it to discriminate between the other bridges with which it communicates: information transmitted via a virtual port is capable of being received at all the peer virtual ports in the subgroup. The part of the group's communications functions corresponding to a subgroup therefore interconnects the subgroup members to provide communication among them that has similar properties to those of the communication provided by a LAN. The support for this LAN-like interconnection is referred to as the *subgroup communications capability.*

6.13.3.2 Groups and subgroups

A group as a whole is also required to be fully and simply connected: that is, each remote bridge attached to a given group shall belong to one or more subgroups, such that each other remote bridge in the group belongs to exactly one of those subgroups.

In a given group, communication takes place between all the bridges' bridge protocol entities, except for those that have had their virtual ports disabled.

This International Standard constrains the interconnection structure of a group only to the extent that any group shall be fully and simply connected via its virtual ports, as described above. The detailed capabilities of any remote bridge, or of any set of remote bridges that can form a group, are at the implementors' choice in terms of, for example,

— The maximum number of virtual ports by which a given remote bridge can attach to a group.
— The maximum size of group in which a given remote bridge can participate.
— The variety of ways in which a group can be structured in terms of subgroups.

An important special case of the possible subgroup structures for a group is that where there is just one subgroup, with each remote bridge attaching to the group by just one virtual port. A group configured in this way is termed a *virtual LAN.* A virtual port that attaches to a virtual LAN is termed a *virtual LAN port.*

When a remote bridge attaches to a group that is not a virtual LAN (i.e., when each remote bridge has multiple virtual ports, and the group consists of multiple subgroups), each of its virtual ports is termed a *subgroup port.* When a subgroup port represents the capability for communication with just one other remote bridge, it is termed an *individual virtual port.* A group in which every remote bridge is attached only by individual virtual ports is termed a *virtual mesh.*
When a subgroup port represents the capability for communication with two or more other remote bridges, it is termed a *multipeer virtual port.* A group containing remote bridges attached by multipeer virtual ports is termed a *mixed-configuration group.*

NOTES

1—A number of the properties of remote bridges that attach to virtual LANs are simpler than those for remote bridges interconnected by multiple subgroups. The requirements in this International Standard therefore frequently distinguish between the virtual LAN case and the other, more general, structures.

2—The following are consequences of the definitions just given:

a) Every virtual port is either a virtual LAN port or a subgroup port.

b) A group containing only two remote bridges is a virtual LAN.

c) An individual virtual port always connects to another individual virtual port, because of the fully and simply connected structure of subgroups; the ports therefore connect their bridges in a two-member subgroup.

d) A mixed-configuration group always contains some individual virtual ports as well as at least one multipeer virtual port.

3—Groups and their virtual port attachments define the static configuration of an RB-LAN. This definition—of how remote bridges are associated together in groups, and of each remote bridge's configuration of virtual ports, including subgroup membership—is essentially an administrative matter, subject of course to any limitations imposed by the technical capabilities of the equipment to be used, and subject to the full-connectivity requirements on subgroups.

4—For a given set of more than two remote bridges and their interconnecting communications equipment, there is not in general a unique configuration either of groups or of virtual ports: actual configurations, and the ranges of possible configurations, depend upon user requirements and upon implementation capabilities, including those for RB-LAN management. (See Annex C for examples.)

6.13.4 The active topology and remote bridge clusters

As defined in 6.4, the active topology of an RB-LAN at any time is the set of communication paths formed by interconnecting the LANs, bridges, and group communications equipment by the forwarding ports. The function of the distributed Spanning Tree Algorithm is to construct an active topology that is simply connected relative to communication between any pair of LAN end stations using given MAC addresses. When a group belongs to an RB-LAN containing other external communication paths between two or more of the remote bridges in the group, it is sometimes necessary to partition the group in order to construct the simply connected active topology.

Dynamic aspects of the configuration of remote bridge groups in an RB-LAN, relating to the selection of virtual ports to be forwarding, are expressed in terms of remote bridge clusters. The Spanning Tree Algorithm and the associated cluster-formation rules (see 7.3) configure the remote bridges of a given group into one or more disjoint subsets. Such a subset is

a) An active cluster consisting of two or more remote bridges, each of which either is relaying frames to the other bridges in the cluster, through its virtual ports that have been selected to be forwarding, or is preparing to perform such relaying following a configuration change; or

b) A degenerate cluster, containing a single isolated remote bridge with one or more virtual ports selected to be forwarding, but with none of its peer virtual ports on other bridges belonging to the same cluster; or

c) A single isolated remote bridge that is not relaying frames, or preparing to relay frames, to any other bridge in the group.

No remote bridge belongs to two clusters of the same group. Consequently, in any group, no relaying of frames occurs through pairs of remote bridges belonging to different clusters of that group.

NOTES

1—This last property, which is a consequence of the Spanning Tree Algorithm and cluster-formation rules, ensures that the active topology for end-station to end-station frame transfer is simply connected.

2—A cluster can, and often will, consist of all the remote bridges in a given group. In particular, this will be the case when there are no better communication paths, external to the group, between any members of the group. The distinction between groups and clusters is necessary only in order to handle the possibility of such external paths.

3—In a mixed-configuration group (neither a virtual LAN nor a virtual mesh), the active topology for relaying of frames is determined by the clusters, rather than by the selection of virtual ports to be forwarding or blocking. It is possible for peer subgroup ports to belong to different clusters, and so to be selected as forwarding; however, frames are never relayed through the two bridges connected by such ports—see 6.5 a), 5.4 h), 8.1, and 8.3. In virtual LAN and virtual mesh configurations, the active topology determined by the clusters in fact corresponds to that determined by the selection of virtual ports to be forwarding or blocking. In any configuration, it is possible that degenerate clusters as in b) will occur; it is an implementation choice whether or not there should be mechanisms to detect this and to suppress the actual transmission of frames across the group, since any frames transmitted will only be discarded on receipt.

4—The fact that a cluster is fully connected, as stated in a), does not mean that direct communication paths are required between each pair of remote bridges, but only that the cluster provides some path between each pair. This would be modeled as a function of the group communications entity (see 6.2.4, and Annex C).

5—A remote bridge that is isolated within one group as in b) or c) could be a member of an active cluster in another group to which it is attached.

6—A remote bridge that belongs to two or more groups can belong to two or more clusters, in different groups, and relays frames between those clusters as necessary.

6.13.5 Functions of virtual ports

Virtual ports provide for management control over the communication between the remote bridges in a given group. Parameters associated with the ports control the Spanning Tree Algorithm's behavior and selective filtering, as described in 6.13.5.1 and 6.13.5.2, respectively.

NOTES

1—These are the only functions of virtual ports. Virtual ports do not relate in any direct way to the physical topology or other features of the underlying non-LAN communications, although the choice of parameter values for specific virtual ports will often reflect some such features.

2—These two kinds of control relate, respectively, to operation of the bridge protocol entity and of the MAC relay entity, and also to objectives e) and f) in 6.13.1.

6.13.5.1 Spanning tree configuration

Virtual ports provide for control over how the Spanning Tree Algorithm can partition the group into clusters and/or isolated remote bridges, in the presence of competing external communication paths joining two or more of the remote bridges. Where multiple paths exist through the RB-LAN, computation of the Spanning Tree Algorithm uses path cost information, which is associated with each port, in deciding which paths are to be blocked in order to provide the necessary simply-connected active topology. Path costs associated with virtual ports can control whether a path through a group becomes blocked—with partitioning of the group into separate clusters or isolated remote bridges—or whether the path through the group remains open, with the external path (e.g., through a different group) becoming blocked.

NOTE—The description here, and throughout consideration of the Spanning Tree Algorithm and Protocol, refers to logical communication paths, not to physical links.

6.13.5.2 Selective filtering

Virtual ports provide for statically specified selective filtering of traffic, so that frames with certain destination MAC addresses received at a given remote bridge can be specified not to be forwarded to certain other remote bridges of the group. Each virtual port of a given remote bridge connects to one or more other remote bridges, in the same subgroup; information entered in the filtering database by the LAN management can determine whether frames with a given destination MAC address are forwarded to the other remote bridges of the subgroup (provided they are in the same cluster), or whether those frames are filtered.

6.13.6 Examples of configurations of remote bridge groups and virtual ports

Figures 6-11 and 6-12 illustrate the concepts relating to static configuration of groups and virtual ports, for some simple cases. Figures 6-13 and 6-14 illustrate some of the consequences for dynamic configuration that arise from the static choices. A more extensive set of examples and explanations can be found in Annex C.

[In all the diagrams, ports are represented by the small rectangles inside the boxes representing bridges; groups are represented as unshaded ellipses. The circles represent subgroups, and the solid lines joining virtual ports and subgroups represent the possible, logical paths along which frames can be relayed. Each of the LB-LANs (L1, L2, etc.) may contain local bridges.]

6.13.6.1 Static configuration examples

Figure 6-11 shows the simplest possible remote bridge group: the pair of remote bridges B1 and B2 have a single LAN port each, attaching to the LAN L1 or L2, and a single virtual port attaching to the group (i.e., representing the communication path to the other member of the pair). The group is a virtual LAN, since each bridge attaches to it by a single virtual port.

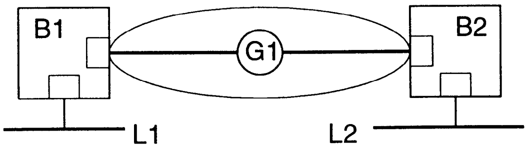
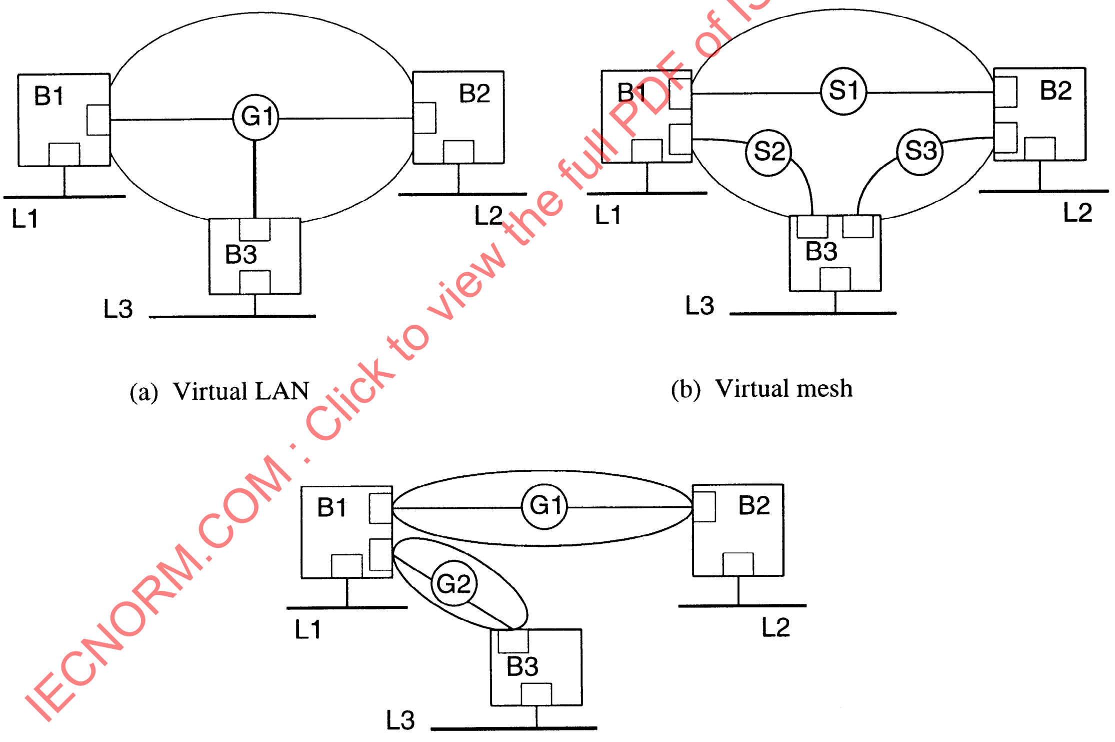


Figure 6-11—Simplest remote bridging configuration: group of two two-port bridges

Figure 6-12 shows some of the possible configurations when a third remote bridge B3 is added, attaching to a LAN L3.



(a) Virtual LAN

(b) Virtual mesh

(c) Two simple groups

Figure 6-12—Some configurations of groups and virtual ports for three remote bridges

Case (a) of Figure 6-12 shows a virtual LAN configuration, in which the three remote bridges form a single subgroup, and each remote bridge attaches to the group by a single virtual port that represents the remote bridge's ability to communicate with both of the others.

Case (b) shows a virtual mesh configuration, where each remote bridge has separate virtual ports, each representing the bridge's ability to communicate with one other remote bridge in the group. Each pair of these peer individual virtual ports connects the corresponding bridges in a two-member subgroup.

Finally, case (c) adds the third remote bridge as a member of a new group, G2. Traffic between L2 and L3 would have to be relayed between the two groups through the virtual ports of B1.

In all the configurations shown, since no other bridges interconnect the LANs, each complete group will operate as a single cluster.

NOTE—It is emphasized again that these configurations express a purely logical view of a remote bridge's ability to communicate selectively with other members of the group. The lines shown joining virtual ports via subgroups do not represent individual physical communications links. Annex C contains examples of how physical configurations could map to a variety of group structures.

6.13.6.2 Spanning tree configuration example

Figure 6-13 shows the addition of another group that can provide communication between L1 and L2, and its possible effects on the full-mesh configuration of Figure 6-12 (b). In order to keep the RB-LAN simply connected, it is necessary for the Spanning Tree Algorithm to break the loop through A1 – A2 – B2 – B1. For the purpose of the examples, it is assumed that the port parameters are such that the B1 – B2 path is broken (otherwise, the B1/B2/B3 configuration operates exactly as in Figure 6-12).

Figure 6-13 shows the two possible ways in which the group could partition, producing a two-bridge cluster and a single isolated bridge. The path costs associated with the ports of B1, B2, and B3 determine which outcome is selected; see 6.13.5.1 and 7.3. (Clusters within groups are represented as shaded ellipses. Dotted lines between virtual ports indicate paths in the group that are not active as part of a cluster; such a path is inactive because at least one of the pair of ports connected by it is not active.)

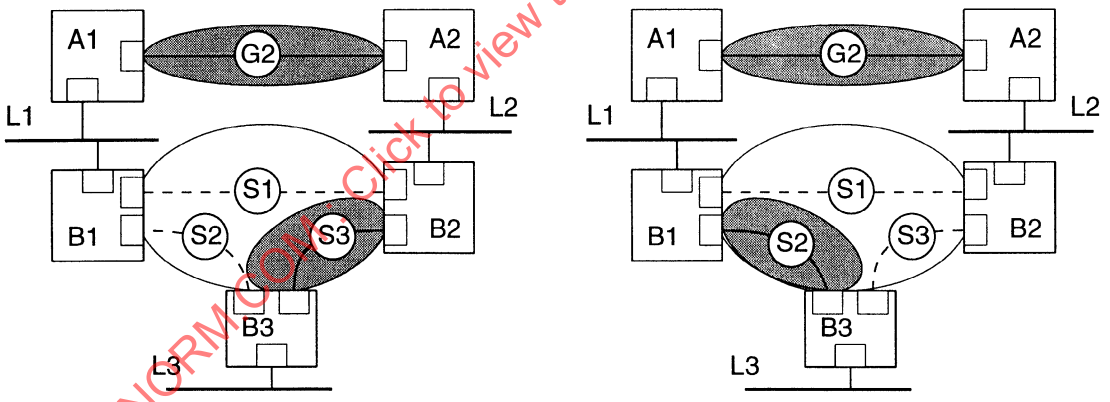


Figure 6-13—Partitioning of a virtual mesh group

6.13.6.3 Static filtering example

Figure 6-14 illustrates the use of static filtering to support partitioning of an RB-LAN for traffic with a given group MAC address. Such a group address might be, for example, for LAN station loading, or for a Network layer routing protocol.

In the illustration, the RB-LAN has LANs attached to four remote bridges, and the multicast traffic is to be split into two disjoint domains, one consisting of the LANs attached to B1 and B4, and the other consisting of those attached to B2 and B3. (The arrows show the required traffic flows for this MAC address.)

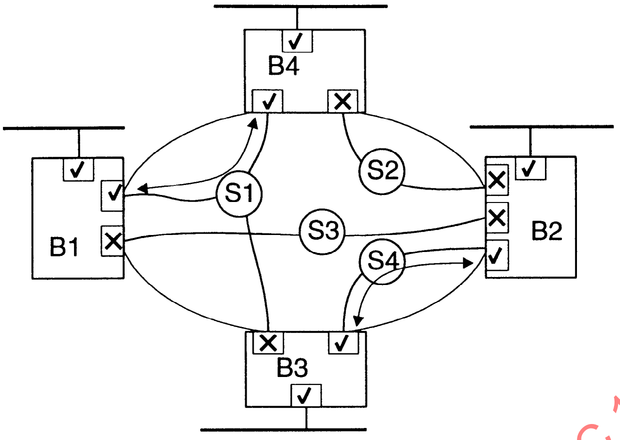


Figure 6-14—Partitioning of a group MAC address space by static filters

Filtering database entries for the MAC address in question need to be established to specify forwarding only through the virtual ports by which B1 and B4 access each other, and through those by which B2 and B3 access each other, with filtering on all other paths. This is possible provided that, as shown, the remote bridges have multiple virtual ports to allow discrimination, by means of the virtual port entries in the filters, between the various destination bridges in the group. In the diagram, ports marked with a cross filter all frames received with the given address, so that they are not forwarded through any other port of the bridge; ports marked with a tick forward all frames received with the given address through all other ticked ports, and not through any crossed ports. (The filtering is controlled by the port maps associated with each port, for the given address; see 6.9.1.)

Note that B3 can participate successfully in this arrangement despite being in the same subgroup, S1, as B1 and B4: the frames forwarded by B1 and B4 are filtered at B3. (In practice, an implementation might well optimize its use of group communications resources by propagating the filtering information from B3 to B1 and B4, so as to avoid actually transmitting frames that will only be discarded on reception.)

7. The Spanning Tree Algorithm and Protocol

7.1 Introduction

Remote bridges participate in the same distributed configuration algorithm as local bridges, and operate the same protocol over the LANs to which their LAN ports are attached; the algorithm and protocol reduce the bridged LAN topology to a single Spanning Tree. The basic Spanning Tree Algorithm that underlies the Spanning Tree Protocol specified in ISO/IEC 10038: 1993 is complemented by extensions to handle the dynamic formation of remote bridge clusters for interconnecting remote bridges.

C.7.1 provides background information on the requirements that the (extended) algorithm is designed to meet. The remainder of this clause states the basic requirements on bridges operating the algorithm (7.2); provides an overview of the algorithm's operation (7.3); defines the states and parameters in terms of which the algorithm is specified (7.4 and 7.5); specifies how those states and parameters are maintained and updated by a remote bridge in support of the algorithm (7.6); specifies how a remote bridge transmits BPDUs on its LAN ports, and responds to BPDUs received on its LAN ports, in its maintenance of the states and parameters (7.7); specifies how management operations relating to the bridge protocol entity affect the states and parameters maintained by a remote bridge (7.8); specifies the encoding and validation of BPDUs transmitted and received on LAN ports (7.9, by reference to ISO/IEC 10038: 1993); and specifies requirements on the performance of remote bridges in an RB-LAN and on the settings of associated parameters (7.10).

NOTE—C.7.3.2 through C.7.3.12 contain several examples of the operation of the Spanning Tree Algorithm in an RB-LAN. With a few exceptions, this clause does not make specific references to C.7.3, but the reader is strongly recommended to refer to those examples as illustrations of the concepts and procedures specified in this clause.

7.2 Requirements of the remote MAC bridges

The requirements for bridge protocol operation are as specified for local MAC bridges in ISO/IEC 10038: 1993, with the additional requirement that there shall be a port identifier for each virtual port, distinct from the port identifier of any other LAN port or virtual port of the same bridge.

The requirements that allow the configuration of the Spanning Tree active topology to be managed are

a) A means of assigning the relative priority of each remote bridge within the set of bridges in the RB-LAN.

b) A means of assigning a relative priority to each LAN port and virtual port within the set of ports of an individual remote bridge.

c) A means of assigning a path cost component to each LAN port and virtual port.

These parameters can be set by management, when bridge management is supported.

The structure of bridge identifiers and port identifiers is as specified in ISO/IEC 10038: 1993.

7.3 General description

7.3.1 Overview

7.3.1.1 The active topology

The Spanning Tree Algorithm, with the cluster extensions, configures the arbitrarily interconnected components of an RB-LAN into an active topology (see 6.4) that is simply connected for end-station to end-station transfers. It does this by selecting the state of each port that is not disabled to be either forwarding or blocking. Frames transmitted by end stations can pass between different LANs and remote bridge groups only via ports that are forwarding; no frames can pass through a port that is blocking. (The state, forwarding or blocking, applies to both directions of possible flow through a port.)

The selection of ports to be made blocking cuts any looping paths that are present in the basic topology. The remaining ports, selected to be made forwarding, interconnect the bridges, LANs, and groups in an active topology in which the forwarding LAN ports are configured as the links in a tree. The root node of the tree is a bridge, and the other nodes are alternating levels of LANs, and bridges or clusters.

Within a cluster, by definition, all the virtual ports by which the remote bridges attach to the cluster are selected to be part of the active topology. However, for the purposes of constructing and maintaining the topology, the Spanning Tree Algorithm selects a subtree within the fully connected topology of the cluster, in which a subset of those virtual ports are the links. The root node of the subtree is the remote bridge nearest to the root of the full tree. The other nodes of the subtree are alternating levels of subgroup communications capabilities, and the other remote bridges of the cluster.

Because the LAN-connected part of the RB-LAN is configured as a tree, and because the forwarding conditions [see 6.7.1 b)] ensure that frames traverse any cluster in a single hop, the active topology as a whole is simply connected for the transfer of user frames between end stations on the LANs.

7.3.1.2 Configuration changes

Changes to priority or path cost values, and addition or removal of ports or bridges, result in general in a new active topology: ports that were previously blocking can become forwarding, and vice versa, to break newly formed loops or to provide alternative paths where connectivity has been lost.

NOTE—A transition from forwarding to blocking is immediate, but it is necessary to introduce a delay between selecting a blocking port to be forwarding and actually making it forwarding; see 7.3.7 and 7.3.8.

7.3.1.3 Relationship between the spanning tree and the active topology

Computation of the active topology can be considered to take place in three logical phases

a) Computation of the basic spanning tree

b) Formation of clusters

c) Selection of port states

In the first phase, the basic Spanning Tree Algorithm of ISO/IEC 10038: 1993 is used to construct a strict tree of bridges, LANs, and subgroup communications capabilities linked by bridge ports. This assigns one of three spanning tree roles to each enabled bridge port.

In the second phase, no further action is required for LAN ports or for virtual LAN ports. For subgroup ports, this phase determines (on the basis of the spanning tree structure) how remote bridges are to be dynamically configured together into clusters.

In the third phase, the spanning tree roles assigned to LAN ports and virtual LAN ports, and the cluster membership for subgroup ports, are used to determine for each port whether its state is to be forwarding or blocking.

NOTES

1—These phases involve only successive phases of processing applied to the same configuration information. Thus, for example, there are no protocol exchanges required between the phases, in a realization of the computation.

2—The second phase is needed in order to handle remote bridges that are attached to their groups by two or more virtual ports. To achieve the full relaying connectivity that characterizes a cluster [see 6.13.4 a)], such a bridge needs to be able to forward frames through all of its virtual ports that provide communication with other remote bridges in the same cluster. In general, applying the LAN-port rules on the basis of spanning tree roles assigned to the virtual ports would only select a subset as forwarding. The determination of cluster membership, and the associated rules for selecting port states, allow the correct full set of virtual ports to be made forwarding.

7.3.1.4 Relationship to protocols

The Spanning Tree Algorithm itself is an abstract algorithm applied to the arbitrarily configured priorities and path costs of all the (non-disabled) bridges and ports in the RB-LAN. The algorithm is given a practical realization by means of the Spanning Tree Protocol operating over LANs, and by equivalent protocols operating in remote bridge groups (one such protocol is the optional Extended Spanning Tree Protocol specified in Clauses 12 and 13). These protocols allow a distributed computation of the spanning tree, by each bridge independently using locally configured information in conjunction with information exchanged with its adjacent bridges.

The abstract algorithm is described in 7.3.2; the principles of how the protocols are used to exchange information between bridges in support of the Spanning Tree Algorithm, and of how ports change state between forwarding and blocking, are specified in 7.3.3 through 7.3.8. More detailed specifications of the requirements on the information to be recorded and exchanged by bridges in order to support the algorithm are in 7.4 through 7.6, and 7.8. The particular mapping of the logical information and operations to the transfer of BPDUs via the LAN ports of a remote bridge is specified in 7.7; as observed on the LAN, this operation of the Spanning Tree Protocol is identical to that specified for local MAC bridges in ISO/IEC 10038: 1993.

7.3.2 Computation of the active topology

7.3.2.1 Spanning tree fundamentals: bridge priorities and path costs

The stable active topology of an RB-LAN is determined by

a) The unique bridge identifier associated with each bridge.

b) The path cost associated with each port.

c) The port identifier associated with each port.

Each bridge in an RB-LAN has a unique bridge identifier containing a configurable priority component, and the set of all such identifiers is totally ordered (i.e., given any pair of distinct bridge identifiers, one is always of higher priority than the other). The bridge with the highest-priority identifier is selected as the root bridge of the Spanning Tree.

Every LAN port and virtual port in the RB-LAN has a configurable path cost associated with it. Given that the root is determined, as above, the spanning tree structure is primarily determined by these path costs.

Each bridge, LAN, and subgroup in the RB-LAN has an associated root path cost. For each possible path through the ports, LANs, and subgroups of the RB-LAN from the root bridge to the bridge, etc., in question, the path cost is calculated: this is the sum of the path cost values of each LAN port and virtual port at which a frame would be received if it traversed the path from the root to the bridge, etc., in question. The root path cost is the lowest cost among the possible paths. (The way in which the algorithm performs tie-breaking to select a single path, when there are multiple paths of equal lowest cost, is specified in 7.3.2.3.)

The spanning tree then consists of the complete set of least-cost paths from the root to each other bridge, etc., in the RB-LAN.

NOTE—The computation guarantees that the set of least-cost paths does indeed form a tree.

7.3.2.2 Spanning tree roles: designated ports, root ports, alternate ports, and designated bridges

Each bridge port that belongs to a spanning tree path is assigned the role of *designated port* if a frame traversing that path from the root would be transmitted via the port, and is assigned the role of *root port* if such a frame would be received via the port. All other enabled bridge ports are assigned the role of *alternate port*.

A bridge that has one or more designated ports is termed the *designated bridge* for each LAN and subgroup to which a designated port attaches. Each LAN or subgroup has a single designated bridge, which is the adjacent node of the spanning tree in the direction of the root.

Each bridge other than the root bridge has exactly one root port, which is the port that offers the least-cost path from the bridge to the root (after application of any tie-breaking needed, see 7.3.2.3).

Each root port connects, via a LAN or subgroup communications capability, to a unique designated port, on the adjacent bridge on the least-cost path to the root. Designated ports also attach to LANs that form leaf-nodes of the spanning tree (i.e., LANs that have no bridges attached to them by root ports); and they can attach as isolated ports to remote bridge groups, when their peer remote bridges do not belong to the same cluster.

Each alternate port also connects (via a LAN or subgroup communications capability) to a unique designated port on an adjacent bridge. However, the path through that port and bridge to the root has a cost that is higher than the root path cost, or equal to the root path cost where tie-breaking has been invoked.

7.3.2.3 Tie-breaking

It is possible, and in practice could be quite likely, that a bridge will find two or more of its ports offering equal lowest root path costs. In this case, the unique bridge identifiers of the adjacent bridges on the respective paths to the root determine which path is selected; the path through the highest-priority bridge wins.

It is still possible to have multiple paths of equal cost and priority after applying the above discrimination (e.g., where an adjacent bridge is attached by two ports to the same interconnecting LAN). In this case, selection is on the basis of the unique port identifiers associated with each of the ports on the adjacent bridge through which the least-cost paths pass; again, the highest-priority port identifier wins.

Finally, in the event that the above second stage of tie-breaking still yields multiple least-cost paths, the port identifiers of the candidate root ports are used; as before, the highest-priority port identifier wins.

NOTE—The last tie-breaking possibility is required only for LAN ports and not for virtual ports, since if two virtual ports on a given remote bridge connect to the same adjacent remote bridge, they can do so only as members of two different groups and, therefore, they connect to different virtual ports on the adjacent remote bridge.

7.3.2.4 Cluster formation, primary bridges

Each remote bridge cluster consists of a set of remote bridges belonging to the same group and forming a subtree of the spanning tree. The remote bridge nearest to the root bridge is termed the *primary bridge* for the cluster. The cluster consists of those remote bridges, in the same group as the primary bridge, that have least-cost paths to the root passing through the primary bridge.

7.3.2.5 Port state selection

Root ports and designated ports are selected to be made forwarding, whether they are LAN ports or virtual ports.

Each LAN port or virtual LAN port that is an alternate port is selected to be made blocking.

A subgroup port that is an alternate port is selected to be made blocking if it is known that none of its peer virtual ports attaches to the same cluster as does the port itself. A subgroup port that is an alternate port is selected to be made forwarding if it has a peer virtual port that attaches to the same cluster, or if it is not known that there is no such peer virtual port.

NOTE—In general, full information about the cluster attachments of an alternate port's peer virtual ports is not necessarily readily available for a subgroup port, except in two cases: (a) when all the ports by which a bridge attaches to a group have been selected as alternate ports, and (b) when the port is an individual virtual port. In case (a), the bridge is isolated in the group, and all the virtual ports are selected as blocking. In case (b), the virtual port is selected as blocking if its peer virtual port attaches to a different cluster, and as forwarding if the peer virtual port attaches to the same cluster. Peer virtual ports can belong to different clusters, when (and only when) the subgroup to which they attach has been partitioned because of an alternative path external to the group (see Annex C, particularly Figure C-10).

7.3.3 Protocol support for the Spanning Tree Algorithm

The Spanning Tree Algorithm is implemented by a combination of local computations in the bridges of an RB-LAN, and exchanges of information between adjacent bridges. This International Standard specifies those computations and exchanges for remote bridges, in terms primarily of

a) Logical parameter information maintained by each remote bridge.
b) The structure of logical information exchanged between remote bridges.
c) Transmission and reception, at LAN ports on remote bridges, of BPDUs of the Spanning Tree Protocol specified for local bridges in ISO/IEC 10038: 1993.
d) The possible observation, using bridge management, of many of the parameter values in item a).

Taken together, items a) through c) specify constraints on the operation of any protocols used by bridges in support of the Spanning Tree Algorithm. These constraints ensure that the spanning tree is constructed, maintained, and reconfigured correctly, and that the protocol operations by remote bridges via their LAN ports are compatible with operation of the Spanning Tree Protocol by local bridges. The Spanning Tree Protocol itself satisfies these constraints, in relation to the operation of local bridges. The optional Extended Spanning Tree Protocol specified in Clauses 12 and 13 is one way in which remote bridges can satisfy these constraints when implementing support for the Spanning Tree Algorithm.

The three major aspects to any such protocol operation in support of the Spanning Tree Algorithm are as follows:

— Determination and maintenance of the stable active topology
— Detection, or instigation, of topology changes
— Notification of topology changes

The first of these involves each bridge in receiving and transmitting configuration information; received information can cause recomputation of a bridge's local parameter information, with related changes in its subsequently transmitted information. (See 7.3.4.)

The second is largely a local matter for each bridge, depending on timeouts and on local bridge management operations; it will often be followed by recomputation of the active topology. (See 7.3.5.)

The third again involves bridges cooperatively receiving and transmitting information to ensure that topology changes are correctly notified and acted upon. (See 7.3.6.)

The purpose of computing the spanning tree is to control the relaying of MAC frames by selecting ports to be in forwarding or blocking states (7.3.1.1, 7.3.2.5). Because there are delays in propagating information about configuration changes in any real RB-LAN, the state of a port cannot always be changed immediately according to changes in its spanning tree role: some additional rules apply (see 7.3.7 and 7.3.8).

7.3.4 Determining the active topology

7.3.4.1 Configuration messages and Configuration BPDUs

The topology information that bridges communicate among each other is logically structured in *configuration messages*. A configuration message expresses the information originated by the root and propagated throughout the bridged LAN, being updated by each bridge that propagates it.

A configuration message is transferred over a LAN by encoding the transmitting bridge's current information in a Configuration BPDU, transferred in a MAC frame, as specified in ISO/IEC 10038: 1993. Configuration message information is also propagated within each remote bridge group; when the optional Extended Spanning Tree Protocol is used, the information is similarly conveyed in the (extended) Configuration BPDUs of that protocol. BPDUs are not relayed by bridges, but the configuration message information in them can be used by a bridge in calculating its own configuration messages for transmission, and can stimulate that transmission.

7.3.4.2 Spanning-tree priorities

Configuration messages are ordered in priority, according to four components. In order of decreasing significance in comparing priority values, the components are as follows:

— Identifier of the originating root bridge
— Root path cost for the transmitting bridge
— Identifier of the transmitting bridge
— Identifier of the port through which the message was transmitted

(For all numerical components, a lower numerical value indicates a higher priority.) A four-component priority value of this kind is termed a *spanning-tree priority*.

The notation {*Root* : *Cost* : *Bridge* : *Port*} is used, with appropriate substitutions for the four components, when it is desired to describe spanning-tree priorities concisely.

When conveyed in a configuration message, a spanning-tree priority value is termed a *message priority*.

Three other related kinds of spanning-tree priority value are also defined, with their components derived differently, as follows.

Each port in the RB-LAN has a spanning-tree priority associated with it, termed the *designated priority* of the port. For a port with the spanning tree role of designated port, the designated priority is the value used as the message priority in configuration messages transmitted through the port. For a port with the spanning tree role of root port or alternate port, the designated priority is the highest-priority message priority contained in configuration messages that have been received on the port.

The *root path priority* for a port is derived from the port's designated priority by adding the port's path cost value to the second component (cost) of the designated priority. This priority value is computed for temporary use by a bridge when updating its configuration.

The *update priority* for a port *P* on bridge *B*, which is also computed for temporary use by a bridge when updating its configuration, is defined as:

{ root identifier : root path cost of *B* : bridge identifier of *B* : port identifier of *P* }.

7.3.4.3 Cluster identifiers

Within a remote bridge group, a configuration message also carries a *cluster identifier* assigned by the remote bridge believed to be the primary bridge of a cluster, in order to allow the remote bridges in the group to determine their membership of the appropriate clusters. This information is also associated with each virtual port that has the spanning tree role of root port or alternate port, along with the corresponding designated priority.

A cluster identifier consists of the following two components:

a) The unique bridge address of the primary bridge (i.e., the constant part of the bridge identifier).

b) An integer value known as the *cluster index.*

The cluster index is assigned by the primary bridge to ensure that distinct clusters for which it is, or has been, the primary bridge can be distinctly identified (see 7.6.2.4.1).

NOTE—The range of values needed for the cluster index, in order to ensure distinctness, is typically quite small. The two optional protocol encodings specified in 10.6 and 12.5.2.2 both allow a range of 0–65535, which is more than adequate.

In a virtual LAN, a variable cluster index is not needed and the value zero is always used; further, the cluster identifiers can be implicit in the message priority and designated priority values, since the Bridge component of each such value always identifies the primary bridge.

A unique null value for cluster identifiers is defined, consisting of a null (all zero) bridge address and a cluster index value of zero.

7.3.4.4 Basic propagation mechanisms, and determination of port roles

The following four basic mechanisms are used to achieve timely propagation throughout the RB-LAN of the necessary information to allow all LAN ports and virtual ports to determine their state (blocking or forwarding):

a) A bridge that believes itself to be the root (e.g., following initialization) originates configuration messages on all the LANs to which it is attached, by sending Configuration BPDUs at regular intervals. A remote bridge that believes itself to be the root also conveys equivalent, up-to-date configuration message information to all other remote bridges belonging to the same group (or groups).

b) A bridge that receives a configuration message, on any port, conveying better or equally good information (i.e., received message priority higher than or equal to the port's previous designated priority) accepts that received information as the basis for updating its own configuration.

   1) The receiving port's new designated priority is set to the received message priority.

   2) The root port is selected as the port that has the highest root path priority among the ports that are not designated ports (if two ports have the same root path priority, the port with higher-priority port identifier is selected, as in 7.3.2.3).

   3) Other ports are then selected as designated ports or alternate ports, according to whether the update priority value is higher or lower than the current designated priority; for a port selected as a designated port, the update priority value is used as the new designated priority.

c) A bridge that receives a configuration message on what it decides is its root port conveying better or equally good information (i.e., highest root path priority among the ports that are not designated ports) passes that information on through its designated ports (to all the LANs for which it believes itself to be the designated bridge, and to all the clusters in which it believes itself to be the designated bridge for one or more subgroups).

d) 1) A bridge that receives inferior information (i.e., received message priority lower than the port's designated priority) on a LAN port that it considers to be the designated port on a LAN to which it is attached transmits its own information in reply, to reaffirm its role as designated bridge to all other bridges attached to that LAN.

2) The remote bridges in a given group exchange information to maintain the consistency of the group's configuration in terms of clusters, both in steady-state operation and during reconfigurations of the cluster structure; when the optional Extended Spanning Tree Protocol is used, this is done by means of Configuration BPDUs.

NOTE—The comparisons of spanning-tree priorities in b) and c) implement the selection of least-cost paths to the root, as in 7.3.2.1, and the first two stages of the tie-breaking in 7.3.2.3.

7.3.4.5 Determination of cluster membership

A remote bridge considers itself to be the primary bridge for a cluster if it has one or more designated ports attaching to a particular group, but no root port attaching to that group. Such a bridge assigns its own cluster identifier, and includes it in each configuration message transmitted on the designated ports as in 7.3.4.3.

NOTE 1—A remote bridge considers itself to be the primary bridge either because it is itself the root, or because its root port attaches to a LAN or to a different group.

When a remote bridge has its root port attaching to a group, it uses the cluster identifier received in the current configuration message for the root port as the identifier of the cluster to which it belongs. The bridge then includes this identifier in each configuration message transmitted on any designated port that attaches to the same group, as in 7.3.4.3.

If a remote bridge has only alternate ports attaching it to a particular group, it does not belong to any cluster in that group, and the null cluster identifier is assigned.

Whenever a new (higher or equal spanning-tree priority) configuration message is accepted as in 7.3.4.4 b) on a virtual port that is an alternate port, the cluster identifier received is associated with the port, in case a reconfiguration causes the port to become the new root port. This cluster identifier is also used, in some circumstances, in determining whether the alternate port is to be in blocking or forwarding state (see the Note in 7.3.2.5).

NOTES

2—The above description applies to steady-state operation, when the active topology and cluster membership are not changing. Additional parameters and mechanisms are needed to handle changes in the cluster configuration (see 7.3.8).

3—See C.7.3.2 through C.7.3.6 for illustrative examples.

7.3.5 Reconfiguration

To allow for reconfiguration of the RB-LAN when components are added or removed, or when management changes are made to parameters determining the topology, the topology information propagated through the RB-LAN has a limited lifetime. This is effected by including the age of the information conveyed (i.e., the time elapsed since the root originated the configuration message) in each configuration message transmitted. Every bridge stores the most recent message priority values received on its root port and each of its alternate ports, and monitors the age of that information against a maximum age that is included by the root, and propagated, in configuration messages. In normal stable operation, the regular generation of configuration messages by the root ensures that this topology information stored at the ports (i.e., the designated priority values) does not time out.

If the bridge times out the information held for an alternate port, it makes that port a designated port, by setting the port's designated priority to the update priority for the port. That new priority information is then propagated in configuration messages through the port onto the attached LAN or subgroup.

If the root port information is timed out, the bridge makes that port a designated port in the same way as for an alternate port. If all the bridge's ports are designated ports, or if the bridge's identifier is of higher priority than the root bridge component of the highest-priority root path priority, the bridge reconfigures by attempting to become the root itself; otherwise, the bridge updates its configuration as in 7.3.4.4 b) 2) and b) 3) to select another port as the new root port, etc.

Management changes to parameter values associated with a bridge or its ports can also require that bridge to reconfigure, as follows:

a) If a port is disabled, or has its path cost changed, a full update of the bridge's configuration is performed, as in the case of the root port timing out.

b) If the priority part of a bridge's identifier is changed, the corresponding (third) component of the designated priority for each designated port is set to the new value of the bridge's identifier, and a full update of the configuration is performed.

c) If the priority part of a port's identifier is changed, then
   1) If the port is a designated port, the fourth component of the designated priority is set to the new value of the port's identifier.
   2) If the port is a LAN port that is an alternate port, and if the update priority corresponding to the new value of the port's identifier is higher than the port's designated priority, the port is made a designated port.
   3) Otherwise, if neither 1) nor 2) applies, no updating of the configuration is required.

7.3.6 Notifying topology change

In normal stable operation, station location information in the filtering database only needs to change as a result of stations being physically relocated: it may therefore be desirable to employ a long ageing time for dynamic entries in the filtering database.

However, when the active topology of an RB-LAN reconfigures, end stations can appear to move, from the point of view of a bridge. This is true even if the states of the ports on that bridge have not changed. It is necessary for station location information to be re-learned following a change in active topology, even if only part of the RB-LAN has reconfigured.

A bridge that detects a change in active topology notifies the root of the change, using reliable protocol mechanisms, and the root then communicates the change to all bridges in its subsequent configuration messages for a suitable period of time (the Topology Change Time). The bridges use a short time for ageing out dynamic entries in the filtering database, until received configuration messages no longer indicate the topology change.

The topology changes that are to be detected by a bridge are

a) The bridge becoming, or attempting to become, the root.

b) A bridge port being put into Forwarding state (see 7.3.7, 7.3.8, 7.4.4), provided that the bridge has at least one designated port.

c) A bridge port being put into Blocking state from Forwarding or Learning state (see 7.3.7, 7.3.8, 7.4).

When a bridge with its root port attached to a LAN changes the active topology as above, it transmits a Topology Change Notification BPDU on the LAN to which the root port is attached, as specified in ISO/ IEC 10038: 1993. This is repeated at intervals until acknowledged (by the LAN's designated bridge, in a Configuration BPDU).

Similarly, when a remote bridge with a virtual port as its root port changes the active topology, it transmits equivalent information over the cluster through the root port, and awaits acknowledging information from the designated bridge. Whether or not such a transmission needs to be repeated at intervals depends upon the nature of the communication service provided across the cluster; however, the transmission always needs to be repeated if the root port changes, or if the designated bridge changes.

In each case, if the designated bridge for the LAN or subgroup is not itself the root, it passes the received notification toward the root using whichever of the above mechanisms is appropriate for its root port, and transmits a configuration message containing an acknowledgment via the relevant designated port.

7.3.7 Changing port state

Since there are propagation delays in passing configuration message information through an RB-LAN, there cannot be a sharp transition from one active topology to another. Topology changes can take place at different times in different parts of the RB-LAN, and moving a port directly from non-participation in the active topology to the forwarding state would risk having temporary data loops, with consequent duplication and misordering of frames. It is also desirable, before starting to forward frames, to allow other bridges the opportunity to reply to inferior configuration message information.

Ports that have been newly selected for addition to the active topology therefore wait for new topology information to propagate throughout the RB-LAN, and for the frame lifetime to expire for any frames forwarded using the old active topology, before starting to forward frames.

During this time it is also desirable for the bridge to time out station location information in the filtering database, since this may no longer be true, and to learn new station location information in order to reduce the effect of initial flooding of frames when a port enters the Forwarding state.

When a port is selected to be added to the active topology, therefore, it is first put into the Listening state, where it waits for receipt of configuration message information that would return it to the Blocking state. After a suitable delay (the *forward delay*), the port is put into the Learning state. In the Learning state, the port still does not forward frames, but station location information acquired by the learning process from frames received on the port is included in the filtering database. After another forward delay period, the port moves into full operation in the Forwarding state, thus becoming part of the active topology.

The length of the forward delay period is set at the root bridge, and included in configuration messages for propagation throughout the RB-LAN.

7.3.8 Changes in cluster configuration

7.3.8.1 Kinds of change in cluster configuration

There are propagation delays in passing configuration message information among the remote bridges belonging to a given group. When the cluster configuration of the group changes, the possible effects of these delays on the transfer of cluster identifiers need to be taken into account in determining the new cluster membership, because of the consequences for port state selection.

Two kinds of change in cluster configuration need to be considered. (In practice, both kinds could occur together, affecting different remote bridges.)

a) Changes in the spanning tree configuration can cause the primary bridge for a cluster to change, without the overall membership of the cluster changing. The (unchanged) cluster is then identified by a new cluster identifier, and all the remote bridges in the cluster need to change over to using that new identifier in place of the previous one.

b) Changes in the spanning tree configuration can cause one or more remote bridges belonging to the group to change cluster membership. The possible changes are as follows:

    1) A remote bridge can transfer from one cluster in the group to another.
    2) An isolated remote bridge can become a member of a cluster.
    3) A member of a cluster can become an isolated remote bridge.

When a remote bridge is affected by a change as in a), its virtual ports that stay attached to the same cluster can all remain in Forwarding state (or in Listening or Learning, if applicable), since there is no change in the local part of the active topology formed by those ports and their active peer ports.

Conversely, when as in b) a remote bridge transfers to one cluster from another, or from none, there is a significant change in the active topology; consequently, the virtual ports by which the remote bridge attaches to the new cluster need to go through the full delay periods of the Listening and Learning states.

NOTE—Except in a virtual LAN, a remote bridge cannot immediately assume the validity of new cluster information accepted on one of its virtual ports, or generated when the remote bridge first considers itself to be the primary bridge for a cluster. More up-to-date information could be acquired soon afterwards, as a result of the global effects on the group of whatever change is occurring. If a previously forwarding virtual port had meanwhile been put into the Blocking state, and were then selected to remain after all as a forwarding port in the same cluster as previously, the transitions through the Listening and Learning states would cause an unnecessary break in the service availability for frames relayed through that virtual port.

7.3.8.2 Cluster configuration changes for virtual LANs

In a virtual LAN, there can be only one cluster, consisting of all remote bridges attaching to the virtual LAN by a designated port or root port; cluster membership is exactly determined by the spanning tree roles of the virtual LAN ports. The normal selection of ports to be forwarding or blocking (see 7.3.2.5) is sufficient to handle both kinds of change in cluster configuration.

7.3.8.3 Cluster configuration changes for other groups

In groups that are not virtual LANs, additional mechanisms are used to stabilize the process of changing cluster relationships. These allow a remote bridge to determine, following a change, whether it is still a member of the same cluster as before, or whether it has newly become a member of a different cluster.

For any group to which it attaches, a remote bridge always has a single established cluster identifier value which it considers to be the identifier of the cluster, if any, to which it belongs. This is the bridge's *Current Cluster Identifier* parameter for the group. When the cluster configuration has reached steady state, the Current Cluster Identifier value is the only cluster identifier value needed, and this is the value conveyed in any configuration messages that the bridge transmits within the group during steady state (7.3.4.3).

To handle the aspects of cluster reconfiguration outlined in 7.3.8.1, a bridge uses two additional cluster identifier parameters for each group: the *New Cluster Identifier* and *Old Cluster Identifier* parameters. The values of these parameters can differ from the Current Cluster Identifier value during reconfigurations, but are both equal to the Current Cluster Identifier value when the cluster configuration (as perceived by the bridge) is stable. Also, a second cluster identifier parameter is included in configuration messages transmitted during reconfigurations, and is recorded with the other port information at any virtual port that receives and accepts such a configuration message.

At a given bridge, a cluster reconfiguration occupies a *reclustering* period before the Current Cluster Identifier is updated, and in some cases a further *overlap* period after updating of the Current Cluster Identifier. Outside these periods, the bridge considers the cluster configuration to be stable.

7.3.8.3.1 Cluster reconfiguration: New Cluster Identifier and reclustering period

A bridge's New Cluster Identifier parameter for a group takes the value most recently selected by the bridge as a candidate for updating the Current Cluster Identifier. If all the virtual ports attaching to the group are selected as alternate ports, the New Cluster Identifier takes the null value. Otherwise, the New Cluster Identifier value is either a cluster identifier accepted in a configuration message received on the root port, when the root port attaches to the group, or a cluster identifier created by the bridge when it considers itself to have become a primary bridge in the group (7.3.4.5).

A cluster reconfiguration starts, for the bridge, when the New Cluster Identifier value is first observed to differ from the Current Cluster Identifier value. The bridge then enters the reclustering period, of duration equal to the *reclustering delay*, during which it can receive or generate further cluster identifier information on any of its virtual ports attaching to the group, updating its New Cluster Identifier as appropriate. Spanning tree information in configuration messages is processed in the usual way, but the states of virtual ports attaching to the group are not changed.

The New Cluster Identifier value is conveyed in any configuration messages that the bridge transmits within the group (7.3.4.3) during the reclustering period. A bridge that accepts such a configuration message on its root port uses the received New Cluster Identifier value to set its own New Cluster Identifier parameter for the group, as described above. Also, a bridge that accepts such a configuration message on an alternate port records the value with the port, as in 7.3.4.5.

NOTE 1—New Cluster Identifier values are therefore propagated promptly within the group, to ensure that the new cluster configuration is established as rapidly as possible.

At the end of the reclustering period, the value of the New Cluster Identifier at that time is accepted as valid: the Current Cluster Identifier is updated to this value, and the virtual ports attaching to the group are selected accordingly to be blocking or forwarding (7.3.2.5). If the bridge belongs to a different cluster from that, if any, to which it belonged before the reclustering period, all the virtual ports selected to be forwarding are put into the Listening state (7.3.7); if any such port is already in the Listening state, its forward delay restarts at this time.

The value of the reclustering delay is set at the primary bridge, and included in configuration messages for propagation throughout the group.

NOTES

2—The reclustering delay and New Cluster Identifier are used to smooth changes in the Current Cluster Identifier. Provided that the reclustering delay is long enough, a single change in the configuration of the group as a whole will result in at most a single change in the bridge's Current Cluster Identifier, although the New Cluster Identifier could take several values during the course of the reconfiguration.

3—Items d) and e) of 7.3.8.3.2 describe how a bridge decides whether it belongs to the same cluster as before, at the end of a reclustering period. Values for reclustering delay are specified in 7.10.5.

7.3.8.3.2 Continuity of forwarding: Old Cluster Identifier and overlap period

When a remote bridge discovers at the end of a reclustering period that it belongs to the same cluster as before, but with a different cluster identifier because of a change in the cluster's primary bridge, it enters an overlap period. The overlap period is of duration equal to the reclustering delay, but ends early if a new cluster reconfiguration starts as in 7.3.8.3.1.

During an overlap period, the previous established cluster identifier (i.e., the value of the Current Cluster Identifier parameter on entry to the reclustering period) is retained, as the value of the Old Cluster Identifier parameter for the same group. At the end of the full overlap period, on return to stable state, the Old Cluster Identifier parameter is updated to be equal to the value of the Current Cluster Identifier.

Updating of the Old Cluster Identifier is not performed on early exit from overlap to a new reclustering period, but is deferred until exit from the reclustering period. If that exit is again to overlap, the Old Cluster Identifier is updated (as already specified) to retain the previous value of the Current Cluster Identifier.

If exit from any reclustering period is directly to stable state, the Old Cluster Identifier is set equal to the newly established value of the Current Cluster Identifier.

The Old Cluster Identifier value is used in the following ways:

a) It is conveyed as an additional parameter of configuration messages transmitted within the group when the transmitting bridge considers itself to be a primary bridge.

b) A bridge that accepts a configuration message on its root port or an alternate port records the received Old Cluster Identifier value with the port.

c) An Old Cluster Identifier value received and recorded on the root port as in b) is conveyed as in a) in configuration messages transmitted on any designated ports attaching to the same group as the root port.

d) At the end of a reclustering period, a bridge that is neither an isolated bridge nor a primary bridge compares the value of the group's Old Cluster Identifier with the Old Cluster Identifier value received and recorded on the root port as in c). The bridge determines that it still belongs to the same cluster as it did before the reclustering period if the values are equal and non-null; otherwise, it belongs to a different cluster.

e) At the end of a reclustering period, a primary bridge determines that it belongs to the same cluster as it did before the reclustering period if, and only if, the value of the group's Old Cluster Identifier is not null.

f) For an individual virtual port selected as an alternate port, the received New Cluster Identifier and Old Cluster Identifier values are both used in deciding whether or not the peer virtual port attaches to the same cluster (see 7.3.2.5).

g) Mapping of the cluster_id parameter for transmitted frames, and checking of the cluster_id parameter in received frames as in 6.5 a), uses both the Old Cluster Identifier and Current Cluster Identifier values, as specified in 8.3.2 and 8.3.3.

Items b), f), and g) apply in stable state and during a reclustering period, as well as in overlap. Items a) and c) apply during reclustering and overlap periods.

NOTES

1—Items a), b), and c) propagate a primary bridge's Old Cluster Identifier value throughout its cluster. This allows each other bridge in the cluster to determine, as in d), whether or not it has remained in the same cluster through a reconfiguration. Note that the propagation of new and old cluster identifiers occurs during both reclustering and overlap periods, in order to avoid race conditions affecting tests such as d) and g).

2—The mechanisms that use Old Cluster Identifier values ensure that relaying continues uninterrupted throughout changes in cluster configuration of the first kind identified in 7.3.8.1 (i.e., between two bridges that remain in the same cluster, but with a change of cluster identifier because of a change of primary bridge—note that the new primary bridge must have been a member of the same cluster, before the reconfiguration). The test in d) allows ports that are forwarding to remain forwarding, rather than go needlessly through the Listening and Learning states. The cluster_id mapping and matching referred to in g) ensures that relayed frames continue to be forwarded and accepted within the cluster, but are discarded if received at any port that does not attach to the cluster.

3—In a virtual LAN, the reclustering mechanisms and parameters can be considered to apply in degenerate form. The reclustering delay is always zero; correspondingly, the New Cluster Identifier and Old Cluster Identifier values are never different from the Current Cluster Identifier value, all being formed from the bridge identifier of the primary bridge and cluster index zero.

7.4 Port states

The operation of an individual LAN port or virtual port of a remote bridge is described in terms of the state of the port and the processes that provide and support the functions necessary for the operation of the bridge.

The state of each port governs the processing of received frames (6.5), the submission of frames for transmission (6.6), and the possible inclusion of the port in the active topology of the RB-LAN.

The following are specified for each of the five possible states for a port (Blocking, Listening, Learning, Forwarding, and Disabled):

- The purpose of the state.
- Whether the forwarding process (6.7) discards received frames.
- Whether the forwarding process submits forwarded frames for transmission.
- How the learning process (6.8) processes received frames.
- Whether the bridge protocol entity (6.10) transmits protocol information through the port for configuration messages and topology change notifications.
- Under what conditions the port enters and leaves the state.

In all states except Disabled, the bridge protocol entity includes the port in its computation of the active topology, and processes received configuration messages and topology change notifications.

7.4.1 Blocking

Purpose — A port in the Blocking state does not participate in frame relay, thus preventing frame duplication arising through multiple paths that could otherwise exist in the active topology of the RB-LAN.

Forwarding process — Discards received frames.
— Does not submit frames for transmission.

Learning process — Does not add station location information to the filtering database.

Bridge protocol entity — Does not transmit protocol information if it is a LAN port, or if it is a virtual port and the optional Extended Spanning Tree Protocol is used.

Entry — Following initialization of the remote bridge.
— On management action from Disabled state.
— From Listening, Learning, or Forwarding states through operation of the Spanning Tree Algorithm, or at the end of a reclustering period.

Exit

— To Listening state on expiry of the configuration message age.

— To Listening state on the bridge receiving new configuration message information that selects the port as the root port or a designated port.

— For a subgroup port, to Listening state at the end of a reclustering period when the port is selected as attaching the bridge to a cluster.

— To Listening state when management action to change the Bridge Priority or a port's Priority or path cost results in selection of the port as the root port or a designated port.

— To Disabled state by management action.

7.4.2 Listening

Purpose

— A port in the Listening state is preparing to participate in frame relay. Frame relay is temporarily disabled to prevent temporary loops that could occur as the active topology changes. Learning is disabled since information acquired could be valid in the old active topology but not in the new.

Forwarding process

— Discards received frames.

— Does not submit frames for transmission.

Learning process

— Does not add station location information to the filtering database.

Bridge protocol entity

— Transmits protocol information as necessary.

Entry

— From Blocking state, as in 7.4.1.

— For a subgroup port, from Listening, Learning, or Forwarding state at the end of a reclustering period when the port is selected as attaching the bridge to a different cluster from that to which it previously belonged.

Exit

— To Learning state after a period of time equal to forward delay from entry to the state.

— For a LAN port or virtual LAN port, to Blocking state when the port is selected as an alternate port through operation of the Spanning Tree Algorithm.

— For a subgroup port, to Blocking state when the port is selected as an alternate port that does not attach the bridge to a cluster, through operation of the Spanning Tree Algorithm or at the end of a reclustering period.

— For a subgroup port, to (the start of) Listening state at the end of a reclustering period as at "Entry" above.

— To Blocking or Disabled state by management action.

7.4.3 Learning

Purpose

— A port in the Learning state is preparing to participate in frame relay. Frame relay is temporarily disabled to prevent temporary loops that could occur as the active topology changes. Learning is enabled to allow information to be acquired prior to frame relay in order to reduce the number of frames unnecessarily relayed.

Forwarding process

— Discards received frames.

— Does not submit frames for transmission.

Learning process — Adds station location information to the filtering database.

Bridge protocol entity — Transmits protocol information as necessary.

Entry — From Listening state after a period of time equal to forward delay in that state.

Exit
- To Forwarding state after a period of time equal to forward delay from entry to the state.
- For a LAN port or virtual LAN port, to Blocking state when the port is selected as an alternate port through operation of the Spanning Tree Algorithm.
- For a subgroup port, to Blocking state when the port is selected as an alternate port that does not attach the bridge to a cluster, through operation of the Spanning Tree Algorithm or at the end of a reclustering period.
- For a subgroup port, to Listening state at the end of a reclustering period as in 7.4.2.
- To Blocking or Disabled state by management action.

7.4.4 Forwarding

Purpose — A port in the Forwarding state is participating in frame relay.

Forwarding process
- Can forward received frames.
- Can submit frames for transmission.

Learning process — Adds station location information to the filtering database.

Bridge protocol entity — Transmits protocol information as necessary.

Entry — From Learning state after a period of time equal to forward delay in that state.

Exit
- For a LAN port or virtual LAN port, to Blocking state when the port is selected as an alternate port, through operation of the Spanning Tree Algorithm.
- For a subgroup port, to Blocking state when the port is selected as an alternate port that does not attach the bridge to a cluster, through operation of the Spanning Tree Algorithm or at the end of a reclustering period.
- For a subgroup port, to Listening state at the end of a reclustering period as in 7.4.2.
- To Blocking or Disabled state by management action.

7.4.5 Disabled

Purpose — A port in the Disabled state does not participate in frame relay or in the operation of the Spanning Tree Algorithm and Protocol.

Forwarding process
- Discards received frames.
- Does not submit frames for transmission.

Learning process — Does not add station location information to the filtering database.

Bridge protocol entity — Does not transmit protocol information.

Entry and exit — As a result of management operations only.

7.5 Protocol parameters

NOTE—The parameters specified here are abstract information elements; questions of formats or encodings are a matter for particular protocols that convey this information. This International Standard specifies three such protocol encodings; see 7.9 (for spanning tree BPDUs on LANs), 10.6 (for optional remote management), and 12.5.2 (for the optional Extended Spanning Tree Protocol).

7.5.1 Parameters of configuration messages

Except as noted for the three parameters specific to configuration messages conveyed across remote bridge groups, the following parameters correspond exactly to those specified, with the same names, as parameters of Configuration BPDUs in ISO/IEC 10038: 1993.

7.5.1.1 Root Identifier

The Root Identifier parameter takes as its value the unique bridge identifier (7.5.3.7) of the bridge assumed to be the root by the bridge transmitting the configuration message. This parameter is the most significant component of the message priority (7.3.4.2) conveyed in the configuration message. It is conveyed to enable all bridges to agree on the root.

7.5.1.2 Root Path Cost

The Root Path Cost parameter takes as its value the cost of the path from the transmitting bridge to the root bridge denoted by the Root Identifier. This parameter is the second most significant component of the message priority conveyed in the configuration message. It is conveyed to enable a receiving bridge to determine which of the bridges attached to the LAN or group on which the configuration message was received offers the lowest cost path to the root for that LAN or group.

7.5.1.3 Bridge Identifier

The Bridge Identifier parameter takes as its value the unique bridge identifier (7.5.3.7) of the bridge transmitting the configuration message. This parameter is the third most significant component of the message priority conveyed in the configuration message. It is used when a receiving bridge needs to perform the first-level tie-breaking between paths of equal lowest cost to the root (see 7.3.2.3).

7.5.1.4 Port Identifier

The Port Identifier parameter takes as its value the port identifier, unique within the transmitting bridge (7.5.4.1.1), of the port through which the configuration message was transmitted. This parameter is the least significant component of the message priority conveyed in the configuration message. It is used when a receiving bridge needs to perform the second-level tie-breaking between paths of equal lowest cost to the root (see 7.3.2.3; in a remote bridge, this tie-break is needed only when the transmitting and receiving remote bridges are adjacent in two or more groups, or groups and LANs).

7.5.1.5 Topology Change Acknowledgment

The Topology Change Acknowledgment parameter is a flag set in a configuration message transmitted in response to a topology change notification received on a designated port. This parameter is conveyed to allow a reliable acknowledged protocol to operate for notifying the root of changes in active topology.

7.5.1.6 Topology Change

The Topology Change parameter is a flag set by the root in all configuration messages transmitted for a period of time following the notification or detection of a topology change. The purpose of this parameter, and its use to select a short ageing time for dynamic entries in the filtering database, are as specified for the corresponding parameter of Configuration BPDUs in ISO/IEC 10038: 1993.

7.5.1.7 Message Age

The Message Age parameter takes as its value the age of the configuration message, being the time elapsed between generation by the root of the configuration message containing the information propagated in this configuration message, and the transmission of this configuration message. This parameter is conveyed to enable a bridge to discard configuration information whose age exceeds the maximum age permitted for the RB-LAN (which is propagated in the Max Age parameter; see 7.5.1.8).

7.5.1.8 Timer parameters

A configuration message contains Max Age, Hello Time, and Forward Delay parameters: these are as specified for Configuration BPDUs in ISO/IEC 10038: 1993.

7.5.1.9 New Cluster Identifier

In a configuration message transmitted via a subgroup port, the New Cluster Identifier parameter takes the value of the New Cluster Identifier parameter for the group to which the port attaches. This parameter is conveyed, within groups only, as part of the information that enables a receiving remote bridge to determine its appropriate cluster membership. The structure of the cluster identifier is as specified in 7.3.4.3.

7.5.1.10 Old Cluster Identifier

In a configuration message transmitted via a subgroup port, the Old Cluster Identifier parameter takes the value of the Old Cluster Identifier parameter for the primary bridge of the cluster to which the port attaches. The primary bridge of the cluster originates this parameter; other bridges in the cluster record the value received on their root ports (7.5.2.2), and propagate it through any designated ports. This parameter is conveyed, within groups only, as part of the information that enables a receiving remote bridge to determine its appropriate cluster membership. The structure of the cluster identifier is as specified in 7.3.4.3.

7.5.1.11 Reclustering Delay

In a configuration message transmitted via a subgroup port, the Reclustering Delay parameter takes the value to be used for the reclustering delay (7.3.8.3.1, 7.3.8.3.2) by each bridge belonging to the same cluster as that indicated in the configuration message. The primary bridge of the cluster originates this parameter, as the value of its Primary Reclustering Delay parameter; other bridges in the cluster record the value received on their root ports, and propagate it through any designated ports. This parameter is conveyed, within groups only, in support of the ability to manage the timer values.

7.5.2 Parameters of topology change notifications

As specified in ISO/IEC 10038: 1993, no parameters are conveyed in Topology Change Notification BPDUs on LANs, or on virtual LANs. The following parameter is conveyed with topology change notifications in remote bridge groups that are not virtual LANs, to ensure that, when groups are partitioned, the notifications propagate toward the root only along spanning tree paths.

7.5.2.1 Cluster Identifier

The Cluster Identifier parameter takes the value of the Current Cluster Identifier parameter (7.5.5.2) for the group to which the transmitting remote bridge is attached by the subgroup port on which the topology change notification was transmitted.

7.5.3 Remote bridge parameters

The following parameters are maintained by the bridge protocol entity independently of the individual ports. The specification given here is equivalent to that for local bridges in ISO/IEC 10038: 1993, extended to encompass virtual ports and the transmission of protocol information across remote bridge groups as well as in BPDUs on LANs.

7.5.3.1 Designated Root

The Designated Root parameter takes as its value the unique bridge identifier (7.5.3.7) of the bridge assumed to be the root. This parameter is used as the value of the Root Identifier parameter in all configuration message information transmitted.

7.5.3.2 Root Path Cost

The Root Path Cost parameter takes as its value the cost of the path to the root from this bridge. When the bridge is the root, this parameter has value zero. Otherwise, it is equal to the sum of the values of the Designated Cost and Path Cost parameters held for the root port. This parameter is used to test the value of the Root Path Cost parameter conveyed in received configuration message information, and as the value of the Root Path Cost parameter in transmitted configuration message information.

7.5.3.3 Root Port

The Root Port parameter takes as its value the port identifier (7.5.4.1.1) of the port that offers the lowest cost path to the root (i.e., for which the sum of the values of the Designated Cost and Path Cost parameters held for the port is the lowest). The root port is selected as specified in 7.3.4.4 and 7.6.8. This parameter is used to identify the port through which the path to the root is established; it is not significant when the bridge is the root, and is then set to zero.

7.5.3.4 Max Age

The Max Age parameter takes as its value the maximum age of received configuration message information before it is discarded.

7.5.3.5 Hello Time

The Hello Time parameter takes as its value the time interval between successive transmissions of configuration message information by a bridge that is attempting to become the root or which is the root.

7.5.3.6 Forward Delay

The Forward Delay parameter takes as its value the time spent by a port in the Listening state and the Learning state before moving to the Learning or Forwarding state, respectively. It is also the value used for the ageing time of dynamic entries in the filtering database while received configuration messages indicate a topology change.

7.5.3.7 Bridge Identifier

The Bridge Identifier parameter takes as its value the unique bridge identifier of the bridge. This parameter is used as the value of the Bridge Identifier parameter in all configuration messages transmitted by the bridge, and as the value of the bridge's Designated Root parameter when the bridge is the root or when the bridge is attempting to become the root.

This parameter consists of two parts, one of which is derived from the unique bridge address and assures the uniqueness of the bridge identifier in the RB-LAN, and the other of which allows adjustment of the priority of the bridge identifier and is taken as the more significant part in priority comparison. When bridge management is supported, the priority part of this parameter can be updated by management action.

7.5.3.8 Bridge Max Age

The Bridge Max Age parameter takes as its value the value to be used for the Max Age parameter when the bridge is the root or is attempting to become the root. When bridge management is supported, this parameter can be updated by management action.

7.5.3.9 Bridge Hello Time

The Bridge Hello Time parameter takes as its value the value to be used for the Hello Time parameter when the bridge is the root or is attempting to become the root. When bridge management is supported, this parameter can be updated by management action.

7.5.3.10 Bridge Forward Delay

The Bridge Forward Delay parameter takes as its value the value to be used for the Forward Delay parameter when the bridge is the root or is attempting to become the root. When bridge management is supported, this parameter can be updated by management action.

7.5.3.11 Topology Change Detected

Topology Change Detected is a Boolean parameter, with its value set to True to record that a topology change has been detected by or notified to the bridge. When set to True, this parameter is used to stimulate transmission of topology change notifications toward the root when the bridge is not itself the root; and to set the value of the Topology Change parameter for the bridge to True if the bridge is, or becomes, the root. Transmission is subject to a reliable acknowledgment mechanism, as in 7.3.6 and 7.5.1.5; on a LAN, the Topology Change Notification BPDUs are transmitted at regular intervals of Bridge Hello Time, until acknowledged.

7.5.3.12 Topology Change

Topology Change is a Boolean parameter, with its value set to record

- For a bridge that is not the root, whether or not the most recently accepted configuration message indicates a change in the active topology.
- For the root, whether or not a change in topology has been detected within the preceding Topology Change Time period.

This parameter is used to propagate the indication of topology change in transmitted configuration messages, and to determine whether the short (forward delay) or long (ageing time) timeout value is to be used for dynamic entries in the filtering database.

7.5.3.13 Topology Change Time

The Topology Change Time parameter takes as its value the time period for which the bridge originates configuration messages indicating topology change following detection of a topology change, when the bridge is the root. The value of this parameter is equal to the sum of the values of the bridge's Bridge Max Age and Bridge Forward Delay parameters.

7.5.3.14 Hold Time

The Hold Time parameter takes as its value the minimum time period to elapse between the transmission of Configuration BPDUs through a given LAN port: at most one Configuration BPDU shall be transmitted in any Hold Time period. This parameter is a fixed parameter, with its value as specified in ISO/IEC 10038: 1993.

7.5.4 Port parameters

7.5.4.1 Port parameters present for all LAN ports and virtual ports

The following parameters are maintained by the bridge protocol entity for each LAN port and virtual port. The specification given here is equivalent to that for bridge ports of local bridges in ISO/IEC 10038: 1993, extended to apply to virtual ports and to transmission of protocol information across remote bridge groups as well as in BPDUs on LANs.

7.5.4.1.1 Port Identifier

The Port Identifier parameter takes as its value the identifier of the port, unique among the ports of this bridge. This parameter is used as the value of the Port Identifier parameter of all configuration messages transmitted through the port.

The parameter consists of two parts. One part bears a fixed relationship to the physical or logical support of the port by real world equipment; this part assures the uniqueness of the port identifier among the ports of a single bridge, and is a small integer assigned in the range from one upwards. The other part of the parameter allows adjustment of the priority of the port and is taken as the more significant part in priority comparisons. When bridge management is supported, the priority part of this parameter can be updated by management.

7.5.4.1.2 State

The State parameter records the current state of the port (Disabled, Blocking, Listening, Learning, or Forwarding); see 7.4.1 through 7.4.5. When bridge management is supported, this parameter can be updated by management.

7.5.4.1.3 Path Cost

The Path Cost parameter takes as its value the contribution of the path through the port, when the port is the root port, to the total cost of the path to the root for this bridge. When the port is not a designated port, this parameter is used in calculating the value of the port's root path priority, to determine the root port for the bridge. When bridge management is supported, this parameter can be updated by management.

7.5.4.1.4 Designated Root

The Designated Root parameter takes as its value the unique bridge identifier of the bridge recorded as the root in the Root Identifier parameter of configuration messages transmitted by the designated bridge on the LAN or subgroup to which the port is attached. This parameter is the most significant component of the port's designated priority, used in testing the message priority values in received configuration messages.

7.5.4.1.5 Designated Cost

- For a designated port, the Designated Cost parameter takes as its value the path cost (equal to the root path cost of the bridge) offered to the LAN or subgroup to which the port is attached.
- Otherwise, the Designated Cost parameter takes as its value the cost of the path to the root offered by the designated port on the LAN or subgroup to which this port is attached.

This parameter is the second most significant component of the port's designated priority, used in testing the message priority values in received configuration messages.

7.5.4.1.6 Designated Bridge

The Designated Bridge parameter takes as its value the unique bridge identifier (7.5.3.7) of

- For a designated port, the bridge to which the port belongs.
- Otherwise, the bridge believed to be the designated bridge for the LAN or subgroup to which this port is attached.

This parameter is the third most significant component of the port's designated priority, used in testing the message priority values in received configuration messages. It can also be used by management in discovering the topology of the RB-LAN.

7.5.4.1.7 Designated Port

The Designated Port parameter takes as its value the port identifier (7.5.4.1.1) of the bridge port, on the designated bridge (as in 7.5.4.1.6), through which the designated bridge transmits the configuration message information stored by this port. This parameter is the least significant component of the port's designated priority, used in testing the message priority values in received configuration messages. It can also be used by management in discovering the topology of the RB-LAN.

7.5.4.1.8 Topology Change Acknowledge

The Topology Change Acknowledge parameter takes the value to be used for the Topology Change Acknowledgment flag in the next configuration message to be transmitted via the port (see 7.6.10).

7.5.4.2 Port parameters present only for subgroup ports

The following parameters are maintained by the bridge protocol entity for each subgroup port, but not for any LAN port or virtual LAN port. They are used in determining the cluster membership appropriate for the remote bridge to which the subgroup port belongs, and in selecting the appropriate port state on changes of cluster membership.

7.5.4.2.1 Peer New Cluster Identifier

For a root port or alternate port, the Peer New Cluster Identifier parameter takes the value of the New Cluster Identifier parameter in the current configuration message received from the designated bridge (as in 7.5.4.1.6) for the subgroup to which this port attaches. For a designated port, the parameter takes the null cluster identifier value.

7.5.4.2.2 Peer Old Cluster Identifier

For a root port or alternate port, the Peer Old Cluster Identifier parameter takes the value of the Old Cluster Identifier parameter in the current configuration message received from the designated bridge (as in 7.5.4.1.6) for the subgroup to which this port attaches. For a designated port, the parameter takes the null cluster identifier value.

7.5.5 Remote bridge group parameters

The following parameters are maintained for each remote bridge group, other than virtual LANs, to which the remote bridge belongs. They are used in determining the cluster membership appropriate for the remote bridge within each group.

7.5.5.1 Reclustering State

Reclustering State is a three-valued state parameter, used to stabilize changes of cluster identifier, and to maintain relaying through cluster reconfigurations. Its values are

— *Stable*. This value is used when neither of the other values applies (always, for a virtual LAN).

— *Reclustering*. This value is set when the remote bridge is in a reclustering period for the group.

— *Overlap*. This value is set at the end of a reclustering period when the bridge remains a member of the same cluster as before but with a new Current Cluster Identifier value; unless another reclustering period occurs first, this value remains set for a further time equal to the reclustering delay.

7.5.5.2 Current Cluster Identifier

The Current Cluster Identifier parameter takes as its value the cluster identifier of the cluster, if any, to which the remote bridge considers itself to belong within the group; it takes the null cluster identifier value if the remote bridge considers itself to belong to no cluster in the group. This parameter is used in mapping to the cluster_id parameters of M-UNITDATA request primitives (see 8.3.2), and in matching the cluster_id parameters of M-UNITDATA indication primitives (see 8.3.3).

7.5.5.3 New Cluster Identifier

The New Cluster Identifier parameter takes the cluster identifier value most recently selected by the bridge for its membership of the group. The value can differ from the Current Cluster Identifier value only when the Reclustering State parameter is Reclustering. This parameter is used as the value of the New Cluster Identifier parameter in configuration messages transmitted via any designated ports by which the remote bridge attaches to the group.

7.5.5.4 Old Cluster Identifier

When the Reclustering State parameter is Stable, the Old Cluster Identifier parameter takes the same value as the Current Cluster Identifier parameter. When the Reclustering State parameter is Overlap, this parameter takes the value to which the Current Cluster Identifier parameter was set at the time immediately before the Reclustering State parameter was last changed from Stable or Overlap to Reclustering. This parameter is used as the value of the Old Cluster Identifier parameter in configuration messages transmitted when the remote bridge is the primary bridge for a cluster; also in mapping to the cluster_id parameters of M-UNITDATA request primitives (see 8.3.2), and in matching the cluster_id parameters of M-UNITDATA indication primitives (see 8.3.3).

7.5.5.5 Reclustering Delay

The Reclustering Delay parameter takes as its value the time for which the bridge waits, following detection of a possible reclustering, before selecting its new cluster membership. After selecting its new cluster membership, the bridge retains information about its previous cluster membership for the same period again, unless it detects a further possible reclustering.

7.5.5.6 Primary Reclustering Delay

The Primary Reclustering Delay parameter records the value to be used, subject to compliance with 7.10.5, for the Reclustering Delay parameter when the bridge is the primary bridge for a cluster in the group. When bridge management is supported, the value of this parameter can be updated by management.

7.6 Elements of parameter computation for a remote bridge

The elements of computation specified here do not affect, and are not affected by, any port with port state set to Disabled.

7.6.1 Accepting received configuration message information at a port

7.6.1.1 Conditions for acceptance

When a configuration message is received at a LAN port or virtual port, it is accepted if and only if

a) The value of the message's Message Age parameter is less than that of its Max Age parameter, and

b) The message priority { *Root Identifier* : *Root Path Cost* : *Bridge Identifier* : *Port Identifier* } received is higher than or equal to the current value of the designated priority associated with the receiving port, i.e., { *Designated Root* : *Designated Cost* : *Designated Bridge* : *Designated Port* }

7.6.1.2 Actions on acceptance

When a configuration message is accepted according to 7.6.1.1,

a) The designated priority information associated with the port is updated to be equal to the received message priority, and

b) If the receiving port is a subgroup port, the cluster information for the port (Peer New Cluster Identifier and Peer Old Cluster Identifier) is similarly updated to be equal to the received cluster information (New Cluster Identifier and Old Cluster Identifier, respectively).

7.6.2 Updating the bridge's configuration

When one of the conditions in 7.6.2.1 occurs, the actions specified in 7.6.2.2, 7.6.2.3, and 7.6.2.4 are performed, in that order, to update the parameters that determine the roles that the bridge and its ports play in the spanning tree, and the cluster membership of the bridge within each remote bridge group. This configuration update is followed by the actions for port state selection specified in 7.6.4.2.

NOTE—The order in which the specified actions are performed is important, since the specifications for later actions are often written so as to use parameter values that have been updated by earlier actions.

7.6.2.1 Conditions for configuration update

Whenever there are changes in the values of bridge or port parameters that can affect the spanning-tree roles of the ports, a full update of the bridge's configuration takes place. Configuration updating therefore occurs on

a) Acceptance of a configuration message (as in 7.6.1).

b) Timing out of the configuration information associated with a port.

c) Management action to disable a port.

d) Management action to set the bridge's priority.

e) Management action to set the path cost of a port.

7.6.2.2 Actions for root selection

7.6.2.2.1 Selecting the bridge as the root

If every port on the bridge either is a designated port (i.e., with designated bridge parameter equal to the bridge's Bridge Identifier parameter) or has the value of its Designated Root parameter of priority lower than or equal to the value of the bridge's Bridge Identifier parameter, the bridge selects itself as the root, as follows:

a) The values of the Root Port and Root Path Cost parameters are both set to zero.

b) The value of the Designated Root parameter is set to the value of the Bridge Identifier.

c) The values of the Max Age, Hello Time, and Forward Delay parameters are set to the values of the Bridge Max Age, Bridge Hello Time, and Bridge Forward Delay parameters, respectively.

7.6.2.2.2 Selecting another bridge as the root

Otherwise, of the bridge's ports that are not designated ports and that have designated bridge values of higher priority than the bridge's bridge identifier

a) The port with the highest root path priority (7.3.4.2) is selected as the root port, unless two or more ports have equal highest root path priority; in that case,

b) The selected root port is the port, among those tied with equal highest root path priority, with the highest-priority port identifier value.

The bridge's Root Port parameter is set to the value of the Port Identifier parameter for the selected port; the bridge's Root Path Cost parameter is set to the value of the second component (path cost) of the root path priority; and the bridge's Designated Root parameter is set to the value of the Designated Root parameter for the selected port.

7.6.2.3 Actions for designated port selection

Each port for which the update priority (7.3.4.2), computed using the newly selected root information, is higher than or equal to the port's designated priority is selected to be a designated port: the port's designated priority is set equal to the update priority.

7.6.2.4 Actions for cluster selection

For each non-virtual-LAN group to which the remote bridge attaches, the actions in 7.6.2.4.1 and 7.6.2.4.2 are performed.

7.6.2.4.1 Setting the New Cluster Identifier

The New Cluster Identifier for the group is set to

a) If the root port attaches to the group, the Peer New Cluster Identifier of the root port.

b) Otherwise, if any designated port attaches to the group, then

   1) The Current Cluster Identifier for the group, if that has the value of its bridge address component equal to the unique bridge address of the remote bridge.

   2) If not 1), the New Cluster Identifier for the group, if that has the value of its bridge address component equal to the unique bridge address of the remote bridge.

   3) If neither 1) nor 2), a new cluster identifier produced from the unique bridge address of the remote bridge and a cluster index value that shall be chosen so that the new cluster identifier is different from any cluster identifier (produced by the remote bridge) used within the group during the previous period of twice the forward delay time (see 7.3.4.3 for the structure of cluster identifiers).

c) Otherwise (when all ports attaching to the group are alternate ports), the null cluster identifier.

NOTE—In case b) 1), the remote bridge considers itself to be continuing as the primary bridge of the cluster to which it currently belongs. In case b) 3), and then subsequently in case b) 2), the remote bridge considers itself as a candidate to become a new primary bridge. In case c), the remote bridge is isolated in the group.

7.6.2.4.2 Setting the Reclustering State

If the value of the New Cluster Identifier set as in 7.6.2.4.1 is not equal to the value of the Current Cluster Identifier parameter for the group, then

a) If the Reclustering State parameter for the group is set to Stable or Overlap,

   1) The Reclustering State is set to Reclustering, and
   2) The reclustering period starts at this time.

b) Otherwise (when the value of the Reclustering State parameter is Reclustering), no further action is required.

7.6.3 Recording additional configuration message information for the bridge and group

Whenever the port on which a configuration message was accepted (7.6.1.1) is selected as the root port by the configuration update that follows (7.6.2.2.2),

a) The values of the bridge's Topology Change, Max Age, Hello Time, and Forward Delay parameters are updated by setting them equal to the parameters of the same names in the received configuration message.

b) The bridge's Topology Change Detected parameter is set to False if the configuration message has its Topology Change Acknowledgment parameter set to True.

c) If the root port is a subgroup port, the value of the Reclustering Delay parameter for the group to which the port attaches is updated by setting it equal to the Reclustering Delay parameter of the received configuration message; if the Reclustering State parameter for the group is set to Reclustering, the reclustering period in progress shall not be affected by any change in the parameter value, but shall continue on the basis of the value current at the start of the period.

7.6.4 Port state selection

7.6.4.1 Conditions for performing port state selection

The states of the ports can change, and port state selection is therefore carried out, immediately following any updating of designated priority and cluster information parameter values, on

a) Any of the conditions that cause updating of the bridge's configuration (7.6.2.1).

b) Management action to initialize the bridge (7.8.1).

c) Management action to enable a port from Disabled state, or to reset a port from any other state (7.8.2).

d) Management action to set a port's priority (7.8.5).

NOTE—The state of a subgroup port can also change as specified in 7.6.6 and 7.6.7.

7.6.4.2 Procedure for port state selection

For each LAN port and virtual LAN port on the bridge, and for each subgroup port attached to a remote bridge group for which the Reclustering State parameter is set to Stable or Overlap,

a) If the port is the root port or a designated port then

   1) If the port state is Blocking, the port state is set to Listening.
   2) If the port state is Forwarding, Listening, or Learning, the port state remains unchanged.

b) If the port is a LAN port or virtual LAN port that has been selected as an alternate port then

   1) If the port state is Listening, Learning, or Forwarding, the port state is set to Blocking.
   2) Otherwise, the port state remains unchanged.

c) If the port is a subgroup port that has been selected as an alternate port, and if

   1) Each other virtual port attaching to the same group has also been selected as an alternate port, or
   2) The port is an isolated alternate subgroup port as determined by 7.6.8,

   the port state is set to Blocking.

d) Otherwise (i.e., when a subgroup port has been selected as an alternate port, and the remote bridge is not isolated in the group, and the port could have peer virtual ports attaching to the same cluster),

   1) If the port state is Blocking, the port state is set to Listening.
   2) Otherwise, the port state (Forwarding, Listening, or Learning) remains unchanged.

NOTE—When the active topology is in a steady state, ports remain selected for the active topology by a) 2) and d) 2); ports that are not part of the active topology remain selected by b) 2) and c).

7.6.5 Cluster resolution

At the end of any reclustering period for a group, the Reclustering State and cluster information parameters for the group are set as specified in items a) through d) below.

NOTES

1—Cluster resolution uses the value of New Cluster Identifier that has been set by the most recent cluster selection action, as specified in 7.6.2.4.1.

2—Cluster resolution is always followed by the actions for cluster port state selection specified in 7.6.6.

a) If the value of the New Cluster Identifier parameter for the group is the null cluster identifier,

   1) The Reclustering State parameter is set to Stable.
   2) The Old Cluster Identifier and Current Cluster Identifier parameters are both set to the null cluster identifier.

NOTE—In this case, the bridge is to be isolated in the group.

b) If the value of the New Cluster Identifier parameter for the group is non-null and equal to the value of the Current Cluster Identifier parameter,

   1) The Reclustering State parameter is set to Stable.
   2) The Old Cluster Identifier is set to the value of the Current Cluster Identifier.

NOTE—In this case, the bridge remains in the same cluster as before, with the same cluster identifier as before.

c) If the value of the New Cluster Identifier parameter for the group is non-null and not equal to the value of the Current Cluster Identifier parameter, and if the bridge's root port attaches to the group and the value of the root port's Peer Old Cluster Identifier parameter is not equal to the value of the Current Cluster Identifier,

   1) The Reclustering State parameter is set to Stable.
   2) The Old Cluster Identifier and Current Cluster Identifier parameters are both set to the value of the New Cluster Identifier.

NOTE—In this case, the bridge is to be in a different cluster from before.

d) If the value of the New Cluster Identifier parameter for the group is non-null and not equal to the value of the Current Cluster Identifier parameter, if the value of the Current Cluster Identifier parameter is non-null, and if either the bridge is the primary bridge for the cluster (identified by the New Cluster Identifier) or the bridge's root port attaches to the group and the value of the root port's Peer Old Cluster Identifier parameter is equal to the value of the Current Cluster Identifier,

   1) If the bridge is the primary bridge for the cluster, the Reclustering Delay parameter for the group is set to the value of the Primary Reclustering Delay parameter for the group, adjusted if necessary to comply with condition (2) specified in 7.10.5.

2) The Reclustering State parameter is set to Overlap.
3) The Old Cluster Identifier parameter is set to the value of the Current Cluster Identifier and the Current Cluster Identifier parameter is set to the value of the New Cluster Identifier.
4) The overlap period, of length equal to the reclustering delay, starts at this time.

NOTE—In this case, the bridge remains in the same cluster as before, but with a different cluster identifier.

7.6.6 Cluster port state selection

At the end of any cluster resolution actions, after the Old Cluster Identifier and Current Cluster Identifier parameters have been updated as necessary, the port states of the subgroup ports attaching to the group are set as specified in 7.6.6.1, 7.6.6.2, or 7.6.6.3.

NOTES

1—Port state selection (7.6.4.2) does not change the states of ports attaching to a group that is in a reclustering period; cluster port state selection ensures that at the end of every such period the states of all the ports attaching to the group are updated in accordance with the spanning tree roles at that time.

2—Selection of 7.6.6.2 or 7.6.6.3 depends upon information derived during cluster resolution. An implementation that followed this text's separation of the functions in 7.6.5 and 7.6.6 would have to pass information about the cluster resolution outcomes of 7.6.5 a) through d) to the cluster port state selection. Alternatively, an implementation could invoke the relevant particular piece of processing for 7.6.6 directly from the corresponding processing for 7.6.5 a) through d).

7.6.6.1 Bridge in no cluster

When the bridge belongs to no cluster in the group [7.6.5 a)], each port has its port state set to Blocking.

7.6.6.2 Bridge in same cluster as before

When the bridge attaches to the same cluster as it did before the reclustering [7.6.5 b), 7.6.5 d)], the port state of each virtual port attaching to the group is set as follows:

a) If the virtual port has been selected as an alternate port and it is an isolated alternate subgroup port as determined by 7.6.8, then the port state is set to Blocking.

b) Otherwise, if the port state is Forwarding, Listening, or Learning, the port state remains unchanged.

c) Otherwise (i.e., when the port state is Blocking), the port state is set to Listening.

NOTES

1—Case b) applies to all ports that were previously attached to the cluster and have not been selected as isolated ports; case c) applies to ports that were previously isolated ports but have now been selected to attach to the cluster.

2—Case b) applies to ports that have been previously selected to be forwarding, but are still in the Listening or Learning state, as well as to ports that have already reached the Forwarding state.

7.6.6.3 Bridge in different cluster

When the bridge attaches to a different cluster from that, if any, to which it attached before the reclustering [7.6.5 c)], the port state of each virtual port attaching to the group is set as follows:

a) If the virtual port has been selected as an alternate port that is an isolated alternate subgroup port as determined by 7.6.8, the port state is set to Blocking.

b) Otherwise,

1) If the port state is Blocking, Learning, or Forwarding, the port state is set to Listening.
2) Otherwise, when the port state is Listening, the port state remains Listening but the forward delay period before entry to the Learning state is measured from this time instead of from the previous starting time.

7.6.7 Cluster confirmation

At the end of any overlap period for a group [see 7.6.5 d)]

a) The Reclustering State parameter for the group is set to Stable.

b) The Old Cluster Identifier parameter is set to the value of the Current Cluster identifier parameter.

c) Each virtual port that has been selected as an alternate port and is an isolated alternate subgroup port as determined by 7.6.8 has its port state set to Blocking.

7.6.8 Test for an isolated alternate subgroup port

A subgroup port that has been selected as an alternate port is isolated in the group

a) If it is an individual virtual port and

1) The value of the Current Cluster Identifier parameter for the group is not equal to the value of either the Peer New Cluster Identifier or the Peer Old Cluster Identifier parameter of the port, and
2) The value of the Old Cluster Identifier parameter for the group is not equal to the value of either the Peer New Cluster Identifier or the Peer Old Cluster Identifier parameter of the port; or

b) If it is determined (by implementation-dependent means outside the scope of this International Standard) that the port has no peer virtual port(s) attaching to the cluster that corresponds to the value of the Current Cluster Identifier parameter for the group.

7.6.9 Topology change detection

7.6.9.1 By a bridge that is not the root

A bridge that is not the root sets its Topology Change Detected parameter (7.5.3.11) whenever

a) A port changes state to Forwarding (7.4.4).

b) A port changes state from Learning or Forwarding to Blocking (7.6.4.2, 7.6.6.1 through 7.6.6.3, 7.6.7).

c) A Topology Change Notification BPDU is received on a LAN port that is a designated port.

d) A topology change notification is received on a virtual LAN port that is a designated port.

e) A topology change notification is received on a subgroup port that is a designated port, from a remote bridge in the same cluster as that to which the receiving port is attached (i.e., with the Cluster Identifier parameter of the topology change notification equal to the Current Cluster Identifier parameter for the group).

The Topology Change Detected parameter is reset as specified in 7.6.3.

7.6.9.2 By the root

A bridge that has been selected as the root sets its Topology Change parameter (7.5.3.12) when

a) It is newly selected as the root, except on initialization of the bridge.

b) A topology change notification is received on a designated port, provided that if the receiving port is a subgroup port, the topology change notification is from a bridge in the same cluster, as in 7.6.9.1 item e).

As long as the bridge remains the root, the Topology Change parameter remains set for a period equal to the Topology Change Time (7.5.3.13), starting from the most recent event a) or b) that sets the parameter; the Topology Change and Topology Change Detected parameters are then both reset.

7.6.10 Topology change acknowledgment

The Topology Change Acknowledgment flag parameter is set for a port whenever the port is a designated port and a topology change notification is received on it from a bridge attached to the same LAN or cluster.

The Topology Change Acknowledgment flag is reset when a configuration message is transmitted via the port.

NOTE—It is necessary to specify the operation of topology change acknowledgment at the logical, protocol-independent level, in order to handle the situations that can arise when the designated bridge and/or root bridge change, possibly more than once, between notification and acknowledgment of a topology change.

7.7 Spanning Tree Protocol operation at LAN ports

The operations specified here do not apply to any port with port state set to Disabled.

7.7.1 Protocol operation at a designated port

A LAN port that has been selected as a designated port shall operate the Spanning Tree Protocol as follows.

7.7.1.1 Transmission of Configuration BPDUs

a) Configuration BPDUs shall be transmitted via the port, normally at intervals of approximately Hello Time but subject to the additions and qualifications in b) and c).

b) A Configuration BPDU shall be transmitted within a period of Hold Time, following reception at the port of a Topology Change Notification BPDU or of a Configuration BPDU containing a message priority lower than the designated priority of the port. The transmitted BPDU may be one of the normal sequence as in a), or may be additional—as would be required, for example, if the received BPDU occurs soon after a BPDU has been transmitted as in a).

c) When the bridge is not the root, gaps in the sequence of normal transmissions a) may occur as a result of the root port failing to receive configuration message information at the normal regular intervals. Conversely, additional transmissions may occur as a result of the root port receiving additional configuration messages—such as would occur, for example, if the root port's designated bridge transmitted additional configuration messages as in b).

7.7.1.2 Stopping transmission

The port shall change its role from being designated port, in accordance with 7.6.2, if a Configuration BPDU is received at the port conveying a message priority higher than the designated priority of the port.

7.7.1.3 Parameters of Configuration BPDUs

The parameters of each transmitted Configuration BPDU shall convey the configuration message parameters, as follows:

a) The message priority parameters (7.5.1.1 through 7.5.1.4) shall take the values of the port's Designated Priority parameters (7.5.4.1.4 through 7.5.4.1.7).

b) The Message Age parameter (7.5.1.7) shall take the value zero if the bridge is the root, and otherwise a value that is not less than the sum of the Message Age value received most recently at the root port and the time elapsed from receipt of the configuration message conveying that value to transmission of the Configuration BPDU.

c) The value of the Message Age parameter when the bridge is not the root shall not overestimate the time from receipt of the original configuration message on the root port to transmission of the Configuration BPDU by more than a stated value for maximum Message Age increment overestimate (see ISO/IEC 10038: 1993, as referenced by 7.10).

d) The Max Age, Hello Time, Forward Delay, and Topology Change parameters (7.5.1.6, 7.5.1.8) shall take the values of the corresponding bridge parameters (7.5.3.4 through 7.5.3.6, 7.5.3.12).

e) The Topology Change Acknowledgment parameter shall be set if the Configuration BPDU is the first one transmitted since reception of a Topology Change Notification BPDU [7.7.1.1 b), 7.6.10].

7.7.2 Protocol operation at a root port

A LAN port that has been selected as the root port shall operate the Spanning Tree Protocol as follows.

7.7.2.1 Configuration BPDUs

No Configuration BPDU shall be transmitted via the port, provided that an acceptable Configuration BPDU (as in 7.6.1.1) is received at the port within a time interval of (Max Age – Message Age) from receipt of the previous such BPDU.

7.7.2.2 Ceasing to be root port

If an acceptable Configuration BPDU (as in 7.6.1.1) is not received at the port within the time interval (Max Age – Message Age) from receipt of the most recently accepted Configuration BPDU, the port ceases to be the root port. The port's Designated Priority is set equal to the update priority for the port (7.3.4.2), and the bridge's configuration is then updated as specified in 7.6.2.

7.7.2.3 Transmission of Topology Change Notification BPDUs

A Topology Change Notification BPDU shall be transmitted via the port on the bridge's Topology Change Detected flag being set, or on the port being newly selected as the root port when that flag is set. Further Topology Change Notification BPDUs shall be transmitted at intervals of Bridge Hello Time as long as the Topology Change Detected flag remains set (see 7.6.3).

7.7.3 Protocol operation at an alternate port

A LAN port that has been selected as an alternate port shall operate the Spanning Tree Protocol as specified in 7.7.2.1 and 7.7.2.2 for a root port. It shall not transmit Topology Change Notification BPDUs.

7.8 Management of the bridge protocol entity

The following management operations can affect the bridge and port parameters, and hence the roles of a bridge and its ports in the spanning tree; the effects are as specified in the subclauses referenced.

- Initialization of the bridge (7.8.1)
- Enabling a port from Disabled state, or resetting (re-enabling) a port from any other state (7.8.2)
- Disabling a port (7.8.3)
- Changing the priority part of the bridge's identifier (7.8.4)
- Changing the priority part of a port's identifier (7.8.5)
- Changing the path cost associated with a port (7.8.6)

7.8.1 Initialization

7.8.1.1 Setting the bridge parameters

The bridge parameters are initialized as follows:

a) The values of the Root Port and Root Path Cost parameters are both set to zero.

b) The value of the Designated Root parameter is set to the value of the bridge identifier.

c) The values of the Max Age, Hello Time, and Forward Delay parameters are set to the values of the Bridge Max Age, Bridge Hello Time, and Bridge Forward Delay parameters, respectively.

d) The Topology Change Detected flag and Topology Change flag parameters are both reset to False.

7.8.1.2 Setting the ports' parameters

For each of the bridge's ports, the port parameters are set as follows:

a) The port's Designated Priority is set equal to the update priority for the port (7.3.4.2).

b) The port state is set to Blocking if the port is to be enabled following initialization, and otherwise is set to Disabled.

c) The Topology Change Acknowledge flag is reset to False.

d) For a subgroup port, the Peer New Cluster Identifier and Peer Old Cluster Identifier are set to the null cluster identifier.

7.8.1.3 Setting the remote bridge groups' parameters

For each non-virtual-LAN remote bridge group of which the bridge is a member

a) The Reclustering State parameter is set to Stable.

b) If all of the virtual ports attaching to the group have port state set to Disabled, the Current Cluster Identifier, New Cluster Identifier, and Old Cluster Identifier parameters are all set to the null cluster identifier; otherwise, all three parameters are set to the same new cluster identifier value, determined as specified in 7.6.2.4.1 b) 3).

7.8.1.4 Enabling the ports

The procedure for port state selection (7.6.4.2) is performed.

7.8.2 Enable Port

The port's parameters are set as follows:

a) The port's Designated Priority is set equal to the update priority for the port (7.3.4.2).

b) The port state is set to Blocking.

c) The Topology Change Acknowledge flag is reset to False.

d) For a subgroup port, the Peer New Cluster Identifier and Peer Old Cluster Identifier are set to the null cluster identifier.

The procedure for port state selection (7.6.4.2) is then performed.

7.8.3 Disable Port

The port's parameters are set as follows:

a) The port's Designated Priority is set equal to the update priority for the port (7.3.4.2).

b) The port state is set to Disabled.

c) The Topology Change Acknowledge flag is reset to False.

The bridge's configuration is then updated as specified in 7.6.2.

7.8.4 Set Bridge Priority

The new value of the bridge identifier (7.5.3.7) is calculated. For each port that has been selected as a designated port, the Designated Bridge parameter is set to the new value of the bridge identifier; the bridge's Bridge Identifier parameter is then set to the new value.

The bridge's configuration is then updated as specified in 7.6.2. If the bridge is selected as the root by the configuration update, the bridge's Topology Change flag parameter is set.

7.8.5 Set Port Priority

The new value of the port identifier (7.5.4.1.1) is calculated. If the port has been selected as a designated port, the Designated Port parameter is set to the new value of the port identifier; the Port Identifier parameter is then set to the new value.

If the port is a LAN port for which the value of the Designated Bridge parameter is equal to the value of the bridge's bridge identifier, and the new value of the port identifier is of higher priority than the value recorded as the designated port, then

a) The port's Designated Priority is set equal to the update priority for the port (7.3.4.2).

b) The procedure for port state selection (7.6.4.2) is performed.

NOTE—This happens when the bridge has two or more ports attached to the same LAN, with the bridge as designated bridge for the LAN, and the update of the port priority causes a change in the port selected as the designated port.

7.8.6 Set Path Cost

The Path Cost parameter for the port is set to the new value, and the bridge's configuration is then updated as specified in 7.6.2.

7.9 Encoding and validation of BPDUs on LANs

Each Configuration BPDU and Topology Change Notification BPDU transmitted through a LAN port shall be encoded as a sequence of octets as specified in ISO/IEC 10038: 1993. Each BPDU received on a LAN port of a remote bridge shall be validated as specified in ISO/IEC 10038: 1993.

7.10 Performance

This subclause specifies requirements on the performance of bridges in an RB-LAN and on the setting of the parameters of the Spanning Tree Algorithm and Protocol. 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 RB-LAN components interoperate.

It specifies absolute maximum values and ranges of applicable 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.

7.10.1 Requirements

The requirements for correct operation are met by placing constraints on

a) The maximum bridge diameter of the RB-LAN, derived from the maximum number of bridges between any two points of attachment of end stations (for remote bridges, a weighting function applies—see 7.10.3).

b) For local bridging, the maximum bridge transit delay, defined as the maximum time elapsing between reception and transmission of a forwarded frame by a bridge, with frames that would otherwise exceed this limit being discarded.

c) For local bridging, the maximum BPDU transmission delay, defined as the maximum delay prior to the transmission of a BPDU on a LAN following the need to transmit such a BPDU arising, as specified in 7.7.

d) For local bridging, the maximum Message Age increment overestimate that may be made to the value of the Message Age parameter in BPDUs transmitted on LANs or to the age of stored configuration message information.

e) For remote bridging, the group maximum transit delay, defined as the maximum time elapsing between reception of a frame by a remote bridge and transmission of a forwarded frame by an adjacent remote bridge in the same group.

f) For remote bridging, the group maximum message propagation delay, defined as the maximum time between either reception of a configuration message on the root port of a remote bridge that is a primary bridge or expiry of the Hello Timer in a remote bridge that is the root bridge, and transmission of a corresponding Configuration BPDU on a Designated LAN port by an adjacent remote bridge in the same cluster.

g) For remote bridging, the group maximum Message Age increment overestimate, defined as the maximum amount by which the difference between Message Age values transmitted and received or originated as in f) can exceed the actual propagation delay incurred; the same limit applies to Message Age information stored with ports selected as alternate ports.

h) The values of the Bridge Hello Time, Bridge Max Age, Bridge Forward Delay, and Hold Time parameters.

Additionally, a bridge shall not

— Underestimate the increment to the Message Age parameter in transmitted configuration messages.

— Underestimate forward delay.

— Overestimate the Hello Time interval when acting as the root.

7.10.2 Parameter values

Recommended default, absolute maximum, and ranges of values for parameters are as specified in ISO/IEC 10038: 1993, subject to the following extensions and conditions that relate specifically to the operation of remote bridges (7.10.3 through 7.10.6).

7.10.3 Bridge diameter and effective bridge count

ISO/IEC 10038: 1993 uses bridge diameter as a convenient measure for the maximum transit delay and frame lifetime in a bridged LAN. This is based upon assumptions about transmission rates and media access delays that relate to the values typical for LANs. In particular, delay is assumed to occur in bridges, but to be negligible for the transfer of frames on the media between bridges.

In an RB-LAN, if the speed and/or transit delay of the non-LAN communications for a given group is significantly worse than would be expected on a LAN—producing transit delays of the order, say, of more than 100 ms—it is recommended that the remote bridge group be treated for configuration purposes as contributing a value greater than two to the diameter of the RB-LAN. (The minimum value of two for a group's contribution to the diameter corresponds simply to counting the two remote bridges on each path across the group, in the relaying model of Clause 6.)

The contribution of a group to the total diameter is known as its *effective bridge count*. An appropriate value for the effective bridge count (which is not required to take only integral values) may be derived from Table 7-1, working back from actual transit delay and propagation delay times. Alternatively, the default assignment

$$\textit{effective bridge count} = (2 + 50\,000/n)$$

may be made, where n bits per second is the limiting aggregate speed of the attachment of any bridge in the group to the non-LAN communications equipment.

7.10.4 Remote bridging transit and propagation delays

For remote bridging, Table 7-1 of this International Standard applies in place of ISO/IEC 10038: 1993, Table 4-2.

Table 7-1—Remote bridging transit and propagation delays

Parameter	Recommended value	Absolute maximum
Group maximum transit delay	1.0 × EBC	10.0
Group maximum message propagation delay	1.0 × EBC	15.0
Group maximum Message Age increment overestimate	2.0	8.0

All times in seconds
EBC = effective bridge count

7.10.5 Reclustering delay parameters, and relationships with forward delay

The permitted range and recommended default value for the Primary Reclustering Delay parameter are as specified in Table 7-2.

Table 7-2—Primary Reclustering Delay values

Parameter	Default value	Range
Primary Reclustering Delay	3.0	2.0–6.0

All times in seconds

The Primary Reclustering Delay should be chosen as an estimated upper bound on the expected time that it will typically take for new cluster membership information to propagate throughout the group (see 7.3.8.3). Its value is used in setting the reclustering delay that is used when the bridge is the primary bridge for the group [7.6.5 d)].

A remote bridge that attaches to one or more non-virtual-LAN groups shall ensure, for each such group, that the value of the Primary Reclustering Delay parameter satisfies the relationship:

$$\textit{Primary Reclustering Delay} \le \textit{Bridge Forward Delay} - 1.0\ \text{s} \qquad (1)$$

A primary bridge for a non-virtual-LAN group shall ensure that the value of the Reclustering Delay parameter for the group satisfies the relationship:

$$\textit{Reclustering Delay} \le \textit{Forward Delay} - 1.0\ \text{s} \qquad (2)$$

NOTES

1—In most practical cases, the Bridge Forward Delay values will exceed the Primary Reclustering Delay values by considerably more than the 1.0 s in condition (1), and similarly for Forward Delay and Reclustering Delay in condition (2). The minimum figure applies to an RB-LAN of diameter 2, (i.e., with one group and no local bridges).

2—Conditions (1) and (2) derive from considering how the performance characteristics of the whole RB-LAN are affected by the presence of one or more groups that can undergo reclustering periods when the RB-LAN topology changes (see C.7.10).

7.10.6 Path costs for virtual ports

The formula specified in 4.10.2 of ISO/IEC 10038: 1993 for default values for the Path Cost parameter shall apply also to virtual ports; the range of permitted values and minimum permitted value for path cost shall be as specified in Table 7-3.

NOTES

1—Where an RB-LAN is configured in such a way that the path costs of virtual ports can have a significant influence on the Spanning Tree structure, it is likely that values other than those given by the default formula will need to be assigned.

2—The extension of the range of permitted values from that in Table 4-5 of ISO/IEC 10038: 1993 accommodates attachment speeds down to 1000 bits/s under the default formula.

Table 7-3—Path cost parameter values

Parameter	Minimum value	Range
Path Cost	1	1–100 000

8. Relaying by remote bridges

Detailed mechanisms for support of the Internal Sublayer Service are outside the scope of this International Standard. Any such mechanisms shall provide the Internal Subnetwork Service defined in 5.4, in accordance with the relaying connectivity requirement specified in 8.1, the QOS requirements in 8.2, and the cluster integrity requirements in 8.3.

8.1 Relaying connectivity requirement

Any remote bridge cluster shall be fully connected with respect to transfer of MAC frames via the Internal Sublayer Service between virtual ports that are in the Forwarding state (i.e., when a MAC frame originating outside the cluster is received at a given remote bridge, the cluster shall support transfer of the frame to any other remote bridge in the cluster for forwarding outside the cluster through the appropriate ports of the other remote bridges). See 6.7.1.

NOTES

1—An important particular case is the flooding of a frame with an unknown destination MAC address to all other remote bridges in the cluster (see 6.7.1).

2—The requirement is stated with respect to clusters, not groups, since LAN-to-LAN relaying of MAC frames occurs only via clusters. However, since formation of clusters within a group is dynamic, fulfilling the requirement will in general involve considerations relevant to the group as a whole.

There is no requirement to transfer a frame across the cluster to any remote bridge that would filter the frame on all its other ports; implementations are free to optimize their use of cluster bandwidth in their own ways.

Conversely, there is no requirement either to transfer, or to avoid transferring, frames across a group between remote bridges belonging to different clusters, when the group has partitioned. However, any such frames are discarded on receipt as specified in 6.5: the cluster_id parameters of M-UNITDATA primitives are present to allow modeling of this behavior (see 8.3).

Failure to maintain full relaying connectivity shall result in virtual ports on one or more of the remote bridges being disabled, so that any remaining remote bridges form a fully connected set with respect to their support for relaying.

A failure of relaying connectivity shall be detected and acted upon within a stated maximum time. It is recommended that this time be the current Max Age value.

In the event of such a failure, an implementation of a group may divide the remote bridges into two or more subsets each operating as a separate group, provided that each of those new groups satisfies the connectivity requirement.

NOTES

3—In general, mechanisms additional to those used in maintaining the Spanning Tree will be needed to monitor the relaying connectivity, since Spanning Tree traffic does not flow along all the paths relevant for MAC frames.

4—Connectivity is only required, by the first paragraph of 8.1, for MAC-frame traffic that actually needs to flow through a cluster. If there is no applied user-traffic load for an extended period, therefore, measurement of the time to detect a failure of connectivity need not start until traffic resumes. This allows, for example, the bridges in a group to support relayed traffic over switched connections, with suitably short set-up delays, which are established on demand and released after periods of inactivity.

8.2 Relaying QOS requirements

8.2.1 Misordering and duplication

There shall be no misordering or duplication of frames transferred, with a given user_priority and with a given destination_address and source_address, between the virtual ports connecting any pair of remote bridges.

8.2.2 Loss

For each pair of peer virtual ports attaching to a given cluster, the probability that a correctly relayed M-UNITDATA indication does not occur at one virtual port following an M-UNITDATA request at the peer virtual port shall obey

$$\textit{Probability (non-occurrence of correctly relayed M-UNITDATA indication)} \leq 0.005$$

provided that

a) The Guaranteed Bridge Relaying Rate of each remote bridge involved in the relaying operation is not exceeded; and

b) The relayed frame is not subject to discard on the basis of either filtering or MSDU-size information in the bridges.

A correctly relayed M-UNITDATA indication is one in which the values of the destination_address, source_address, and user_data parameters are the same as the values of those parameters in the corresponding M-UNITDATA request, and in which the frame_check_sequence parameter, if present, has the value that corresponds to those parameter values for the MAC type indicated, at the indicated user_priority.

8.2.3 Undetected error rate

a) The probability that an M-UNITDATA indication with no frame_check_sequence parameter has a destination_address, source_address, or user_data parameter with a value different from that of the same parameter in the corresponding M-UNITDATA request (see C.5.3.7) shall obey

$$\textit{Probability (erroneous M-UNITDATA indication)} \leq 5 \times 10^{-14} \textit{ per octet of MSDU length}$$

b) The probability that an M-UNITDATA indication with a frame_check_sequence parameter has a destination_address, source_address, or user_data parameter with a value different from that of the same parameter in the corresponding M-UNITDATA request, but the frame_check_sequence parameter has the value that corresponds to the parameter values in the M-UNITDATA indication, for the MAC type indicated at the indicated user_priority (see C.5.3.7), shall also obey

$$\textit{Probability (erroneous M-UNITDATA indication)} \leq 5 \times 10^{-14} \textit{ per octet of MSDU length}$$

8.2.4 Transit delay

The transit delay which shall be used in assigning an effective bridge count to a group as specified in 7.10.3 is the elapsed time between receiving the last bit of a frame at a LAN port of one bridge of the group and transmitting the first bit of a corresponding relayed frame at a LAN port of another bridge of the group.

8.2.5 Maximum MSDU size

Any group shall support transfer of MSDUs between a given pair of its remote bridges, of sizes at least up to and including the largest MSDU size that is supported by both bridges, on some LAN port of each bridge.

8.2.6 Priority

The group may support an access_priority mechanism, with up to eight levels of priority (use of such a mechanism is as specified in 5.4 and 6.7.4).

8.3 Cluster integrity requirements

Correct operation of an RB-LAN requires that frames be relayed through any group only between pairs of bridges belonging to the same cluster. This requirement is expressed, in terms of the cluster_id parameters of the Internal Sublayer Service, in 6.5 a).

The remainder of this subclause specifies requirements on the mapping between abstract cluster_id parameters and the cluster identifier values created and exchanged by the bridges in a group, as specified in Clause 7. The specification is in terms of correspondences between cluster_id values and cluster identifier values or pairs of cluster identifier values.

NOTE—The concept of mapping or correspondence here is important: implementations are not required to transmit full 8-octet cluster identifiers with every frame. However, to comply with 6.5 a), implementations are required to be able to associate all transmitted and received frames with corresponding full cluster identifier values. Annex C includes some notes on ways in which such correspondences could be realized in implementations; see, for example, C.8.2.4.

8.3.1 Virtual LANs

In a virtual LAN, cluster membership is determined exactly by the spanning tree roles of the virtual ports. There can be only one cluster in the group, and any bridge with a designated port or root port attaching to the virtual LAN belongs to the cluster. Mapping of the cluster_id parameters is automatic, as specified in 7.3.4.3 for virtual LANs. No explicit exchange of cluster identifier information is needed.

8.3.2 Frames transmitted on subgroup ports

Any frame transmitted on a subgroup port shall have a cluster_id corresponding to both the Current Cluster Identifier and Old Cluster Identifier values for the group,

NOTES

1—When the Reclustering State for the group is Stable, or during a reclustering period that was entered from Stable state, the Current and Old Cluster Identifier values are equal, so a transmitted frame's cluster_id corresponds in fact to only a single cluster identifier value. During an overlap period, or during a reclustering period that was entered from an overlap period, a transmitted frame's cluster_id corresponds to two different cluster identifier values.

2—The New Cluster Identifier value for a group is never mapped to a cluster_id parameter. An updated New Cluster Identifier value is not established as valid for use in relaying frames until the end of the reclustering period. Use of only established values in mapping to cluster_id is necessary to preserve correctness (otherwise, for example, there could be leakage of frames relayed prematurely on the new active topology, using the new but not yet established cluster identifier, and a port state of Forwarding that will change to Listening on establishing the new cluster membership).

8.3.3 Frames received on subgroup ports

At a subgroup port that is in the Forwarding state (6.7.1), a received frame, with a cluster_id value corresponding to its transmitter's Current and Old Cluster Identifier values, shall be accepted if and only if

a) The received frame's Current Cluster Identifier value is equal to the Current Cluster Identifier value at the receiving bridge, or

b) The received frame's Old Cluster Identifier value is equal to the Current Cluster Identifier value at the receiving bridge, or

c) The received frame's Current Cluster Identifier value is equal to the Old Cluster Identifier value at the receiving bridge.

NOTES

1—When the receiving bridge's Old Cluster Identifier and Current Cluster Identifier values are equal, condition c) is clearly equivalent to condition a). Similarly, when the received frame's Current and Old Cluster Identifier values are the same, condition b) is equivalent to condition a).

2—A received frame is discarded if there is no match for equality between either of its Current and Old Cluster Identifier values and the Current and Old Cluster Identifier values at the receiving bridge, or if the Old Cluster Identifier values are equal but the Current Cluster Identifier values are unequal.

3—The simple formulation of the conditions a) through c) relies upon correct behavior by bridges when transmitting frames and when performing cluster selection and cluster port state selection. For example, in Overlap state, a frame should never be received that has a matching Old Cluster Identifier but a different Current Cluster Identifier, since this represents a change in cluster membership on the part of the transmitting bridge: that bridge's ports should therefore be in the Listening or Learning state, with no frames being transmitted.

4—See Annex C, and particularly C.8.2.4, for further discussion of relaying and cluster identification.

9. Bridge management

9.1 Management functions

The management functions defined for remote bridges are a compatible extension of those defined in ISO/IEC 10038: 1993 for local bridges. Management functions relating to LAN ports are identical to those in ISO/IEC 10038: 1993; management functions relating to ports in general are extended to apply equally to virtual ports and LAN ports. Some functions applicable to virtual ports require details that differ from the corresponding ones for LAN ports (e.g., allowable values for certain parameters). Additional management functions are defined to describe a remote bridge's view of the remote bridge group(s) of which it is a member.

9.2 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 remote MAC bridge are

a) The bridge management entity (9.4; 6.11).

b) The individual MAC entities associated with each LAN port (9.5; 6.2, 6.5, 6.6).

c) The forwarding process of the MAC relay entity (9.6; 6.2, 6.3, 6.7).

d) The filtering database of the MAC relay entity (9.7; 6.3, 6.9).

e) The bridge protocol entity (9.8; 6.10, Clause 7).

The management of each of these resources is specified in terms of managed objects and operations.

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, etc., that convey or otherwise represent this information. This International Standard specifies one such protocol encoding in 10.6 (for optional remote management).

9.3 Data types

The data types used are as specified in ISO/IEC 10038: 1993.

9.4 Bridge management entity

The bridge management entity is described in 6.11. The managed objects for the bridge management entity are

a) The Bridge Configuration managed object.

b) The Port Configuration managed objects, one for each LAN port and virtual port.

c) The Group Configuration managed objects, one for each group that is not a virtual LAN.

9.4.1 Bridge Configuration

The Bridge Configuration managed object models the operations that modify or enquire about the configuration of the remote bridge's resources. There is a single Bridge Configuration managed object per remote bridge.

The operations that can be performed on the Bridge Configuration object are Discover Bridge, Read Bridge, Set Bridge Name, and Reset Bridge.

9.4.1.1 Discover Bridge

9.4.1.1.1 Purpose

To solicit configuration information about the bridges in the bridged LAN.

9.4.1.1.2 Inputs

An Inclusion Range and Exclusion List, as specified in ISO/IEC 10038: 1993.

9.4.1.1.3 Outputs

For each bridge responding to the Discover Bridge operation, the outputs are those specified in 9.4.1.2.3 for the Read Bridge operation on a single bridge.

9.4.1.2 Read Bridge

9.4.1.2.1 Purpose

To obtain configuration information about a specific bridge.

9.4.1.2.2 Inputs

None.

9.4.1.2.3 Outputs

a) Bridge Address—the MAC address for the remote bridge, from which the bridge identifier used by the Spanning Tree Algorithm and Protocol is derived.

b) Bridge Name—a text string of up to 32 characters, of locally determined significance.

c) Number of LAN Ports—the total number of LAN ports.

d) LAN Port Addresses—a list specifying for each LAN port

   1) Port Number—the number of the LAN port.
   2) Port Address—the specific MAC address of the individual MAC entity associated with the port.

e) Number of Virtual LANs—the number of virtual LANs to which the bridge belongs.

f) Number of Non-Virtual-LAN Groups—the number of non-virtual-LAN remote bridge groups to which the bridge belongs.

g) Number of Subgroup Ports—the total number of virtual ports attaching to non-virtual-LAN groups.

h) Virtual Port List—a list specifying for each virtual port

   1) Port Number—the number of the virtual port.

   2) Group Number—zero for a virtual LAN port, and otherwise the group number (see 9.4.3) for the group to which the subgroup port attaches.

3) Virtual Port Type, an indication of whether the virtual port is

— A virtual LAN port, or
— An individual virtual port, or
— A multipeer virtual port, or
— A dummy virtual port.

i) Uptime—count in seconds of the time elapsed since the remote bridge was last reset or initialized.

9.4.1.3 Set Bridge Name

9.4.1.3.1 Purpose

To associate a text string, readable by the Read Bridge operation, with a bridge.

9.4.1.3.2 Inputs

Bridge Name—a text string of up to 32 characters.

9.4.1.3.3 Outputs

None.

9.4.1.4 Reset Bridge

9.4.1.4.1 Purpose

To reset the specified bridge. The filtering database is cleared and initialized with the entries specified in the permanent database; the Initialization procedure (7.8.1) is used to initialize the bridge protocol entity.

9.4.1.4.2 Inputs

None.

9.4.1.4.3 Outputs

None.

9.4.2 Port Configuration

The Port Configuration managed object models the operations that modify or enquire about the configuration of the ports of a remote bridge. There is a fixed set of LAN ports and virtual ports per remote bridge, each identified by a permanently allocated, non-zero, port number.

The allocated port numbers are not required to be consecutive. Also, some port numbers may be dummy entries, with no actual LAN port or virtual port (e.g., to allow for expansion of the bridge by addition of further MAC interfaces in the future, or because the implementation provides for some measure of dynamic reconfiguration of remote bridge groups). Such dummy ports shall support the Port Configuration management operations, and other port-related management operations in a manner consistent with the port being permanently disabled.

The information provided by the Port Configuration consists only of summary data indicating its name and type. The forwarding process managed object maintains port-specific counter information pertaining to the number of packets forwarded, filtered, and in error, and also information about the transmission priorities for ports; the Bridge Protocol Entity managed object maintains state information for each port.

The management operations that can be performed on each Port Configuration object are Read Port and Set Port Name.

9.4.2.1 Read Port

9.4.2.1.1 Purpose

To obtain general information regarding a specific LAN port or virtual port.

9.4.2.1.2 Inputs

Port Number—the number of the LAN port or virtual port.

9.4.2.1.3 Outputs

a) Port Name—a text string of up to 32 characters, of locally determined significance.

b) Port Type, either

   1) For a LAN port, the MAC entity type (ISO/IEC 8802-3; ISO/IEC 8802-4; ISO/IEC 8802-5; ISO/IEC 9314-2; ISO/IEC 8802-6; other); or
   2) For a virtual port, an indication of the type of non-LAN communications support.

9.4.2.2 Set Port Name

9.4.2.2.1 Purpose

To associate a text string, readable by the Read Port operation, with a LAN port or a virtual port.

9.4.2.2.2 Inputs

a) Port Number.

b) Port Name—a text string of up to 32 characters.

9.4.2.2.3 Outputs

None.

9.4.3 Group Configuration

The Group Configuration managed object models the operations that modify or enquire about the configuration information held by a remote bridge about a given non-virtual-LAN group to which it belongs. A given remote bridge belongs to a fixed set of such groups, each identified by a permanently allocated *group number*. The groups are numbered consecutively starting from 1, so that the highest allocated group number value is equal to the Number of Non-Virtual-LAN Groups in the Bridge Configuration, 9.4.1.2.3 f).

NOTE—Group numbers are local to a bridge: they are allocated purely for use in management operations, and are not exchanged between remote bridges as part of the topology computation, or in association with relaying of frames.

The information provided by the Group Configuration consists only of summary data indicating its name and the set of virtual ports by which the bridge attaches to the group. The Bridge Protocol Entity managed object maintains detailed parameter information for each group to which the remote bridge belongs.

9.4.3.1 Read Group

9.4.3.1.1 Purpose

To obtain general information regarding a bridge's attachments to a specific non-virtual-LAN group.

9.4.3.1.2 Inputs

Group Number—the number that identifies the group, within the set of non-virtual-LAN groups to which the remote bridge belongs.

9.4.3.1.3 Outputs

a) Group Name—a text string of up to 32 characters, of locally determined significance.

b) Virtual Ports—a list specifying the port numbers of each virtual port by which the remote bridge attaches to the group.

9.4.3.2 Set Group Name

9.4.3.2.1 Purpose

To associate a text string, readable by the Read Group operation, with the information held by a remote bridge about a given non-virtual-LAN group to which it belongs.

9.4.3.2.2 Inputs

a) Group Number.

b) Group Name—a text string of up to 32 characters.

9.4.3.2.3 Outputs

None.

9.5 MAC entities

The management operations and facilities provided for the MAC entities are those specified in the Layer Management standards for the individual MACs.

9.6 Forwarding Process

The Forwarding Process managed object maintains port-specific counter information pertaining to the number of packets forwarded, filtered, and in error, and also information about the transmission priorities for ports.

The Forwarding Process managed object consists of the Port Counters managed objects and the Transmission Priority managed objects (one object of each class for each port).

9.6.1 The Port Counters

The Port Counters object models the operations that can be performed on the port counters of the Forwarding Process managed object. There are multiple instances of the Port Counters object, one for each LAN port and virtual port of a given remote bridge.

9.6.1.1 Read Forwarding Port Counters

9.6.1.1.1 Purpose

To read the forwarding counters associated with a specific LAN port or virtual port.

9.6.1.1.2 Inputs

Port Number.

9.6.1.1.3 Outputs

a) Frames Received—count of valid frames received, as M-UNITDATA indication primitives.

b) Discard Inbound—count of valid frames received that were discarded (filtered) by the forwarding process.

c) Forward Outbound—count of frames forwarded to the associated MAC entity or subgroup.

d) Discard Lack of Buffers—count of frames that were to be transmitted but were discarded due to lack of buffers.

e) Discard Transit Delay Exceeded—count of frames that were to be transmitted but were discarded due to the maximum bridge transit delay being exceeded.

f) Discard on Error—count of frames that were to be forwarded on the associated MAC or subgroup but could not be transmitted (e.g., because the frame was too large).

g) Discard on Error Details—a list of 16 elements, each containing the source MAC address of a frame and the reason why the frame was discarded (frame too large, the only currently defined reason for error). The list is maintained as a circular buffer.

9.6.2 Transmission Priority

Management operations on Transmission Priority control how frame priority is handled for each transmitting port, in accordance with 6.7.4. The management operations that can be performed are Read Transmission Priority and Set Transmission Priority.

9.6.2.1 Read Transmission Priority

9.6.2.1.1 Purpose

To read the settings of the parameters governing the use of priority for relayed frames.

9.6.2.1.2 Inputs

Port Number.

9.6.2.1.3 Outputs

a) Outbound User Priority, in the range 0–7.

b) Outbound Access Priority, in the range 0–7.

9.6.2.2 Set Transmission Priority

9.6.2.2.1 Purpose

To set the parameters governing the use of priority for relayed frames.

9.6.2.2.2 Inputs

a) Port Number.

b) Outbound User Priority, in the range 0–7.

c) Outbound Access Priority, in the range 0–7.

9.6.2.2.3 Outputs

None.

9.7 The filtering database

The filtering database is described in 6.9: it contains information used by the forwarding process (6.7) in deciding through which ports of the remote bridge frames are to be forwarded. The definition here is equivalent to that in ISO/IEC 10038: 1993 for local bridges, with virtual ports included on an equal basis with LAN ports—see 9.7.5.1.2 c).

Managed objects relating to the filtering database are

a) The Filtering Database managed object.

b) The Static Entry managed objects.

c) The Dynamic Entry managed objects.

d) The Permanent Database managed object.

9.7.1 Filtering Database managed object

The Filtering Database managed object models the operations that can be performed on, or affect, the filtering database as a whole. There is a single Filtering Database object per remote bridge.

The management operations that can be performed on the Database are Read Filtering Database and Set Filtering Database Ageing Time, as defined in ISO/IEC 10038: 1993; and the Create Filtering Entry, Delete Filtering Entry, Read Filtering Entry, and Read Filtering Entry Range operations defined in 9.7.5.

9.7.2 Static Entry

A Static Entry object models the operations that can be performed on a single static entry in the filtering database. The set of Static Entry objects within the filtering database changes under management control. A Static Entry supports the Create Filtering Entry, Delete Filtering Entry, and Read Filtering Entry operations defined in 9.7.5.

9.7.3 Dynamic Entry

A Dynamic Entry models the operations that can be performed on a single dynamic entry in the filtering database (i.e., one that is created by the learning process). A Dynamic Entry supports the Delete Filtering Entry and Read Filtering Entry operations defined in 9.7.5.

9.7.4 Permanent Database

The Permanent Database managed object models the operations that can be performed on, or affect, the permanent database. There is a single permanent database per filtering database.

The management operations that can be performed on the Permanent Database are Read Permanent Database, as defined in ISO/IEC 10038: 1993; and the Create Filtering Entry, Delete Filtering Entry, Read Filtering Entry, and Read Filtering Entry Range operations defined in 9.7.5.

9.7.5 General filtering database operations

9.7.5.1 Create Filtering Entry

9.7.5.1.1 Purpose

To create an entry in the filtering database or permanent database. Only static entries may be created.

9.7.5.1.2 Inputs

a) Identifier—filtering database or permanent database.

b) Address—the MAC address of the entry.

c) Port Map—a list specifying for each port

   1) Inbound Port—the port number of the LAN port or virtual port.
   2) Outbound Ports—a set of Boolean indicators, one for each LAN port and virtual port. If a member of Outbound Ports is True, the entry permits forwarding to the associated port; otherwise, frames are filtered. The member of Outbound Ports that represents the Inbound Port takes the value False.

9.7.5.1.3 Outputs

None.

9.7.5.2 Delete Filtering Entry

The Delete Filtering Entry operation is as specified in ISO/IEC 10038: 1993.

9.7.5.3 Read Filtering Entry

9.7.5.3.1 Purpose

To read an entry from the filtering database or permanent database.

9.7.5.3.2 Inputs

a) Identifier—filtering database or permanent database.

b) Address—MAC address of the desired entry.

9.7.5.3.3 Outputs

a) Address—MAC address of the desired entry.

b) Entry Type—either Dynamic or Static.

c) Either

   1) Port Number, if the Entry Type is Dynamic, or
   2) Port Map, as defined in 9.7.5.1.2, if the Entry Type is Static.

9.7.5.4 Read Filtering Entry Range

9.7.5.4.1 Purpose

To read a range of entries from the filtering database or permanent database.

Since the number of values to be returned in the requested range may exceed the capacity of the service data unit conveying the management response, the returned entry range is identified. The indices that define the ranges take on values from zero up to Filtering Database Size minus one.

9.7.5.4.2 Inputs

a) Identifier—filtering database or permanent database.

b) Start Index—inclusive starting index of the desired entry range.

c) Stop Index—inclusive ending index of the desired entry range.

9.7.5.4.3 Outputs

a) Start Index—inclusive starting index of the returned entry range.

b) Stop Index—inclusive ending index of the returned entry range.

c) For each filtering entry returned

   1) Address—MAC address of the desired entry.

   2) Entry Type—either Dynamic or Static.

   3) Either
      i) Port Number, if the Entry Type is Dynamic, or
      ii) Port Map, as defined in 9.7.5.1.2, if the Entry Type is Static.

9.8 The bridge protocol entity

The bridge protocol entity is described in 6.10, and is responsible for performing the Spanning Tree Algorithm and Protocol as specified in Clauses 7 and 12.

The managed objects relating to the bridge protocol entity are

a) The Bridge Protocol Entity managed object.

b) The Bridge Port managed objects, one for each LAN port and virtual port.

c) The Remote Bridge Group managed objects, one for each group of which the remote bridge is a member.

9.8.1 Bridge Protocol Entity managed object

The Bridge Protocol Entity managed object models the operations that can be performed upon, or enquire about, the operation of the Spanning Tree Algorithm and Protocol. There is a single Bridge Protocol Entity object per remote bridge. The definition here is compatible with that in ISO/IEC 10038: 1993 for local bridges, extended only to allow virtual ports on an equal basis with LAN ports.

The management operations that can be performed on the Bridge Protocol Entity are Read Bridge Protocol Parameters and Set Bridge Protocol Parameters.

9.8.1.1 Read Bridge Protocol Parameters

9.8.1.1.1 Purpose

To obtain information regarding the remote bridge's bridge protocol entity.

9.8.1.1.2 Inputs

None.

9.8.1.1.3 Outputs

a) Bridge Identifier—as defined in 7.5.3.7.

b) Time Since Topology Change—count in seconds of the time elapsed since the Topology Change parameter for the remote bridge (7.5.3.12) was last True.

c) Topology Change Count—count of the number of times the Topology Change parameter has been set (i.e., has changed value from False to True) since the remote bridge was initialized.

d) Topology Change (7.5.3.12).

e) Designated Root (7.5.3.1).

f) Root Path Cost (7.5.3.2).

g) Root Port (7.5.3.3).

h) Max Age (7.5.3.4).

i) Hello Time (7.5.3.5).

j) Forward Delay (7.5.3.6).

k) Bridge Max Age (7.5.3.8).

l) Bridge Hello Time (7.5.3.9).

m) Bridge Forward Delay (7.5.3.10).

n) Hold Time (7.5.3.14).

9.8.1.2 Set Bridge Protocol Parameters

9.8.1.2.1 Purpose

To modify parameters in the remote bridge's bridge protocol entity, in order to control the configuration of the spanning tree, and/or to tune the reconfiguration time to suit a specific topology.

9.8.1.2.2 Inputs

a) Bridge Max Age—the new value (7.5.3.8).

b) Bridge Hello Time—the new value (7.5.3.9).

c) Bridge Forward Delay—the new value (7.5.3.10).

d) Bridge Priority—the new value of the priority part of the bridge identifier (7.5.3.7).

9.8.1.2.3 Outputs

None.

9.8.1.2.4 Behavior

The input parameter values are checked for compliance with the relationships specified in 4.10.2 of ISO/IEC 10038: 1993. If they do not comply, or if the value of Bridge Max Age or Bridge Forward Delay is less than the lower limit of the range specified in Table 4-3 of ISO/IEC 10038: 1993, no action shall be taken for any of the parameters. If the value of any of Bridge Max Age, Bridge Forward Delay, or Bridge Hello Time is outside the range specified in Table 4-3 of ISO/IEC 10038: 1993, the remote bridge is not required to take any action.

Otherwise, the remote bridge's Bridge Max Age, Bridge Hello Time, and Bridge Forward Delay parameters are set to the supplied values. The Set Bridge Priority procedure (7.8.4) is used to set the priority of the Bridge Identifier to the supplied value.

9.8.2 Bridge Port

A Bridge Port managed object models the operations on an individual LAN port or virtual port that relate to operation of the Spanning Tree Algorithm and Protocol. There is one Bridge Port object for each allocated port number value (see 9.4.2), each such object being permanently identified by its corresponding port number value.

The management operations that can be performed on a Bridge Port managed object are Read Port Parameters, Force Port State, and Set Port Parameters.

9.8.2.1 Read Port Parameters

9.8.2.1.1 Purpose

To obtain information regarding a specific port.

9.8.2.1.2 Inputs

Port Number—the number that identifies the LAN port or virtual port (9.4.2).

9.8.2.1.3 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) (7.4, 7.5.4.1.2).

c) Port Identifier—as defined in 7.5.4.1.1.

d) Path Cost (7.5.4.1.3).

e) Designated Root (7.5.4.1.4).

f) Designated Cost (7.5.4.1.5).

g) Designated Bridge (7.5.4.1.6).

h) Designated Port (7.5.4.1.7).

i) Topology Change Acknowledge (7.5.4.1.8).

The following output values are returned only for a subgroup port:

j) Peer New Cluster Identifier (7.5.4.2.1).

k) Peer Old Cluster Identifier (7.5.4.2.2).

9.8.2.2 Force Port state

9.8.2.2.1 Purpose

To force the specified port into the Disabled state, or to re-enable or reset the port by forcing it into the Blocking state.

9.8.2.2.2 Inputs

a) Port Number—the number that identifies the LAN port or virtual port.

b) State—either Disabled or Blocking (7.4, 7.5.4.1.2).

9.8.2.2.3 Outputs

None.

9.8.2.2.4 Behavior

If the selected state is Disabled, the Disable Port procedure (7.8.3) is used for the specified port. If the selected state is Blocking, the Enable Port procedure (7.8.2) is used.

9.8.2.3 Set Port Parameters

9.8.2.3.1 Purpose

To modify parameters for a port in order to control the configuration of the spanning tree.

9.8.2.3.2 Inputs

a) Port Number—the number that identifies the LAN port or virtual port.

b) Path Cost—the new value (7.5.4.1.3).

c) Port Priority—the new value of the priority field of the port identifier (7.5.4.1.1).

9.8.2.3.3 Outputs

None.

9.8.2.3.4 Behavior

The Set Path Cost procedure (7.8.6) is used to set the Path Cost parameter for the specified port. The Set Port Priority procedure (7.8.5) is used to set the priority part of the port identifier to the supplied value.

9.8.3 Remote Bridge Group

A Remote Bridge Group managed object models the operations relating to operation of the Spanning Tree Algorithm and cluster membership, for a given non-virtual-LAN remote bridge group of which the remote bridge is a member.

The management operations that can be performed on a Remote Bridge Group managed object are Read Group Parameters and Set Reclustering Delay.

9.8.3.1 Read Group Parameters

9.8.3.1.1 Purpose

To obtain information regarding the bridge's view of a particular non-virtual-LAN group.

9.8.3.1.2 Inputs

Group Number—the number that identifies the group (9.4.3).

9.8.3.1.3 Outputs

a) Reclustering State (7.5.5.1).

b) Current Cluster Identifier (7.5.5.2).

c) New Cluster Identifier (7.5.5.3).

d) Old Cluster Identifier (7.5.5.4).

e) Reclustering Delay (7.5.5.5).

f) Primary Reclustering Delay (7.5.5.6).

9.8.3.2 Set Primary Reclustering Delay

9.8.3.2.1 Purpose

To modify the value of the Primary Reclustering Delay parameter.

9.8.3.2.2 Inputs

a) Group Number—the number that identifies the group (9.4.3).

b) Primary Reclustering Delay—the new value desired for the Primary Reclustering Delay parameter.

9.8.3.2.3 Outputs

None.

9.8.3.2.4 Behavior

The input value for the reclustering delay is checked to be within the range specified in 7.10.5. If it is outside that range, no modification of the group parameter values shall occur; otherwise, the Primary Reclustering Delay parameter for the indicated group is set to the supplied value, and if the bridge is the primary bridge for the group, the Reclustering Delay parameter for the group is also set to the supplied value, adjusted if necessary for compliance with condition (2) specified in 7.10.5.

10. Management protocol

When the management facilities provided by a remote MAC bridge are realized through the use of ISO/IEC 15802-2: 1995 or ISO 9596-1: 1991, that realization shall be as specified for local MAC bridges in IEEE Std 802.1j-1996, with the additions specified in 10.1 through 10.6.

10.1 Mapping of operations onto LMMS services

The operations on objects defined in Clause 9 are carried out by means of the LMMS services specified in IEEE Std 802.1j-1996, with the additional mappings to the M-GET and M-SET LMMS services and parameters specified in Tables 10-1 through 10-6.

Table 10-1—Mapping of bridge management entity operations to LMMS services

Management operation	LMMS service element	Managed object class
Read Group	M-GET	Group
Set Group Name	M-SET	Group

Table 10-2—Mapping of bridge protocol entity operations to LMMS services

Management operation	LMMS service element	Managed object class
Read Group Parameters	M-GET	Group
Set Reclustering Delay	M-SET	Group

Table 10-3—Mapping of Read Group operation parameters (see 9.4.3.1)

M-GET parameter name	Req/Ind (operation inputs)	Rsp/Conf (operation outputs)
Base object class	Group	—
Base object instance	Name of Group managed object	—
Scope	Base object alone	—
Filter	Not required	—
Attribute identifier list	Read Group attribute group name	—
Managed object class	—	Group
Managed object instance	—	Name of Group managed object
Attribute list	—	Names / values of all attributes that are members of the Read Group attribute group (10.5.11)

Table 10-4—Mapping of Set Group Name operation parameters (see 9.4.3.2)

M-SET parameter name	Req/Ind (operation inputs)
Base object class	Group
Base object instance	Name of Group managed object
Scope	Base object alone
Filter	Not required
Modification list	Group Name attribute name and desired replacement value

Table 10-5—Mapping of Read Group Parameters operation parameters (see 9.8.3.1)

M-GET parameter name	Req/Ind (operation inputs)	Rsp/Conf (operation outputs)
Base object class	Group	—
Base object instance	Name of Group managed object	—
Scope	Base object alone	—
Filter	Not required	—
Attribute identifier list	Read Group Parameters attribute group name	—
Managed object class	—	Group
Managed object instance	—	Name of Group managed object
Attribute list	—	Names / values of all attributes that are members of the Read Group Parameters attribute group (10.5.12)

Table 10-6—Mapping of Set Reclustering Delay operation parameters (see 9.8.3.2)

M-SET parameter name	Req/Ind (operation inputs)
Base object class	Group
Base object instance	Name of Group managed object
Scope	Base object alone
Filter	Not required
Modification list	Primary Reclustering Delay attribute name and desired replacement value

10.2 Managed object containment structure

The containment structure specified in IEEE Std 802.1j-1996 applies to the managed objects for remote bridges, with the following addition.

Zero or more Group managed objects are contained in each MAC Bridge DLE managed object. The Group managed object models the manageable properties of a single group, other than a virtual LAN, to which a remote bridge attaches. These entries are instantiated at initialization, and are not created or deleted dynamically.

Figure 10-1 shows the resulting managed object containment structure for a remote MAC bridge. Elements indicated in the figure by dashed lines show the Data Link layer containment structure defined in ISO/IEC 10742: 1994.

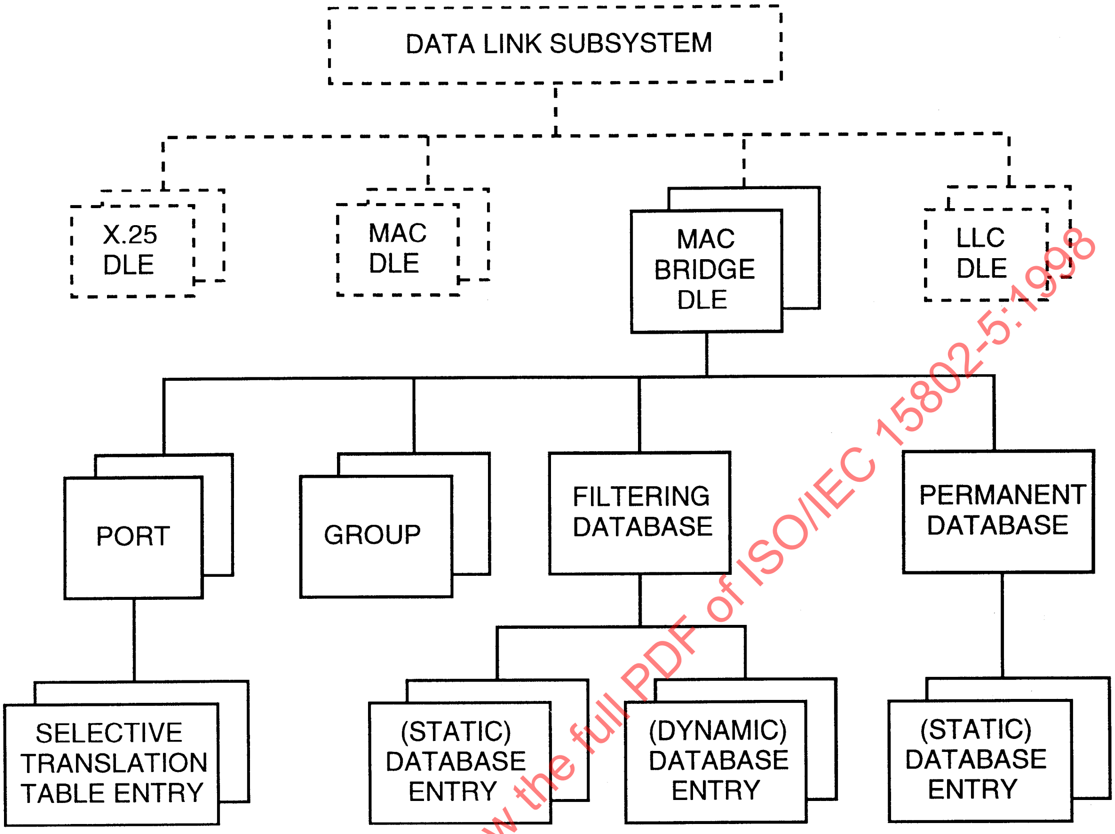


Figure 10-1—Managed object containment structure for a remote bridge

NOTE—The managed object classes defined in IEEE Std 802.1j-1996 are MAC Bridge DLE, Port, Selective Translation Table Entry, Permanent Database, Filtering Database, and Database Entry. The MAC Bridge DLE managed object combines the characteristics of the Bridge Configuration object (9.4.1) and the Bridge Protocol Entity object (9.8.1). A Port managed object combines the characteristics of the Port Configuration object (9.4.2), the Port Counters object (9.6.1), the Transmission Priority object (9.6.2), and the Bridge Port object (9.8.2). The Database Entry managed object combines the characteristics of the Static Entry and Dynamic Entry objects (9.7.2, 9.7.3).

10.3 Additions to MAC Bridge DLE managed object class definition

10.3.1 Remote MAC Bridge conditional package definition

The following conditional package is defined for the MAC Bridge DLE managed object.

```
pRemoteMACBridge PACKAGE
    BEHAVIOUR
        bRemoteMACBridge BEHAVIOUR
            DEFINED AS !Present when a MAC bridge supports remote MAC bridging!
        ;
    ;
    ATTRIBUTES
        aBridgeNumVirtualLANs               GET,
        aBridgeNumNonVLANGroups             GET,
        aBridgeNumSubgroupPorts             GET,
        aBridgeVirtualPorts                 GET
    ;
REGISTERED AS {Bridge.package remoteMACBridge (0)};
```

10.3.2 Bridge Number of Virtual LANs attribute definition

```
aBridgeNumVirtualLANs ATTRIBUTE
    WITH ATTRIBUTE SYNTAX Bridge.BridgeNumVirtualLANs ;
    MATCHES FOR EQUALITY, ORDERING ;
    BEHAVIOUR
        bBridgeNumVirtualLANs BEHAVIOUR
            DEFINED AS  !See 9.4.1.2.3 e). !
        ;
    ;
REGISTERED AS {Bridge.attribute bridgeNumVirtualLANs (52)};
```

10.3.3 Bridge Number of Non-Virtual-LAN Groups attribute definition

```
aBridgeNumNonVLANGroups ATTRIBUTE
    WITH ATTRIBUTE SYNTAX Bridge.BridgeNumNonVLANGroups ;
    MATCHES FOR EQUALITY, ORDERING ;
    BEHAVIOUR
        bBridgeNumNonVLANGroups BEHAVIOUR
            DEFINED AS  !See 9.4.1.2.3 f). !
        ;
    ;
REGISTERED AS {Bridge.attribute bridgeNumNonVLANGroups (53)};
```

10.3.4 Bridge Number of Subgroup Ports attribute definition

```
aBridgeNumSubgroupPorts ATTRIBUTE
    WITH ATTRIBUTE SYNTAX Bridge.BridgeNumSubgroupPorts ;
    MATCHES FOR EQUALITY, ORDERING ;
    BEHAVIOUR
        bBridgeNumSubgroupPorts BEHAVIOUR
            DEFINED AS  !See 9.4.1.2.3 g). !
        ;
    ;
REGISTERED AS {Bridge.attribute bridgeNumSubgroupPorts (54)};
```

10.3.5 Bridge Virtual Ports attribute definition

```
aBridgeVirtualPorts ATTRIBUTE
    WITH ATTRIBUTE SYNTAX Bridge.BridgeVirtualPorts ;
    MATCHES FOR EQUALITY;
    BEHAVIOUR
        bBridgeVirtualPorts BEHAVIOUR
            DEFINED AS  !See 9.4.1.2.3 h). !
        ;
    ;
REGISTERED AS {Bridge.attribute bridgeVirtualPorts (55)};
```

10.4 Additions to Port managed object class definition

10.4.1 Subgroup Port Parameters conditional package definition

The following conditional package is defined for the Port managed object.

```
pSubgroupPort PACKAGE
    BEHAVIOUR
        bSubgroupPort BEHAVIOUR
            DEFINED AS !Present when the Port managed object is for a subgroup port!
        ;
    ;
    ATTRIBUTES
        aPeerNewClusterId                       GET,
        aPeerOldClusterId                       GET
    ;
REGISTERED AS {Bridge.package subgroupPort (1)};
```

10.4.2 Peer New Cluster Identifier attribute definition

```
aPeerNewClusterId ATTRIBUTE
    WITH ATTRIBUTE SYNTAX Bridge.ClusterIdentifier ;
    MATCHES FOR EQUALITY, ORDERING ;
    BEHAVIOUR
        bPeerNewClusterId BEHAVIOUR
            DEFINED AS  !See 9.8.2.1.3 j), 7.5.4.2.1. !
        ;
    ;
REGISTERED AS {Bridge.attribute peerNewClusterId (56)};
```

10.4.3 Peer Old Cluster Identifier attribute definition

```
aPeerOldClusterId ATTRIBUTE
    WITH ATTRIBUTE SYNTAX Bridge.ClusterIdentifier ;
    MATCHES FOR EQUALITY, ORDERING ;
    BEHAVIOUR
        bPeerOldClusterId BEHAVIOUR
            DEFINED AS  !See 9.8.2.1.3 k), 7.5.4.2.2. !
        ;
    ;
REGISTERED AS {Bridge.attribute peerOldClusterId (57)};
```

10.5 Group managed object class and attributes

10.5.1 Group managed object class definition

```
oGroup MANAGED OBJECT CLASS
    DERIVED FROM “ISO/IEC 10165-2”:top;
    CHARACTERIZED BY
        pGroup PACKAGE
            BEHAVIOUR
                bGroup BEHAVIOUR
                    DEFINED AS            !The Group managed object class combines the
                                          characteristics of the Group Configuration managed
                                          object and the Remote Bridge Group managed object,
                                          as defined in 9.4.3 and 9.8.3. !
                ;
            ;
            ATTRIBUTES
                aGroupNumber              GET,              -- Naming attribute
                aGroupName                GET-REPLACE,
                aGroupVirtualPorts        GET,
                aReclusteringState        GET,
                aCurrentClusterId         GET,
                aNewClusterId             GET,
                aOldClusterId             GET,
                aReclusteringDelay        GET,
                aPrimaryReclusteringDelay GET-REPLACE
            ;
            ATTRIBUTE GROUPS
                agReadGroup,
                agReadGroupParameters
            ;
REGISTERED AS {Bridge.moClass group (6)};
```

```
nbGroup-MACBridgeDLE NAME BINDING
    SUBORDINATE OBJECT CLASS            oGroup AND SUBCLASSES;
    NAMED BY SUPERIOR OBJECT CLASS      oMACBridgeDLE AND SUBCLASSES;
    WITH ATTRIBUTE                      aGroupNumber;
REGISTERED AS {Bridge.nameBinding group-MACBridgeDLE (8)};
```

10.5.2 Group Number attribute definition

```
aGroupNumber ATTRIBUTE
    WITH ATTRIBUTE SYNTAX Bridge.GroupNumber ;
    MATCHES FOR EQUALITY, ORDERING ;
    BEHAVIOUR
        bGroupNumber BEHAVIOUR
            DEFINED AS  !See 9.4.3, 9.4.3.1.2. !
        ;
    ;
REGISTERED AS {Bridge.attribute groupNumber (58)};
```

10.5.3 Group Name attribute definition

```
aGroupName ATTRIBUTE
    WITH ATTRIBUTE SYNTAX Bridge.GroupName ;
    MATCHES FOR EQUALITY ;
    BEHAVIOUR
        bGroupName BEHAVIOUR
            DEFINED AS  !See 9.4.3.1.3 a). !
        ;
    ;
REGISTERED AS {Bridge.attribute groupName (59)};
```

10.5.4 Group Virtual Ports attribute definition

```
aGroupVirtualPorts ATTRIBUTE
    WITH ATTRIBUTE SYNTAX Bridge.GroupVirtualPorts ;
    MATCHES FOR EQUALITY ;
    BEHAVIOUR
        bGroupVirtualPorts BEHAVIOUR
            DEFINED AS  !See 9.4.3.1.3 b). !
        ;
    ;
REGISTERED AS {Bridge.attribute groupVirtualPorts(60)};
```

10.5.5 Reclustering State attribute definition

```
aReclusteringState ATTRIBUTE
    WITH ATTRIBUTE SYNTAX Bridge.ReclusteringState ;
    MATCHES FOR EQUALITY ;
    BEHAVIOUR
        bReclusteringState BEHAVIOUR
            DEFINED AS  !See 9.8.3.1.3 a), 7.5.5.1. !
        ;
    ;
REGISTERED AS {Bridge.attribute reclusteringState(61)};
```

10.5.6 Current Cluster Identifier attribute definition

```
aCurrentClusterId ATTRIBUTE
    WITH ATTRIBUTE SYNTAX Bridge.ClusterIdentifier ;
    MATCHES FOR EQUALITY ;
    BEHAVIOUR
        bCurrentClusterId BEHAVIOUR
            DEFINED AS  !See 9.8.3.1.3 b), 7.5.5.2. !
        ;
    ;
REGISTERED AS {Bridge.attribute currentClusterId (62)};
```

10.5.7 New Cluster Identifier attribute definition

```
aNewClusterId ATTRIBUTE
    WITH ATTRIBUTE SYNTAX Bridge.ClusterIdentifier ;
    MATCHES FOR EQUALITY ;
    BEHAVIOUR
        bNewClusterId BEHAVIOUR
            DEFINED AS  !See 9.8.3.1.3 c), 7.5.5.3. !
        ;
    ;
REGISTERED AS {Bridge.attribute newClusterId (63)};
```

10.5.8 Old Cluster Identifier attribute definition

```
aOldClusterId ATTRIBUTE
    WITH ATTRIBUTE SYNTAX Bridge.ClusterIdentifier ;
    MATCHES FOR EQUALITY ;
    BEHAVIOUR
        bOldClusterId BEHAVIOUR
            DEFINED AS  !See 9.8.3.1.3 d), 7.5.5.4. !
        ;
    ;
REGISTERED AS {Bridge.attribute oldClusterId (64)};
```

10.5.9 Reclustering Delay attribute definition

```
aReclusteringDelay ATTRIBUTE
    WITH ATTRIBUTE SYNTAX Bridge.ReclusteringDelay ;
    MATCHES FOR EQUALITY, ORDERING ;
    BEHAVIOUR
        bReclusteringDelay BEHAVIOUR
            DEFINED AS  !See 9.8.3.1.3 e), 7.5.5.5. !
        ;
    ;
REGISTERED AS {Bridge.attribute reclusteringDelay(65)};
```

10.5.10 Primary Reclustering Delay attribute definition

```
aPrimaryReclusteringDelay ATTRIBUTE
    WITH ATTRIBUTE SYNTAX Bridge.ReclusteringDelay ;
    MATCHES FOR EQUALITY, ORDERING ;
    BEHAVIOUR
        bPrimaryReclusteringDelay BEHAVIOUR
            DEFINED AS  !See 9.8.3.1.3 f), 7.5.5.6. !
        ;
    ;
REGISTERED AS {Bridge.attribute primaryReclusteringDelay(66)};
```

10.5.11 Read Group attribute group definition

```
agReadGroup ATTRIBUTE GROUP
    GROUP ELEMENTS
        aGroupName,
        aGroupVirtualPorts
    ;
    FIXED ;
    DESCRIPTION   !This attribute group is used in the mapping of the Read Group operation, as
                  defined in Table 10-3.!
    ;
REGISTERED AS {Bridge.attributeGroup readGroup (8)};
```

10.5.12 Read Group Parameters attribute group definition

```
agReadGroupParameters ATTRIBUTE GROUP
    GROUP ELEMENTS
        aReclusteringState,
        aCurrentClusterId,
        aNewClusterId,
        aOldClusterId,
        aReclusteringDelay,
        aPrimaryReclusteringDelay
    ;
    FIXED ;
    DESCRIPTION   !This attribute group is used in the mapping of the Read Group Parameters
                  operation, as defined in Table 10-5.!
    ;
REGISTERED AS {Bridge.attributeGroup readGroupParameters (9)};
```

10.6 ASN.1 definitions

The definitions of data types and data values given below are added to the Bridge ASN.1 module specified in IEEE Std 802.1j-1996, resulting in version 2 of the module, as follows.

```
Bridge (iso(1) member-body(2) us(840) ieee802dot1D(10009) asn1Module(2) bridgeDefinitions(0)
        version2(1))

DEFINITIONS ::=  BEGIN

IMPORTS
        everything
FROM Bridge {iso(1) member-body(2) us(840) ieee802dot1D(10009) asn1Module(2)
                bridgeDefinitions(0) version1(0)}
                --              All definitions from IEEE Std 802.1j-1996
; -- End of IMPORTS

BridgeNumVirtualLANs    ::=  INTEGER                  --  Range 0–255
BridgeNumNonVLANGroups  ::=  INTEGER                  --  Range 0–255
BridgeNumSubgroupPorts  ::=  INTEGER                  --  Range 0–255
BridgeVirtualPorts      ::=  SET OF { SEQUENCE      { PortNumber,
                                                       GroupNumber,
                                                       VirtualPortType } }
ClusterIdentifier       ::=  SEQUENCE    { ClusterIndex,
                                             BridgeAddress }
ClusterIndex            ::=  INTEGER                  --  Range 0–65535
GroupName               ::=  GraphicString            --  Max 32 characters
GroupNumber             ::=  INTEGER                  --  Range 1–255
GroupVirtualPorts       ::=  SET OF { PortNumber }
ReclusteringDelay       ::=  INTEGER                  --  Units of 1/256 seconds
ReclusteringState       ::=  INTEGER     { Stable (0),
                                             Reclustering (1),
                                             Overlap (2) }
VirtualPortType         ::=  INTEGER     { VLANport (0),
                                             IndVirtPort (1),
                                             MultipeerVirtPort (2),
                                             DummyVirtPort (3) }

END
```

11. Performance

The performance parameters specified in ISO/IEC 10038: 1993 for local bridges apply also to remote bridges (i.e., the Guaranteed Port Filtering Rate for each LAN port, and the Guaranteed Bridge Relaying Rate for the remote bridge).

12. The Extended Spanning Tree Protocol

12.1 Introduction

This clause specifies an optional protocol that may be used by a set of remote bridges, to support the Spanning Tree Algorithm over one or more remote bridge groups. The protocol is a straightforward extension of the Spanning Tree Protocol for use on LANs, specified in ISO/IEC 10038: 1993.

The specification makes use of the definitions of the logical protocol parameters, contained in 7.5.

Much of the protocol is specified by reference to the specification in ISO/IEC 10038: 1993 for local bridges. When provisions of ISO/IEC 10038: 1993 are invoked in this way, references in them to parameters and elements of procedure are to be interpreted as referring in the first instance to the corresponding specifications in this International Standard, rather than to those in ISO/IEC 10038: 1993 itself.

Where the phrase “extended for designated virtual ports” is used, this means that there is a reference in the ISO/IEC 10038: 1993 text to the designated port for the LAN to which a (LAN) port is attached; when applied to a virtual port, that reference shall be taken to mean a virtual port whose spanning tree role is designated port.

NOTE—Where a remote bridge attaches to a virtual LAN, the remote bridge’s operation of the Extended Spanning Tree Protocol is precisely equivalent to operation of the Spanning Tree Protocol as specified in ISO/IEC 10038: 1993, with virtual ports included on an equal basis with LAN ports. The additional parameters and elements of procedure specific to the Extended Spanning Tree Protocol are not required, since they are defined to enable the protocol to support the more general interconnection configurations where remote bridges attach to groups by multiple virtual ports. The only extension of the Spanning Tree Protocol in this case is its application to virtual LAN ports as well as to LAN ports.

12.2 Bridge-protocol support in remote bridge groups

Specification of detailed mechanisms for support of the Extended Spanning Tree Protocol over specific non-LAN communications technologies is outside the scope of this International Standard. Such mechanisms shall provide the subnetwork service defined in 12.2.1, in accordance with the connectivity requirement specified in 12.2.2.

Transmission and reception of BPDUs are mapped to the subnetwork service as specified in 12.2.3.

12.2.1 RB-Protocol Subnetwork Service

The service to be provided within a group in order to support the Extended Spanning Tree Protocol is an unacknowledged connectionless-mode service, operating in broadcast mode within each subgroup, as follows.

12.2.1.1 Primitives and parameters

The primitives and parameters that describe the RB-Protocol Subnetwork Service are

```
RBP-UNITDATA request    ( destination_RB_address,
                          source_RB_address,
                          RBP_service_data_unit )
```

RBP-UNITDATA indication (destination_RB_address,
source_RB_address,
RBP_service_data_unit)

The destination_RB_address parameter in an RBP-UNITDATA request issued at a virtual port identifies all bridges in the subgroup to which the requesting virtual port belongs.

The source_RB_address parameter in an RBP-UNITDATA request identifies the remote bridge at which the primitive is issued.

The RBP_service_data_unit parameter in an RBP-UNITDATA request is a sequence of four or more octets.

An RBP-UNITDATA request results in one corresponding RBP-UNITDATA indication primitive at each other remote bridge in the same subgroup, subject to a small probability of loss (see 12.2.1.2.2). The indication primitives occur at the peer virtual ports of the requesting virtual port, except possibly at (some or all) peer ports that are disabled.

The parameters of each corresponding RBP-UNITDATA indication primitive take the same values as those in the original RBP-UNITDATA request primitive.

No corresponding RBP-UNITDATA indication primitive occurs at any virtual port attached to a different subgroup, or to a different group.

NOTES

1—This last characteristic is essential for correct operation of the Extended Spanning Tree Protocol. Propagation of BPDUs has to be confined to a single subgroup, in the same way that it is confined to a single LAN for the Spanning Tree Protocol of ISO/IEC 10038: 1993.

2—Whether or not a protocol providing this service needs to convey explicit information in support of the subgroup-addressing requirements will depend upon the characteristics of the group's supporting communications capabilities. In the case of a group consisting of two bridges and a point-to-point link, addressing can be entirely implicit. In the case of a network supporting several groups, it could well be necessary to convey group or subgroup identification information in PDUs supporting this service, in order to ensure correct occurrence of corresponding indication primitives.

12.2.1.2 Quality of service (QOS)

NOTE—The QOS specified here relates only to the conveyance of BPDUs of the Extended Spanning Tree Protocol. It is independent of the QOS requirements for relaying, specified in 8.2.

12.2.1.2.1 Misordering and duplication

The subnetwork service does not misorder or duplicate information transferred between a given pair of remote bridges (i.e., the order of a sequence of RBP-UNITDATA indication primitives at a virtual port, all of which have the same source_RB_address, is the same as the order of the corresponding request primitives). The definition in 12.2.1.1 of the occurrence of RBP-UNITDATA indication primitives resulting from RBP-UNITDATA requests precludes duplication.

12.2.1.2.2 Loss

The probability that a corresponding RBP-UNITDATA indication will not occur after an RBP-UNITDATA request shall not be higher than 0.005.

12.2.1.2.3 Undetected error rate

The probability that the parameters of an RBP-UNITDATA indication differ from those of the corresponding RBP-UNITDATA request shall not exceed 1.5×10^{-5} for an RBP_service_data_unit length of 54 octets.

NOTE—This probability can be assured, for example, by use of a 16-bit cyclic redundancy check, or by use of a pair of modulo-255 checksum octets calculated as specified in ISO/IEC 8473-1: 1994[4] or ISO/IEC 8073: 1992.[5]

12.2.1.2.4 Transit delay

The transit delay of the subnetwork service is the mean elapsed time between an RBP-UNITDATA request and any corresponding RBP-UNITDATA indication. This is used in assigning an effective bridge count to the group as specified in 7.10.3.

12.2.1.2.5 Throughput

The subnetwork service is required to support a mean throughput at any virtual port of the order of at most

$$440\ (n-1)\ \text{bit/s}$$

of RBP_service_data_units where there are *n* remote bridges in the group. The requirement is likely to be substantially less than that in many practical configurations.

NOTE—See C.12.2 for discussion of this requirement.

12.2.1.2.6 SDU size

The subnetwork service is required to support transfer of RBP_service_data_units of size up to 54 octets.

12.2.2 Group connectivity requirement

The set of active bridges in any group shall be fully connected with respect to provision of the RB-Protocol Subnetwork Service (i.e., any active bridge in the group shall be able to communicate via the subnetwork service with each of the other remote bridges in the group, via the virtual port attaching to the appropriate subgroup). A bridge is defined to be active in a group if the virtual ports attaching it to the group are not disabled.

The connectivity represented in this way for the RB-Protocol Subnetwork Service shall reflect the relaying connectivity, as in 8.1. Where a group is partitioned into multiple clusters (including isolated remote bridges), the RB-Protocol Subnetwork Service is only required to reflect the potential connectivity between remote bridges belonging to different clusters in the group. This potential connectivity would be realized in the event that a reconfiguration and consequent reclustering called for establishment of relaying between previously separated bridges.

Failure to maintain the full RB-Protocol Subnetwork Service connectivity shall result in one or more of the remote bridges ceasing to be active (as defined above), so that any remaining active remote bridges form a fully connected set.

[4] ISO/IEC 8473-1: 1994, Information technology—Protocol for providing the connectionless-mode network service: Protocol specification.

[5] ISO/IEC 8073: 1992, Information technology—Telecommunications and information exchange between systems—Open Systems Interconnection—Protocol for providing the connection-mode transport service.

In the event of such a failure, the remote bridges of a group may be divided into two or more subsets, each operating as a separate group, provided that each of those new groups satisfies the connectivity requirement.

Each remote bridge shall detect connectivity failure within a stated maximum time; it is recommended that this time be the current Max Age value.

12.2.3 Mapping of BPDU transfer to the RB-Protocol Subnetwork Service

Each BPDU is conveyed across the group as the complete RBP_service_data_unit parameter of an RBP-UNITDATA request and, hence, of the corresponding RBP-UNITDATA indication primitive(s).

12.3 Protocol parameters and timers

The logical information in configuration messages and topology change notifications (7.5.1, 7.5.2) is transferred between the bridge protocol entities of individual bridges in the concrete form of BPDUs. This subclause specifies (12.3.1, 12.3.2) how those parameters are conveyed in the two corresponding types of BPDUs: Configuration BPDUs and Topology Change Notification BPDUs. The encoding of these parameters and additional information elements in BPDUs is specified in 12.4 and 12.5.

Each bridge protocol entity maintains the remote bridge parameters, port parameters, and remote bridge group parameters specified in 7.5.3, 7.5.4, and 7.5.5, respectively. The specification of the Extended Spanning Tree Protocol makes use of some additional parameters, and of a number of timers for the remote bridge and for each port. The parameters and timers are specified in 12.3.3 through 12.3.8.

12.3.1 Parameters of Configuration BPDUs

The parameters contained in Configuration BPDUs transmitted and received at LAN ports and virtual LAN ports are those specified in 7.5.1.1 through 7.5.1.8 for configuration messages. Configuration BPDUs transmitted and received at virtual LAN ports may also contain parameters specified in 7.5.1.9 through 7.5.1.11. The parameters contained in Configuration BPDUs transmitted and received at subgroup ports are those specified in 7.5.1.1 through 7.5.1.11.

12.3.2 Parameters of Topology Change Notification BPDUs

Topology Change Notification BPDUs transmitted and received at LAN ports and virtual LAN ports do not have any parameters. Topology Change Notification BPDUs transmitted and received at subgroup ports shall have the parameters specified in 12.3.2.1 through 12.3.2.4.

12.3.2.1 Cluster Identifier

The Cluster Identifier parameter is as specified in 7.5.2.1.

12.3.2.2 Originating Bridge Identifier

The Originating Bridge Identifier parameter takes as its value the unique bridge identifier of the bridge that detected the topology change that originally gave rise to this notification. This parameter is absent, or takes the null bridge identifier value, when that information is not available.

12.3.2.3 Port Identifier

When a non-null Originating Bridge Identifier is present, the Port Identifier parameter shall be present, and takes the value of the port identifier for the port whose change of state to Forwarding or Blocking gave rise to this notification; otherwise, this parameter is not significant.

12.3.2.4 Reason

When a significant Port Identifier parameter is present, the Reason parameter shall be present, and indicates the reason for the notification associated with the port's change of state. Values are from the set

- Port state Forwarding
- Port state Blocking, condition unspecified
- Port state Blocking, spanning tree
- Port state Blocking, isolated remote bridge
- Port state Blocking, partitioned cluster
- Port state Blocking, management reset

where the new state of the port is identified in the value.

12.3.3 Remote bridge parameters

The parameters that describe a remote bridge operating the Extended Spanning Tree Protocol are those specified in 7.5.3, with no additions.

12.3.4 Remote bridge timers

12.3.4.1 Hello Timer

The Hello Timer serves to ensure periodic transmission of configuration message information by the remote bridge when it is, or is attempting to become, the root. The timeout value of the timer is that of the bridge's Bridge Hello Time parameter.

12.3.4.2 Topology Change Notification Timer

The Topology Change Notification Timer serves to ensure that the designated bridge on the LAN or cluster to which the bridge's root port is attached is notified of any detected topology change. The timeout value of the timer is that of the bridge's Bridge Hello Time parameter.

12.3.4.3 Topology Change Timer

The Topology Change Timer serves to determine the time period for which the remote bridge transmits Configuration BPDUs with the Topology Change flag set, following the detection of a topology change, when it is the root. The timeout value of the timer is that of the bridge's Topology Change Time parameter.

12.3.5 Port parameters

The parameters that describe a LAN port or virtual port of a remote bridge operating the Extended Spanning Tree Protocol are as specified in 7.5.4, with the additional Configuration Pending parameter as specified in 12.3.5.1.

12.3.5.1 Configuration Pending

The Configuration Pending parameter is a Boolean parameter, which is set to record that a Configuration BPDU is to be transmitted on expiry of the Hold Timer for the port. This parameter is used, in conjunction with the Hold Timer for the port, to ensure that Configuration BPDUs are not transmitted too frequently, but that up-to-date information is transmitted.

12.3.6 Port timers

12.3.6.1 Message Age Timer

The Message Age Timer serves to measure the age of the received configuration message information recorded for a root port or alternate port, and to ensure that this information is discarded when its age exceeds the value of the Max Age parameter recorded by the remote bridge. The timeout value of this timer is that of the bridge's Max Age parameter.

12.3.6.2 Forward Delay Timer

The Forward Delay Timer determines the time spent by a port in the Listening and Learning states. The timeout value of this timer is that of the bridge's Forward Delay parameter.

12.3.6.3 Hold Timer

The Hold Timer serves to ensure that Configuration BPDUs are not transmitted too frequently through any port. The timeout value of this timer is that of the bridge's Hold Time parameter.

12.3.7 Remote bridge group parameters

The parameters that describe a remote bridge group, other than a virtual LAN, over which the Extended Spanning Tree Protocol is operated are those specified in 7.5.5, with no additions. A virtual LAN requires no parameters at a bridge additional to those for the virtual port by which the bridge attaches to the virtual LAN.

12.3.8 Reclustering Delay Timer

There is one Reclustering Delay Timer for each remote bridge group that is not a virtual LAN. The timer implements the reclustering period, and also the overlap period that follows a reclustering period (7.3.8). The timeout value for each period is the value of the bridge's Reclustering Delay parameter for the group.

12.4 Encoding of BPDUs on LANs

A remote bridge that operates the Extended Spanning Tree Protocol over its virtual ports shall encode BPDUs transmitted on its LAN ports, and validate BPDUs received on its LAN ports, as specified in ISO/IEC 10038: 1993.

12.5 Encoding of Extended Spanning Tree Protocol BPDUs in remote bridge groups

NOTE—The Extended Spanning Tree Protocol specified here is identified as version 1 of the Spanning Tree Protocol, where the basic Spanning Tree Protocol (specified for LANs in ISO/IEC 10038: 1993) is identified as version 0. The encoding specified for BPDUs of the Extended Spanning Tree Protocol is a compatible extension of the encoding specified for BPDUs in ISO/IEC 10038: 1993, using the same value of protocol identifier but a different value of protocol version identifier. In a virtual LAN, version 0 BPDUs can be used, encoded exactly as specified in ISO/IEC 10038: 1993; see 12.5.1.3 of this International Standard.

12.5.1 BPDU structure

12.5.1.1 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 in which they occur in an RBP_service_data_unit parameter (see 12.2.1.1). The bits in an octet are numbered from 1 to 8, where bit 1 is the low-order bit.

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

When octet values are represented as sequences of bits in 12.5.2 and 12.5.3, the bits within each octet are shown with bit 8 to the left and bit 1 to the right (grouped in fours to improve legibility).

12.5.1.2 BPDU components

Each BPDU of the Extended Spanning Tree Protocol shall consist of, in order

- A BPDU header
- A set of version 0 BPDU parameters
- A version 1 parameter-length indicator
- A set of version 1 BPDU parameters

12.5.1.2.1 BPDU headers

Each BPDU header contains a protocol identifier, a protocol version identifier, and a BPDU type.

The protocol identifier value is encoded in the initial octets of each BPDU, and is that specified by ISO/IEC 10038: 1993 for use by bridge protocol entities operating the Spanning Tree Algorithm and Protocol. This International Standard places no further restriction on the structure, encoding, or use of BPDUs with different values of the protocol identifier, should these be specified by other standard protocols.

The protocol version identifier distinguishes BPDUs of the Extended Spanning Tree Protocol, version 1, from those of the basic Spanning Tree Protocol, version 0.

The BPDU type takes the same values as specified in ISO/IEC 10038: 1993, and determines the parameters that the BPDU can contain.

12.5.1.2.2 Version 0 BPDU parameters

For each BPDU type, the version 0 parameters are exactly the parameters specified for that BPDU type in ISO/IEC 10038: 1993, encoded in the same way.

NOTE—In particular, the set of version 0 parameters for a given BPDU type consists of a fixed number of parameters, occurring in a fixed order and each of fixed length.

12.5.1.2.3 Version 1 parameter length

Immediately following the version 0 parameters, each BPDU contains an indicator of the length in octets of the set of version 1 parameters present.

NOTE—Use of this explicit length serves two functions:

a) It provides for the future specification of further extensions of the protocol, as further new versions, with the flexibility for such versions either to include the version 1 parameters in the same way as version 1 includes the version 0 parameters, or to add new parameters directly to the version 0 set; in the latter case, use of a version 1 parameter length of zero would be all that was required in order to maintain compatibility of encoding.

b) It provides for some of the version 1 parameters to be optionally present, according to the needs of the transmitting bridge protocol entity.

12.5.1.2.4 Version 1 parameters

The set of version 1 parameters that can be present is determined by the BPDU type of each BPDU. Each parameter is encoded in one or more octets and is of fixed length. The order of occurrence of the version 1 parameters in each given type of BPDU is fixed; parameters may be omitted from the end of the set, with default values applying, as specified in 12.5.3.

12.5.1.3 BPDUs in virtual LANs

BPDUs transmitted through a virtual LAN port may be structured as specified above, or they may be structured as specified in ISO/IEC 10038: 1993, with protocol version identifier for version 0 and without the version 1 parameter length and parameters. When version 1 BPDUs are transmitted through a virtual LAN port, the version 1 parameters are set to default values as specified in 12.6.1.3 c) and 12.6.6.3 c).

12.5.2 Encoding of parameter types

12.5.2.1 Encoding of parameter types common to version 0 and version 1

The following BPDU header elements and version 0 parameters are encoded as specified in ISO/IEC 10038: 1993.

— Protocol identifiers
— Protocol version identifiers
— BPDU types
— Flags
— Bridge identifiers
— Root path cost
— Port identifiers
— Timer values

The bridge identifier null value shall be encoded as eight octets, each assigned the value 0000 0000.

12.5.2.2 Encoding of cluster identifiers

A cluster identifier shall be encoded as a sequence of eight octets. The null cluster identifier is encoded as eight octets each assigned the value 0000 0000. Otherwise, the first two octets encode the cluster index, considered as a 16-bit binary number; the remaining six octets encode the globally unique bridge address, in exactly the same way as for a bridge identifier.

12.5.2.3 Encoding of topology change notification reasons

A reason value for a topology change notification shall be encoded as a single octet, according to Table 12-1.

Table 12-1—Encoding of reasons for topology change notifications

Reason value	Octet encoding
Port state became Forwarding	0000 1000
Port state became Blocking, condition unspecified	0000 0111
Port state became Blocking, spanning tree	0000 0010
Port state became Blocking, isolated remote bridge	0000 0011
Port state became Blocking, partitioned cluster	0000 0100
Port state became Blocking, management reset	0000 0101

12.5.3 Formats and parameters for BPDUs of the Extended Spanning Tree Protocol

12.5.3.1 Format and parameters of Configuration BPDUs

The format of the Configuration BPDU is shown in Figure 12-1. Each transmitted Configuration BPDU shall contain the following parameters (7.5.1) and no others:

a) The protocol identifier is encoded in octets 1 and 2 of the BPDU as specified in ISO/IEC 10038: 1993.

b) The protocol version identifier is encoded in octet 3 of the BPDU. It takes the value 0000 0001, which identifies the protocol version as the Extended Spanning Tree Protocol.

c) The BPDU type is encoded in octet 4 of the BPDU as specified in ISO/IEC 10038: 1993.

d) The Topology Change Acknowledgment flag, Topology Change flag, Root Identifier, Root Path Cost, Bridge Identifier, Port Identifier, Message Age, Max Age, Hello Time, and Forward Delay parameters (7.5.1.1 through 7.5.1.8) are encoded in octets 4 through 35 of the BPDU as specified in ISO/IEC 10038: 1993.

e) The version 1 parameter length is encoded in octet 36 of the BPDU. It takes the value 18 or 10, depending upon whether or not the Old Cluster Identifier parameter is present.

f) The New Cluster Identifier (7.5.1.9) is encoded in octets 37 through 44 of the BPDU.

g) The Reclustering Delay (7.5.1.11) is encoded in octets 45 and 46 of the BPDU.

h) The Old Cluster Identifier (7.5.1.10), when present, is encoded in octets 47 through 54 of the BPDU. Absence of this parameter from the BPDU (version 1 parameter length equal to 10) indicates that its value is equal to the value of the New Cluster Identifier parameter.

	Octet
Protocol identifier	1 2
Protocol version identifier	3
BPDU type	4
Flags	5
Root Identifier	6 | | 13
Root Path Cost	14 | 17
Bridge Identifier	18 | | 25
Port Identifier	26 27
Message Age	28 29
Max Age	30 31
Hello Time	32 33
Forward Delay	34 35
Version 1 parameter length	36
New Cluster Identifier	37 | | 44
Reclustering Delay	45 46
Old Cluster Identifier *	47 | | 54

* Optional, when value equal to that of the New Cluster Identifier parameter.

Figure 12-1—Configuration BPDU parameters and format

12.5.3.2 Format and parameters of Topology Change Notification BPDUs

The format of the Topology Change Notification BPDU is shown in Figure 12-2. Each transmitted Topology Change Notification BPDU shall contain the following parameters (7.5.2) and no others:

a) The protocol identifier is encoded in octets 1 and 2 of the BPDU as specified in ISO/IEC 10038: 1993.

b) The protocol version identifier is encoded in octet 3 of the BPDU. It takes the value 0000 0001, which identifies the protocol version as the Extended Spanning Tree Protocol.

c) The BPDU type is encoded in octet 4 of the BPDU as specified in ISO/IEC 10038: 1993.

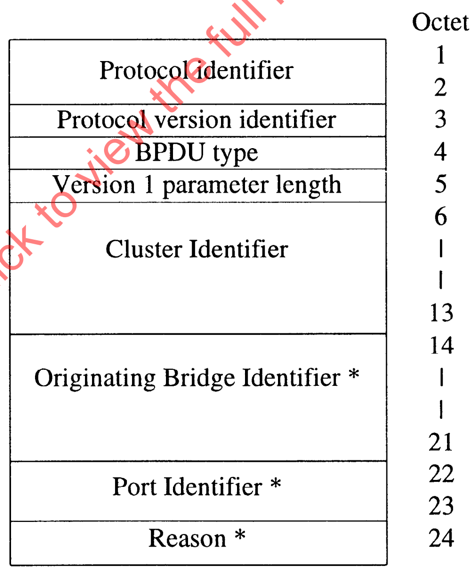
d) The version 1 parameter length is encoded in octet 5 of the BPDU. It takes the value 19 if the parameters f) through h) are present, and the value 8 otherwise.

e) The Cluster Identifier (7.5.2.1) is encoded in octets 6 through 13 of the BPDU.

f) The Originating Bridge Identifier (12.3.2.2), when present, is encoded in octets 14 through 21 of the BPDU. Absence of this parameter from the BPDU (version 1 parameter length equal to 8) indicates that its value is equal to the null value.

g) The Port Identifier (12.3.2.3), when present, is encoded in octets 22 and 23 of the BPDU. The value of this parameter is not significant if the Originating Bridge Identifier is present but has the null value.

h) The Reason (12.3.2.4), when present, is encoded in octet 24 of the BPDU. This parameter shall be present if the Originating Bridge Identifier is present; its value is not significant if the Originating Bridge Identifier is present but has the null value.



* Optional, as a set—either all three parameters are present, or none is present.

Figure 12-2—Topology Change Notification BPDU parameters and format

12.5.4 Validation of received BPDUs

A bridge protocol entity that operates the Extended Spanning Tree Protocol shall process a BPDU received on a virtual port as specified in 12.7 if and only if the BPDU contains at least four octets and the protocol identifier has the value specified for BPDUs, and

a) The protocol version identifier has the value 0000 0001, the BPDU type denotes a Configuration BPDU, and the BPDU contains either

— at least 46 octets with the version 1 parameter length value equal to 10, or
— at least 54 octets with the version 1 parameter length value equal to 18; or

b) The protocol version identifier has the value 0000 0001, the BPDU type denotes a Topology Change Notification BPDU, and the BPDU contains

— at least 13 octets with the version 1 parameter length value equal to 8, or
— at least 24 octets with the version 1 parameter length value equal to 19; or

c) The protocol version identifier has a value other than 0000 0000 or 0000 0001, the BPDU type denotes a Configuration BPDU, and the BPDU contains either

— at least 46 octets with the version 1 parameter length value equal to 10, or
— at least 54 octets with the version 1 parameter length value equal to 18; or

d) The protocol version identifier has a value other than 0000 0000 or 0000 0001, the BPDU type denotes a Topology Change Notification BPDU, and the BPDU contains

— at least 13 octets with the version 1 parameter length value equal to 8, or
— at least 24 octets with the version 1 parameter length value equal to 19; or

e) The protocol version identifier has any octet value other than 0000 0001, the BPDU type denotes a Configuration BPDU, the receiving port is a virtual LAN port, and the BPDU contains at least 35 octets; or

f) The protocol version identifier has any octet value other than 0000 0001, the BPDU type denotes a Topology Change Notification BPDU, and the receiving port is a virtual LAN port.

In cases a) through d), any octets that are present beyond the last parameter indicated by the version 1 parameter length value are ignored, as far as processing according to this International Standard is concerned. In cases e) and f), any octets that are present beyond octet 35 or octet 4, respectively, are similarly ignored.

In cases a) through d) when the receiving virtual port is a virtual LAN port, the values in the Version 1 parameter fields are ignored, and are not checked for consistency with the default information.

12.6 Elements of procedure

The elements of procedure specified in this subclause apply to a bridge protocol entity's operation both of the Spanning Tree Protocol through LAN ports and of the Extended Spanning Tree Protocol through virtual ports. These elements of procedure do not apply to any port with port state set to Disabled.

12.6.1 Transmit Configuration BPDU

12.6.1.1 Purpose

To convey knowledge of the designated root, root path cost, designated bridge, designated port, and the values of protocol timers to other bridge ports attached to the same LAN or subgroup as the port on which the Configuration BPDU is transmitted; also, for virtual ports, to convey knowledge of the transmitting bridge's cluster membership.

12.6.1.2 Use

As specified in ISO/IEC 10038: 1993.

12.6.1.3 Procedure

a) For a LAN port, or if the Hold Timer for a virtual port is active, the procedure is as specified in ISO/IEC 10038: 1993.

b) Otherwise, for a virtual port for which the Hold Timer is not active

  1) A Configuration BPDU shall be transmitted through the virtual port within a time of maximum BPDU transmission delay (as specified in ISO/IEC 10038: 1993) after any invocation of this procedure.

  2) The Configuration Pending and Topology Change Acknowledge flag parameters for the virtual port are reset.

  3) The Hold Timer for the virtual port is started.

c) The Configuration BPDU transmitted through a virtual port shall have parameters set as follows:

  1) The Configuration BPDU's Root Identifier, Root Path Cost, Bridge Identifier, Port Identifier, Message Age, Max Age, Hello Time, Forward Delay, Topology Change Acknowledgment, and Topology Change parameters shall be set as specified in ISO/IEC 10038: 1993.

  2) If the Configuration BPDU is to be transmitted through a virtual LAN port, the transmitted BPDU may be a version 0 BPDU (see 12.5.1.3). Alternatively, it may be be a version 1 BPDU with parameters as in 3) through 5) below, using default values for the group parameters as follows:

    i) The value for each cluster identifier parameter is constructed from the bridge's bridge address and a cluster index of zero.
    ii) The Reclustering Delay value is zero.

  3) The Configuration BPDU's New Cluster Identifier parameter shall be set to the value of the New Cluster Identifier parameter for the group to which the virtual port attaches.

  4) The Configuration BPDU's Old Cluster Identifier parameter shall be set to

    i) The value of the Old Cluster Identifier parameter for the group to which the virtual port attaches, if the bridge is the primary bridge for the cluster, and otherwise to
    ii) The value of the Peer Old Cluster Identifier parameter for the root port.

  5) The Configuration BPDU's Reclustering Delay parameter shall be set to the value of the Reclustering Delay parameter for the group to which the virtual port attaches.

12.6.2 Record configuration information

12.6.2.1 Purpose

As specified in ISO/IEC 10038: 1993.

12.6.2.2 Use

As specified in ISO/IEC 10038: 1993.

12.6.2.3 Procedure

As specified in ISO/IEC 10038: 1993; in addition, for a subgroup port, the Peer New Cluster Identifier and Peer Old Cluster Identifier parameters are set to the values, respectively, of the New Cluster Identifier and Old Cluster Identifier parameters conveyed in the received Configuration BPDU.

12.6.3 Record configuration timeout values

12.6.3.1 Purpose

As specified in ISO/IEC 10038: 1993.

12.6.3.2 Use

As specified in ISO/IEC 10038: 1993.

12.6.3.3 Procedure

As specified in ISO/IEC 10038: 1993; in addition, for a subgroup port, the Reclustering Delay parameter for the group to which the port attaches is set to the value of the Reclustering Delay parameter conveyed in the received Configuration BPDU.

12.6.4 Configuration BPDU generation

12.6.4.1 Purpose

To convey, to bridges attached to each LAN and subgroup for which the bridge is the designated bridge, knowledge of the designated root, root path cost, designated bridge, and designated port, and the values of protocol timers. In addition to convey, to remote bridges attached to each subgroup in which the bridge is a designated bridge, knowledge of the bridge’s cluster membership within the relevant group.

12.6.4.2 Use

As specified in ISO/IEC 10038: 1993.

12.6.4.3 Procedure

As specified in ISO/IEC 10038: 1993, extended for designated virtual ports.

12.6.5 Reply to Configuration BPDU

As specified in ISO/IEC 10038: 1993, extended for designated virtual ports.

12.6.6 Transmit Topology Change Notification BPDU

12.6.6.1 Purpose

As specified in ISO/IEC 10038: 1993.

12.6.6.2 Use

As specified in ISO/IEC 10038: 1993.

12.6.6.3 Procedure

A Topology Change Notification BPDU shall be transmitted via the root port within a time of maximum BPDU transmission delay. (See 4.10.2 of ISO/IEC 10038: 1993.)

a) If the root port is a LAN port, the transmitted BPDU has no parameters (7.5.2).

b) If the root port is a virtual LAN port, the BPDU may be a version 0 BPDU with no parameters (12.5.1.3), or it may be a version 1 BPDU with parameters as in c) and d).

c) If the root port is a subgroup port, the Cluster Identifier parameter of the transmitted BPDU takes the value of the Current Cluster Identifier for the group to which the port attaches. If the root port is a virtual LAN port, the Cluster Identifier parameter takes the default value constructed from the bridge address component of the port's Designated Bridge parameter and a cluster index value of zero.

d) The Originating Bridge Identifier, Port Identifier, and Reason parameters may be omitted, or equivalently set to null values, or they may be set to values indicating the cause of the topology change notification.

12.6.7 Configuration update

12.6.7.1 Purpose

As specified in ISO/IEC 10038: 1993.

12.6.7.2 Use

As specified in ISO/IEC 10038: 1993, extended for designated virtual ports.

12.6.7.3 Procedure

As specified in ISO/IEC 10038: 1993; in addition, the procedure for cluster selection (12.6.17) shall be used to determine the cluster membership for each non-virtual-LAN group.

12.6.8 Root selection

As specified in ISO/IEC 10038: 1993, extended for designated virtual ports, and with the reference to "the bridge identifier recorded as the designated bridge for the LAN to which it is attached" interpreted for a virtual port as "the bridge identifier recorded as the designated bridge by the virtual port."

12.6.9 Designated port selection

As specified in ISO/IEC 10038: 1993, extended for designated virtual ports.

12.6.10 Become designated port

As specified in ISO/IEC 10038: 1993, extended for designated virtual ports.

12.6.11 Port state selection

12.6.11.1 Purpose

To select the relaying state of each of the remote bridge's ports, based upon updated configuration information that indicates, for each port, its role in the activc topology of the RB-LAN (see 7.3.1.3, 7.3.2.2, 7.3.2.5); also to ensure that the Configuration Pending and Topology Change Acknowledge flags are cleared, and the Message Age Timers and Forward Delay Timers are stopped, when necessary.

NOTE—This procedure does not alter the relaying state for any subgroup ports attaching to a group that is undergoing a reclustering delay when the procedure is invoked. Port states are selected at the end of the reclustering delay: see 12.6.19 and 12.6.20.

12.6.11.2 Use

As specified in ISO/IEC 10038: 1993.

12.6.11.3 Procedure

For each of the bridge's LAN ports and virtual LAN ports, and for each of its subgroup ports that attaches to a group for which the Reclustering State parameter is set to Stable or Overlap:

a) If the port is a LAN port or a virtual LAN port, or a subgroup port that is a root port or designated port, the procedure is as specified in ISO/IEC 10038: 1993 extended for designated virtual ports.

b) Otherwise (when the port is a subgroup port that has been selected as an alternate port)

 1) The Configuration Pending flag parameter and the Topology Change Acknowledge flag parameter for the port are reset.

 2) If

  i) Each other virtual port attaching to the same group has also been selected as an alternate port, or

  ii) The port is an isolated alternate subgroup port as determined by 7.6.8,

 then the make blocking procedure (12.6.13) is used for the port, and otherwise—when neither i) nor ii) holds—the make forwarding procedure (12.6.12) is used for the port.

12.6.12 Make forwarding

As specified in ISO/IEC 10038: 1993.

12.6.13 Make blocking

As specified in ISO/IEC 10038: 1993.

12.6.14 Topology change detection

12.6.14.1 Purpose

As specified in ISO/IEC 10038: 1993.

12.6.14.2 Use

a) On receipt of a Topology Change Notification BPDU on a port that is the designated port for the LAN or subgroup to which it is attached.

b) When a bridge port is put into the Forwarding state following expiry of the Forward Delay Timer for the port, provided that the remote bridge has at least one port that is the designated port for a LAN or subgroup.

c) When a bridge port in either the Forwarding or Learning state is put into the Blocking state.

d) When the bridge becomes the root.

12.6.14.3 Procedure

As specified in ISO/IEC 10038: 1993.

12.6.15 Topology change acknowledged

12.6.15.1 Purpose

As specified in ISO/IEC 10038: 1993.

12.6.15.2 Use

Following receipt on the root port of a Configuration BPDU with the Topology Change Acknowledgment flag parameter set, from the bridge recorded as the designated bridge for the root port.

12.6.15.3 Procedure

As specified in ISO/IEC 10038: 1993.

12.6.16 Acknowledge topology change

As specified in ISO/IEC 10038: 1993, extended for designated virtual ports.

12.6.17 Cluster selection

12.6.17.1 Purpose

To determine the most up-to-date cluster membership information for each non-virtual-LAN group to which the remote bridge attaches, including creation of a new cluster identifier when the remote bridge first becomes the primary bridge for a cluster.

12.6.17.2 Use

As part of the configuration update procedure.

12.6.17.3 Procedure

The cluster information for each non-virtual-LAN group is set as specified in 7.6.2.4. If those actions include setting the Reclustering State parameter for any group to Reclustering [7.6.2.4.2 a) 1)] the Reclustering Delay Timer for that group is started.

12.6.18 Cluster resolution

12.6.18.1 Purpose

To determine the cluster, if any, to which a remote bridge belongs following a change in the observed cluster information for a non-virtual-LAN group.

12.6.18.2 Use

On expiry of the Reclustering Delay Timer for a group for which the Reclustering State is Reclustering.

12.6.18.3 Procedure

a) If the bridge is newly selected as the primary bridge for the cluster [7.6.5 d)], the Reclustering Delay parameter for the group is set as specified in 7.6.5 d).

b) The Reclustering State and cluster identifier parameters for the group are set as specified in 7.6.5 a) through d).

c) If the Reclustering State is set to Overlap [7.6.5 d)], the Reclustering Delay Timer for the group is started; otherwise, the Reclustering Delay Timer remains stopped.

d) The procedure for cluster port state selection is then performed (12.6.19).

12.6.19 Cluster port state selection

12.6.19.1 Purpose

To select the relaying state of each subgroup port attaching to a group that has completed a reclustering operation. Also to ensure that the Configuration Pending and Topology Change Acknowledge flags are cleared and Message Age Timers and Forward Delay Timers are stopped when necessary.

12.6.19.2 Use

After the cluster information has been updated by the cluster resolution procedure.

12.6.19.3 Procedure

The port flags and Message Age Timers of the virtual ports attaching to the group are adjusted as in a). The port state parameters of the virtual ports attaching to the group are set as in b), c), or d), according to the actions selected in cluster resolution, 7.6.5 a) through d).

a) For each virtual port attaching to the group:

   1) If the port has been selected as the root port or an alternate port, the Configuration Pending and Topology Change Acknowledge flag parameters for the port are reset, to False.

   2) Otherwise (when the port has been selected as a designated port), the port's Message Age Timer is stopped, if it is running.

b) If the remote bridge attaches to no cluster in the group [7.6.5 a)], i.e., if none of its virtual ports attaching to the group is the root port or a designated port, the make blocking procedure (12.6.13) is used for each of the virtual ports attaching to the group.

c) If the remote bridge attaches to the same cluster as it did before the reclustering [7.6.5 b), 7.6.5 d)], the port state of each virtual port attaching to the group is set as follows:

   1) If the virtual port has been selected as an alternate port and it is an isolated alternate subgroup port as determined by 7.6.8, the make blocking procedure (12.6.13) is used for the port.

   2) Otherwise, the make forwarding procedure (12.6.12) is used for the port.

d) If the remote bridge attaches to a different cluster from that, if any, to which it attached before the reclustering [7.6.5 c)], the port state of each virtual port attaching to the group is set as follows:

   1) If the virtual port has been selected as an alternate port that is an isolated alternate subgroup port as determined by 7.6.8, the make blocking procedure (12.6.13) is used for the port.

   2) Otherwise, the port state is set to Listening and the Forward Delay Timer is restarted if it is running, or started if it is not running.

12.6.20 Cluster confirmation

12.6.20.1 Purpose

To establish new steady-state cluster information for a non-virtual-LAN group, following a reclustering that leaves the bridge in the same cluster as that to which it previously belonged but with a different cluster identifier applicable.

12.6.20.2 Use

On expiry of the Reclustering Delay Timer for a group for which the Reclustering State is Overlap.

12.6.20.3 Procedure

The Reclustering State parameter for the group is set to Stable, and the Old Cluster Identifier parameter for the group is set equal to the value of the Current Cluster Identifier parameter. For each virtual port that has been selected as an alternate port and is an isolated alternate subgroup port as determined by 7.6.8, the make blocking procedure is performed (12.6.13).

12.7 Operation of the protocol

A bridge protocol entity in a remote bridge shall

a) Communicate with its peer entities in other bridges (local and remote) by the transmission of BPDUs,

b) Update stored protocol variables and timers, and

c) Change the state of the LAN ports and virtual ports.

following the reception of BPDUs and the expiry of bridge and port Timers, as required by the specification in 12.7.1 through 12.7.9. In any case of ambiguity, reference shall be made to the Procedural Model (Clause 13) which is the definitive description of the operation of the protocol by a bridge protocol entity. None of the procedures specified apply to ports with port state set to Disabled.

12.7.1 Received Configuration BPDU

a) If a Configuration BPDU is received that conveys protocol information superseding that already held for the port on which the BPDU was received (i.e., if the received message priority is higher than or equal to the port's designated priority as defined in 7.3.4.2), the sequence of procedures is as specified in ISO/IEC 10038: 1993.

b) If a Configuration BPDU is received that does not convey protocol information superseding that held for the port on which it was received, and if the receiving port is the designated port for the LAN or subgroup to which it is attached, then the reply to Configuration BPDU procedure (12.6.5) is used.

12.7.2 Received Topology Change Notification BPDU

a) When a Topology Change Notification BPDU is received on a LAN port or virtual LAN port, the procedure is as specified in ISO/IEC 10038: 1993, extended for designated virtual ports.

b) When a Topology Change Notification BPDU is received on a subgroup port and the value of the received Cluster Identifier parameter is equal to the value of the Current Cluster Identifier parameter for the group to which the subgroup port attaches, then the procedure is as specified in ISO/IEC 10038: 1993, extended for designated virtual ports.

12.7.3 Hello Timer expiry

The procedure is as specified in ISO/IEC 10038: 1993.

12.7.4 Message Age Timer expiry

The procedure is as specified in ISO/IEC 10038: 1993.

12.7.5 Forward Delay Timer expiry

a) If the state of the port for which the Forward Delay Timer has expired was Listening, the procedure is as specified for this case in ISO/IEC 10038: 1993.

b) Otherwise, if the state of the port for which the Forward Delay Timer has expired was Learning, then

1) Thc port state is set to Forwarding.

2) If the bridge has at least one port that is the designated port for a LAN or subgroup, the topology change detection procedure (12.6.14) is invoked.

12.7.6 Topology Change Notification Timer expiry

The procedure is as specified in ISO/IEC 10038: 1993.

12.7.7 Topology Change Timer expiry

The procedure is as specified in ISO/IEC 10038: 1993.

12.7.8 Hold Timer expiry

The procedure is as specified in ISO/IEC 10038: 1993.

12.7.9 Reclustering Delay Timer expiry

If the value of the Reclustering State parameter for the group whose Reclustering Delay Timer has expired is Reclustering, the cluster resolution procedure (12.6.18) is used for the group, followed by the cluster port state selection procedure (12.6.19); if the value of the Reclustering State parameter is Overlap, the cluster confirmation procedure (12.6.20) is used for the group.

12.8 Management of the bridge protocol entity

12.8.1 Initialization

a) The Designated Root parameter held by the bridge is set equal to the value of the bridge identifier, and the values of the Root Path Cost and Root Port parameters held by the bridge are set to zero.

b) The Max Age, Hello Time, and Forward Delay parameters held by the bridge are set to the values of the Bridge Max Age, Bridge Hello Time, and Bridge Forward Delay parameters, respectively.

c) The Topology Change Detected and Topology Change flag parameters for the bridge are reset, and the Topology Change Notification Timer and Topology Change Timer are stopped, if running.

d) For each of the bridge's ports

   1) The Become designated port procedure (12.6.10) is invoked to assign values to the Designated Root, Designated Cost, Designated Bridge, and Designated Port parameters for the port.
   2) The port state is set to Blocking if the port is to be enabled following initialization, and otherwise is set to Disabled.
   3) The Topology Change Acknowledge and Configuration Pending flag parameters are reset.
   4) Each of the Message Timer, Forward Delay Timer, and Hold Timer is stopped, if running.
   5) For a subgroup port, the Peer New Cluster Identifier and Peer Old Cluster Identifier parameters are each set to the null cluster identifier value.

e) For each group to which the remote bridge attaches, other than virtual LANs

   1) The Reclustering State parameter is set to Stable.
   2) The Reclustering Delay Timer is stopped, if running.
   3) The Reclustering Delay parameter is set to the value of the Primary Reclustering Delay parameter, adjusted if necessary to comply with condition (2) specified in 7.10.5.
   4) i) If all the virtual ports attaching to the group have port state set to Disabled, the Current Cluster Identifier, New Cluster Identifier, and Old Cluster Identifier parameters for the group are all set to the null cluster identifier value; otherwise,
      ii) The Current Cluster Identifier, New Cluster Identifier, and Old Cluster Identifier parameters for the group are all set to the same new cluster identifier value, determined as in 7.6.2.4.1 b) 3).

f) The port state selection procedure (12.6.11) is invoked to select the state of each of the bridge's ports.

g) The Configuration BPDU generation procedure (12.6.4) is invoked and the Hello Timer (12.3.4.1) is started.

12.8.2 Enable port

For a subgroup port, the Peer New Cluster Identifier and Peer Old Cluster Identifier parameters are each set to the null cluster identifier value, and the cluster selection procedure (12.6.17) is then invoked.

For any bridge port, the remainder of this operation is then as specified in ISO/IEC 10038: 1993.

12.8.3 Disable port

This operation is as specified in ISO/IEC 10038: 1993.

12.8.4 Set bridge priority

This operation is as specified in ISO/IEC 10038: 1993, extended for designated virtual ports.

12.8.5 Set port priority

This operation is as specified in ISO/IEC 10038: 1993, extended for designated virtual ports.

12.8.6 Set path cost

This operation is as specified in ISO/IEC 10038: 1993.

13. The Extended Spanning Tree Protocol: Procedural model

```
/***************************************************************************
*
*  SPANNING TREE ALGORITHM AND PROTOCOL: EXTENDED SPANNING TREE PROTOCOL
*
*  The basis for the procedural model is as specified in 4.9 of ISO/IEC 10038:
*  1993. The modifications required are indicated below by the use of
*  "#ifdef ESTP". With three exceptions, where the original ISO/IEC 10038: 1993
*  code appears following a "#else", the modifications consist simply of
*  additional code or declarations.
*
*  Subclause references without parentheses are to this International Standard.
*  Subclause references within parentheses are the original references to
*  ISO/IEC 10038: 1993; these all appear outside the new code introduced by the
*  "#ifdef ESTP" directives; in the context of this International Standard they
*  are to be interpreted according to the correspondence documented in Annex D
*  of this International Standard.
*
***************************************************************************/

#define ESTP


/***************************************************************************
*  DEFINED CONSTANTS
***************************************************************************/

#define Zero        0
#define One         1

#define False       0
#define True        1

/** port states. **/

#define Disabled       0                                  /* 7.4.5  (4.4.5) */
#define Listening      1                                  /* 7.4.2  (4.4.2) */
#define Learning       2                                  /* 7.4.3  (4.4.3) */
#define Forwarding     3                                  /* 7.4.4  (4.4.4) */
#define Blocking       4                                  /* 7.4.1  (4.4.1) */
```

```
#ifdef ESTP
/****  ESTP : kinds of port.  **/

#define LAN_port       0
#define VLAN_port      1
#define Ind_virt_port  2
#define Mp_virt_port   3

/****  ESTP : group reclustering states,
              and constant for set_reclustering_state.  **/

#define Stable         1
#define Reclustering   2
#define Overlap        3

#define One_Second ((Time) 1.0)

#endif

/** BPDU type constants **/

#define Config_bpdu_type     0
#define Tcn_bpdu_type      128

/** pseudo-implementation constants. **/

#define No_of_ports 2
        /* arbitrary choice, to allow the code below to compile */

#define All_ports No_of_ports+1
        /* ports start at 1, arrays in C start at 0 */

#define Default_path_cost 10
        /* arbitrary */

#define Message_age_increment 1
        /* minimum increment possible to avoid underestimating age, allows
           for BPDU transmission time */

#define No_port 0
        /* reserved value for Bridge's root port parameter indicating no
           root port, used when Bridge is the root */

#ifdef ESTP
/****  ESTP : pseudo-implementation constants for groups  **/

#define No_of_groups 2
        /* arbitrary choice, to allow the code below to compile */

#define All_groups No_of_groups+1
        /* groups start at 1, arrays in C start at 0 */

#endif
```

```
/******************************************************************************
*  TYPEDEFS, STRUCTURES AND UNION DECLARATIONS
******************************************************************************/

/** basic types. **/

typedef int Int;        /* to align with case stropping convention used
                           here. Types and defined constants have their
                           initial letters capitalized. */

typedef Int Boolean;    /* : (True, False) */

typedef Int State;      /* : (Disabled, Listening, Learning,
                              Forwarding, Blocking) */

#ifdef ESTP
/****  ESTP : kinds of ports, reclustering states  **/

typedef Int Kind;       /* : (LAN_port, VLAN_port,
                              Ind_virt_port, Mp_virt_port) */

typedef Int Reclustering_state;  /* (Stable, Reclustering, Overlap) */

#endif


/** BPDU encoding types defined in 12.4 and 12.5.2, including references
         from 12.5.2.1 to ISO/IEC 10038: 1993 Section 5, are:

Protocol_version      (5.2.2)

Bpdu_type             (5.2.3)

Flag                  (5.2.4)

Identifier            (5.2.5)   (****  ESTP : 12.5.2.2)

Cost                  (5.2.6)

Port_id               (5.2.7)

Time                  (5.2.8)

Reason                (****  ESTP : 12.5.2.3)

**/

#include "types.c"      /* defines BPDU encoding types */
```

```
/** Configuration BPDU Parameters, 7.5.1   **/

typedef struct
{
   Bpdu_type   type;

   Identifier root_id;                                   /* 7.5.1.1         */

   Cost       root_path_cost;                            /* 7.5.1.2         */

   Identifier bridge_id;                                 /* 7.5.1.3         */

   Port_id    port_id;                                   /* 7.5.1.4         */

   Time       message_age;                               /* 7.5.1.7         */

   Time       max_age;                                   /* 7.5.1.8         */

   Time       hello_time;                                /* 7.5.1.8         */

   Time       forward_delay;                             /* 7.5.1.8         */

   Flag       topology_change_acknowledgment;            /* 7.5.1.5         */

   Flag       topology_change;                           /* 7.5.1.6         */

#ifdef ESTP
/****   ESTP : additional parameters   **/

   Flag       version_1;                                 /* 12.5.1.2.1      */

   Identifier new_cluster_id;                            /* 7.5.1.9         */

   Identifier old_cluster_id;                            /* 7.5.1.10        */

   Time       recluster_delay;                           /* 7.5.1.11        */

#endif

} Config_bpdu;


/** Topology Change Notification BPDU Parameters, 7.5.2   **/

typedef struct
{
   Bpdu_type   type;

#ifdef ESTP
/****   ESTP : additional parameters   **/

   Flag       version_1;                                 /* 12.5.1.2.1      */

   Identifier cluster_id;                                /* 7.5.2.1         */

   Identifier originating_bridge;                        /* 12.3.2.2        */

   Int        originating_port;                          /* 12.3.2.3        */

   Reason     tcn_reason;                                /* 12.3.2.4        */

#endif

} Tcn_bpdu;
```

```
/** Bridge Parameters, 7.5.3  **/

typedef struct
{
   Identifier designated_root;                     /* 7.5.3.1        */

   Cost       root_path_cost;                      /* 7.5.3.2        */

   Int        root_port;                           /* 7.5.3.3        */

   Time       max_age;                             /* 7.5.3.4        */

   Time       hello_time;                          /* 7.5.3.5        */

   Time       forward_delay;                       /* 7.5.3.6        */

   Identifier bridge_id;                           /* 7.5.3.7        */

   Time       bridge_max_age;                      /* 7.5.3.8        */

   Time       bridge_hello_time;                   /* 7.5.3.9        */

   Time       bridge_forward_delay;                /* 7.5.3.10       */

   Boolean    topology_change_detected;            /* 7.5.3.11       */

   Boolean    topology_change;                     /* 7.5.3.12       */

   Time       topology_change_time;                /* 7.5.3.13       */

   Time       hold_time;                           /* 7.5.3.14       */

} Bridge_data;
```

```
/** Port Parameters, 7.5.4  **/

typedef struct
{
   Port_id    port_id;                                  /* 7.5.4.1.1      */

   State      state;                                    /* 7.5.4.1.2      */

   Int        path_cost;                                /* 7.5.4.1.3      */

   Identifier designated_root;                          /* 7.5.4.1.4      */

   Int        designated_cost;                          /* 7.5.4.1.5      */

   Identifier designated_bridge;                        /* 7.5.4.1.6      */

   Port_id    designated_port;                          /* 7.5.4.1.7      */

   Boolean    topology_change_acknowledge;              /* 7.5.4.1.8      */

   Boolean    config_pending;                           /* 12.3.5.1       */

#ifdef ESTP
/****   ESTP : additional parameters   **/

   Identifier peer_new_cluster_id;                      /* 7.5.4.2.1      */

   Identifier peer_old_cluster_id;                      /* 7.5.4.2.2      */

#endif

} Port_data;

#ifdef ESTP
/****   ESTP :   Group parameters, 7.5.5   **/

typedef struct
{
   Reclustering_state     reclustering_state;          /* 7.5.5.1        */

   Identifier current_cluster_id;                       /* 7.5.5.2        */

   Identifier new_cluster_id;                           /* 7.5.5.3        */

   Identifier old_cluster_id;                           /* 7.5.5.4        */

   Time       reclustering_delay;                       /* 7.5.5.5        */

   Time       primary_reclustering_delay;               /* 7.5.5.6        */

} Group_data;

#endif
```

```
/** types to support timers for this pseudo-implementation. **/

typedef struct
{
   Boolean    active;          /* timer in use. */

   Time       value;           /* current value of timer, counting up. */

} Timer;


/*******************************************************************************
 *  STATIC STORAGE ALLOCATION
 ******************************************************************************/

Bridge_data   bridge_info;                               /* 7.5.3          */

Port_data     port_info[All_ports];                      /* 7.5.4          */

Config_bpdu   config_bpdu[All_ports];

Tcn_bpdu      tcn_bpdu[All_ports];

Timer         hello_timer;                               /* 12.3.4.1       */

Timer         tcn_timer;                                 /* 12.3.4.2       */

Timer         topology_change_timer;                     /* 12.3.4.3       */

Timer         message_age_timer[All_ports];              /* 12.3.6.1       */

Timer         forward_delay_timer[All_ports];            /* 12.3.6.2       */

Timer         hold_timer[All_ports];                     /* 12.3.6.3       */

#ifdef ESTP
/****   ESTP : static storage for group-related information  **/

Group_data    group_info[All_groups];                    /* 7.5.5          */

Timer         reclustering_delay_timer[All_groups];      /* 12.3.8         */

/** group-related information to support the pseudo-implementation  **/

Identifier    null_cluster_id;                           /* 7.3.4.3        */
              /* needs to be initalized to the correct all-zero value. */

Int           group_number[All_ports];                   /* 9.4.3          */
              /* for each port, zero means LAN or VLAN port, so no group info;
                 non-zero means subgroup port, and is index in Group_data.  */

Kind          port_kind[All_ports];           /* LAN_port/VLAN_port, etc. */
#endif
```

```
/******************************************************************************
*  CODE
******************************************************************************/

/** Elements of Procedure, 12.6 (4.6) **/

transmit_config(port_no)                                    /* 12.6.1 (4.6.1) */
Int      port_no;
{
   if (hold_timer[port_no].active)                          /* 12.6.1.3 a)     */
   {
      port_info[port_no].config_pending = True;             /* 12.6.1.3 a)     */
   }
   else                                                     /* 12.6.1.3 c) 1) */
   {                                                        /* (4.6.1.3.2)     */
      config_bpdu[port_no].type = Config_bpdu_type;

      config_bpdu[port_no].root_id = bridge_info.designated_root;
                                                            /* (4.6.1.3.2(1)) */
      config_bpdu[port_no].root_path_cost = bridge_info.root_path_cost;
                                                            /* (4.6.1.3.2(2)) */
      config_bpdu[port_no].bridge_id = bridge_info.bridge_id;
                                                            /* (4.6.1.3.2(3)) */
      config_bpdu[port_no].port_id = port_info[port_no].port_id;
                                                            /* (4.6.1.3.2(4)) */
      if (root_bridge())
      {
         config_bpdu[port_no].message_age = Zero;           /* (4.6.1.3.2(5)) */
      }
      else
      {
         config_bpdu[port_no].message_age
            =  message_age_timer[bridge_info.root_port].value
              + Message_age_increment;                      /* (4.6.1.3.2(6)) */
      }

      config_bpdu[port_no].max_age = bridge_info.max_age;   /* (4.6.1.3.2(7)) */
      config_bpdu[port_no].hello_time = bridge_info.hello_time;
      config_bpdu[port_no].forward_delay = bridge_info.forward_delay;

      config_bpdu[port_no].topology_change_acknowledgment
        =  port_info[port_no].topology_change_acknowledge;
                                                            /* (4.6.1.3.2(8)) */
      port_info[port_no].topology_change_acknowledge = False;
                                                            /* 12.6.1.3 b) 2) */
                                                            /* (4.6.1.3.2(8)) */
      config_bpdu[port_no].topology_change
        = bridge_info.topology_change;                      /* (4.6.1.3.2(9)) */
```

```
#ifdef ESTP
/****  ESTP : additional BPDU parameters  **/

      if (group_number[port_no] != Zero)
      {
         Group_data *group = & group_info[group_number[port_no]];

         config_bpdu[port_no].new_cluster_id = group->new_cluster_id;
                                                           /* 12.6.1.3 c) 3) */
         config_bpdu[port_no].old_cluster_id
           = (primary_bridge(group_number[port_no])
              ? group->old_cluster_id
              : port_info[bridge_info.root_port].peer_old_cluster_id);
                                                           /* 12.6.1.3 c) 4) */
         config_bpdu[port_no].recluster_delay = group->reclustering_delay;
                                                           /* 12.6.1.3 c) 5) */
      }
#endif

      send_config_bpdu(port_no, &config_bpdu[port_no]);

      port_info[port_no].config_pending = False;           /* 12.6.1.3 b) 2) */
                                                           /* (4.6.1.3.2(10))*/
      start_hold_timer(port_no);                           /* 12.6.1.3 b) 3) */
   }                                                       /* (4.6.1.3.2(11))*/
}
        /* where

        send_config_bpdu(port_no,bpdu)
        Int         port_no;
        Config_bpdu *bpdu;

        is a pseudo-implementation specific routine which transmits
        the bpdu on the specified port within the specified time.

        */

        /* and */

        Boolean root_bridge()
        {
           return(bridge_info.designated_root == bridge_info.bridge_id);
        }

#ifdef ESTP
/****  ESTP : additional ESTP routine  **/
        /* and */

        Boolean primary_bridge(group_no)
        Int group_no;
        {
           return(group_number[bridge_info.root_port] != group_no);
        }
#endif
```

```
Boolean supersedes_port_info(port_no, config)           /* 7.3.4.2 (4.6.2.2)*/
Int port_no;
Config_bpdu *config;
{
   return (
   (  config->root_id
      <  port_info[port_no].designated_root               /* (4.6.2.2.1)     */
   )
   ||
   (  (  config->root_id
         == port_info[port_no].designated_root
      )
      &&
      (  (  config->root_path_cost
            <  port_info[port_no].designated_cost         /* (4.6.2.2.2)     */
         )
         ||
         (  (  config->root_path_cost
               == port_info[port_no].designated_cost
            )
            &&
            (  (  config->bridge_id
                  <  port_info[port_no].designated_bridge  /* (4.6.2.2.3)     */
               )
               ||
               (  (  config->bridge_id
                     == port_info[port_no].designated_bridge
                  )                                        /* (4.6.2.2.4)     */
                  &&
                  (  (  config->bridge_id != bridge_info.bridge_id
                     )                                     /* (4.6.2.2.4(1)) */
                     ||
                     (  config->port_id
                        <= port_info[port_no].designated_port
                     )                                     /* (4.6.2.2.4(2)) */
   )  )  )  )  )  )
          );
}


record_config_information(port_no, config)                 /* 12.6.2 (4.6.2) */
Int port_no;
Config_bpdu *config;
{
   port_info[port_no].designated_root = config->root_id;   /* (4.6.2.3.1)     */
   port_info[port_no].designated_cost = config->root_path_cost;
   port_info[port_no].designated_bridge = config->bridge_id;
   port_info[port_no].designated_port = config->port_id;

#ifdef ESTP
/****  ESTP : peer cluster identifiers   **/

   if (group_number[port_no] != Zero)
   {                                                       /* 12.6.2.3        */

      port_info[port_no].peer_new_cluster_id = config->new_cluster_id;
      port_info[port_no].peer_old_cluster_id = config->old_cluster_id;
   }
#endif

   start_message_age_timer(port_no, config->message_age);  /* (4.6.2.3.2)     */
}
```

```
record_config_timeout_values(config)                       /* 12.6.3 (4.6.3) */
Config_bpdu *config;
{
   bridge_info.max_age = config->max_age;                  /* (4.6.3.3)      */
   bridge_info.hello_time = config->hello_time;
   bridge_info.forward_delay = config->forward_delay;
   bridge_info.topology_change = config->topology_change;

#ifdef ESTP
/****  ESTP : if root port is a subgroup port  **/

   if (group_number[bridge_info.root_port] != Zero)
   {
      group_info[group_number[bridge_info.root_port]].reclustering_delay
       = config->recluster_delay;                          /* 12.6.3.3       */
   }
#endif

}


config_bpdu_generation()                                   /* 12.6.4 (4.6.4) */
{
   Int port_no;

   for (port_no = One; port_no <= No_of_ports; port_no++)  /* (4.6.4.3)      */
   {
      if (  designated_port(port_no)                       /* (4.6.4.3)      */
            &&
            (port_info[port_no].state != Disabled)
         )
      {
         transmit_config(port_no);                         /* (4.6.4.3)      */
      }                                                    /* (4.6.1.2)      */
   }
}

        /* where */

        Boolean designated_port(port_no)
        Int port_no;
        {
           return (   (  port_info[port_no].designated_bridge
                         == bridge_info.bridge_id
                      )
                      &&
                      (  port_info[port_no].designated_port
                          == port_info[port_no].port_id
                      )
                  );
        }


reply(port_no)                                             /* 12.6.5 (4.6.5) */
Int port_no;
{
   transmit_config(port_no);                               /* (4.6.5.3)      */
}
```

```
transmit_tcn()                                             /* 12.6.6 (4.6.6) */
{
   Int port_no;

   port_no = bridge_info.root_port;
   tcn_bpdu[port_no].type = Tcn_bpdu_type;

#ifdef ESTP
/****  ESTP : add cluster id  **/

   if (group_number[port_no] != Zero)
      tcn_bpdu[port_no].cluster_id                         /* 12.6.6.3 c)    */
        = group_info[group_number[port_no]].current_cluster_id;
#endif

   send_tcn_bpdu(port_no, &tcn_bpdu[bridge_info.root_port]);/* (4.6.6.3)     */

}

        /* where

        send_tcn_bpdu(port_no,bpdu)
        Int         port_no;
        Tcn_bpdu    *bpdu;

        is a pseudo-implementation specific routine which transmits
        the bpdu on the specified port within the specified time.

        */


configuration_update()                                     /* 12.6.7 (4.6.7) */
{
   root_selection();                                       /* (4.6.7.3.1)    */
                                                           /* (4.6.8.2)      */

   designated_port_selection();                            /* (4.6.7.3.2)    */
                                                           /* (4.6.9.2)      */
#ifdef ESTP
/****  ESTP : select cluster membership  **/

   cluster_selection();                                    /* 12.6.7.3       */
                                                           /* 12.6.17.2      */
#endif
}
```

```
root_selection()                                           /* 12.6.8 (4.6.8) */
{
   Int root_port;
   Int port_no;

   root_port = No_port;

   for (port_no = One; port_no <= No_of_ports; port_no++)     /* (4.6.8.3.1) */
   {
      if (  (  (!designated_port(port_no))
               &&
               (port_info[port_no].state != Disabled)
               &&
               (port_info[port_no].designated_root < bridge_info.bridge_id)
            )
            &&
            (  (root_port == No_port)
               ||
               (  port_info[port_no].designated_root
                  <  port_info[root_port].designated_root  /* (4.6.8.3.1(1)) */
               )
               ||
               (  (  port_info[port_no].designated_root
                     ==  port_info[root_port].designated_root
                  )
                  &&
                  (  (  (  port_info[port_no].designated_cost
                           + port_info[port_no].path_cost
                        )
                        <
                        (  port_info[root_port].designated_cost
                           + port_info[root_port].path_cost
                        )                                  /* (4.6.8.3.1(2)) */
                     )
                     ||
                     (  (  (  port_info[port_no].designated_cost
                              + port_info[port_no].path_cost
                           )
                           ==
                           (  port_info[root_port].designated_cost
                              + port_info[root_port].path_cost
                           )
                        )
                        &&
                        (  (  port_info[port_no].designated_bridge
                              <  port_info[root_port].designated_bridge
                           )                               /* (4.6.8.3.1(3)) */
                           ||
                           (  (  port_info[port_no].designated_bridge
                                 ==  port_info[root_port].designated_bridge
                              )
                              &&
                              (  (  port_info[port_no].designated_port
                                    <  port_info[root_port].designated_port
                                 )                         /* (4.6.8.3.1(4)) */
                                 ||
                                 (  (  port_info[port_no].designated_port
                                       == port_info[root_port].designated_port
                                    )
                                    &&
                                    (  port_info[port_no].port_id
                                       <  port_info[root_port].port_id
                                    )                      /* (4.6.8.3.1(5)) */
         )  )  )  )  )  )  )  )  )
      {
         root_port = port_no;
      }
   }

   bridge_info.root_port = root_port;                       /* (4.6.8.3.1)    */

   if (root_port == No_port)                                /* (4.6.8.3.2)    */
   {

      bridge_info.designated_root = bridge_info.bridge_id;
                                                            /* (4.6.8.3.2(1)) */
      bridge_info.root_path_cost = Zero;                    /* (4.6.8.3.2(2)) */
   }
   else                                                     /* (4.6.8.3.3)    */
   {
      bridge_info.designated_root = port_info[root_port].designated_root;
                                                            /* (4.6.8.3.3(1)) */
      bridge_info.root_path_cost = (  port_info[root_port].designated_cost
                                      + port_info[root_port].path_cost
                                   );                       /* (4.6.8.3.3(2)) */

   }
}


designated_port_selection()                                 /* 12.6.9 (4.6.9) */
{
   Int port_no;

   for (port_no = One; port_no <= No_of_ports; port_no++)   /* (4.6.9.3)      */
   {
      if (  designated_port(port_no)                        /* (4.6.9.3.1)    */
            ||
            (
               port_info[port_no].designated_root
               != bridge_info.designated_root               /* (4.6.9.3.2)    */
            )
            ||
            (  bridge_info.root_path_cost
               <  port_info[port_no].designated_cost
            )                                               /* (4.6.9.3.3)    */
            ||
            (  (  bridge_info.root_path_cost
                  == port_info[port_no].designated_cost
               )
               &&
               (  (  bridge_info.bridge_id
                     <  port_info[port_no].designated_bridge
                  )                                         /* (4.6.9.3.4)    */
                  ||
                  (  (  bridge_info.bridge_id
                              == port_info[port_no].designated_bridge
                     )
                     &&
                     (  port_info[port_no].port_id
                        <=  port_info[port_no].designated_port
                     )                                      /* (4.6.9.3.5)    */
         )  )  )  )
      {
         become_designated_port(port_no);                   /* (4.6.10.3.2.2) */
      }
   }
}
```

```
become_designated_port(port_no)                          /* 12.6.10(4.6.10) */
Int port_no;
{
    port_info[port_no].designated_root = bridge_info.designated_root;
                                                          /* (4.6.10.3.1)   */

    port_info[port_no].designated_cost = bridge_info.root_path_cost;
                                                          /* (4.6.10.3.2)   */

    port_info[port_no].designated_bridge = bridge_info.bridge_id;
                                                          /* (4.6.10.3.3)   */

    port_info[port_no].designated_port = port_info[port_no].port_id;
                                                          /* (4.6.10.3.4)   */
}
```

```
port_state_selection()                                   /* 12.6.11 (4.6.11) */
{
   Int port_no;

   for (port_no = One; port_no <= No_of_ports; port_no++)
   {
#ifdef ESTP
/****  ESTP : no port-state change during reclustering delay  **/

      Int group_no = group_number[port_no];
      if (  (group_no != Zero)
            &&                                              /* 12.6.11.3      */
            (group_info[group_no].reclustering_state == Reclustering)
         )
         { }
      else if (port_no == bridge_info.root_port)            /* (4.6.11.3.1)   */
#else
      if (port_no == bridge_info.root_port)                 /* (4.6.11.3.1)   */
#endif
      {
         port_info[port_no].config_pending = False;         /* (4.6.11.3.1(1))*/
         port_info[port_no].topology_change_acknowledge = False;

         make_forwarding(port_no);                          /* (4.6.11.3.1(2))*/
      }
      else if (designated_port(port_no))                    /* (4.6.11.3.2)   */
      {
         stop_message_age_timer(port_no);                   /* (4.6.11.3.2(1))*/

         make_forwarding(port_no);                          /* (4.6.11.3.2(2))*/
      }
      else                                                  /* (4.6.11.3.3)   */
      {
         port_info[port_no].config_pending = False;         /* (4.6.11.3.3(1))*/
         port_info[port_no].topology_change_acknowledge = False;

#ifdef ESTP
/****  ESTP : alternate ports   **/
         if (   (group_no == Zero)                          /* 12.6.11.3 a)   */
                ||
                isolated_in(group_no)                       /* 12.6.11.3 b)2)*/
                ||
                isolated_alternate_vp(port_no)              /* 12.6.11.3 b)2) */
            )
            make_blocking(port_no);
         else
            make_forwarding(port_no);
#else
         make_blocking(port_no);                            /* (4.6.11.3.3(2))*/
#endif
      }
   }
}
```

```
make_forwarding(port_no)                                 /* 12.6.12 (4.6.12) */
Int port_no;
{
   if (port_info[port_no].state == Blocking)               /* (4.6.12.3)     */
   {
      set_port_state(port_no, Listening);                  /* (4.6.12.3.1)   */

      start_forward_delay_timer(port_no);                  /* (4.6.12.3.2)   */
   }
}


make_blocking(port_no)                                   /* 12.6.13 (4.6.13) */
Int port_no;
{
   if (   (port_info[port_no].state != Disabled)
          &&
          (port_info[port_no].state != Blocking)
      )                                                    /* (4.6.13.3)     */
   {
      if (   (port_info[port_no].state == Forwarding)
             ||
             (port_info[port_no].state == Learning)
         )
      {
         topology_change_detection();                      /* (4.6.13.3.1)   */
                                                           /* (4.6.14.2.3)   */
      }

      set_port_state(port_no, Blocking);                   /* (4.6.13.3.2)   */

      stop_forward_delay_timer(port_no);                   /* (4.6.13.3.3)   */
   }
}

        /* where */

        set_port_state(port_no, state)
        Int port_no;
        State state;
        {
              port_info[port_no].state = state;
        }
```

```
topology_change_detection()                                  /* 12.6.14 (4.6.14) */
{
    if (root_bridge())                                         /* (4.6.14.3.1)    */
    {
        bridge_info.topology_change = True;                    /* (4.6.14.3.1(1))*/

        start_topology_change_timer();                         /* (4.6.14.3.1(2))*/
    }

    else if (bridge_info.topology_change_detected == False) /* (4.6.14.3.2)    */
    {
        transmit_tcn();                                        /* (4.6.14.3.2(1))*/

        start_tcn_timer();                                     /* (4.6.14.3.2(2))*/
    }

    bridge_info.topology_change_detected = True;               /* (4.6.14.3.3)    */
}


topology_change_acknowledged()                               /* 12.6.15 (4.6.15) */
{
    bridge_info.topology_change_detected = False;              /* (4.6.15.3.1)    */

    stop_tcn_timer();                                          /* (4.6.15.3.2)    */
}


acknowledge_topology_change(port_no)                         /* 12.6.16 (4.6.16) */
Int port_no;
{
    port_info[port_no].topology_change_acknowledge = True;   /* (4.6.16.3.1)    */

    transmit_config(port_no);                                  /* (4.6.16.3.2)    */
}
```

```
#ifdef ESTP
/****  ESTP : cluster-membership functions  **/

cluster_selection()                                  /* 12.6.17          */
{
   Int group_no;

   for (group_no = One; group_no <= No_of_groups; group_no++)
   {                                               /* 12.6.17.3, 7.6.2.4 */
      Group_data *group = & group_info[group_no];
      if (group_number[bridge_info.root_port] == group_no)
         group->new_cluster_id                       /* 7.6.2.4.1 a)     */
           = port_info[bridge_info.root_port].peer_new_cluster_id;

      else if (isolated_in(group_no))
         group->new_cluster_id = null_cluster_id;    /* 7.6.2.4.1 c)     */

      else if (bridge_addresses_match(bridge_info.bridge_id,
                                    group->current_cluster_id))
         group->new_cluster_id = group->current_cluster_id; /* 7.6.2.4.1 b)1)*/

      else if (bridge_addresses_match(bridge_info.bridge_id,
                                    group->new_cluster_id))
         ;    /* no action, new cluster id is OK as it is:  7.6.2.4.1 b)2)*/

      else                                            /* 7.6.2.4.1 b)3)*/
         group->new_cluster_id = create_cluster_id(group_no);

      if (  (group->new_cluster_id != group->current_cluster_id)
            &&
            (group->reclustering_state != Reclustering)
         )                                            /* 7.6.2.4.2     */
      {
         group->reclustering_state = Reclustering;    /* 7.6.2.4.2 a)1)*/
         if (primary_bridge(group_no))
            set_reclustering_delay(group);            /* 7.10.5 (2)    */
         start_reclustering_delay_timer(group_no);    /* 7.6.2.4.2 a)2)*/
      }
   }
}

        /* where

        Boolean bridge_addresses_match(bridge_id,cluster_id)
        Identifier  port_no;
        Identifier  cluster_id;

        is a pseudo-implementation specific routine which compares
        the bridge-address components of its arguments.

        */


        /* and

        Identifier create_cluster_id(group_no)
        Int group_no;

        is a pseudo-implementation specific routine which creates a new
        cluster identifier value for the group.

        */
```

```
#define No_cluster 0
#define Same_cluster 1
#define Different_cluster 2

cluster_resolution(group_no)                               /* 12.6.18, 7.6.5 */
Int group_no;
{
   Group_data *group = & group_info[group_no];

   if (group->new_cluster_id == null_cluster_id)
   {                                                      /* 7.6.5 a)         */
      group->reclustering_state = Stable;
      group->old_cluster_id = null_cluster_id;
      group->current_cluster_id = null_cluster_id;
      cluster_port_state_selection(group_no, No_cluster);     /* 12.6.19.2    */
   }
   else if (group->new_cluster_id == group->current_cluster_id)
   {                                                      /* 7.6.5 b)         */
      group->reclustering_state = Stable;
      group->old_cluster_id = group->current_cluster_id;
      cluster_port_state_selection(group_no, Same_cluster);   /* 12.6.19.2    */
   }
   else if (!primary_bridge(group_no)
            &&
            (port_info[bridge_info.root_port].peer_old_cluster_id
              != group->current_cluster_id)
           )
   {                                                      /* 7.6.5 c)         */
      group->reclustering_state = Stable;
      group->old_cluster_id = group->new_cluster_id;
      group->current_cluster_id = group->new_cluster_id;
      cluster_port_state_selection(group_no, Different_cluster);
                                                                /* 12.6.19.2 */
   }
   else
   {                                                      /* 7.6.5 d)         */
      if (primary_bridge(group_no)) set_reclustering_delay(group);
      group->reclustering_state = Overlap;
      group->old_cluster_id = group->current_cluster_id;
      group->current_cluster_id = group->new_cluster_id;
      cluster_port_state_selection(group_no, Same_cluster);   /* 12.6.19.2    */
      start_reclustering_delay_timer(group_no);
   }
}
```

```
cluster_port_state_selection(group_no, no_same_diff)     /* 12.6.19, 7.6.6   */
Int group_no;
Int no_same_diff;
{
   Int port_no;
   for (port_no = One; port_no <= No_of_ports; port_no++)
   {                                                      /* 12.6.19.3       */
      if (group_number[port_no] == group_no)
      {                                                   /* 12.6.19.3 a)    */
         if (designated_port(port_no))
            stop_message_age_timer(port_no);              /* 12.6.19.3 a)2)  */
         else
         {                                                /* 12.6.19.3 a)1)  */
            port_info[port_no].config_pending = False;
            port_info[port_no].topology_change_acknowledge = False;
         }
         if (no_same_diff == No_cluster)
            make_blocking(port_no);                       /* 12.6.19.3 b)    */
         else if (no_same_diff == Same_cluster)
         {                                                /* 12.6.19.3 c)    */
            if (isolated_alternate_vp(port_no))
               make_blocking(port_no);                    /* 12.6.19.3 c)1)  */
            else
               make_forwarding(port_no);                  /* 12.6.19.3 c)2)  */
         }
         else                                             /* 12.6.19.3 d)    */
         {
            if (isolated_alternate_vp(port_no))
               make_blocking(port_no);                    /* 12.6.19.3 d)1)  */
            else
            {                                             /* 12.6.19.3 d)2)  */
               port_info[port_no].state = Listening;
               stop_forward_delay_timer(port_no);
               start_forward_delay_timer(port_no);
            }
         }
      }
   }
}

#undef No_cluster
#undef Same_cluster
#undef Different_cluster


cluster_confirmation(group_no)                             /* 12.6.20, 7.6.7 */
Int group_no;
{
   Int port_no;
   Group_data *group = & group_info[group_no];

   group->reclustering_state = Stable;                    /* 7.6.7 a)        */
   group->old_cluster_id = group->current_cluster_id;     /* 7.6.7 b)        */

   for (port_no = One; port_no <= No_of_ports; port_no++)
   {
      if (  (group_number[port_no] == group_no)
            &&
            (isolated_alternate_vp(port_no))
         )
         make_blocking(port_no);                          /* 7.6.7 c)        */
   }
}
```

```
set_reclustering_delay(group)                             /* 7.10.5 (2)      */
Group_data *group;
{
   Time rc_delay_limit = bridge_info.forward_delay - One_Second;

   if (group->primary_reclustering_delay <= rc_delay_limit)
      group->reclustering_delay = group->primary_reclustering_delay;
   else
      group->reclustering_delay = rc_delay_limit;
}


Boolean isolated_in(group_no)
Int group_no;
{
   Int port_no;

   if (group_number[bridge_info.root_port] == group_no)
      return(False);

   for (port_no = One; port_no <= No_of_ports; port_no++)
   {
      if (  (group_number[port_no] == group_no)
            &&
            designated_port(port_no)
         )
         return(False);
   }
   return(True);
}


Boolean isolated_alternate_vp(port_no)
Int port_no;
{
   Group_data *group = & group_info[group_number[port_no]];

   if (  (port_kind[port_no] == Ind_virt_port)
         &&
         (group->current_cluster_id != port_info[port_no].peer_new_cluster_id)
         &&
         (group->current_cluster_id != port_info[port_no].peer_old_cluster_id)
      )
      return(True);
   else
      return(local_knowledge_says_isolated(port_no));
}

         /* where

         Boolean local_knowledge_says_isolated(port_no)
         Int port_no;

         is a pseudo_implementation_specific routine which returns True if
         there is local knowledge available that the port is isolated in
         its group, and returns False otherwise.

         */

#endif
```

```
/** Operation of the Protocol (4.7) **/

received_config_bpdu(port_no, config)                      /* 12.7.1 (4.7.1) */
Int port_no;
Config_bpdu *config;
{
   Boolean root;

   root = root_bridge();

   if (port_info[port_no].state != Disabled)
   {
      if (supersedes_port_info(port_no, config))           /* 12.7.1 a)      */
      {                                                    /* (4.6.2.2)      */
         record_config_information(port_no, config);       /* (4.7.1.1.1)    */
                                                           /* (4.6.2.2)      */
         configuration_update();                           /* (4.7.1.1.2)    */
                                                           /* (4.6.7.2.1)    */
         port_state_selection();                           /* (4.7.1.1.3)    */
                                                           /* (4.6.11.2.1)   */

         if ((!root_bridge()) && root)                     /* (4.7.1.1.4)    */
         {
            stop_hello_timer();

            if (bridge_info.topology_change_detected)      /* (4.7.1.1.5)    */
            {
               stop_topology_change_timer();

               transmit_tcn();                             /* (4.6.6.1)      */

               start_tcn_timer();
            }
         }

         if (port_no == bridge_info.root_port)
         {
            record_config_timeout_values(config);          /* (4.7.1.1.6)    */
                                                           /* (4.6.3.2)      */
            config_bpdu_generation();                      /* (4.6.4.2.1)    */

            if (config->topology_change_acknowledgment)    /* (4.7.1.1.7)    */
            {
               topology_change_acknowledged();             /* (4.6.15.2)     */
            }
         }
      }
      else if (designated_port(port_no))                   /* 12.7.1 b)      */
      {
         reply(port_no);                                   /* (4.7.1.2.1)    */
      }                                                    /* (4.6.5.2)      */
   }
}
```

```
received_tcn_bpdu(port_no, tcn)                            /* 12.7.2 (4.7.2) */
Int port_no;
Tcn_bpdu *tcn;
{
   if (port_info[port_no].state != Disabled)
   {
#ifdef ESTP
/****  ESTP : check received cluster_id on subgroup port  **/

      if (  (designated_port(port_no))
            &&
            (  (group_number[port_no]  == 0)                /* 12.7.2 a)      */
               ||
               (tcn->cluster_id                             /* 12.7.2 b)      */
                  == group_info[group_number[port_no]].current_cluster_id)
            )
         )
#else
      if (designated_port(port_no))
#endif
      {
         topology_change_detection();                       /* (4.7.2.1)      */
                                                            /* (4.6.14.2.1)   */
         acknowledge_topology_change(port_no);              /* (4.7.2.2)      */
      }                                                     /* (4.6.16.2)     */
   }
}


hello_timer_expiry()                                        /* 12.7.3 (4.7.3) */
{
   config_bpdu_generation();                                /* (4.6.4.2.2)    */

   start_hello_timer();
}
```

```
message_age_timer_expiry(port_no)                          /* 12.7.4 (4.7.4) */
Int port_no;
{
   Boolean root;

   root = root_bridge();

   become_designated_port(port_no);                        /* (4.7.4.1)      */
                                                           /* (4.6.10.2.1)   */
   configuration_update();                                 /* (4.7.4.2)      */
                                                           /* (4.6.7.2.2)    */
   port_state_selection();                                 /* (4.7.4.3)      */
                                                           /* (4.6.11.2.2)   */

   if ((root_bridge()) && (!root))                         /* (4.7.4.4)      */
   {
      bridge_info.max_age = bridge_info.bridge_max_age;    /* (4.7.4.4.1)    */
      bridge_info.hello_time = bridge_info.bridge_hello_time;
      bridge_info.forward_delay = bridge_info.bridge_forward_delay;

      topology_change_detection();                         /* (4.7.4.4.2)    */
                                                           /* (4.6.14.2.4)   */
      stop_tcn_timer();                                    /* (4.7.4.4.3)    */

      config_bpdu_generation();                            /* (4.7.4.4.4)    */
                                                           /* (4.6.4.4.3)    */
      start_hello_timer();
   }
}


forward_delay_timer_expiry(port_no)                        /* 12.7.5 (4.7.5) */
Int port_no;
{
   if (port_info[port_no].state == Listening)              /* 12.7.5.1       */
   {
      set_port_state(port_no, Learning);                   /* (4.7.5.1.1)    */

      start_forward_delay_timer(port_no);                  /* (4.7.5.1.2)    */
   }

   else if (port_info[port_no].state == Learning)          /* 12.7.5.2       */
   {
      set_port_state(port_no, Forwarding);                 /* 12.7.5.2(a)    */

      if (designated_for_some_port())                      /* 12.7.5.2(b)    */
      {
         topology_change_detection();                      /* 12.6.14.2(b)   */
      }
   }
}
```

```
        /* where */

        Boolean designated_for_some_port()
        {
           Int port_no;

           for (port_no = One; port_no <= No_of_ports; port_no++)
           {
              if ( port_info[port_no].designated_bridge
                   == bridge_info.bridge_id
                 )
              {
                 return(True);
              }
           }

           return(False);
        }


tcn_timer_expiry()                                          /* 12.7.6 (4.7.6) */
{
      transmit_tcn();                                       /* (4.7.6.1)      */

      start_tcn_timer();                                    /* (4.7.6.2)      */
}


topology_change_timer_expiry()                              /* 12.7.7 (4.7.7) */
{
      bridge_info.topology_change_detected = False;         /* (4.7.7.1)      */

      bridge_info.topology_change = False;                  /* (4.7.7.2)      */
}

hold_timer_expiry(port_no)                                  /* 12.7.8 (4.7.8) */
Int port_no;
{
   if (port_info[port_no].config_pending)
   {
      transmit_config(port_no);                             /* (4.7.8.1)      */
   }                                                        /* (4.6.1.2.3)    */
}


#ifdef ESTP
/****  ESTP : end of reclustering delay or overlap period  **/

reclustering_delay_timer_expiry(group_no)                   /* 12.7.9         */
Int group_no;
{
   if (group_info[group_no].reclustering_state == Reclustering)
      cluster_resolution(group_no);                         /* 12.6.18        */
   else
      cluster_confirmation(group_no);                       /* 12.6.20        */
}

#endif
```

```
/** Management of the Bridge Protocol Entity (4.8) **/
initialisation()                                            /* 12.8.1         */
{
   Int port_no;

#ifdef ESTP
/****  ESTP : for-loop control for group initialization  **/
   Int group_no;
#endif

   bridge_info.designated_root = bridge_info.bridge_id;     /* 12.8.1 a)      */
   bridge_info.root_path_cost = Zero;
   bridge_info.root_port = No_port;

   bridge_info.max_age = bridge_info.bridge_max_age;        /* 12.8.1 b)      */
   bridge_info.hello_time = bridge_info.bridge_hello_time;
   bridge_info.forward_delay = bridge_info.bridge_forward_delay;

   bridge_info.topology_change_detected = False;            /* 12.8.1 c)      */
   bridge_info.topology_change = False;
   stop_tcn_timer();
   stop_topology_change_timer();

   for (port_no = One; port_no <= No_of_ports; port_no++)   /* 12.8.1 d)      */
   {

      initialize_port(port_no);

   }

#ifdef ESTP
/****  ESTP : initialize group information  **/

   for (group_no = One; group_no <= No_of_groups; group_no++)
   {                                                        /* 12.8.1 e)      */
      initialize_group(group_no);
   }
#endif

   port_state_selection();                                  /* 12.8.1 f)      */

   config_bpdu_generation();                                /* 12.8.1 g)      */
   start_hello_timer();
}
```

```
   initialize_port(port_no)
   Int port_no;
   {

      become_designated_port(port_no);                     /* 12.8.1 d)1)     */

      set_port_state(port_no, Blocking);                   /* 12.8.1 d)2)     */

      port_info[port_no].topology_change_acknowledge = False;
                                                           /* 12.8.1 d)3)     */

      port_info[port_no].config_pending = False;           /* 12.8.1 d)3)     */

      stop_message_age_timer(port_no);                     /* 12.8.1 d)4)     */

      stop_forward_delay_timer(port_no);                   /* 12.8.1 d)4)     */

      stop_hold_timer(port_no);                            /* 12.8.1 d)4)     */

#ifdef ESTP
/****  ESTP : initialize peer cluster id's  **/

      if (group_number[port_no] != 0)
      {                                                    /* 12.8.1 e)5)     */
         port_info[port_no].peer_new_cluster_id = null_cluster_id;
         port_info[port_no].peer_old_cluster_id = null_cluster_id;
      }
#endif

   }


#ifdef ESTP
/****  ESTP : initialize group information  **/

   initialize_group(group_no)                              /* 12.8.1 e)       */
   Int group_no;
   {
      Int port_no;
      Group_data *group = & group_info[group_no];

      group->reclustering_state = Stable;                  /* 12.8.1 e)1)     */
      stop_reclustering_delay_timer(group_no);             /* 12.8.1 e)2)     */
      set_reclustering_delay(group);                       /* 12.8.1 e)3)     */
      group->new_cluster_id = null_cluster_id;             /* 12.8.1 e)4)i)   */

      for (port_no = One; port_no <= No_of_ports; port_no++)
      {
         if ( (group_number[port_no] == group_no)
              &&
              (port_info[port_no].state != Disabled)
            )
         {                                                 /* 12.8.1 e)4)ii) */
            group->new_cluster_id = create_cluster_id(group_no);
            break;
         }
      }
      group->current_cluster_id = group->new_cluster_id;   /* 12.8.1 e4)      */
      group->old_cluster_id = group->new_cluster_id;       /* 12.8.1 e4)      */
   }

#endif
```

```
enable_port(port_no)                                       /* 12.8.2 (4.8.2) */
Int port_no;
{
   initialize_port(port_no);

#ifdef ESTP
/****  ESTP : cluster selection needed on enabling a subgroup port  */

   if (group_number[port_no] != 0)  cluster_selection();   /* 12.8.2         */

#endif
   port_state_selection();                                 /* (4.8.2.7)      */
}


disable_port(port_no)                                      /* 12.8.3 (4.8.3) */
Int port_no;
{
   Boolean root;

   root = root_bridge();

   become_designated_port(port_no);                        /* (4.8.3.1)      */

   set_port_state(port_no, Disabled);                      /* (4.8.3.2)      */

   port_info[port_no].topology_change_acknowledge = False; /* (4.8.3.3)      */

   port_info[port_no].config_pending = False;              /* (4.8.3.4)      */

   stop_message_age_timer(port_no);                        /* (4.8.3.5)      */

   stop_forward_delay_timer(port_no);                      /* (4.8.3.6)      */

   configuration_update();

   port_state_selection();                                 /* (4.8.3.7)      */

   if ((root_bridge()) && (!root))                         /* (4.8.3.8)      */
   {
      bridge_info.max_age = bridge_info.bridge_max_age;    /* (4.8.3.8.1)    */
      bridge_info.hello_time = bridge_info.bridge_hello_time;
      bridge_info.forward_delay = bridge_info.bridge_forward_delay;

      topology_change_detection();                         /* (4.8.3.8.2)    */

      stop_tcn_timer();                                    /* (4.8.3.8.3)    */

      config_bpdu_generation();                            /* (4.8.3.8.4)    */

      start_hello_timer();
   }
}
```

```
set_bridge_priority(new_bridge_id)                          /* 12.8.4 (4.8.4) */
Identifier new_bridge_id;                                   /* (4.8.4.1)      */
{
   Boolean root;
   Int port_no;

   root = root_bridge();

   for (port_no = One; port_no <= No_of_ports; port_no++)   /* (4.8.4.2)      */
   {
      if (designated_port(port_no))
      {
         port_info[port_no].designated_bridge = new_bridge_id;
      }
   }

   bridge_info.bridge_id = new_bridge_id;                   /* (4.8.4.3)      */

   configuration_update();                                  /* (4.8.4.4)      */

   port_state_selection();                                  /* (4.8.4.5)      */

   if ((root_bridge()) && (!root))                          /* (4.8.4.6)      */
   {
      bridge_info.max_age = bridge_info.bridge_max_age;     /* (4.8.4.6.1)    */
      bridge_info.hello_time = bridge_info.bridge_hello_time;
      bridge_info.forward_delay = bridge_info.bridge_forward_delay;

      topology_change_detection();                          /* (4.8.4.6.2)    */

      stop_tcn_timer();                                     /* (4.8.4.6.3)    */

      config_bpdu_generation();                             /* (4.8.4.6.4)    */

      start_hello_timer();
   }
}


set_port_priority(port_no, new_port_id)                     /* 12.8.5 (4.8.5) */
Int port_no;
Port_id new_port_id;                                        /* (4.8.5.1)      */
{
   if (designated_port(port_no))                            /* (4.8.5.2)      */
   {
      port_info[port_no].designated_port = new_port_id;
   }

   port_info[port_no].port_id = new_port_id;                /* (4.8.5.3)      */

   if ( (  bridge_info.bridge_id                            /* (4.8.5.4)      */
            == port_info[port_no].designated_bridge
        )
        &&
        (  port_info[port_no].port_id
           <  port_info[port_no].designated_port
        )
      )
   {
      become_designated_port(port_no);                      /* (4.8.5.4.1)   */

      port_state_selection();                               /* (4.8.5.4.2)   */
   }
}
```

```
set_path_cost(port_no, path_cost)                           /* 12.8.6 (4.8.6) */
Int port_no;
Cost path_cost;
{
   port_info[port_no].path_cost = path_cost;                /* (4.8.6.1)      */

   configuration_update();                                  /* (4.8.6.2)      */

   port_state_selection();                                  /* (4.8.6.3)      */
}


/** pseudo-implementation specific timer running support **/

tick()
{
   Int port_no;

#ifdef ESTP
#****   ESTP : add awareness of groups   **/
   Int group_no;
#endif

   if (hello_timer_expired())
   {  hello_timer_expiry();
   }

   if (tcn_timer_expired())
   {  tcn_timer_expiry();
   }

   if (topology_change_timer_expired())
   {  topology_change_timer_expiry();
   }

   for (port_no = One; port_no <= No_of_ports; port_no++)
   {
      if (forward_delay_timer_expired(port_no))
      {
         forward_delay_timer_expiry(port_no);
      }
      if (message_age_timer_expired(port_no))
      {
         message_age_timer_expiry(port_no);
      }
      if (hold_timer_expired(port_no))
      {
         hold_timer_expiry(port_no);
      }
   }

#ifdef ESTP
/****   ESTP : reclustering delay timer   **/

   for (group_no = One; group_no <= No_of_groups; group_no++)
   {
      if (reclustering_delay_timer_expired(group_no)
      {
         reclustering_delay_timer_expiry(group_no);
      }
   }
#endif

}
```

```
/* where */

start_hello_timer()
{  hello_timer.value = (Time) Zero;
   hello_timer.active = True;
}

stop_hello_timer()
{  hello_timer.active = False;
}

Boolean hello_timer_expired()
{  if (hello_timer.active && (++hello_timer.value >= bridge_info.hello_time))
   {   hello_timer.active = False;
       return(True);
   }
   return(False);
}

start_tcn_timer()
{  tcn_timer.value = (Time) Zero;
   tcn_timer.active = True;
}

stop_tcn_timer()
{  tcn_timer.active = False;
}

Boolean tcn_timer_expired()
{  if (tcn_timer.active && (++tcn_timer.value >=
bridge_info.bridge_hello_time))
   {   tcn_timer.active = False;
       return(True);
   }
   return(False);
}

start_topology_change_timer()
{  topology_change_timer.value = (Time) Zero;
   topology_change_timer.active = True;
}

stop_topology_change_timer()
{  topology_change_timer.active = False;
}

Boolean topology_change_timer_expired()
{  if ( topology_change_timer.active
        && ( ++topology_change_timer.value
             >= bridge_info.topology_change_time
           )
       )
   {   topology_change_timer.active = False;
       return(True);
   }
   return(False);
}
```

```
start_message_age_timer(port_no, message_age)
Int  port_no;
Time message_age;
{  message_age_timer[port_no].value = message_age;
   message_age_timer[port_no].active = True;
}

stop_message_age_timer(port_no)
Int  port_no;
{  message_age_timer[port_no].active = False;
}

Boolean message_age_timer_expired(port_no)
Int port_no;
{  if (message_age_timer[port_no].active &&
         (++message_age_timer[port_no].value >= bridge_info.max_age))
   {  message_age_timer[port_no].active = False;
      return(True);
   }
      return(False);
}

start_forward_delay_timer(port_no)
Int port_no;
{  forward_delay_timer[port_no].value = Zero;
   forward_delay_timer[port_no].active = True;
}

stop_forward_delay_timer(port_no)
Int port_no;
{  forward_delay_timer[port_no].active = False;
}

Boolean forward_delay_timer_expired(port_no)
Int port_no;
{  if (forward_delay_timer[port_no].active &&
       (++forward_delay_timer[port_no].value >= bridge_info.forward_delay))
   {   forward_delay_timer[port_no].active = False;
       return(True);
   }
   return(False);
}

start_hold_timer(port_no)
Int port_no;
{  hold_timer[port_no].value = Zero;
   hold_timer[port_no].active = True;
}

stop_hold_timer(port_no)
Int port_no;
{  hold_timer[port_no].active = False;
}

Boolean hold_timer_expired(port_no)
Int port_no;
{  if (hold_timer[port_no].active &&
       (++hold_timer[port_no].value >= bridge_info.hold_time))
   {   hold_timer[port_no].active = False;
       return(True);
   }
   return(False);
}
```

```
#ifdef ESTP
/****  ESTP : reclustering delay timers  **/

start_reclustering_delay_timer(group_no)
Int group_no;
{  reclustering_delay_timer[group_no].value = Zero;
   reclustering_delay_timer[group_no].active = True;
}

stop_reclustering_delay_timer(group_no)
Int group_no;
{  reclustering_delay_timer[group_no].active = False;
}

Boolean reclustering_delay_timer_expired(group_no)
Int group_no;
{  if (reclustering_delay_timer[group_no].active &&
       (++reclustering_delay_timer[group_no].value
         >= group_info[group_no].reclustering_delay))
   {   reclustering_delay_timer[group_no].active = False;
       return(True);
   }
   return(False);
}

#endif

/** pseudo-implementation specific transmit routines **/

#include "transmit.c"
```

Annex A

(normative)

PICS proforma[1]

A.1 Introduction

The supplier of a protocol implementation which is claimed to conform to ISO/IEC 15802-5: 1998 shall complete the following Protocol Implementation Conformance Statement (PICS) proforma.

A completed PICS proforma is the PICS for the implementation in question. The PICS is a statement of which capabilities and options of the protocol have been implemented. The PICS can have a number of uses, including use

- By the protocol implementor, as a checklist to reduce the risk of failure to conform to the standard through oversight.
- By the supplier and acquirer—or potential acquirer—of the implementation, as a detailed indication of the capabilities of the implementation, stated relative to the common basis for understanding provided by the standard PICS proforma.
- By the user—or potential user—of the implementation, as a basis for initially checking the possibility of interworking with another implementation (note that, while interworking can never be guaranteed, failure to interwork can often be predicted from incompatible PICS's).
- By a protocol tester, as the basis for selecting appropriate tests against which to assess the claim for conformance of the implementation.

A.2 Abbreviations and special symbols

A.2.1 Status symbols

M	mandatory
O	optional
O.*n*	optional, but support of at least one of the group of options labeled by the same numeral *n* is required
X	prohibited
pred:	conditional-item symbol, including predicate identification: see A.3.4
¬	logical negation, applied to a conditional item's predicate

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

A.2.2 General abbreviations

ESTP	Extended Spanning Tree Protocol
N/A	not applicable
PICS	Protocol Implementation Conformance Statement
[FDDI]	reference to ISO 9314-2: 1989

A.3 Instructions for completing the PICS proforma

A.3.1 General structure of the PICS proforma

The first part of the PICS proforma, Implementation identification and protocol summary, is to be completed as indicated with the information necessary to identify fully both the supplier and the implementation.

The main part of the PICS proforma is a fixed-format questionnaire, divided into several subclauses, each containing a number of individual items. Answers to the questionnaire items are to be provided in the rightmost column, either by simply marking an answer to indicate a restricted choice (usually Yes or No), or by entering a value or a set or range of values. (Note that there are some items where two or more choices from a set of possible answers can apply: all relevant choices are to be marked.)

Each item is identified by an item reference in the first column; the second column contains the question to be answered; the third column contains the reference or references to the material that specifies the item in the main body of this International Standard. The remaining columns record the status of the item (whether support is mandatory, optional, or conditional), and provide the space for the answers; see also A.3.4.

A supplier may also provide, or be required to provide, further information, categorized as either Additional Information or Exception Information. When present, each kind of further information is to be provided in a further subclause of items labeled A*i* or X*i* respectively for cross-referencing purposes, where *i* is any unambiguous identification for the item (e.g., simply a numeral); there are no other restrictions on its format and presentation.

A completed PICS proforma, including any Additional Information and Exception Information, is the Protocol Implementation Conformation Statement for the implementation in question.

NOTE—Where an implementation is capable of being configured in more than one way, a single PICS may be able to describe all such configurations. However, the supplier has the choice of providing more than one PICS, each covering some subset of the implementation's configuration capabilities, in case that makes for easier and clearer presentation of the information.

A.3.2 Additional Information

Items of Additional Information allow a supplier to provide further information intended to assist in the interpretation of the PICS. It is not intended or expected that a large quantity will be supplied, and a PICS can be considered complete without any such information. Examples might be an outline of the ways in which a (single) implementation can be set up to operate in a variety of environments and configurations, or information about aspects of the implementation that are outside the scope of this International Standard but that have a bearing upon the answers to some items.

References to items of Additional Information may be entered next to any answer in the questionnaire, and may be included in items of Exception Information.

A.3.3 Exception Information

It may occasionally happen that a supplier will wish to answer an item with mandatory status (after any conditions have been applied) in a way that conflicts with the indicated requirement. No pre-printed answer will be found in the Support column for this: instead, the supplier shall write the missing answer into the Support column, together with an X*i* reference to an item of Exception Information, and shall provide the appropriate rationale in the Exception item itself.

An implementation for which an Exception item is required in this way does not conform to this International Standard.

NOTE—A possible reason for the situation described above is that a defect in this International Standard has been reported, a correction for which is expected to change the requirement not met by the implementation.

A.3.4 Conditional status

A.3.4.1 Conditional items

The PICS proforma contains a number of conditional items. These are items for which both the applicability of the item itself, and its status if it does apply (mandatory or optional) are dependent upon whether or not certain other items are supported.

Where a group of items is subject to the same condition for applicability, a separate preliminary question about the condition appears at the head of the group, with an instruction to skip to a later point in the questionnaire if the “Not Applicable” answer is selected. Otherwise, individual conditional items are indicated by a conditional symbol in the Status column.

A conditional symbol is of the form “**pred:** *S*”where **pred** is a predicate as described in A.3.4.2, and *S* is a status symbol, M or O.

If the value of the predicate is true (see A.3.4.2), the conditional item is applicable, and its status is indicated by the status symbol following the predicate; the answer column is to be marked in the usual way. If the value of the predicate is false, the “Not Applicable” (N/A) answer is to be marked.

A.3.4.2 Predicates

A predicate is one of the following:

a) An item-reference for an item in the PICS proforma: the value of the predicate is true if the item is marked as supported, and is false otherwise.

b) A predicate-name, for a predicate defined (at the end of A.5) as a Boolean expression constructed by combining item-references using the Boolean operator OR: the value of the predicate is true if one or more of the items is marked as supported.

c) The logical negation symbol “¬” prefixed to an item-reference or predicate-name: the value of the predicate is true if the value of the predicate formed by omitting the “¬” symbol is false, and vice versa.

Each item whose reference is used in a predicate or predicate definition, or in a preliminary question for grouped conditional items, is indicated by an asterisk in the Item column.

A.4 PICS proforma—ISO/IEC 15802-5: 1998: Identification

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	

NOTES

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.

2—The terms Name and Version should be interpreted appropriately to correspond with a supplier's terminology (e.g., Type, Series, Model).

A.4.2 Protocol summary, ISO/IEC 15802-5: 1998

Identification of protocol specification	**ISO/IEC 15802-5: 1998, Remote MAC Bridging**
Identification of amendments and corrigenda to the PICS proforma that 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 15802-5: 1998.)	No □ Yes □

Date of Statement	

A.5 Major capabilities and options

Item	Protocol Feature	References	Status	Support
	Communications Support			
	Which MAC types are supported on LAN ports, in conformance with the relevant MAC Standards?			
CP1	CSMA/CD, ISO/IEC 8802-3		O.1	Yes □ No □
CP2	Token Bus, ISO/IEC 8802-4		O.1	Yes □ No □
CP3	Token Ring, ISO/IEC 8802-5		O.1	Yes □ No □
*CP4	FDDI, ISO 9314-2		O.1	Yes □ No □
CP5	DQDB, ISO/IEC 8802-6		O.1	Yes □ No □
	Which types of LAN MAC frames can be relayed on the virtual ports?			
CV1	CSMA/CD		O.2	Yes □ No □
CV2	Token Bus		O.2	Yes □ No □
CV3	Token Ring		O.2	Yes □ No □
CV4	FDDI		O.2	Yes □ No □
CV5	DQDB		O.2	Yes □ No □
CVx	Which types of non-LAN communication are supported on the virtual ports?			A___
	What group structures are supported:	6.13.3		
*CG1	— Virtual LAN?		O.3	Yes □ No □
*CG2	— Virtual mesh?		O.3	Yes □ No □
*CG3	— Mixed configuration?		O.3	Yes □ No □
CGp	Is an access priority mechanism supported across the non-LAN communications?	5.4 e), 8.2.6	O	Yes □ No □
CLLC	Is LLC Type 1 supported on the LAN ports in conformance with ISO/IEC 8802-2?	6.2, 6.3, 6.12	M	Yes □
RF	Relay and filtering of frames (A.6)	6.5–6.7, 8	M	Yes □
RA1	Does the bridge filter frames with equal source and destination addresses?	6.7.1, 6.7.2	O.4	Yes □ No □
RA2	Does the bridge not filter frames with equal source and destination addresses?	6.7.1, 6.7.2	O.4	Yes □ No □
*RPm	Does the bridge support management of the priority of relayed frames?	6.7.4	O	Yes □ No □

Item	Protocol Feature	References	Status	Support
FI	Maintenance of filtering information (A.7)	6.8, 6.9	M	Yes □
FMr	Can the filtering database be read by management?	6.9	O	Yes □ No □
*FMu	Can static entries be created and deleted?	6.9.1	O	Yes □ No □
FMp	Can static entries be created and deleted in the permanent database?	6.9.3	O	Yes □ No □
FMa	Can the bridge be configured to use an arbitrarily chosen value from the specified range of values for ageing time?	6.9.2	O	Yes □ No □
FRl	Is filtering information exchanged with other remote bridges belonging to the same group(s)?	6.9	O	Yes □ No □
AD	Addressing (A.8)	6.12	M	Yes □
*AU	Can the bridge be configured to use 48-bit universally administered MAC addresses?	6.12	O.5	Yes □ No □
*AL	Can the bridge be configured to use 48-bit locally administered MAC addresses?	6.12	O.5	Yes □ No □
ST	Spanning Tree Algorithm (A.9)	7.3–7.6, 7.8	M	Yes □
SLp	Spanning Tree Protocol on LAN ports	7.7, 7.9	M	Yes □
*SMt	Does the bridge support management of the spanning tree priority parameters?	7.2	O	Yes □ No □
*SMp	Does the bridge support management of the spanning tree protocol timers?	7.10	O	Yes □ No □
	Extended Spanning Tree Protocol (A.10):			
*EP1	ESTP on virtual LAN ports, Version 0	9, 13	**CG1:** O	N/A □ Yes □ No □
*EP2	ESTP on virtual LAN ports, Version 1	9, 13	**CG1:** O	N/A □ Yes □ No □
*EP3	ESTP on subgroup ports, Version 1	9, 13	**CGs:** O	N/A □ Yes □ No □
*BM	Bridge management operations (A.11)	9	O	Yes □ No □
*BMp	LAN/MAN management protocol	10	**BM:** O	N/A □ Yes □ No □

Predicate definitions: **CGs** = CG2 OR CG3
EPv1 = EP2 OR EP3

A.6 Relay and filtering of frames

Item	Protocol Feature	References	Status	Support
Rme	Are received frames with MAC errors discarded?	6.5	M	Yes □
Rlp	Are correctly received frames submitted to the learning process?	6.5	M	Yes □
Rlc	Are all frames addressed to LAN ports of the bridge submitted to LLC?	6.5	M	Yes □
Rud	Are frames of type user_data_frame the only type of frame relayed on LAN ports?	6.5, 6.6	M	Yes □
Rnr	Are request_with_no_response frames the only frames relayed on LAN ports?	6.5, 6.6	M	Yes □
Rq1	Are relayed frames queued for transmission only under the conditions in 6.7.1?	6.7.1	M	Yes □
Rq2	Is the order of relayed frames of given user priority between a given pair of MAC addresses preserved?	6.7.1, 6.7.3, 8.2.1	M	Yes □
Rq3	Is a relayed frame submitted for transmission only once to the MAC entity or MAC service support functions for each port to which it is forwarded?	6.7.3	M	Yes □
Rq4	Are queued frames discarded if a port leaves Forwarding state?	6.7.3	M	Yes □
Rtd	Is a maximum bridge transit delay enforced?	6.7.3	M	Yes □
Rue	Is the undetected frame error rate greater than that achievable by preservation of the FCS where possible?	5.3.1	X	No □
Rfp	Is the FCS preserved for frames relayed between ports supporting the same MAC type?	5.4 g)	O	Yes □ No □
Rp1	Is the user priority of relayed frames preserved where possible?	6.7.4	M	Yes □
Rp2	Is the user priority set to Outbound User Priority where appropriate?	6.7.4	M	Yes □
Rp3	Is the access priority, when supported, set to the user priority?	6.7.4	O.6	Yes □ No □
Rp4	Is the access priority, when supported, set to the Outbound Access Priority?	6.7.4	O.6	Yes □ No □
Rp5	Can the bridge use the specified default values of Outbound Access Priority?	6.7.4	M	Yes □
	If FDDI LAN ports are not supported, item CP4, mark N/A and continue at item Rpm1:			N/A □
Rfd1	Is an M-UNITDATA indication generated on receipt of a valid frame transmitted by the LAN port's local MAC entity?	5.5, [FDDI]	X	No □
Rfd2	Is only Asynchronous service used?	[FDDI]	M	Yes □
Rfd3	On receiving a frame from an FDDI ring for forwarding, does the bridge set the C indicator?	5.5, [FDDI]	O	Yes □ No □
Rfd4	On receiving a frame from an FDDI ring for forwarding, does the bridge leave the C indicator unaltered?	5.5, [FDDI]	O	Yes □ No □

Item	Protocol Feature	References	Status	Support
Rpm1	If management of priority is supported, can the Outbound User Priority be set to each value in the range of values specified, for each port?	6.7.4	**RPm**: M	N/A □ Yes □
Rpm2	If management of priority is supported, can the Outbound Access Priority be set to each value in the range of values specified, for each port?	6.7.4	**RPm**: M	N/A □ Yes □
Rpm3	If management of priority is not supported, is the Outbound User Priority for each transmission port set to the specified default value?	6.7.4	¬**RPm**: M	N/A □ Yes □
Rpm4	If management of priority is not supported, is the Outbound Access Priority for each transmission port set to the specified default value?	6.7.4	¬**RPm**: M	N/A □ Yes □

A.7 Maintenance of filtering information

Item	Protocol Feature	References	Status	Support
Fse	Does the filtering database support static entries?	6.9	M	Yes □
Fde	Does the filtering database support dynamic entries?	6.9	M	Yes □
Fd1	Are dynamic entries created and updated if and only if the port state permits?	6.8, 7.4	M	Yes □
Fd2	Are dynamic entries made on receipt of frames with a group source address?	6.8, 6.9.2	X	No □
Fd3	Can a dynamic entry be created that conflicts with an existing static entry?	6.8, 6.9	X	No □
Fd4	Are dynamic entries removed from the filtering database if not updated for the ageing time period?	6.9.2	M	Yes □
Fd5	Does the creation of a static entry remove any dynamic entry for the same address?	6.9	M	Yes □
Fs1	Does each static entry specify a MAC address and an outbound port map for each inbound port?	6.9.1	M	Yes □
Fs2	Does each dynamic entry specify a MAC address and a port number?	6.9.2	M	Yes □
Fip	Is the filtering database initialized with the entries contained in the permanent database?	6.9.3	M	Yes □
	If static entries cannot be created and deleted, item FMu, mark N/A and continue at item Fma:			N/A □
Fms1	Can static entries be made for individual MAC addresses?	6.9.1	M	Yes □
Fms2	Can static entries be made for group MAC addresses?	6.9.1	M	Yes □
Fms3	Can a static entry be made for the broadcast MAC address?	6.9.1	M	Yes □
Fma	Can the bridge be configured to use the default value recommended for ageing time?	6.9.2	O	Yes □ No □

A.8 Addressing

Item	Protocol Feature	References	Status	Support
Dl1	Does each LAN port have a separate MAC address?	6.12	M	Yes □
Dg1	Do all BPDUs transmitted on LANs use the same destination group MAC address?	6.12, 7.2	M	Yes □
Dg2	Do all BPDUs transmitted on LANs use the Bridge Protocol Group Address when universally administered addresses are used?	6.12, 7.2	**AU**: M	N/A □ Yes □
Dl2	Is the source MAC address of each BPDU transmitted on a LAN the address of the transmitting port?	6.12	M	Yes □
Dlu	Is the bridge address a universally administered MAC address?	6.12, 7.2	M	Yes □
Dra	Does the bridge relay any frames addressed to any of the Reserved Addresses?	6.12	X	No □
Dmp	Is bridge management accessible through each LAN port using the MAC address of the port and the assigned LLC address?	6.12	**BM**: O	N/A □ Yes □ No □
Dmg	Is bridge management accessible through all LAN ports using the All LANs Bridge Management Group Address?	6.12	**BM**: O	N/A □ Yes □ No □
Dbp	Is the bridge address the MAC address of port 1?	6.12	**AU**: O	N/A □ Yes □ No □
*Dpa	Are group addresses additional to the Reserved Addresses preconfigured in the permanent database?	6.12	O	Yes □ No □
Dpd	Can the additional preconfigured entries in the permanent database be deleted?	6.12	**Dpa**: O	N/A □ Yes □ No □
Dbpe	Can a group MAC address be assigned to identify the bridge protocol entity?	7.2	**AL**: M	N/A □ Yes □
Dpi	Does each port (LAN port and virtual port) have a distinct identifier?	7.2, 7.5.4.1.1	M	Yes □

A.9 Spanning Tree Algorithm

A.9.1 Bridge, group, and port parameters

Item	Protocol Feature	References	Status	Support
Sbp	Are all of the following bridge parameters maintained?	7.5.3	M	Yes □
	— Designated Root	7.5.3.1		
	— Root Path Cost	7.5.3.2		
	— Root Port	7.5.3.3		
	— Max Age	7.5.3.4		
	— Hello Time	7.5.3.5		
	— Forward Delay	7.5.3.6		
	— Bridge Identifier	7.5.3.7		
	— Bridge Max Age	7.5.3.8		
	— Bridge Hello Time	7.5.3.9		
	— Bridge Forward Delay	7.5.3.10		
	— Topology Change Detected	7.5.3.11		
	— Topology Change	7.5.3.12		
	— Topology Change Time	7.5.3.13		
	— Hold Time	7.5.3.14		
Sgp	Are all of the following group parameters maintained for each non-virtual-LAN group?	7.5.5	**CGs:** M	N/A □ Yes □
	— Reclustering State	7.5.5.1		
	— Current Cluster Identifier	7.5.5.2		
	— New Cluster Identifier	7.5.5.3		
	— Old Cluster Identifier	7.5.5.4		
	— Reclustering Delay	7.5.5.5		
	— Primary Reclustering Delay	7.5.5.6		
Spp	Are all of the following port parameters maintained for each LAN port and virtual port?	7.5.4.1	M	Yes □
	— Port Identifier	7.5.4.1.1		
	— State	7.5.4.1.2		
	— Path Cost	7.5.4.1.3		
	— Designated Root	7.5.4.1.4		
	— Designated Cost	7.5.4.1.5		
	— Designated Bridge	7.5.4.1.6		
	— Designated Port	7.5.4.1.7		
	— Topology Change Acknowledge	7.5.4.1.8		
Ssp	Are both of the following port parameters maintained for each subgroup port?	7.5.4.2	**CGs:** M	N/A □ Yes □
	— Peer New Cluster Identifier	7.5.4.2.1		
	— Peer Old Cluster Identifier	7.5.4.2.2		

A.9.2 Spanning Tree Protocol at LAN ports

Item	Protocol Feature	References	Status	Support
Slr	Are protocol parameters maintained, and BPDUs transmitted at LAN ports, as required on each of the following events? — Received Configuration BPDU on LAN port or received configuration message on virtual port — Received topology change notification on LAN port (as BPDU) or virtual port	7.7, 7.5.3, 7.5.4, 7.5.5, 7.6 7.6.1, 7.6.2 7.6.9, 7.7.1.3	M	Yes □
Slt	Are protocol parameters maintained, and BPDUs transmitted at LAN ports, as required in relation to the following timer parameters? — Hello Time — Topology Change Time — Hold Time — Max Age and received Message Age — Forward Delay — Reclustering Delay	7.7, 7.5.3, 7.5.4, 7.5.5, 7.6 7.7.1.1 7.7.1.3 7.7.1.1.1 7.6.2.1 7.4.2, 7.4.3 7.6.5, 7.6.6	M	Yes □
Sln	Are Topology Change Notification BPDUs retransmitted at the specified intervals at any LAN port that is the root port?	7.7.2.3	M	Yes □
Slm	Are protocol parameters modified, and BPDUs transmitted at LAN ports, as required by the following operations? — Initialization — Enable Port — Disable Port — Set Bridge Priority — Set Port Priority — Set Path Cost	7.7, 7.5.3, 7.5.4, 7.5.5 7.8.1 7.8.2 7.8.3 7.8.4 7.8.5 7.8.6	M	Yes □

A.9.3 Spanning tree timers

Item	Protocol Feature	References	Status	Support
St1	Does the bridge underestimate the increment to the Message Age parameter in transmitted BPDUs at LAN ports and configuration messages at virtual ports?	7.10.1	X	No □
St2	Does the bridge underestimate forward delay?	7.10.1	X	No □
St3	Does the bridge overestimate the Hello Time interval?	7.10.1	X	No □
St4	Does the bridge use the specified value for Hold Time?	7.10.2	M	Yes □

A.9.4 Management of spanning tree topology and timers

Item	Protocol Feature	References	Status	Support
	If management of the spanning tree priority parameters, item SMp, is not supported, mark N/A and go to item Smt1:			N/A ☐
Smp1	Can the relative priority of the bridge be set?	7.2, 7.5.3.7, 7.8.4	M	Yes ☐
Smp2	Can the relative priority of each port be set?	7.2, 7.5.4.1.1, 7.8.5	M	Yes ☐
Smp3	Can the path cost for each port be set?	7.2, 7.5.4.1.3, 7.8.6	M	Yes ☐
	If management of the Spanning Tree protocol timers, item SMt, is not supported, mark N/A and go to A.9.5:			N/A ☐
Smt1	Can Bridge Max Age be set to the full range of values specified?	7.10.2, 7.5.3.8	M	Yes ☐
Smt2	Can Bridge Hello Time be set to the full range of values specified?	7.10.2, 7.5.3.9	M	Yes ☐
Smt3	Can Bridge Forward Delay be set to the full range of values specified?	7.10.2, 7.5.3.10	M	Yes ☐
Smt4	Can Primary Reclustering Delay be set to the full range of values specified, for each non-virtual-LAN group?	7.10.5, 7.5.5.6	CGs: M	N/A ☐ Yes ☐

A.9.5 BPDUs transmitted and received at LAN ports

Item	Protocol Feature	References	Status	Support
Slb1	Are all of the following BPDU parameter types encoded as specified? — Protocol identifier — Protocol version identifier — BPDU type — Flags — Bridge identifiers — Root path cost — Port identifiers — Timer values	7.9	M	Yes ☐
Slb2	Do Configuration BPDUs transmitted at LAN ports have the format and parameters specified?	7.9	M	Yes ☐
Slb3	Do Topology Change Notification BPDUs transmitted at LAN ports have the format and parameters specified?	7.9	M	Yes ☐
Slb4	Are BPDUs received at LAN ports validated as specified?	7.9	M	Yes ☐

A.10 Extended Spanning Tree Protocol

If the Extended Spanning Tree Protocol is not supported, items EP1–EP3, mark N/A and go to A.11: N/A □

Item	Protocol Feature	References	Status	Support
Ebt	Are all of the following bridge timers maintained? — Hello Timer — Topology Change Notification Timer — Topology Change Timer	13, 12.3.4.1 12.3.4.2 12.3.4.3	M	Yes □
Epp	Is thc Configuration Pending port parameter maintained for each port?	12.3.5.1, 13	M	Yes □
Ept	Are all of the following port timers maintained for each port? — Message Age Timer — Forward Delay Timer — Hold Timer	13, 12.3.6.1 12.3.6.2 12.3.6.3	M	Yes □
Egt	Is the Reclustering Delay Timer maintained for each non-virtual-LAN group?	12.3.8, 13	M	Yes □
Ert	Are protocol parameters and timers maintained, and BPDUs transmitted at LAN ports and virtual ports, as required on each of the following events? — Received Configuration BPDU — Received Topology Change Notification BPDU — Hello Timer expiry — Message Timer expiry — Forward Delay Timer expiry — Topology Change Notification Timer expiry — Topology Change Timer expiry — Hold Timer expiry — Reclustering Delay Timer expiry	7.5.3, 7.5.4, 7.5.5, 7.6, 13 12.7.1 12.7.2 12.7.3 12.7.4 12.7.5 12.7.6 12.7.7 12.7.8 12.7.9	M	Yes □
Ebm	Are protocol parameters and timers maintained, and BPDUs transmitted at LAN ports and virtual ports, as required for each of the following operations? — Initialization — Enable Port — Disable Port — Set Bridge Priority — Set Port Priority — Set Path Cost	7.5.3, 7.5.4, 7.5.5, 7.6, 13 12.8.1 12.8.2 12.8.3 12.8.4 12.8.5 12.8.6	M	Yes □

Item	Protocol Feature	References	Status	Support
Etx1	Are all of the following BPDU parameter types encoded as specified, in BPDUs transmitted at virtual ports? — Protocol identifier — Protocol version identifier — BPDU type — Flags — Bridge identifiers — Root path cost — Port identifiers — Timer values — Cluster identifiers — Reason	12.5.2 12.5.2.1 12.5.2.1 12.5.2.1 12.5.2.1 12.5.2.1 12.5.2.1 12.5.2.1 12.5.2.1 12.5.2.2 12.5.2.3	M	Yes □
Etx2	Do Configuration BPDUs transmitted at subgroup ports have the format and parameters specified for Version 1 BPDUs?	12.5.3.1, Figure 12-1	**EP3**: M	N/A □ Yes □
Etx3	Do Configuration BPDUs transmitted at virtual LAN ports have the format and parameters specified for Version 1 BPDUs?	12.5.3.1, Figure 12-1	**EP2**: M	N/A □ Yes □
Etx4	Do Configuration BPDUs transmitted at virtual LAN ports have the format and parameters specified for Version 0 BPDUs?	12.5.3.1, Figure 12-1	**EP1**: M	N/A □ Yes □
Etx5	Do Topology Change Notification BPDUs transmitted at subgroup ports have the format and parameters specified for Version 1 BPDUs?	12.5.3.2, Figure 12-2	**EP3**: M	N/A □ Yes □
Etx6	Do Topology Change Notification BPDUs transmitted at virtual LAN ports have the format and parameters specified for Version 1 BPDUs?	12.5.3.2, Figure 12-2	**EP2**: M	N/A □ Yes □
Etx7	Do Topology Change Notification BPDUs transmitted at virtual LAN ports have the format and parameters specified for Version 0 BPDUs?	12.5.3.2, Figure 12-2	**EP1**: M	N/A □ Yes □
Etx8	Are the optional diagnostic parameters supported in Topology Change Notification BPDUs?	12.3.2.2 – 12.3.2.4	**EPv1**: O	Yes □ No □
Erx	Are received BPDUs validated as specified?	12.5.4	M	Yes □
Ess1	In each group, do the mechanisms for transfer of BPDUs between remote bridges conform to the QOS requirements for the RB-Protocol Subnetwork Service, in respect of the following? — Misordering and duplication — Loss — Undetected error rate	12.2.1, 12.2.1.1, 12.2.1.2.1 12.2.1.2.2 12.2.1.2.3	M	Yes □
Ess2	In each group, do the mechanisms for transfer of BPDUs between remote bridges meet the group connectivity requirement?	12.2.1, 12.2.2	M	Yes □

A.11 Bridge management

If bridge management is not supported, item BM, mark N/A and go to A.12: N/A □

Item	Protocol Feature	References	Status	Support
	Are the following bridge management operations supported as specified?			
Mb1	Discover Bridge	9.4.1.1	M	Yes □
Mb2	Read Bridge	9.4.1.2	M	Yes □
Mb3	Set Bridge Name	9.4.1.3	M	Yes □
Mb4	Reset Bridge	9.4.1.4	M	Yes □
Mp1	Read Port	9.4.2.1	M	Yes □
Mp2	Set Port Name	9.4.2.2	M	Yes □
Mg1	Read Group, if subgroup ports supported	9.4.3.1	CGs: M	N/A □ Yes □
Mg2	Set Group Name, if subgroup ports supported	9.4.3.2	CGs: M	N/A □ Yes □
Mp4	Read Forwarding Port Counters	9.6.1.1	M	Yes □
Mp5	Read Transmission Priority	9.6.2.1	M	Yes □
Mp6	Set Transmission Priority	9.6.2.2	M	Yes □
Mf1	Read Filtering Database	9.7.1	M	Yes □
Mf2	Set Filtering Database Ageing Time	9.7.1	M	Yes □
Mf3	Read Permanent Database	9.7.4	M	Yes □
Mf4	Create Filtering Entry	9.7.5.1	M	Yes □
Mf5	Delete Filtering Entry	9.7.5.2	M	Yes □
Mf6	Read Filtering Entry	9.7.5.3	M	Yes □
Mf7	Read Filtering Entry Range	9.7.5.4	M	Yes □
Mbp1	Read Bridge Protocol Parameters	9.8.1.1	M	Yes □
Mbp2	Set Bridge Protocol Parameters	9.8.1.2	M	Yes □
Mpp1	Read Port Parameters	9.8.2.1	M	Yes □
Mpp2	Force Port State	9.8.2.2	M	Yes □
Mpp3	Set Port Parameters	9.8.2.3	M	Yes □
Mgp1	Read Group Parameters, if subgroup ports supported	9.8.3.1	CGs: M	N/A □ Yes □
Mgp2	Set Reclustering Delay, if subgroup ports supported	9.8.3.2	CGs: M	N/A □ Yes □

If ISO/IEC 15802-2 LAN/MAN management is not supported, item BMp, mark N/A and go to A.12: N/A □

Item	Protocol Feature	References	Status	Support
Mlm	Does the bridge conform to ISO/IEC 15802-2: 1995?	10	M	Yes □
Mop	Are the management operations mapped to LMMS services and parameters as specified?	10.1–10.6	M	Yes □

A.12 Performance and parameter values

Specify the values of the parameters indicated.

Item	Parameter	References	Status	Values Supported
Pgr	Guaranteed Bridge Relaying Rate G_R, and the associated measurement interval T_R	11	M	G_R = ____________ frames/s T_R = ____________ s
Pgf	Guaranteed Port Filtering Rate G_F, and the associated measurement interval T_F, for each LAN port: identify each port clearly, by port number or other means.	11	M	
	Port(s):			G_F = ____________ frames/s T_F = ____________ s
	Port(s):			G_F = ____________ frames/s T_F = ____________ s
	Port(s):			G_F = ____________ frames/s T_F = ____________ s
	Port(s):			G_F = ____________ frames/s T_F = ____________ s
	Port(s):			G_F = ____________ frames/s T_F = ____________ s
Psf	Size of the filtering database	6.9	M	________ entries
Psp	Size of the permanent database	6.9	M	________ entries
Pfr	Time to detect failure of the relaying connectivity requirement	8.1	M	Range: from ________ s to ________ s Default ________ s
Pfg	Time to detect failure of the group connectivity requirement	12.2.2	M	Range: from ________ s to ________ s Default ________ s

Annex B

(normative)

Allocation of object identifier values

B.1 Introduction

This annex contains a summary of all object identifier values allocated by this International Standard.

Each table shows allocations related to a specific category of information object. The heading of the table identifies the category of information object, and shows the invariant part of the object identifier value allocated to the entries in the table. The column marked Arc shows the value allocated to the arc subsequent to the invariant part, which completes the object identifier value allocated. The column marked Purpose contains a text description of the information object and a reference to the definition of the object in this International Standard. The column marked Status shows the status of the allocated values, using the following notation:

R Reserved. The object identifier value is reserved for future use by this International Standard.

C Current. The object identifier value has been allocated to an information object that is identified within the current revision of this International Standard.

B.2 Allocation tables

Allocations for Standard-specific extensions
Invariant part of object identifier value =
{iso(1) member-body(2) us(840) ieee802dot1D(10009) standardSpecificExtensions(0)}

Arc	Purpose	Status
N/A	N/A	N/A

Allocations for functional unit packages
Invariant part of object identifier value =
{iso(1) member-body(2) us(840) ieee802dot1D(10009) functionalUnitPackage(1)}

Arc	Purpose	Status
N/A	N/A	N/A