



**IEEE Standard for
Local and metropolitan area networks—**

**Virtual Bridged Local Area Networks
Amendment 7: Provider Backbone Bridges**

IEEE Computer Society

Sponsored by the
LAN/MAN Standards Committee

IEEE
3 Park Avenue
New York, NY 10016-5997, USA
14 August 2008

IEEE Std 802.1ah™-2008
(Amendment to
IEEE Std 802.1Q™-2005)

IEEE Std 802.1ah™-2008

(Amendment to
IEEE Std 802.1Q™-2005)

**IEEE Standard for
Local and metropolitan area networks—**

Virtual Bridged Local Area Networks Amendment 7: Provider Backbone Bridges

Sponsor

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

Approved 12 June 2008

IEEE-SA Standards Board

Abstract: This amendment defines an architecture and bridge protocols for interconnection of multiple Provider Bridged Networks, allowing a Provider to support up to 2^{24} service instances.

Keywords: Bridged Local Area Networks, LANs, local area networks, MAC Bridges, MANs, metropolitan area networks, MRP, MSTP, Multiple Registration Protocol, Multiple Spanning Tree Protocol, Multiple VLAN Registration Protocol, MVRP, Provider Bridged Local Area Networks, Virtual Bridged Local Area Networks, virtual LANs

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

Copyright © 2008 by the Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 14 August 2008. Printed in the United States of America.

IEEE and 802 are registered trademarks in the U.S. Patent & Trademark Office, owned by The Institute of Electrical and Electronics Engineers, Incorporated.

PDF: ISBN 973-07381-5762-7 STD95803
Print: ISBN 973-07381-5763-4 STDPD95803

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

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

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

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

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

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

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration. A statement, written or oral, that is not processed in accordance with the IEEE-SA Standards Board Operations Manual shall not be considered the official position of IEEE or any of its committees and shall not be considered to be, nor be relied upon as, a formal interpretation of the IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position, explanation, or interpretation of the IEEE.

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

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

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

Introduction

This introduction is not part of IEEE Std 802.1ah-2008, IEEE Standards for Local and Metropolitan Area Networks—Virtual Bridged Local Area Networks—Amendment 7: Provider Backbone Bridges.

Notice to users

Laws and regulations

Users of these documents should consult all applicable laws and regulations. Compliance with the provisions of this standard does not imply compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Copyrights

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

Updating of IEEE documents

Users of IEEE standards should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect. In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit the IEEE Standards Association website at <http://ieeexplore.ieee.org/xpl/standards.jsp>, or contact the IEEE at the address listed previously.

For more information about the IEEE Standards Association or the IEEE standards development process, visit the IEEE-SA website at <http://standards.ieee.org>.

Errata

Errata, if any, for this and all other standards can be accessed at the following URL: <http://standards.ieee.org/reading/ieee/updates/errata/index.html>. Users are encouraged to check this URL for errata periodically.

Interpretations

Current interpretations can be accessed at the following URL: <http://standards.ieee.org/reading/ieee/interp/index.html>.

Patents

Attention is called to the possibility that implementation of this amendment may require use of subject matter covered by patent rights. By publication of this amendment, no position is taken with respect to the existence or validity of any patent rights in connection therewith. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this amendment are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

Participants

At the time this amendment was submitted, the IEEE 802.1 Working Group had the following membership:

Tony Jeffree, *Chair*

Paul Congdon, *Vice Chair*

Stephen Haddock, *Chair, Interworking Task Group*

Paul Bottorff, Stephen Haddock, and Muneyoshi Suzuki, *Editors*

Osama Aboul-Magd	Michael Johas Teener	Dan Romascanu
Zehavit Alon	Prakash Kashyap	Moran Roth
Caitlin Bestler	Hal Keen	Jessy V. Rouyer
Jan Bialkowski	Keti Kilcrease	Ali Sajassi
Rob Boatright	Yongbum Kim	Joseph Salowey
Jean-Michel Bonnamy	Philippe Klein	Panagiotis Saltsidis
Rudolf Brandner	Mike Ko	Satish Sathe
Craig W. Carlson	Raghu Kondapalli	John Sauer
Frank Chao	Bruce Kwan	Mick Seaman
Rao Cherukuri	Kari Laihonen	Koichiro Seto
Taesik Cheung	Yannick Le Goff	Himanshu Shah
Alex Conta	Gael Mace	Ravi Shenoy
Diego Crupnicoff	Ben Mack-Crane	Nurit Sprecher
Claudio Desanti	David Martin	Kevin B. Stanton
Linda Dunbar	Riccardo Martinotti	Bob Sultan
Hesham Elbakoury	Marco Mascitto	George Swallow
David Elie-Dit-Cosaque	Alan McGuire	Attila Takacs
Janos Farkas	Menucher Menuchery	Francois Tallet
Don Fedyk	John Messenger	John Terry
Felix Feifei Feng	Dinesh Mohan	Pat Thaler
Norm Finn	Kevin Nolish	Oliver Thorp
Bob Frazier	Hiroshi Ohta	Fred Tuck
John Fuller	David Olsen	Maarten Vissers
Geoffrey Garner	Shlomo Ovadia	Manoj Wadekar
Anoop Ghanwani	Don Pannell	Yan Wang
Franz Goetz	Glenn Parsons	Brian Weis
Eric Gray	Ken Patton	Bert Wijnen
Ken Grewal	Joe Pelissier	Ludwig Winkel
Craig Gunther	Hayim Porat	Michael D. Wright
Mitch Gusat	Max Pritikin	Chien-Hsien Wu
Brian Hassink	Ananda Rajagopal	Ken Young
Brian Hausauer	Karen Randall	
Asif Hazarika	Robert Roden	
Romain Insler	Guenter Roeck	
Paul Hongkyu Jeong	Josef Roes	

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

Osama Aboul-Magd	John Hawkins	Maximilian Riegel
Toru Aihara	Atsushi Ito	Jessy V. Rouyer
Butch Anton	Raj Jain	Randall Safier
Danilo Antonelli	Tony Jeffree	John Sargent
Hugh Barrass	Piotr Karocki	John Sauer
Parag Bhatt	Stuart J. Kerry	Veselin Skendzic
Tomo Bogataj	Yongbum Kim	Kapil Sood
Paul Bottorff	Thomas Kurihara	Amjad Soomro
Michael Bouchet	John Lemon	Nurit Sprecher
James Carlo	William Lumpkins	Manikantan Srinivasan
Juan Carreon	Thomas Mack-Crane	Thomas Starai
Keith Chow	David Martin	Marian Stasney
Charles Cook	Jonathon McLendon	Alourdes Sully
Arjan De Heer	John Messenger	Bob Sultan
Russell Dietz	Gary Michel	Michael Johas Teener
Sourav Dutta	Dinesh Mohan	Pat Thaler
Don Fedyk	Jose Morales	Oliver Thorp
Prince Francis	Michael S. Newman	Thomas Tullia
Yukihiro Fujimoto	Nick S. A. Nikjoo	Mark-Rene Uchida
Devon Gayle	Glenn Parsons	Ludwig Winkel
Sergiu Goma	Haim Porat	Derek Woo
C. Guy	Venkatesha Prasad	Ken Young
Stephen Haddock		Oren Yuen

When the IEEE-SA Standards Board approved this standard on 12 June 2008, it had the following membership:

Robert M. Grow, *Chair*

Thomas Prevost, *Vice Chair*

Steve M. Mills, *Past Chair*

Judith Gorman, *Secretary*

Victor Berman	Jim Hughes	Chuck Powers
Richard DeBlasio	Richard H. Hulett	Narayanan Ramachandran
Andy Drozd	Young Kyun Kim	Jon Walter Rosdahl
Mark Epstein	Joseph L. Koepfinger*	Robby Robson
Alexander Gelman	John Kulick	Anne-Marie Sahazizia
William R. Goldbach	David J. Law	Malcolm V. Thaden
Arnold M. Greenspan	Glenn Parsons	Howard L. Wolfman
Kenneth S. Hanus	Ronald C. Petersen	Don Wright

*Member Emeritus

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

Satish K. Aggarwal, *NRC Representative*

Michael H. Kelly, *NIST Representative*

Don Messina

IEEE Standards Program Manager, Document Development

Michael D. Kipness

IEEE Standards Program Manager, Technical Program Development

Contents

1. Overview.....	2
1.1 Scope.....	2
2. Normative references	3
3. Definitions	5
4. Abbreviations.....	7
5. Conformance.....	9
5.2 Conformant components and equipment	9
5.3 Protocol Implementation Conformance Statement (PICS).....	9
5.7 I-component conformance	9
5.8 B-component conformance.....	10
5.11 Backbone Edge Bridge conformance	11
6. Support of the MAC Service in VLANs.....	13
6.6 Internal Sublayer Service.....	13
6.8 Enhanced Internal Sublayer Service	15
6.9 Support of the EISS	17
6.10 Support of the EISS by Provider Instance Ports	17
6.11 Support of the EISS by Customer Backbone Ports.....	20
6.18 Backbone Service Instance Multiplex Entity.....	24
7. Principles of network operation.....	29
7.4 Locating end stations	29
8. Principles of bridge operation.....	31
8.8 The Filtering Database.....	31
9. Tagged frame format	33
9.5 Tag Protocol Identification	33
9.8 Backbone Service Instance Tag Control Information.....	34
12. Bridge management	37
12.2 ~~Managed objects~~ VLAN-aware bridge objects.....	37
12.3 Data types	37
12.7 Filtering Database.....	37
12.8 Bridge Protocol Entity	38
12.14 CFM entities	39
12.15 Backbone Core Bridge management	43
12.16 Backbone Edge Bridge management.....	43
13. The Multiple Spanning Tree Protocol (MSTP)	59
13.12 Port Role assignments.....	59
13.19 Multiple Spanning Tree state machines.....	59
13.21 State machine timers.....	61
13.25 State machine conditions and parameters.....	61

13.26	State machine procedures	62
13.27	The Port Timers state machine	63
13.28	Port Receive state machine	63
13.31	Port Transmit state machine	63
13.32	Port Information state machine	64
13.37	Port Receive Pseudo Information state machine	65
13.40	Virtual Instance Port Spanning Tree operation.....	67
18.	Principles of Connectivity Fault Management operation	69
19.	CFM Entity operation	71
19.4	Maintenance Point addressing	71
20.	Connectivity Fault Management protocols.....	73
25.	Support of the MAC Service by Provider Backbone Bridged Networks	75
25.1	Service transparency	77
25.2	Customer service interface.....	77
25.3	Port-based service interface	77
25.4	S-tagged service interface.....	79
25.5	I-tagged service interface	82
25.6	Service instance segregation	84
25.7	Service instance selection and identification	84
25.8	Service priority and drop eligibility selection.....	84
25.9	Service access protection	85
26.	Principles of Provider Backbone Bridged Network operation	89
26.1	Provider Backbone Bridged Network overview	89
26.2	Provider Backbone Bridged Network example	90
26.3	Backbone VLAN connectivity.....	92
26.4	Backbone addressing	93
26.5	Detection of connectivity loops through attached networks.....	95
26.6	Scaling of Provider Backbone Bridges	96
26.7	Network Management.....	96
26.8	Connectivity Fault Management in Provider Backbone Bridges	97
Annex A	(normative) PICS proforma.....	105
Annex B	(informative) Shared and independent VLAN learning	109

Figures

Figure 6-2	I-Component Provider Instance Port.....	18
Figure 6-3	B-Component Customer Backbone Port.....	21
Figure 6-6	Two back-to-back Backbone Service Instance Multiplex Entities	24
Figure 6-7	Backbone Service Instance Multiplex Entities with example CFM shims	25
Figure 9-3	I-TAG TCI Format.....	34
Figure 12-1	Relationship among BEB managed objects	44
Figure 13-9	MSTP state machines—overview and relationships.....	60
Figure 13-11	Port Timers state machine.....	63
Figure 13-13	Port Transmit state machine.....	64
Figure 13-14	Port Information state machine	65
Figure 13-23	Port Receive Pseudo Information state machine.....	66
Figure 25-1	Internal organization of the MAC sublayer in a PBBN	75
Figure 25-2	Provider Backbone Bridge terminology	76
Figure 25-3	Customer service interface types	77
Figure 25-4	Port-based service interface	78
Figure 25-5	Port-based interface equipment.....	79
Figure 25-6	Encapsulated service frames at ISS	80
Figure 25-7	S-tagged service interface	81
Figure 25-8	S-tagged service interface equipment	81
Figure 25-9	I-tagged service interface	83
Figure 25-10	I-tagged service interface equipment	83
Figure 25-11	S-tagged and Port-based service interface access classifications	86
Figure 25-12	I-tagged service interface access protection classifications	87
Figure 26-1	PBBN example.....	91
Figure 26-2	CFM Shim Model	98
Figure 26-3	CFM example applied to a Port-based and S-tagged Service Interface.....	100
Figure 26-4	CFM example applied to an I-tagged Service Interface	101
Figure 26-5	CFM example applied to a hierarchal E-NNI, CBP-PIP Demarc.....	103
Figure 26-6	CFM example applied to a peer E-NNI, CBP-PIP.....	104

Tables

Table 6-7 Encapsulated Addresses type..... 25

Table 9-1 IEEE 802.1Q Ethernet Type allocations..... 34

Table 9-3 Reserved I-SID values 35

Table 26-1 Backbone Service Instance Group address OUI..... 93

IEEE Standard for Local and metropolitan area networks—

Virtual Bridged Local Area Networks

Amendment 7: Provider Backbone Bridges

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

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

[This amendment is based on IEEE Std 802.1Q™-2005.]

Editorial Note

This amendment specifies changes to IEEE Std 802.1Q-2005, as amended by IEEE Std 802.1ad™-2005, IEEE Std 802.1ak™-2007 and IEEE Std 802.1ag™-2007, that enable a provider to scale a Provider Bridged Network by interconnecting many Provider Bridged Networks over a Provider Backbone Bridged Network. The resulting backbone network can support interconnection of up to 2^{24} equivalent of separate LANs, Bridged Local Area Networks, or Virtual Bridged Local Area Networks. Changes are applied to the base text of IEEE Std 802.1Q-2005 as amended by IEEE Std 802.1ad-2005, IEEE Std 802.1ak-2007, and IEEE Std 802.1ag-2007. Text shown in bold italics in this amendment defines the editing instructions for changes to this base text. Three editing instructions are used: **change**, **delete**, and **insert**. **Change** is used to make a change to existing material. The editing instruction specifies the location of the change and describes what is being changed. Changes to existing text may be clarified using ~~strikeout~~ markings to indicate removal of old material, and underline markings to indicate addition of new material). **Delete** removes existing material. **Insert** adds new material without changing the existing material. Insertions may require renumbering. If so, renumbering instructions are given in the editing instruction. Editorial notes will not be carried over into future editions of IEEE Std 802.1Q.

1. Overview

1.1 Scope

Insert the following text and bullets immediately after bullet z:

To allow scaling of Provider Networks to at least 2^{24} Service Virtual LANs, this standard further specifies the operation of Provider Backbone Bridges (PBBs) by means of an architecture and bridge protocols compatible and interoperable with Provider Bridged Network protocols and equipment, allowing interconnection of multiple Provider Bridged Networks. To this end, it

- aa) Introduces Backbone Edge Bridges that, by exchanging backbone frames that encapsulate the addresses, VLAN tags, and data of customer frames, support the virtual, media independent, equivalent of a number of independent instances of the service provided by media dependent frame transmission procedures.
- ab) Extends the parameters of the Internal Sublayer Service (ISS) and Enhanced Internal Sublayer Service (EISS) to include a connection identifier, capable of referencing the backbone addresses and other parameters, used to convey customer frames from one Backbone Edge Bridge (BEB) to all, or one of, the other BEBs supporting a particular backbone service instance.
- ac) Specifies the format of the Backbone Service Instance tag (I-TAG) that encapsulates the customer addresses, and introduces a Backbone Service Instance Identifier (I-SID) that allows each BEB to support a number of backbone service instances and permits the unambiguous identification of up to 2^{24} backbone service instances within a single Provider Backbone Bridged Network (PBBN).
- ad) Provides a model of Backbone Edge Bridge operation in terms of VLAN-aware bridge components that allows the use of Provider Bridges as Backbone Core Bridges, with PBBN traffic carried as frames containing I-TAGs on particular Backbone VLANs (B-VLANs) potentially coexisting with PBN traffic carried as frames without I-TAGs on other Backbone VLANs.
- ae) Specifies the interfaces that a Provider Backbone Bridged Network can provide to transport service frames. These comprise a Port-based service interface that assigns all received untagged and priority-tagged frames to a single S-VLAN transported over a single backbone service instance, an S-tagged service interface capable of mapping individual S-VLANs to different backbone service instances, and an I-tagged service interface capable of mapping frames from one set of backbone service instances to another.
- af) Describes the use of redundant bridges and access LANs to protect backbone service access against failure of any of those systems or components.
- ag) Specifies the management of Backbone Edge Bridges in terms of the model of operation [item ad) above], making use of defined management objects for the individual VLAN-aware bridge components, and adding managed objects to facilitate service creation.
- ah) Describes the use of connectivity fault management (CFM) to detect and isolate faults in the connectivity provided to individual S-VLANs across the PBBN, in the connectivity provided to the group of S-VLANs supported by a single backbone service instance (identified by an I-SID), and in the connectivity provided to individual B-VLANs within the backbone itself.
- ai) Specifies extensions to the Multiple Spanning Tree Protocol (MSTP) to allow network administrators to protect against loops through peered PBBNs without requiring coupling of spanning trees that operate independently for each PBBN.

2. Normative references

Insert the following references at the appropriate point:

IETF RFC 3411 (December 2002), *An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks*, Harrington, D., Presuhn, R., and Wijnen, B.¹

Metro Ethernet Forum Technical Specification MEF 4, Metro Ethernet Network Architecture Framework—Part 1: Generic Framework, May 2004.²

¹IETF documents are available at <http://www.ietf.org>.

²MEF publications are available at <http://www.metroethernetforum.org/TechSpec.htm>.

3. Definitions

Insert the following definitions into Clause 3, in appropriate collating sequence, re-numbering existing/new definitions as appropriate:

3.1 B-component (B-comp): An S-VLAN component with one or more Customer Backbone Ports (CBPs).

3.2 Backbone Core Bridge (BCB): An S-VLAN Bridge used within the core of a Provider Backbone Bridged Network.

3.3 Backbone Edge Bridge (BEB): A system that encapsulates customer frames for transmission across a Provider Backbone Bridged Network.

3.4 backbone MAC address (B-MAC): An individual MAC address associated with a Provider Instance Port and used in creating the MAC header of I-tagged frames transmitted across a Provider Backbone Bridged Network.

3.5 Backbone MAC Frame: A LAN frame addressed with backbone MAC addresses.

3.6 backbone service instance: An instance of the MAC service in a Provider Backbone Bridged Network, provided between two or more Virtual Instance Ports in Backbone Edge Bridges.

3.7 Backbone Service Instance Identifier (I-SID): A field of the Backbone Service Instance tag that identifies the backbone service instance of a frame.

3.8 Backbone Service Instance Drop Eligibility Indicator (I-DEI): A field of the Backbone Service Instance tag that indicates the drop eligibility of a frame in a backbone service instance.

3.9 Backbone Service Instance priority code point (I-PCP): A field of the Backbone Service Instance tag that indicates the priority of a frame in a backbone service instance.

3.10 Backbone Service Instance tag (I-TAG): A tag with a Ethertype value allocated for “IEEE 802.1Q Backbone Service Instance tag type”.

3.11 Backbone VLAN (B-VLAN): A VLAN identified by a Backbone VLAN ID.

3.12 Backbone VLAN drop eligible indicator (B-DEI): A field of a B-TAG that identifies the backbone drop eligibility of the frame.

3.13 Backbone VLAN ID (B-VID): A VLAN identifier conveyed in a B-TAG.

3.14 Backbone VLAN priority code point (B-PCP): A field of a B-TAG that indicates the Backbone VLAN priority of the frame.

3.15 Backbone VLAN tag (B-TAG): An S-TAG used in conjunction with backbone MAC addresses.

3.16 Backbone VLAN tagged frames: Frames that contain a B-TAG immediately following the Source MAC address field.

3.17 Bridge Port: A port supporting the MAC relay entity of a bridge component.

3.18 Customer Backbone Port (CBP): A Backbone Edge Bridge Port that can receive and transmit I-tagged frames for multiple customers, and can assign B-VIDs and translate I-SID on the basis of the received I-SID.

3.19 Customer MAC Address (C-MAC): A MAC address within a Customer Bridged Network (CBN) or Provider Bridged Network. Customer MAC Addresses are carried within an I-TAG in a Provider Backbone Bridged Network.

3.20 I-component (I-comp): An S-VLAN component with one or more Provider Instance Ports (PIPs).

3.21 I-tagged frame: A frame that contains a Backbone Service Instance tag immediately following the Source MAC address field.

3.22 Port: A service access point and the interface stack supporting that service access point.

3.23 Provider Backbone Bridge (PBB): A Backbone Core Bridge or a Backbone Edge Bridge.

3.24 Provider Backbone Bridged Network (PBBN): A network using Backbone Edge Bridges and Backbone Core Bridges to interconnect Provider Bridged Networks and other networks.

3.25 Provider Instance Port (PIP): The set of Virtual Instance Ports that are supported by a single instance of the Internal Sublayer Service (ISS).

3.26 service frame: A LAN frame exchanged between a provider and a customer.

3.27 Virtual Instance Port (VIP): A Bridge Port on an I-component in a Backbone Edge Bridge that provides access to a single backbone service instance.

4. Abbreviations

Insert the following abbreviations, in the appropriate collating sequence:

B-Comp	B-component
B-DEI	Backbone VLAN Drop Eligible Indicator
B-DA	Backbone Destination MAC address
B-MAC	backbone MAC address
B-SA	Backbone Source MAC address
B-PCP	Backbone VLAN Priority Code Point
B-TAG	Backbone VLAN tag
B-VID	Backbone VLAN ID
B-VLAN	Backbone VLAN
BCB	Backbone Core Bridge
BEB	Backbone Edge Bridge
C-DA	Customer Destination MAC address
C-MAC	Customer MAC Address
C-SA	Customer Source MAC address
CBP	Customer Backbone Port
CB	Customer Bridge
CBN	Customer Bridged Network
I-Comp	I-component
I-DEI	Backbone Service Instance Drop Eligible Indicator
I-PCP	Backbone Service Instance priority code point
I-TAG	Backbone Service Instance tag
I-SID	Backbone Service Instance Identifier
PB	Provider Bridge
PBN	Provider Bridged Network
PIP	Provider Instance Port
PBB	Provider Backbone Bridge
PBBN	Provider Backbone Bridged Network
VIP	Virtual Instance Port

5. Conformance

5.2 Conformant components and equipment

Change the line following item a) as follows:

for the following ~~two~~ components that use that core functionality:

Insert the following two components after the list item c) renumbering d)–f):

- d) I-component (5.7);
- e) B-component (5.8);

Change the line following the list as follows:

and for the following systems that include instances of ~~either or both of the above two components~~

Insert the following system after the renumbered list item h):

- i) Backbone Edge Bridge (5.11).

5.3 Protocol Implementation Conformance Statement (PICS)

Change the first paragraph as follows:

A claim of conformance specifies implementation of a C-VLAN component, an S-VLAN component, an I-component, a B-component, or a specific system. A component or system can support multiple claims for a range of possible behaviors.

Insert NOTE 3 following NOTE 2 as follows:

NOTE 3—I-component and B-component PICS both require completion of a PICS for an S-VLAN component.

Insert the following new subclauses 5.7 and 5.8 and renumber the existing subclauses 5.7 VLAN Bridge conformance and 5.8 Provider Bridge conformance as subclauses 5.9 and 5.10.

5.7 I-component conformance

An I-component comprises an S-VLAN component (5.6) with the EISS on each Customer Network Port supported by the use of a Service VLAN tag (6.9, 9.5), and the EISS for each Virtual Instance Port configured on a Provider Instance Port supported by the use of both a Service VLAN tag (6.9, 9.5) and a Backbone Service Instance tag (I-TAG) (6.10, 9.5).

A conformant implementation of an I-component shall

- a) Comprise a single conformant S-VLAN component;
- b) Recognize and use Backbone Service Instance tags (I-TAGs) on one or more Provider Instance Ports (6.10);
- c) Support 1:1 mapping between S-VID values and I-SID values;
- d) Support the connection_identifier parameter on the EISS of each Virtual Instance Port (6.10) and in the Filtering Database (8.8.11);

- e) Support the termination of PBN spanning trees by inhibiting transmission of PBN BPDUs at a Provider Instance Port.

5.7.1 I-component options

A conformant I-component may implement any of the options specified for an S-VLAN component (5.6.1), and may

- a) Support many-to-one mapping from S-VID values to I-SID values;
- b) Support a common spanning tree among PBNs interconnected with one or more backbone service instances by permitting transmission of PBN BPDUs through Virtual Instance Ports at a Provider Instance Port;
- c) Filter frames received at Provider Instance Ports that have the B-SA used by the Provider Instance Port, to prevent such frames from circulating if backbone service instances in two or more attached PBBNs are configured in a loop;
- d) Use back-to-back backbone service instance multiplex entities (6.18) on a PIP to enable monitoring of backbone service instances with Connectivity Fault Management (CFM).

5.8 B-component conformance

A B-component comprises an S-VLAN component with the EISS on each Provider Network Port supported by the use of a Service VLAN tag (6.9, 9.5), and the EISS on each Customer Backbone Port supported by the use of both a Service VLAN tag (6.9, 9.5) and a Backbone Service Instance tag (6.11, 9.5). A conformant implementation of a B-component shall

- a) Comprise a single conformant S-VLAN component;
- b) Recognize and use Backbone Service Instance tags (I-TAGs) on one or more Customer Backbone Ports (6.11);
- c) Terminate PBBN spanning tree by inhibiting transmission of PBBN BPDUs at a Customer Backbone Port.

5.8.1 B-component options

A conformant B-component may implement any of the options specified for an S-VLAN component (5.6.1), and may

- a) Translate I-SID values by supporting the Local-SID field in the Backbone Service Instance table in Customer Backbone Ports (6.11);
- b) Assign B-VID values based on I-SID values by supporting the B-VID field in the Backbone Service Instance table in Customer Backbone Ports (6.11);
- c) Use the Default Backbone Destination field in the Backbone Service Instance table in Customer Backbone Ports (6.11) as the destination MAC address for any frames received with the Backbone Service Instance group address (26.4);
- d) Use back-to-back Backbone Service Instance multiplex entities (6.18) on a CBP to enable monitoring of backbone service instances with CFM.

Insert the following subclause after renumbered subclause 5.10.

5.11 Backbone Edge Bridge conformance

A Backbone Edge Bridge system shall comprise zero or more I-components and zero or one B-component, but at least one I-component or one B-component, supporting externally accessible ports as specified in Clause 25. Each externally accessible port shall be designated as one of, and may be capable of being configured as any of the following:

- a) A Provider Network Port
- b) A Provider Instance Port
- c) A Customer Backbone Port
- d) A Customer Network Port

6. Support of the MAC Service in VLANs

Change the first paragraph of subclause 6.1.4 as follows:

6.1.4 Service access points, interface stacks, and ports

Each service is provided to a single protocol entity at a service access point (SAP) within a system. A given N-entity can support a number of N-SAPs and use one or more (N – 1)-SAPs. The service access point serves to delineate the boundary between protocol specifications and to specify the externally observable relationship between entities operating those protocols. A service access point is an abstraction, and does not necessarily correspond to any concrete realization within a system, but an N-entity often associates management counters with the SAP and provides status parameters that can be used by the (N + 1)-entity using the SAP. Examples include the MAC_Operational and operPointToPoint-MAC status parameters (6.6.2, 6.6.3). Each service access point has an identifier with a value that is local to the system and uniquely identifies the service access point within the system.

Change subclause 6.6 as follows:

6.6 Internal Sublayer Service

The Internal Sublayer Service (ISS) augments the specification of the MAC Service (ISO/IEC 15802-1) with elements necessary to the performance of the relay function. Within an end station, these additional elements are considered to be either below the MAC Service boundary, and pertinent only to the operation of the service provider; or local matters not forming part of the peer-to-peer nature of the MAC Service. The ISS excludes MAC-specific features and procedures whose operation is confined to an individual LAN.

NOTE 1—No new service primitives are defined. The frame_check_sequence, drop_eligible, service_access_point_identifier, and connection_identifier are ~~is~~ added to the list of parameters associated with the MA_UNITDATA.request and MA_UNITDATA.indication primitives.

6.6.1 Service primitives and parameters

The ISS is specified by two unit-data primitives, an M_UNITDATA.indication and an M_UNITDATA.request, together with the parameters of those primitives. Each M_UNITDATA indication corresponds to the receipt of an error-free MAC frame from a LAN. A data request primitive is invoked to transmit a frame to an individual LAN.

```
M_UNITDATA.indication    (
                           destination_address,
                           source_address,
                           mac_service_data_unit,
                           priority,
                           drop_eligible,
                           frame_check_sequence,
                           service_access_point_identifier,
                           connection_identifier
                           )
```

```
M_UNITDATA.request      (
                           destination_address,
                           source_address,
                           mac_service_data_unit,
```

priority,
drop_eligible,
frame_check_sequence,
service_access_point_identifier,
connection_identifier
)

NOTE 1—Detailed specifications of error conditions in received frames are contained in the relevant MAC standards; for example, FCS errors, length errors, non-integral number of octets.

The **destination_address** parameter is the address of an individual MAC entity or a group of MAC entities. The **source_address** parameter is the individual address of the source MAC entity. The **mac_service_data_unit** parameter is the service user data. The default **priority** value is 0. Values 1 through 7 form an ordered sequence of user_priorities, with 1 being the lowest value and 7 the highest.

The **drop_eligible** parameter provides guidance to the recipient of the service indication or of an indication corresponding to the service request, and takes the values True or False. If drop_eligible is True, the parameters of the indication should be discarded in preference to others with drop_eligible False that result in frames queued with the same traffic class. The default **drop_eligible** value is False.

The **frame_check_sequence** parameter is explicitly provided with the M_UNITDATA.indication so that it can be used in a related M_UNITDATA.request. The parameter comprises the FCS value and sufficient information to determine whether the FCS value can be used. If the frame_check_sequence parameter is provided with an M_UNITDATA.request and the receiving and the transmitting service providers

- a) Use the same algorithm to determine the FCS; and
- b) Apply that algorithm to the same fields of the frame, i.e., the FCS coverage is the same; and
- c) The data that is within the coverage of the FCS remains the same;

the transmitting service provider should use the supplied FCS value (6.3.7, 6.8).

NOTE 2—There are two possibilities for recreating a valid FCS. The first is to generate a new FCS by algorithmically modifying the received FCS, based on knowledge of the FCS algorithm and the transformations that the frame has undergone between reception and transmission. The second is to rely on the normal MAC procedures to recalculate the FCS for the outgoing frame. The former approach can be preferable in terms of its ability to protect against increased levels of undetected frame errors. Annex F of IEEE Std 802.1D discusses these possibilities in more detail. The frame_check_sequence parameter of the Enhanced Internal Sublayer Service (6.8) is able to signal the validity, or otherwise, of the FCS; an unspecified value in this parameter in a data request indicates to the transmitting MAC that the received FCS is no longer valid, and the FCS must therefore be recalculated.

~~The identification of the LAN from which particular frames are received is a local matter and not expressed as a parameter of the service primitive.~~

Unlike the other parameters of the service primitives, the **service_access_point_identifier** and **connection_identifier** are not parameters of the peer to peer service. The values of the **service_access_point_identifier** and **connection_identifier** are purely local to the system within which a given service request or service indication occurs, and are not conveyed to the communicating peer system. The values are opaque to the user of the ISS, and are not manipulated by that service user, thus permitting independent operation of entities within the MAC sublayer as well as extending capabilities by the addition of protocol shims (6.1.5). The values are not conveyed in any external protocol, including management protocols; however, management protocols can convey externally visible data related to the service access point or connection. For example, there is a one-to-one association between **service_access_point_identifier** of a Bridge Port used by the MAC Relay Entity and the port number identifying that Bridge Port in management and control protocols.

The **service_access_point_identifier** parameter always identifies the service access point at which the service indication occurs or the service request is made, in the context of the receiving or transmitting system. In the common case of direct support of the ISS by a specific MAC procedure (6.7) it identifies the attached individual LAN. When a protocol entity in the interface stack either uses or supports multiple service access points, the **service_access_point_identifier** parameter associates the service primitive with the appropriate service access point.

NOTE 3—In previous versions of this standard the **service_access_point_identifier** parameter itself was implicit and was not included in the parameter list. The parameter is made explicit in this version to facilitate the description of new protocol entities that support or use multiple service access points.

The **connection_identifier** can be null, and is ignored by any specific MAC procedures except as explicitly specified in 6.7. The **connection_identifier** is used where this standard specifically provides for efficient support of a single service access point by a number of connections, i.e., by dynamically created connectivity associations between peer entities (6.1.8). For example, a Provider Instance Port (6.10) creates connections with peer Provider Instance Ports, and uses the **connection_identifier** to associate the backbone MAC address of the peer Provider Instance Port with Customer MAC Addresses that can be reached through that peer Provider Instance Port (26.4). Following an M_UNITDATA.indication at a given service access point with a given **source_address** and **connection_identifier**, a subsequent M_UNITDATA.request at the same service access point with that source address as its **destination_address** and with the same **connection_identifier** will result in an M_UNITDATA.indication at the peer entity selected by the **connection_identifier** (in the absence of frame loss or reconfiguration of network components). The value of a **connection_identifier** is significant only at a single service access point. Any protocol entity in the interface stack that does not specify use of the **connection_identifier** assigns the **connection_identifier** value (if any) supplied with a request from the user of the protocol entity (or with an indication from the provider of the service used by the protocol entity) to the **connection_identifier** on associated requests made (or indications generated) by the protocol entity.

NOTE 34—The ISS specification in this standard differs from that in IEEE Std 802.1D as it omits the **frame_type** and **access_priority** parameters. The **frame_type** is not required as the receipt of a frame other than a user data frame does not cause a data indication, nor are such frames transmitted by the media independent bridge functions. The mapping of the ISS to particular access methods specified by this standard includes derivation of the **access_priority** parameter (for those media that require it) from the ISS priority parameter.

Change subclause 6.8 as follows:

6.8 Enhanced Internal Sublayer Service

The Enhanced Internal Sublayer Service (EISS) is derived from the Internal Sublayer Service (ISS, see 6.6) by augmenting that specification with elements necessary to the operation of the tagging and untagging functions of a VLAN-aware Bridge (3.70). Within the attached end station, these elements can be considered to be either below the MAC Service boundary, and pertinent only to the operation of the service provider; or local matters not forming part of the peer-to-peer nature of the MAC Service.

The EISS provides the same service status and point-to-point parameters as the ISS (,).

6.8.1 Service primitives

The unit-data primitives that define this service are

EM_UNITDATA.indication (
 destination_address,
 source_address,
 mac_service_data_unit,

priority,
drop_eligible,
vlan_identifier,
frame_check_sequence,
service_access_point_identifier,
connection_identifier,
canonical_format_indicator,
rif_information (optional)
)

EM_UNITDATA.request
(
destination_address,
source_address,
mac_service_data_unit,
priority,
drop_eligible,
vlan_identifier,
frame_check_sequence,
service_access_point_identifier,
connection_identifier,
canonical_format_indicator,
rif_information (optional)
)

The **destination_address**, **source_address**, **mac_service_data_unit**, **priority**, **drop_eligible**, **service_access_point_identifier**, **connection_identifier**, and **frame_check_sequence** parameters are as defined for the ISS.

NOTE 1—Some of the functions supporting the EISS may result in changes to the mac_service_data_unit or other parameters used to construct a frame. The original FCS associated with a frame is invalidated if there are changes to any fields of the frame, if fields are added or removed, or if bit ordering or other aspects of the frame encoding have changed. An invalid FCS is signaled in the E-ISS by an unspecified value in the frame_check_sequence parameter. This signals the need for the FCS to be regenerated according to the normal procedures for the transmitting MAC. The options for regenerating the FCS under these circumstances are discussed in Annex F of IEEE Std 802.1D.

~~The **drop_eligible** parameter provides guidance to the recipient of the service indication or of an indication corresponding to the service request, and takes the values True or False. If drop_eligible is True the parameters of the indication should be discarded in preference to others with drop_eligible False that result in frames queued with the same traffic class.~~

The **vlan_identifier** parameter carries the VLAN identifier (VID).

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

NOTE 2—The meanings of the terms Canonical format and Non-canonical format are discussed in IEEE Std 802.

The **rif_information** parameter is present if a tag header containing a Routing Information Field (RIF) is present (indication primitive) or requested (request primitive). Its value is equal to the value of the RIF.

6.9 Support of the EISS

6.9.1 Data indications

Delete Note 1 in 6.9.1.

Change bullet b) of 6.9.1 as follows:

- b) The initial octets of the mac_service_data_unit ~~do not~~ contain a valid VLAN tag header (9.3) of the type used to support the EISS (9.5), and
 - 1) The VID value is FFF, reserved in Table 9-2 for future implementation use; or
 - 2) The VID value is 000 (indicating priority-tagged), and the Acceptable Frame Types is *Admit Only VLAN-tagged frames*; or
 - 3) The VID value is in the range 001-FFE, and the Acceptable Frame Types is *Admit Only Untagged and Priority-tagged frames*.

Change bullet g) of 6.9.1 as follows:

- g) The received priority value and the drop_eligible parameter value are the values is the value in the M_UNITDATA.indication and the value of the drop_eligible parameter is false.

6.9.2 Data requests

Change the second paragraph as follows:

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

Insert new subclauses 6.10 and 6.11 after 6.9 as follows and re-number the subsequent subclauses:

6.10 Support of the EISS by Provider Instance Ports

The functions specified in this subclause are used in conjunction with the functions specified in 6.9 when the EISS is supported by a Provider Instance Port on an I-component in a Backbone Edge Bridge (Clause 25). The combination of the functions specified in this subclause and those specified in 6.9 use a single instance of the ISS (6.6, 6.7) and provide one or more instances of the EISS. An instance of the EISS is created for each backbone service instance, represented by the Backbone Service Instance Identifier (I-SID), configured on the Provider Instance Port. Each instance of the EISS is a Bridge Port referred to as a Virtual Instance Port. A Provider Instance Port is the set of all Virtual Instance Ports supported by a single ISS (PIP-ISS). Figure 6-2 shows a representation of a Provider Instance Port with three Virtual Instance Ports.

The EISS of each Virtual Instance Port (VIP-EISS) shall be supported by an instance of the functions of 6.9 using the Service VLAN tag type (9.5). The ISS of all of the Virtual Instance Ports (VIP-ISS) shall be supported by a single instance of the functions specified in this subclause using the Backbone Service Instance tag type (9.5).

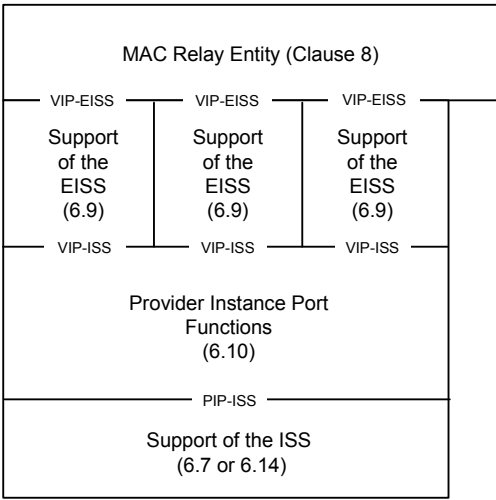


Figure 6-2—I-Component Provider Instance Port

NOTE 1—Higher Layer Entities may interface to any Virtual Instance Port and/or the Provider Instance Port at the service access points represented by any VIP-ISS and/or the PIP-ISS using the Bridge Port Connectivity specified in 8.5.1. Protocol shims such as Connectivity Fault Management Maintenance Points may interface at any VIP-EISS using the EISS Multiplex Entity specified in 6.17, or at the PIP-ISS using the Backbone Service Instance Multiplex Entity specified in 6.18. Figure 26-2 shows a Provider Instance Port with both Bridge Port Connectivity and CFM shims.

Each Provider Instance Port shall have an individual Backbone MAC address referred to as the PIP MAC address (26.4) for use by the functions specified in this subclause.

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

- a) A VIP-ISID parameter for the Backbone Service Instance Identifier of the backbone service instance associated with this Virtual Instance Port;
- b) A Default Backbone Destination;
- c) An adminPointToPointMAC.

The VIP-ISID shall contain a valid I-SID value (9.8), which is configured by management.

The Default Backbone Destination parameter contains a MAC address to be used in the destination_address parameter of a M_UNITDATA.request at the PIP-ISS when a backbone destination address cannot be derived from the connection_identifier parameter of the M_UNITDATA.request from the VIP-ISS. The default value of the Default Backbone Destination is the Backbone Service Instance Group address for the VIP-ISID (generated by concatenating the Provider Backbone Bridge OUI with the VIP-ISID value as specified in 26.4).

The adminPointToPointMAC parameter of the VIP-ISS reflects the point-to-point status of the backbone service instance associated with the VIP. The default value of the adminPointToPointMAC parameter is ForceFalse. The value may be configured by management to ForceTrue when instantiating a VIP for a point-to-point backbone service instance. A value of ForceFalse or Auto results in a operPointToPointMAC value of FALSE; a value of ForceTrue results in a operPointToPointMAC value of TRUE. Whenever the operPointToPointMAC parameter transitions to FALSE, the value for the Default Backbone Destination parameter is set to the Backbone Service Instance Group address for the VIP-ISID. When the operPointToPointMAC parameter is TRUE, the Default Backbone Destination parameter is modified by subsequent M_UNITDATA.indications as specified in 6.10.1 (and described in 26.4.1).

6.10.1 Data indications

On receipt of an M_UNITDATA.indication primitive from the PIP-ISS, the received frame shall be discarded if

- a) The initial octets of the mac_service_data_unit do not contain a valid Backbone Service Instance tag header (I-TAG); or
- b) The Use Customer addresses (UCA) bit in the I-TAG is not zero; or
- c) The Res2 field in the I-TAG is not zero; or
- d) The I-SID value in the I-TAG does not match the VIP-ISID parameter for one of the VIPs supported by this PIP; or
- e) The destination_address parameter of the received M_UNITDATA.indication primitive does not match one of the following:
 - 1) The PIP MAC address; or
 - 2) A Backbone Service Instance Group address; or
 - 3) The broadcast address.

NOTE 2—The generation of I-tagged frames specified in 6.10.2 and the allowed manipulation of backbone addresses specified in 6.11 never result in an I-tagged frame with a broadcast destination address. The broadcast address is included in the previous list on the principle that the broadcast address is a valid address at any reception port by definition.

The received frame may be discarded if

- f) The source_address parameter of the received M_UNITDATA.indication primitive is the PIP MAC address.

Otherwise an M_UNITDATA.indication primitive is invoked at the VIP-ISS with a VIP-ISID value matching the I-SID in the I-TAG. The parameter values of the indication are determined as follows:

The **destination_address** parameter contains the value of the encapsulated C-DA field of the I-TAG.

The **source_address** parameter contains the value of the encapsulated C-SA field of the I-TAG.

The **connection_identifier** parameter references the value of the source_address parameter (B-SA) of the M_UNITDATA.indication primitive received from the PIP-ISS. The connection_identifier parameter in the M_UNITDATA.indication primitive received from the PIP-ISS is ignored.

NOTE 3—The connection_identifier has only local significance meaning that it is used only at a specific PIP. The only requirement imposed by this subclause is that the connection_identifier uniquely identifies an individual backbone MAC address. Whether the connection_identifier takes the value of that address, or a value from which that address can be determined, is an implementation decision that is not externally observable.

If the value of the operPointToPointMAC parameter of the VIP-ISS is TRUE then the Default Backbone Destination parameter is set to the value of the source_address parameter of the M_UNITDATA.indication primitive.

The **service_access_point_identifier** parameter contains a value that uniquely identifies the VIP within the I-component. This is a local value that is used only within the I-component, and is used by the relay functions (Clause 8) as the port number of the VIP. The service_access_point_identifier parameter received in the M_UNITDATA.indication primitive is ignored.

The value of the **drop_eligible** and **priority** parameters are decoded from the I-PCP and I-DEI fields of the I-TAG as described in 6.10.3.

The value of the **mac_service_data_unit** parameter is the value of the received **mac_service_data_unit** following the removal of the I-TAG.

The value of the **frame_check_sequence** parameter is either the received FCS information modified to take into account the changes to the frame, or the unspecified value is used.

6.10.2 Data requests

On invocation of an **M_UNITDATA.request** primitive at any VIP-ISS, a **M_UNITDATA.request** primitive is invoked at the PIP-ISS, with parameter values as follows:

If the **connection_identifier** is not null and the **connection_identifier** references an address retained by the Provider Instance Port, then the value for the **destination_address** is the address referenced by the **connection_identifier**. Otherwise, the value for the **destination_address** is the contents of the Default Backbone Destination parameter of the Virtual Instance Port.

The value for the **source_address** is the PIP MAC address.

The **priority** and **drop_eligible** parameters carry the same value as the corresponding parameter in the VIP-ISS request.

The **connection_identifier** parameter is NULL.

The **mac_service_data_unit** parameter is constructed from the **mac_service_data_unit** of the **M_UNITDATA.request** primitive from the VIP-ISS by prepending an I-TAG as follows:

- a) The encapsulated C-SA field is the value of the **source_address** parameter of the VIP-ISS request;
- b) The encapsulated C-DA field is the value of the **destination_address** parameter of the VIP-ISS request;
- c) The UCA, Res1, and Res2 fields are all zero;
- d) The I-SID field is the value contained in the VIP-ISID parameter;
- e) The I-PCP and I-DEI fields are encoded from the **priority** and **drop_eligible** parameters of the VIP-ISS request as specified in 6.10.3.

The value of the **frame_check_sequence** parameter is either the received FCS information modified to take into account the changes to the frame, or the unspecified value is used.

6.10.3 Priority Code Point encoding

The encoding (and decoding) of the I-PCP and I-DEI fields from (to) the **priority** and **drop_eligible** parameters is accomplished using a Priority Code Point Encoding Table (and Priority Code Point Decoding Table) as described in 6.9.3.

6.11 Support of the EISS by Customer Backbone Ports

The functions specified in this subclause replace the functions specified in 6.9 when the EISS is supported by a Customer Backbone Port in a Backbone Edge Bridge (Clause 25). The functions specified in this subclause use a single instance of the ISS (6.6, 6.7) and provide a single instance of the EISS. The EISS shall be supported by functions using both the Service VLAN tag type (S-TAG) and the Backbone Service I-TAG as specified in 9.5.

NOTE 1—Although the Customer Backbone Port does not transmit or receive S-tagged frames, it supports the Service VLAN tag type in the sense that it is a port on an S-VLAN component and the **vlan_identifier** parameter contains a Service VLAN Identifier.

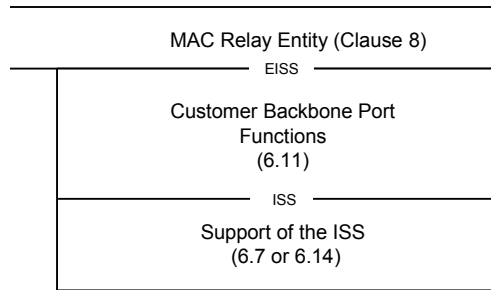


Figure 6-3—B-Component Customer Backbone Port

NOTE 2—Higher Layer Entities may interface to the Customer Backbone Port at the service access point represented by the ISS using the Bridge Port Connectivity specified in 8.5.1. Protocol shims such as Connectivity Fault Management Maintenance Points may interface at the EISS using the EISS Multiplex Entity specified in 6.17, or at the ISS using the Backbone Service Instance Multiplex Entity specified in 6.18. Figure 26-2 shows a Customer Backbone Port with both Bridge Port Connectivity and CFM shims.

The Customer Backbone Port shall support the following parameters for use by these functions:

- a) A PVID parameter for port-based B-VLAN classification;
- b) A Backbone Service Instance table.

The PVID shall contain a valid VID value (Table 9-2) and may be configured by management. If it has not been explicitly configured, the PVID shall assume the value of the default PVID defined in Table 9-2.

The Backbone Service Instance table is configurable by management and has one entry for each backbone service instance supported on the Customer Backbone Port. The Backbone Service Instance table is used to filter frames containing an I-SID value that is not configured on this Customer Backbone Port. It may also be used to provide backbone service instance specific assignment of a B-VID, to translate I-SID values, and/or to translate backbone destination addresses. The Backbone Service Instance table shall contain the following field for each entry:

- c) A Backbone Service Instance Identifier (Backbone-SID). This contains the identifier used in the backbone network for the backbone service instance.

The Backbone Service Instance table may also contain one or more of the following additional fields for each entry:

- d) A Backbone VLAN Identifier (B-VID). If the B-VID field is supported, it shall contain a valid VID value (Table 9-2) for that backbone service instance. The default value is the same as the PVID.
- e) A Local Service Instance Identifier (Local-SID). This field is used to perform a bidirectional 1:1 mapping between the Backbone-SID (contained in the I-TAG in the mac_service_data_unit at the EISS) and the Local-SID (contained in the I-TAG in the mac_service_data_unit at the ISS). The default value of the Local-SID field is the value of the Backbone-SID field.
- f) A Default Backbone Destination. If this field is present, it is used as the destination_address in the EISS indication when the destination_address parameter of the ISS indication is the Backbone Service Instance Group address. It allows the backbone provider to replace the Backbone Service Instance Group address with an individual or group address to limit the distribution of the frame within a Backbone VLAN. The default value is the Backbone Service Instance Group address.

6.11.1 Data indications

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

- a) The initial octets of the mac_service_data_unit do not contain a valid I-TAG; or
- b) The Res2 field in the I-TAG is not zero; or
- c) The Local-SID field of the Backbone Service Instance table is supported and the I-SID value in the I-TAG does not match the Local-SID value in any entry in the Backbone Service Instance table; or
- d) The Local-SID field of the Backbone Service Instance table is not supported and the I-SID value in the I-TAG does not match the Backbone-SID value in any entry in the Backbone Service Instance table.

Otherwise, an EM_UNITDATA.indication primitive is invoked at the EISS with parameter values determined using one entry in the Backbone Service Instance table. If the Local-SID field of the Backbone Service Instance table is supported then the entry with a Local-SID value that matches the I-SID value in the I-TAG is selected. Otherwise, the entry with a Backbone-SID value that matches the I-SID value in the I-TAG is selected. The parameter values are then determined as follows:

The **mac_service_data_unit** parameter is determined as follows:

- e) If the Local-SID field of the Backbone Service Instance table is supported, then value of the **mac_service_data_unit** parameter is the value of the mac_service_data_unit parameter of the ISS indication modified by replacing the value of the I-SID field in the I-TAG with the value of Backbone-SID field in the selected Backbone Service Instance table entry;
- f) Otherwise, the **mac_service_data_unit** parameter contains the same value as the mac_service_data_unit parameter of the ISS indication.

The **destination_address** parameter is determined as follows:

- g) If the destination_address in the ISS indication is the Backbone Service Instance Group address and the Default Backbone Destination field of the Backbone Service Instance table is supported, then the **destination_address** parameter is the Default Backbone Destination value in the selected Backbone Service Instance table entry;
- h) Otherwise, if the destination_address in the ISS indication is the Backbone Service Instance Group address and the Local-SID field of the Backbone Service Instance table is supported, then the **destination_address** parameter is the Backbone Service Instance Group address constructed using the Backbone-SID value in the selected Backbone Service Instance table entry;
- i) Otherwise, the **destination_address** parameter contains the same value as the destination_address parameter of the ISS indication.

The **source_address** parameter contains the same value as the source_address parameter of the ISS indication.

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

- j) If the B-VID field of the Backbone Service Instance table is supported, then the **vlan_identifier** parameter contains the B-VID value in the selected Backbone Service Instance table entry;
- k) Otherwise, the **vlan_identifier** parameter contains the value of the PVID parameter.

The value of the **drop_eligible** and **priority** parameters are determined as follows:

- l) The value of the **drop_eligible** parameter and the received_priority value are decoded from the I-DEI and the I-PCP fields of the I-TAG in the mac_service_data_unit as described in 6.11.3;
- m) The value of the **priority** parameter is then regenerated from the received_priority value, as specified for the received_priority in 6.11.4.

The value of the **frame_check_sequence** parameter is either the received FCS information modified as necessary to take into account changes to the frame, or the unspecified value is used.

6.11.2 Data requests

On invocation of an EM_UNITDATA.request primitive by a user of the EISS, the frame shall be discarded if

- a) The initial octets of the mac_service_data_unit do not contain a valid I-TAG;
- b) The Res2 field in the I-TAG is not zero;
- c) The I-SID value in the I-TAG does not match the Backbone-SID value in any of the entries in the Backbone Service Instance table.

Otherwise, an M_UNITDATA.request primitive is invoked, with parameter values determined using the Backbone Service Instance table entry with a Backbone-SID value that matches the I-SID value in the I-TAG. The parameter values are determined as follows:

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

- d) If the Local-SID field of the Backbone Service Instance table is supported, then **mac_service_data_unit** parameter contains the value of the mac_service_data_unit parameter of the EISS request modified by replacing the I-SID in the I-TAG with the Local-SID value in the selected Backbone Service Instance table entry;
- e) Otherwise, the **mac_service_data_unit** parameter contains the same value as the mac_service_data_unit parameter of the EISS request.

The **destination_address** parameter is determined as follows:

- f) If the Local-SID field of the Backbone Service Instance table is supported and the destination_address in the EISS request is the address of this Customer Backbone Port, or is a group address, then the **destination_address** parameter is the Backbone Service Instance Group address constructed from the Local-SID value in the selected Backbone Service Instance table entry;
- g) If the Local-SID field of the Backbone Service Instance table is not supported and the destination_address in the EISS request is the address of this Customer Backbone Port, or is a group address, then the **destination_address** parameter is the Backbone Service Instance Group address constructed from the Backbone-SID value in the selected Backbone Service Instance table entry;
- h) Otherwise, the **destination_address** parameter contains the same value as the destination_address parameter of the EISS request.

The **source_address** parameter contains the same value as the source_address parameter of the EISS request.

The **priority** and **drop_eligible** parameters carry the same value as the corresponding parameter in the EISS request.

The value of the **frame_check_sequence** parameter is either the received FCS information modified to take into account any changes to the frame, or the unspecified value is used.

6.11.3 Priority Code Point decoding

The decoding of the I-PCP and I-DEI fields to the **priority** and **drop_eligible** parameters is accomplished using a Priority Code Point Decoding Table as described in 6.9.3.

6.11.4 Regenerating priority

The priority of received frames is regenerated using priority information contained in the I-PCP and the Priority Regeneration table for the reception Port. For each reception Port, the Priority Regeneration table has eight entries, corresponding to the eight possible values of priority (0 through 7). Each entry specifies, for the given value of received_priority, the corresponding regenerated value. The priority decoded from the I-PCP field of the I-TAG is taken to be the received_priority.

NOTE—IEEE 802 LAN technologies signal a maximum of eight priority values. Annex G further explains the use of priority values and how they map to traffic classes.

Table 6-5 specifies default regenerated priority values for each of the eight possible values of the received_priority. These default values shall be used as the initial values of the corresponding entries of the Priority Regeneration table for each Port. The table may be configured as described in 6.9.4.

Insert the following subclause 6.18 (following “EISS Multiplex Entity” from IEEE Std 802.1ag-2007), renumbering subsequent subclauses appropriately:

6.18 Backbone Service Instance Multiplex Entity

The Backbone Service Instance Multiplex Entity allows shims defined for the ISS to be instantiated per backbone service instance at a Service Access Point that supports multiple backbone service instances. Figure 6-6 illustrates two Backbone Service Instance Multiplex Entities placed back-to-back. A set of back-to-back Backbone Service Instance Multiplex Entities may be used at the ISS of a Provider Instance Port or Customer Backbone Port to provide per Backbone Service Instance SAPs that support Connectivity Fault Management (CFM) shims at the Backbone Service Instance MD Level as shown in the figures of 26.8.

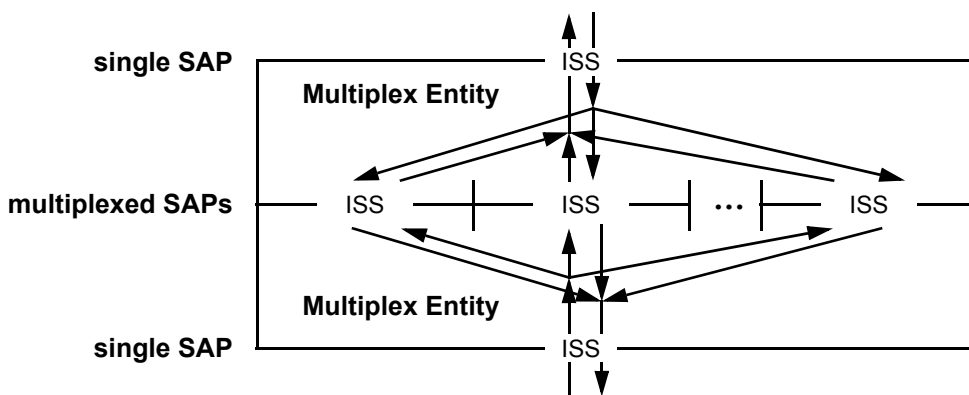


Figure 6-6—Two back-to-back Backbone Service Instance Multiplex Entities

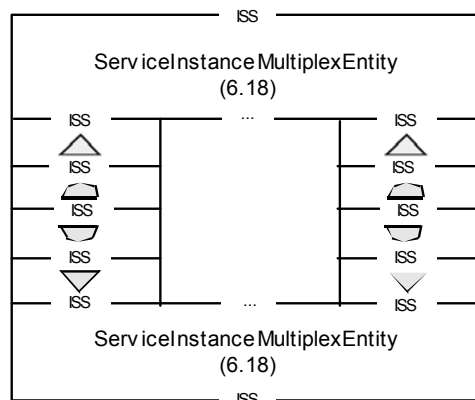


Figure 6-7—Backbone Service Instance Multiplex Entities with example CFM shims

NOTE 1—Figure 6-7 shows a representative placement of CFM shims between back-to-back Backbone Service Instance Multiplex Entities. This is not intended to imply any restriction that this is the only possible configuration of MEPs and MIPs.

A Backbone Service Instance Multiplex Entity has one ISS SAP that supports multiple backbone service instances, and a number of multiplexed ISS SAPs each supporting a single backbone service instance. Each multiplexed SAP has a single assigned I-SID value. Every I-SID value is assigned to a multiplexed SAP, and no I-SID value is assigned to more than one multiplexed SAP.

NOTE 2—There are implicitly 2^{24} multiplexed SAPs—one for each potential I-SID value. This does not require any implementation to support 2^{24} SAPs. The combined structure of back-to-back Backbone Service Instance Multiplex Entities is transparent to any frames with an I-SID value corresponding to a multiplexed SAP that does not have a protocol entity (e.g., CFM) instantiated at that SAP.

In the multiplexing direction, any Request or Indication received from a multiplexed SAP results in the generation of a corresponding Request or Indication to the single ISS SAP, with a Backbone Service Instance tag header (I-TAG) added to the `mac_service_data_unit` as specified in 6.18.2. Similarly, in the demultiplexing direction any Request or Indication received from the single ISS SAP results in the generation of a corresponding Request or Indication at one of the multiplexed SAPs (determined by the I-SID contained in a I-TAG in the `mac_service_data_unit`), with the I-TAG removed from the `mac_service_data_unit` as specified in 6.18.1. When a Backbone Service Instance tag header with the UCA field containing a value of zero is removed, the `C_DA` and `C_SA` fields of the I-TAG remain in the `mac_service_data_unit`. In this case, a new type field is prepended so the `mac_service_data_unit` still begins with a valid Ethertype value. The Encapsulated Addresses type value in Table 6-7 is allocated for this purpose.

Table 6-7—Encapsulated Addresses type

Name	Value
IEEE 802.1Q Encapsulated Addresses type	89-10

NOTE 3—This Ethertype value is used only internally at the Backbone Service Multiplex Entity and does not appear in any frame on a LAN in the PBBN.