INTERNATIONAL STANDARD

ISO/IEC/IEEE 8802-1Q

First edition
2016-03-15

AMENDMENT 6
2019-02

Information technology — Telecommunications and information exchange between systems — Local and metropolitan area networks — Specific requirements —

Part 1Q: Bridges and bridged networks

AMENDMENT 6: Per-stream filtering and policing

Technologies de l'information — Télécommunications et échange d'information entre systèmes — Réseaux locaux et métropolitains — Exigences spécifiques —

Partie 1Q: Ponts et réseaux pontés

AMENDEMENT 6: Régulation & filtrage par flux

Reference number
ISO/IEC/IEEE 8802-1Q:2016/Amd.6:2019(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Fax: +41 22 749 09 47
Email: copyright@iso.org
Website: www.iso.org

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

Email: stds.ipr@ieee.org
Website: www.ieee.org

Published in Switzerland

Foreword

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.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO documents should be noted (see www.iso.org/directives).

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

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

ISO/IEC/IEEE 8802-1Q:2016/Amd.6 was prepared by the LAN/MAN of the IEEE Computer Society (as IEEE Std 802.1Qci-2017) and drafted in accordance with its editorial rules. It was adopted, under the "fast-track procedure" defined in the Partner Standards Development Organization cooperation agreement between ISO and IEEE, by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC6, *Telecommunications and information exchange between systems*.

A list of all parts in the ISO/IEC/IEEE 8802 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

IEEE Std 802.1Qci™-2017
(Amendment to
IEEE Std 802.1Q™-2014
as amended by
IEEE Std 802.1Qca™-2015,
IEEE Std 802.1Qcd™-2015,
IEEE Std 802.1Q-2014/Cor 1-2015,
IEEE Std 802.1Qbv™-2015,
IEEE Std 802.1Qbu™-2016, and
IEEE Std 802.1Qbz™-2016)

IEEE Standard for
Local and metropolitan area networks—

Bridges and Bridged Networks—

Amendment 28: Per-Stream Filtering and Policing

Sponsor

LAN/MAN Standards Committee
of the
IEEE Computer Society

Approved 14 February 2017

IEEE-SA Standards Board

Abstract: Enhancements to the forwarding process that support per-stream filtering and policing are provided in this amendment to IEEE Std 802.1Q-2014.

Keywords: Bridged Local Area Networks, IEEE 802®, IEEE 802.1Q™, IEEE Std 802.1Qbu™, IEEE 802.1Qbv™, IEEE Std 802.1Qbz™, IEEE 802.1Qca™, IEEE 802.1Qcd™, IEEE 802.1Qci™, local area networks (LANs), MAC Bridges, metropolitan area networks, per-stream filtering and policing, PSFP, scheduled traffic, Time-Sensitive Networking, Virtual Bridged Local Area Networks (virtual LANs)

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

Print: ISBN 978-1-5044-4051-6 STD22606
PDF: ISBN 978-1-5044-4052-3 STDPD22606

Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page, appear in all standards and may be found under the heading "Important Notices and Disclaimers Concerning IEEE Standards Documents." They can also be obtained on request from IEEE or viewed at http://standards.ieee.org/IPR/disclaimers.html.

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

IEEE Standards documents (standards, recommended practices, and guides), both full-use and trial-use, are developed within IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association ("IEEE-SA") Standards Board. IEEE ("the Institute") develops its standards through a consensus development process, approved by the American National Standards Institute ("ANSI"), which brings together volunteers representing varied viewpoints and interests to achieve the final product. IEEE Standards are documents developed through scientific, academic, and industry-based technical working groups. Volunteers in IEEE working groups are not necessarily members of the Institute and participate without compensation from IEEE. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

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

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

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

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

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

Translations

The IEEE consensus development process involves the review of documents in English only. In the event that an IEEE standard is translated, only the English version published by IEEE should be considered the approved IEEE standard.

Official statements

A statement, written or oral, that is not processed in accordance with the IEEE-SA Standards Board Operations Manual shall not be considered or inferred to be the official position of IEEE or any of its committees and shall not be considered to be, or be relied upon as, a formal position of IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position of IEEE.

Comments on standards

Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE. However, IEEE does not provide consulting information or advice pertaining to IEEE Standards documents. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Since IEEE standards represent a consensus of concerned interests, it is important that any responses to comments and questions also receive the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to comments or questions except in those cases where the matter has previously been addressed. For the same reason, IEEE does not respond to interpretation requests. Any person who would like to participate in revisions to an IEEE standard is welcome to join the relevant IEEE working group.

Comments on standards should be submitted to the following address:

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

Laws and regulations

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

Copyrights

IEEE draft and approved standards are copyrighted by IEEE under U.S. and international copyright laws. They are made available by IEEE and are adopted for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making these documents available for use and adoption by public authorities and private users, IEEE does not waive any rights in copyright to the documents.

Photocopies

Subject to payment of the appropriate fee, IEEE will grant users a limited, non-exclusive license to photocopy portions of any individual standard for company or organizational internal use or individual, non-commercial use only. To arrange for payment of licensing fees, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Updating of IEEE Standards documents

Users of IEEE Standards documents should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect.

Every IEEE standard is subjected to review at least every ten years. When a document is more than ten years old and has not undergone a revision process, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE standard.

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

Errata

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

Patents

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

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

Participants

At the time of approval of this standard, the IEEE 802.1 Working Group had the following membership:

Glenn Parsons, *Chair*
John Messenger, *Vice Chair*
János Farkas, *Chair, Time-Sensitive Networking Task Group*
Tony Jeffree, *Editor*

SeoYoung Baek
Shenghua Bao
Jens Bierschenk
Steinar Bjørnstad
Christian Boiger
Paul Bottorff
David Chen
Feng Chen
Weiying Cheng
Rodney Cummings
Norman Finn
Geoffrey Garner
Eric W. Gray
Craig Gunther
Marina Gutierrez
Stephen Haddock
Mark Hantel
Patrick Heffernan
Marc Holness
Lu Huang
Michael Johas Teener
Hal Keen
Stephan Kehrer
Philippe Klein
Jouni Korhonen
Yizhou Li
Christophe Mangin
Tom McBeath
James McIntosh
Tero Mustala
Hiroki Nakano
Bob Noseworthy
Donald R. Pannell
Walter Pienciak
Michael Potts
Karen Randall
Maximilian Riegel
Dan Romascanu
Jessy Rouyer
Eero Ryytty
Soheil Samii
Behcet Sarikaya
Frank Schewe
Michael Seaman
Johannes Specht
Wilfried Steiner
Patricia Thaler
Paul Unbehagen
Hao Wang
Karl Weber
Brian Weis
Jordon Woods
Nader Zein
Helge Zinner
Juan Carlos Zuniga

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

Thomas Alexander
Richard Alfvin
Butch Anton
Stefan Aust
Christian Boiger
Nancy Bravin
Ashley Butterworth
William Byrd
Juan Carreon
Yesenia Cevallos
David Chalupsky
Keith Chow
János Farkas
Norman Finn
Michael Fischer
Yukihiro Fujimoto
Joel Goergen
Eric W. Gray
Randall Groves
Craig Gunther
Stephen Haddock
Mark Hantel
Marco Hernandez
Werner Hoelzl
C. Huntley
Noriyuki Ikeuchi
Sergiu Iordanescu
Atsushi Ito
Raj Jain
Tony Jeffree
Michael Johas Teener
Piotr Karocki
Stuart Kerry
Yongbum Kim
John Lemon
Jon Lewis
Arthur H. Light
Michael Lynch
Elvis Maculuba
Roger Marks
Arthur Marris
Jonathon Mclendon
Richard Mellitz
Michael Montemurro
Jeffrey Moore
Charles Moorwood
Matthew Mora
Michael Newman
Nick S. A. Nikjoo
Paul Nikolich
Satoshi Obara
David Olsen
Bansi Patel
Alon Regev
Maximilian Riegel
Robert Robinson
Osman Sakr
Frank Schewe
Michael Seaman
Thomas Starai
Walter Struppler
Mark-Rene Uchida
Prabodh Varshney
George Vlantis
Khurram Waheed
Stephen Webb
Oren Yuen

When the IEEE-SA Standards Board approved this standard on 14 February 2017, it had the following membership:

Jean-Philippe Faure, *Chair*
Vacant position, *Vice Chair*
John D. Kulick, *Past Chair*
Konstantinos Karachalios, *Secretary*

Chuck Adams
Masayuki Ariyoshi
Ted Burse
Stephen Dukes
Doug Edwards
J. Travis Griffith
Gary Hoffman
Michael Janezic
Thomas Koshy
Joseph L. Koepfinger*
Kevin Lu
Daleep Mohla
Damir Novosel
Ronald C. Petersen
Annette D. Reilly
Robby Robson
Dorothy Stanley
Adrian Stephens
Mehmet Ulema
Phil Wennblom
Howard Wolfman
Yu Yuan

*Member Emeritus

Introduction

This introduction is not part of IEEE Std 802.1Qci-2017, IEEE Standard for Local and metropolitan area networks—Bridges and Bridged Networks—Amendment 28: Per-Stream Filtering and Policing.

This amendment to IEEE Std 802.1Q-2014 provides enhancements to the forwarding process that support per-stream filtering and policing.

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

Secretary, IEEE-SA Standards Board
445 Hoes Lane
Piscataway, NJ 08854-4141
USA

Contents

List of figures

List of tables

IEEE Standard for
Local and metropolitan area networks—

Bridges and Bridged Networks—

Amendment 28: Per-Stream Filtering and Policing

(This amendment is based on IEEE Std 802.1Q™-2014 as amended by IEEE Std 802.1Qca™-2015, IEEE Std 802.1Qcd™-2015, IEEE Std 802.1Q-2014/Cor 1-2015, IEEE Std 802.1Qbv™-2015, IEEE Std 802.1Qbu™-2016, and IEEE Std 802.1Qbz™-2016.)

NOTE—The editing instructions contained in this amendment define how to merge the material contained therein into the existing base standard and its amendments to form the comprehensive standard.

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

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

1. Overview

1.3 Introduction

Insert the following item at the end of lettered list, renumbering as necessary:

ch) This standard specifies protocols, procedures, and managed objects that allow for the filtering and policing of individual traffic streams.

2. Normative references

Insert the following references in alphanumeric order:

IEEE Std 802.1CB™, IEEE Standard for Local and Metropolitan Area Networks—Frame Replication and Elimination for Reliability.

MEF Technical Specification 10.3 (MEF 10.3), Ethernet Service Attributes Phase 3.[2]

Delete the reference to MEF 10.3 from the Bibliography [B45] and amend references to it in the text accordingly.

[2]MEF technical specifications are available from the MEF website (http://www.mef.net).

4. Abbreviations

Insert the following abbreviations in alphanumeric order:

IPV	internal priority value specification
PSFP	per-stream filtering and policing

5. Conformance

5.4 VLAN Bridge component requirements

5.4.1 VLAN Bridge component options

Insert new subclause 5.4.1.8 as shown, renumbering as necessary:

5.4.1.8 Per-stream filtering and policing (PSFP) requirements

A VLAN Bridge component implementation that conforms to the provisions of this standard for PSFP shall:

a) Support PSFP as specified in 8.6.5.1 and 8.6.6.1.
b) Support the state machines for stream gate control as specified in 8.6.10.
c) Support the management entities for PSFP as specified in 12.31.

5.13 MAC Bridge component requirements

5.13.1 MAC Bridge component options

Insert new subclause 5.13.1.1 as shown, renumbering as necessary:

5.13.1.1 Per-stream filtering and policing (PSFP) requirements

A MAC Bridge component implementation that conforms to the provisions of this standard for PSFP shall:

a) Support PSFP as specified in 8.6.5.1 and 8.6.6.1.
b) Support the state machines for stream gate control as specified in 8.6.10.
c) Support the management entities for PSFP as specified in 12.31.

Insert the following new subclause at the end of Clause 5, renumbering as necessary:

5.27 End-station requirements—PSFP

An end-station implementation that conforms to the provisions of this standard for PSFP shall:

a) Support PSFP as specified in 8.6.5.1 and 8.6.6.1.
b) Support the state machines for stream gate control as specified in 8.6.10.
c) Support the management entities for PSFP as specified in 12.31.

8. Principles of bridge operation

8.6 The Forwarding Process

Replace Figure 8-11 with the following figure. The effect of this replacement is to change "Ingress (8.6.2)" to "Ingress filtering (8.6.2)" and "Egress (8.6.4)" to "Egress filtering (8.6.4)" to match the subclause headings.

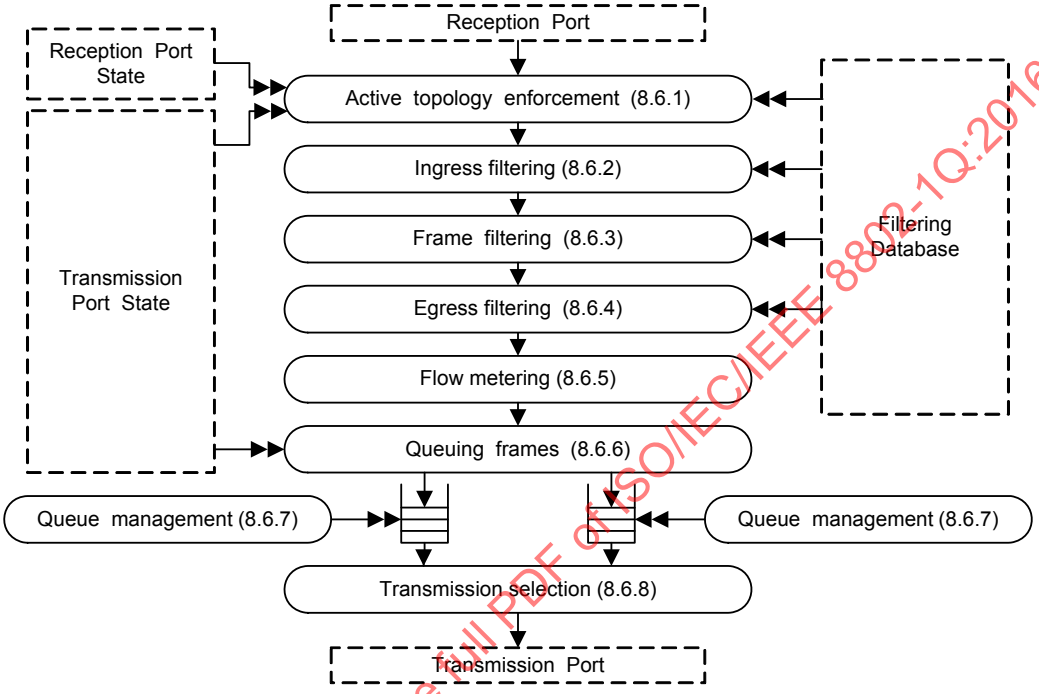

Figure 8-11—Forwarding process functions

8.6.5 Flow classification and metering

Insert new list item e) immediately following list item d), and change the sentence immediately following new item e), renumbering following list items, as shown:

e) connection_identifier

Item c), specifying a VID value, is not applicable to VLAN-unaware MAC Relays. Item e), specifying a connection_identifier, is only applicable to bridges that support PSFP.

Insert new subclause 8.6.5.1, Figure 8-12, and Table 8-5 as shown, renumbering subsequent figures and tables as necessary:

8.6.5.1 Per-stream filtering and policing

A Bridge or an end station may support PSFP capabilities that allow filtering and policing decisions, and subsequent frame queuing decisions (8.6.6.1), to be made on a per-stream basis for received frames.

Support of PSFP requires implementation of the Stream identification function specified in Clause 6 of IEEE Std 802.1CB, as the stream_handle provided by this function is used in the policing and queuing decisions taken by PSFP.

NOTE—The stream_handle specified in Clause 6 of IEEE Std 802.1CB is a sub-parameter of the connection_identifier parameter of the ISS.

PSFP is supported by the following tables:

a) Stream filter instance table (8.6.5.1.1)
b) Stream gate instance table (8.6.5.1.2)
c) Flow meter instance table (8.6.5.1.3)

The relationship between these tables is illustrated in Figure 8-12. The tables and their parameters can be modified by management as specified in 12.31.

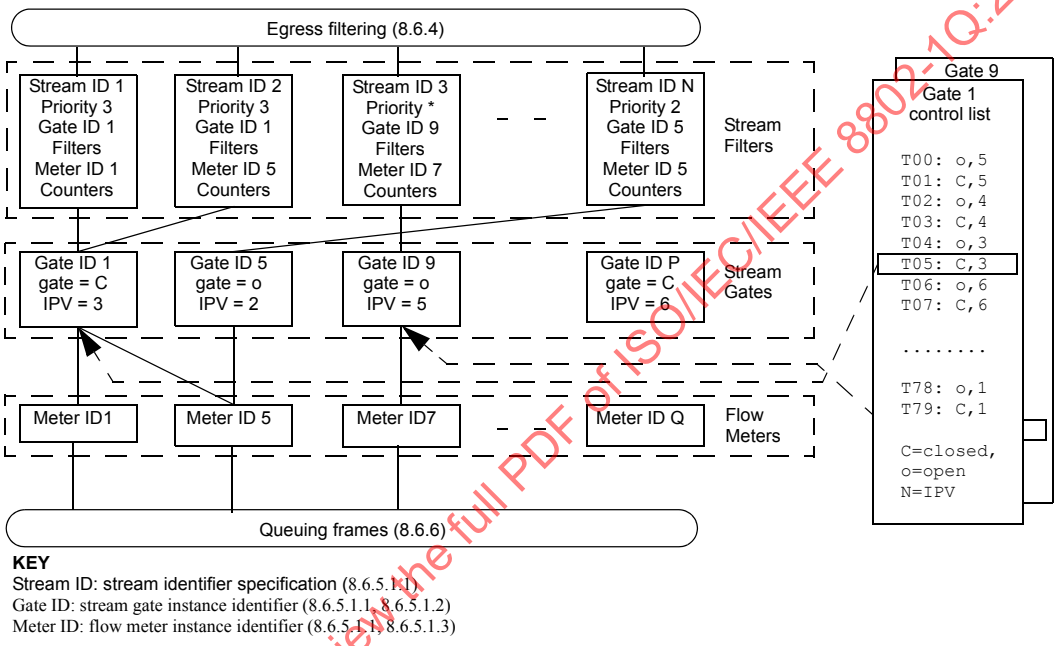

Figure 8-12—Per-stream filtering and policing

8.6.5.1.1 Stream filter instance table

The stream filter instance table consists of an ordered list of *stream filters* that determine the filtering and policing actions that are to be applied to frames received on a specific stream. Each stream filter contains the following elements:

a) A *stream filter instance identifier*. This is an integer value that uniquely identifies the filter instance, and acts as an index to the table. The ordering of the identifier values defines the ordering of the list of stream filters; smaller identifier values appear earlier in the ordered list.
b) A *stream_handle specification*. This can be either of the following:
   1) A single stream_handle value, as specified in IEEE Std 802.1CB.
   2) A wild-card value that matches any stream_handle value.
c) A *priority specification*. This can be either of the following:
   1) A single priority value.
   2) A wild-card value that matches any priority value.

d) A *stream gate instance identifier*. Identifies the stream gate instance (8.6.5.1.2) that is used by the stream filter. A stream gate can be in one of two states:

   1) Open: Frames pass through the gate.

   2) Closed: Frames do not pass through the gate.

e) Zero or more *filter specifications*. The actions specified in a filter specification can result in a frame passing or failing the specified filter. Frames that fail a filter are discarded. The filter specification can include other actions, such as setting the drop_eligible parameter to TRUE. The following filter specifications are currently defined:

   1) *Maximum SDU size*. Frames that exceed this SDU size do not pass the stream filter; frames that do not exceed this SDU size can pass the stream filter if all other filter conditions are met.

NOTE 1—The Maximum SDU size is defined per stream and can therefore differ from the queueMaxSDU specified in 8.6.8.4 of IEEE Std 802.1Qbv. As queueMaxSDU is applied after the stream filters, it is possible that a frame that passes the Maximum SDU size stream filter will later be discarded because its SDU size exceeds queueMaxSDU.

   2) *Flow meter instance identifier*. The identifier of an instance of a flow metering function as specified in 8.6.5. The flow meter instance is an index into a *flow meter instance table* (8.6.5.1.3) that specifies the operating parameters for each flow meter instance. Flow metering is always applied after any other filter specifications that could result in frame discard.

f) Frame counters

   1) A count of frames matching both the stream_handle and priority specifications.

   2) A count of frames that passed the stream gate.

   3) A count of frames that did not pass the stream gate.

   4) A count of frames that passed the Maximum SDU size filter.

   5) A count of frames that did not pass the Maximum SDU size filter.

   6) A count of frames that were discarded as a result of the operation of the flow meter.

g) A *StreamBlockedDueToOversizeFrameEnable* parameter, which takes the value TRUE or FALSE. A value of TRUE indicates that the StreamBlockedDueToOversizeFrame function is enabled; a value of FALSE indicates that the StreamBlockedDueToOversizeFrame function is disabled. The default value of StreamBlockedDueToOversizeFrameEnable is FALSE.

h) A *StreamBlockedDueToOversizeFrame* parameter, which takes the value TRUE or FALSE. If StreamBlockedDueToOversizeFrameEnable is TRUE, a value of TRUE in StreamBlockedDueToOversizeFrame indicates that all frames are to be dropped (i.e., the stream filter behaves as it would if the maximum SDU size were to be set to 0 octets). If StreamBlockedDueToOversizeFrame is FALSE, it has no effect. The default value of StreamBlockedDueToOversizeFrame is FALSE; if any frame is discarded because it exceeds the Maximum SDU size for the stream, then StreamBlockedDueToOversizeFrame is set TRUE.

The value of the *stream_handle* and *priority* parameters associated with a received frame determine which stream filter is selected by the frame, and therefore what combination of filtering and policing actions is applied to the frame. If the stream_handle and priority parameters associated with a received frame match more than one stream filter, the stream filter that is selected is the one that appears earliest in the ordered list. If a received frame's stream_handle and priority does not match any of the stream filters in the table, the frame is processed as would be the case if PSFP was not supported.

NOTE 2—The use of stream identifier and priority, along with the wild-carding rules previously stated, allow configuration possibilities that go beyond PSFP as implied by the subclause title; for example, per-priority filtering and policing, or per-priority per-ingress port filtering and policing can be configured using these rules.

NOTE 3—If it is desired to discard frames that do not match any other stream filter, rather than such frames being processed without filtering, this can be achieved by placing a stream filter at the end of the table, in which the stream_handle and priority are both wild-carded (set to the null value), and where the stream gate instance identifier points at a stream gate that is permanently closed.

8.6.5.1.2 Stream gate instance table

The stream gate instance table contains a set of parameters for each stream gate instance. The parameters for each stream gate instance are as follows:

a) A *Stream gate instance identifier*. An integer value identifying the stream gate instance.

b) An operational and an administrative *stream gate state* (8.6.10.4, 8.6.10.5, 12.31.3). A stream gate can be in one of two states:

   1) Open: Frames are permitted to pass through the gate.

   2) Closed: Frames are not permitted to pass through the gate.

c) An operational and an administrative *internal priority value specification* (*IPV*, 8.6.10.6, 8.6.10.7, 12.31.3). An IPV can be either of the following:

   1) The null value. For a frame that passes through the gate, the priority value associated with the frame is used to determine the frame's traffic class, using the Traffic Class Table as specified in 8.6.6.

   2) An internal priority value. For a frame that passes through the gate, the IPV is used, in place of the priority value associated with the frame, to determine the frame's traffic class, using the Traffic Class Table as specified in 8.6.6.

NOTE 1—A use case for the ability to assign internal priority values can be found in IEEE Std 802.1Qch™ [B1].

d) A *GateClosedDueToInvalidRxEnable* parameter, which takes the value TRUE or FALSE. A value of TRUE indicates that the GateClosedDueToInvalidRx function is enabled; a value of FALSE indicates that the GateClosedDueToInvalidRx function is disabled. The default value of GateClosedDueToInvalidRxEnable is FALSE.

e) A *GateClosedDueToInvalidRx* parameter, which takes the value TRUE or FALSE. If GateClosedDueToInvalidRxEnable is TRUE, a value of TRUE in GateClosedDueToInvalidRx indicates that all frames are dropped (i.e., the gate behaves as if the operational stream gate state is Closed). If GateClosedDueToInvalidRx is FALSE, it has no effect. The default value of GateClosedDueToInvalidRx is FALSE; if any frame is discarded because the gate is in the Closed state, then GateClosedDueToInvalidRx is set TRUE.

NOTE 2—This parameter, in combination with its enable parameter, allow the detection of incoming frames during time periods when the stream gate is in the closed state to result in the stream gate being permanently set to a closed state, until such a time as management action is taken to reset the condition. The intent is to support applications where the transmission and reception of frames across the network is coordinated such that frames are received only when the stream gate is open, and hence, a frame received by the stream gate when it is in the closed state represents an invalid receive condition.

f) A *GateClosedDueToOctetsExceededEnable* parameter, which takes the value TRUE or FALSE. A value of TRUE indicates that the GateClosedDueToOctetsExceeded function is enabled; a value of FALSE indicates that the GateClosedDueToOctetsExceeded function is disabled. The default value of GateClosedDueToOctetsExceededEnable is FALSE.

g) A *GateClosedDueToOctetsExceeded* parameter, which takes the value TRUE or FALSE. If GateClosedDueToOctetsExceededEnable is TRUE, a value of TRUE in GateClosedDueToOctetsExceeded indicates that all frames are dropped (i.e., the gate behaves as if the operational stream gate state is Closed). If GateClosedDueToOctetsExceeded is FALSE, it has no effect. The default value of GateClosedDueToOctetsExceeded is FALSE; if any frame is discarded because there are insufficient IntervalOctetsLeft (8.6.10.8), then GateClosedDueToOctetsExceeded is set TRUE.

h) Optionally, an operational and an administrative *stream gate control list*. If present, these are ordered lists of stream control operations, as specified in Table 8-7. The state machines that control the execution of the operational stream gate control list, along with their variables and procedures, are specified in 8.6.10.

Table 8-7—Stream gate control operations

Operation name	Parameter(s)	Action
SetGateAndIPV	StreamGateState, IPV, TimeInterval, IntervalOctetMax	The StreamGateState parameter specifies a desired state, *open* or *closed*, for the stream gate, and the IPV parameter specifies a desired value of the IPV associated with the stream. On execution, the StreamGateState and IPV parameter values are used to set the operational values of the stream gate state and internal priority specification parameters for the stream. After *TimeInterval* ticks (8.6.9.4.16) has elapsed since the completion of the previous stream gate control operation in the stream gate control list, control passes to the next stream gate control operation. The optional IntervalOctetMax parameter specifies the maximum number of MSDU octets that are permitted to pass the gate during the specified TimeInterval. If the IntervalOctetMax parameter is omitted, there is no limit on the number of octets that can pass the gate.

The administrative values of these parameters are used to determine the initial values of the corresponding operational parameters, and in the case of the administrative stream gate control list parameter, to provide a means of configuring a new control list prior to its installation in a running system.

8.6.5.1.3 Flow meter instance table

The flow meter instance table contains a set of parameters for each flow meter instance. The parameters for each flow meter instance are as specified in *Bandwidth Profile Parameters and Algorithm* in MEF 10.3, plus some additional parameters, as follows:

NOTE—Envelope and Rank, as defined in MEF 10.3, are not used for PSFP; i.e., PSFP uses the reduced functionality algorithm described in 12.2 of MEF 10.3.

a) *Flow meter instance identifier.* An integer value identifying the flow meter instance.

b) *Committed information rate (CIR)*, in bits per second.

c) *Committed burst size (CBS)*, in octets.

d) *Excess Information Rate (EIR)*, in bits per second.

e) *Excess burst size (EBS) per bandwidth profile flow*, in octets.

f) *Coupling flag (CF)*, which takes the value 0 or 1.

g) *Color mode (CM)*, which takes the value *color-blind* or *color-aware*.

h) *DropOnYellow*, which takes the value TRUE or FALSE. A value of TRUE indicates that yellow frames are dropped (i.e., discarded); a value of FALSE indicates that yellow frames will have the drop_eligible parameter set to TRUE.

i) *MarkAllFramesRedEnable,* which takes the value TRUE or FALSE. A value of TRUE indicates that the MarkAllFramesRed function is enabled; a value of FALSE indicates that the MarkAllFramesRed function is disabled. The default value of MarkAllFramesRedEnable is FALSE.

j) *MarkAllFramesRed*, which takes the value TRUE or FALSE. If MarkAllFramesRedEnable is TRUE, a value of TRUE in MarkAllFramesRed indicates that all frames are dropped (i.e., discarded). If MarkAllFramesRed is False, it has no effect. The default value of MarkAllFramesRed is FALSE; if the operation of the flow meter causes any frame to be discarded, then MarkAllFramesRed is set TRUE.

8.6.6 Queuing frames

Insert new subclause 8.6.6.1 as shown:

8.6.6.1 PSFP queuing

If PSFP is supported (8.6.5.1), and the IPV associated with the stream filter that passed the frame is anything other than the null value, then that IPV is used to determine the traffic class of the frame, in place of the frame's priority, via the Traffic Class Table specified in 8.6.6. In all other respects, the queuing actions specified in 8.6.6 are unchanged.

The IPV is used only to determine the traffic class associated with a frame, and hence select an outbound queue; for all other purposes, the received priority is used.

Insert new subclause 8.6.10, its subclauses, and Table 8-8, as shown, renumbering as necessary:

8.6.10 Stream gate control state machines

The execution of the gate operations in a stream gate control list (8.6.5.1.2) is controlled by the three state machines specified in 8.6.9[3]:

a) The Cycle Timer state machine (8.6.9.1);
b) The List Execute state machine (8.6.9.2); and
c) The List Config state machine (8.6.9.3).

One instance of each state machine is instantiated for each stream gate control list associated with instances of stream gates in a Bridge component that supports PSFP. An overview of the operation of these state machines can be found in Figure 8-13.

The operation of these state machines is as defined in 8.6.9, with the exception of the definitions of the ExecuteOperation() procedure, the SetGateStates() procedure, the ListPointer variable, the AdminGateStates variable, and the OperGateStates variable; amended versions of these definitions appear in 8.6.10.1 through 8.6.10.5. Table 8-8 shows the correspondence between the procedures/variables used in 8.6.9 and the PSFP versions of these procedures/variables.

Three additional variables needed by the ExecutePSFPOperation procedure are defined in 8.6.10.6 and 8.6.10.7.

Table 8-8—Scheduled Traffic and PSFP procedures/variables

Procedure/variable name in 8.6.9	PSFP procedure/variable name
ExecuteOperation() (8.6.9.2.1)	ExecutePSFPOperation() (8.6.10.1)
SetGateStates() (8.6.9.2.2)	SetPSFPGateStates() (8.6.10.2)
ListPointer (8.6.9.4.15)	PSFPListPointer (8.6.10.3)
AdminGateStates (8.6.9.4.5)	PSFPAdminGateStates (8.6.10.4)
OperGateStates (8.6.9.4.22)	PSFPOperGateStates (8.6.10.5)

[3]Figure 8-13 and 8.6.9 can be found IEEE Std 802.1Qbv, which is also an amendment to IEEE Std 802.1Q.

8.6.10.1 ExecutePSFPOperation()

The ExecutePSFPOperation() procedure is responsible for fetching the next gate operation from the OperControlList, along with any parameters associated with it, and performing actions based upon the gate operation that has been fetched. The value of the PSFPListPointer variable (8.6.10.3) is used as an index into OperControlList. The procedure processes the operation according to its operation name (Table 8-7) as follows:

a) If the operation name is SetGateAndIPV, then the StreamGateState parameter value associated with the operation is assigned to the PSFPOperGateStates variable (8.6.10.5), the IPV parameter value is assigned to the OperIPV variable (8.6.10.7), and the TimeInterval parameter value associated with the operation is assigned to the TimeInterval variable (8.6.9.4.24). If the TimeInterval parameter value associated with the operation was 0, the TimeInterval variable is assigned the value 1. If there is an IntervalOctetMax parameter associated with the gate operation, then that parameter value is used to set the value of the IntervalOctetsLeft variable (8.6.10.8); otherwise, the IntervalOctetsLeft variable is set to a value greater than the maximum possible number of octets that the gate could pass during TimeInterval.

b) If the operation name is unrecognized, then the PSFPListPointer variable (8.6.9.4.15) is assigned the value of the OperControlListLength variable (8.6.9.4.23) and the TimeInterval variable (8.6.9.4.24) is assigned the value 0.

c) If there is no TimeInterval parameter associated with the operation, then the TimeInterval variable is assigned the value 0.

8.6.10.2 SetPSFPGateStates()

This procedure sets the stream gate state as specified by the value of the PSFPOperGateStates variable (8.6.9.4.22).

8.6.10.3 PSFPListPointer

An integer used as a pointer to entries in the OperControlList (8.6.9.4.19), each entry consisting of a stream gate control operation with its associated parameters (Table 8-7). A value of zero points at the first entry in the list; a value of (OperControlListLength) – 1 points at the last entry.

8.6.10.4 PSFPAdminGateStates

The initial state of the gate associated with the stream gate is set by the List Execute state machine (8.6.9.2) and is determined by the value of the PSFPAdminGateStates variable. The default value of PSFPAdminGateStates is open. The value of PSFPAdminGateStates can be changed by management.

8.6.10.5 PSFPOperGateStates

The current state of the gate associated with the stream gate. PSFPOperGateStates is set by the List Execute state machine (8.6.9.2), and its initial value is determined by the value of the PSFPAdminGateStates variable (8.6.10.4).

8.6.10.6 AdminIPV

The initial value of the OperIPV variable (8.6.10.7) associated with the stream gate is determined by the value of the AdminIPV variable. The default value of AdminIPV variable is the null value. The value of the AdminIPV variable can be changed by management.

8.6.10.7 OperIPV

The current value of the IPV associated with the stream gate. The initial value of OperIPV is set equal to the value of the AdminIPV variable (8.6.10.6). Subsequently, if there is a stream gate control list associated with the stream gate instance, its value is controlled by the contents of the operational stream gate control list and the operation of the List Execute state machine (8.6.9.2).

8.6.10.8 IntervalOctetsLeft

The current value of the IntervalOctetsLeft parameter indicates how many more MSDU octets can be passed by the stream gate during the current TimeInterval. This variable is initialized by the ExecutePSFPOperation() procedure (8.6.10.1). If a frame that would otherwise pass the gate is larger than the current value of IntervalOctetsLeft, it is treated as if the gate is in the *closed* state; i.e., it is discarded. If a frame that would otherwise pass the gate is smaller than the current value of IntervalOctetsLeft, the number of MSDU octets is subtracted from the value of IntervalOctetsLeft.

12. Bridge management

Insert new subclause 12.31, and its subclauses and tables, as shown, renumbering as necessary.

12.31 Managed objects for per-stream filtering and policing

The Bridge enhancements for support of per-stream filtering and policing are defined in 8.6.5.1 and the associated state machines are defined in 8.6.10.

The objects that comprise this managed resource are as follows:

a) The Stream Parameter Table (12.31.1)
b) The Stream Filter Instance Table (12.31.2)
c) The Stream Gate Instance Table (12.31.3)
d) The Flow Meter Instance Table (12.31.4)

12.31.1 The Stream Parameter Table

There is one Stream Parameter Table per Bridge component. The table contains a set of parameters that supports PSFP (8.6.5.1), as detailed in Table 12-30. Tables can be created or removed dynamically in implementations that support dynamic configuration of Bridge components.

Table 12-30—The Stream Parameter Table

Name	Data type	Operations supported[a]	Conformance[b]	References
MaxStreamFilterInstances	integer	R	BE	8.6.5.1, 12.31.2
MaxStreamGateInstances	integer	R	BE	8.6.5.1, 12.31.3
MaxFlowMeterInstances	integer	R	BE	8.6.5.1, 12.31.4
SupportedListMax	integer	R	BE	8.6.5.1, 12.31.4

[a]R= Read only access; RW = Read/Write access.
[b]B = Required for Bridge or Bridge component support of PSFP.
E = Required for end-station support of PSFP.

12.31.1.1 MaxStreamFilterInstances

The maximum number of Stream Filter instances supported by this Bridge component.

12.31.1.2 MaxStreamGateInstances

The maximum number of Stream Gate instances supported by this Bridge component.

12.31.1.3 MaxFlowMeterInstances

The maximum number of Flow Meter instances supported by this Bridge component.

12.31.1.4 SupportedListMax

The maximum value supported by this Bridge component of the AdminControlListLength and OperControlListLength parameters. It is available for use by schedule computation software to determine the Bridge component's control list capacity prior to computation.

12.31.2 The Stream Filter Instance Table

There is one Stream Filter Instance Table per Bridge component. Each table row contains a set of parameters that defines a single Stream Filter (8.6.5.1), as detailed in Table 12-31. The table rows form an ordered list of filter instances, the order being determined by the StreamFilterInstance parameter. Tables can be created or removed dynamically in implementations that support dynamic configuration of Bridge components. Rows in the table can be created or removed dynamically in implementations that support dynamic configuration of stream filters.

Table 12-31—Stream Filter Instance Table

Name	Data type	Operations supported[a]	Conformance[b]	References
StreamFilterInstance	integer	R	BE	8.6.5.1
StreamHandleSpec	stream_handle specification	RW	BE	8.6.5.1
PrioritySpec	priority specification	RW	BE	8.6.5.1
StreamGateInstanceID	integer	RW	BE	8.6.5.1, 8.6.5.1.2
FilterSpecificationList	sequence of FilterSpecification	RW	BE	8.6.5.1, 8.6.5.1.3, 12.31.2.5
MatchingFramesCount	counter	R	BE	8.6.5.1
PassingFramesCount	counter	R	BE	8.6.5.1
NotPassingFramesCount	counter	R	BE	8.6.5.1
PassingSDUCount	counter	R	BE	8.6.5.1
NotPassingSDUCount	counter	R	BE	8.6.5.1
REDFramesCount	counter	R	BE	8.6.5.1
StreamBlockedDueToOversizeFrameEnable	Boolean	RW	BE	8.6.5.1, 8.6.5.1.1
StreamBlockedDueToOversizeFrame	Boolean	RW	BE	8.6.5.1, 8.6.5.1.1

[a]R = Read only access; RW = Read/Write access.
[b]B = Required for Bridge or Bridge component support of PSFP.
E = Required for end-station support of PSFP.

12.31.2.1 StreamFilterInstance

An integer index value that determines the place of the stream filter in the ordered list of stream filter instances. The values of StreamFilterInstance are ordered according to their integer value; smaller values appear earlier in the ordered list.

12.31.2.2 stream_handle specification data type

The stream_handle specification data type allows either of the following to be represented:

a) A stream_handle value, represented as an integer.
b) The wild card value.

12.31.2.3 priority specification data type

The priority specification data type allows either of the following to be represented:

a) A priority value, represented as an integer.
b) The wild card value.

12.31.2.4 StreamGateInstance

The StreamGateInstance parameter identifies the stream gate (12.31.3) that is associated with the stream filter. The relationship between stream filters and stream gates is many to one; a given stream filter can be associated with only one stream gate, but there can be multiple stream filters associated with a given stream gate.

12.31.2.5 FilterSpecification data type

The FilterSpecification data type can represent the following:

a) An integer value representing a Maximum SDU size (8.6.5.1).
b) An integer value representing a flow meter instance identifier (8.6.5.1, 8.6.5.1.3).

12.31.3 The Stream Gate Instance Table

There is one Stream Gate Instance Table per Bridge component. Each table row contains a set of parameters that defines a single Stream Gate (8.6.5.1.2), as detailed in Table 12-32. Tables can be created or removed dynamically in implementations that support dynamic configuration of Bridge components. Rows in the table can be created or removed dynamically in implementations that support dynamic configuration of stream gates.

Table 12-32—The Stream Gate Instance Table

Name	Data type	Operations supported[a]	Conformance[b]	References
StreamGateInstance	integer	R	BE	8.6.5.1, 8.6.5.1.2
PSFPGateEnabled	Boolean	RW	BE	8.6.9.4.14
PSFPAdminGateStates	PSFPgateStatesValue	RW	BE	8.6.10.4, 12.31.3.2.1
PSFPOperGateStates	PSFPgateStatesValue	R	BE	8.6.10.5, 12.31.3.2.1
PSFPAdminControlListLength	unsigned integer	RW	BE	8.6.9.4.6, 12.31.3.2
PSFPOperControlListLength	unsigned integer	R	BE	8.6.9.4.23, 12.31.3.2
PSFPAdminControlList	sequence of PSFP-GateControlEntry	RW	BE	8.6.9.4.2, 12.31.3.2, 12.31.3.2.2
PSFPOperControlList	sequence of PSFP-GateControlEntry	R	BE	8.6.9.4.19, 12.31.3.2, 12.31.3.2.2
PSFPAdminCycleTime	RationalNumber	RW	BE	8.6.9.4.3, 12.29.1.3
PSFPOperCycleTime	RationalNumber (seconds)	R	BE	8.6.9.4.20, 12.29.1.3
PSFPAdminCycleTimeExtension	Integer (nanoseconds)	RW	BE	8.6.9.4.4
PSFPOperCycleTimeExtension	Integer (nanoseconds)	R	BE	8.6.9.4.21
PSFPAdminBaseTime	PTPtime	RW	BE	8.6.9.4.1, 12.29.1.4
PSFPOperBaseTime	PTPtime	R	BE	8.6.9.4.18, 12.29.1.4
PSFPConfigChange	Boolean	RW	BE	8.6.9.4.7
PSFPConfigChangeTime	PTPtime	R	BE	8.6.9.4.9, 12.29.1.4
PSFPTickGranularity	Integer (tenths of nanoseconds)	R	BE	8.6.9.4.16
PSFPCurrentTime	PTPtime	R	BE	8.6.9.4.10, 12.29.1.4
PSFPConfigPending	Boolean	R	BE	8.6.9.3, 8.6.9.4.8
PSFPConfigChangeError	Integer	R	BE	8.6.9.3.1

Table 12-32—The Stream Gate Instance Table *(continued)*

Name	Data type	Operations supported[a]	Conformance[b]	References
PSFPAdminIPV	IPV	RW	BE	8.6.5.1.2, 8.6.10.6, 12.31.3.3
PSFPOperIPV	IPV	RW	BE	8.6.5.1.2, 8.6.10.7, 12.31.3.3
PSFPGateClosedDueToInvalidRx-Enable	Boolean	RW	BE	8.6.5.1.2
PSFPGateClosedDueToInvalidRx	Boolean	RW	BE	8.6.5.1.2
PSFPGateClosedDueToOctets-ExceededEnable	Boolean	RW	BE	8.6.5.1.2
PSFPGateClosedDueToOctetsExceeded	Boolean	RW	BE	8.6.5.1.2

[a]R= Read only access; RW = Read/Write access.
[b]B = Required for Bridge or Bridge component support of PSFP.
E = Required for end-station support of PSFP.

12.31.3.1 StreamGateInstance

An integer table index that allows the stream gate to be referenced from Stream Filter Instance Table entries.

12.31.3.2 The gate control list structure and data types

The AdminControlList and OperControlList are ordered lists containing AdminControlListLength or OperControlListLength entries, respectively. Each entry represents a gate operation as defined in Table 8-7. Each entry in the list is structured as a GateControlEntry (12.31.3.2.2).

12.31.3.2.1 PSFPgateStatesValue

The PSFPgateStatesValue indicates the desired gate state, *open* or *closed*, for the stream gate.

12.31.3.2.2 PSFPGateControlEntry

A PSFPGateControlEntry consists of an operation name, followed by three parameters associated with the operation, as detailed in Table 8-7. The first parameter is a PSFPgateStatesValue (12.31.3.2.1); the second parameter is an IPV value (12.31.3.2.3), and the third parameter is a timeIntervalValue (12.31.3.2.4).

12.31.3.2.3 IPV value

The IPV value indicates the IPV (12.31.3.3) to be associated with frames that pass the gate (8.6.10.7).

12.31.3.2.4 timeIntervalValue

An unsigned integer, denoting a TimeInterval in nanoseconds (see TimeInterval in Table 8-7).

12.31.3.3 The Internal priority value specification (IPV) data type

The IPV data type represents an IPV value (8.6.5.1.2); this is either the null value or an internal priority value.

12.31.4 The Flow Meter Instance Table

There is one Flow Meter Instance Table per Bridge component. Each table row contains a set of parameters that defines a single Flow Meter Instance (8.6.5.1), as detailed in Table 12-33. Tables can be created or removed dynamically in implementations that support dynamic configuration of Bridge components. Rows in the table can be created or removed dynamically in implementations that support dynamic configuration of flow meters.

Table 12-33—The Flow Meter Instance Table

Name	Data type	Operations supported[a]	Conformance[b]	References
FlowMeterInstanceID	integer	R	BE	8.6.5.1, 8.6.5.1.3
CIR	integer, bit/s	RW	BE	8.6.5.1, 8.6.5.1.3
CBS	integer, octets	RW	BE	8.6.5.1, 8.6.5.1.3
EIR	integer, bit/s	RW	BE	8.6.5.1, 8.6.5.1.3
EBS	integer, octets	RW	BE	8.6.5.1, 8.6.5.1.3
CF	integer, 0 or 1	RW	BE	8.6.5.1, 8.6.5.1.3
CM	enumerated, color-blind or color-aware	RW	BE	8.6.5.1, 8.6.5.1.3
DropOnYellow	Boolean	RW	BE	8.6.5.1, 8.6.5.1.3
MarkAllFramesRedEnable	Boolean	RW	BE	8.6.5.1, 8.6.5.1.3
MarkAllFramesRed	Boolean	RW	BE	8.6.5.1, 8.6.5.1.3

[a]R= Read only access; RW = Read/Write access.
[b]B = Required for Bridge or Bridge component support of PSFP.
E = Required for end-station support of PSFP.

17. Management Information Base (MIB)

17.2 Structure of the MIB

Insert new subclause 17.2.24 and Table 17-30 at the end of 17.2, as shown, renumbering as necessary.

17.2.24 Structure of the IEEE8021-PSFP-MIB

The IEEE8021-PSFP-MIB provides for configuration of PSFP (8.6.5, 8.6.5.1, 8.6.10) on reception Ports. Table 17-30 indicates the relationship between the SMIv2 objects defined in the MIB module (17.7.24) and managed objects defined in 12.31.

Table 17-30—IEEE8021-PSFP-MIB Structure and relationship to this standard

MIB table	MIB object	Reference
ieee8021PSFPStreamFillterParameters subtree		
ieee8021PSFPStreamFilterTable		Stream Filter Instance Table, 8.6.5, 8.6.5.1, 12.31.2
	ieee8021PSFPStreamFilterInstance	StreamFilterInstance, 8.6.5, 8.6.5.1, 12.31.2
	ieee8021PSFPStreamHandleSpec	StreamHandleSpec, 8.6.5, 8.6.5.1, 12.31.2
	ieee8021PSFPPrioritySpec	PrioritySpec, 8.6.5, 8.6.5.1, 12.31.2
	ieee8021PSFPStreamGateInstanceID	StreamGateInstanceID, 8.6.5, 8.6.5.1, 12.31.2
	ieee8021PSFPFilterSpecificationList	FilterSpecificationList, 8.6.5, 8.6.5.1, 12.31.2
	ieee8021PSFPMatchingFramesCount	MatchingFramesCount, 8.6.5, 8.6.5.1, 12.31.2
	ieee8021PSFPPassingFramesCount	PassingFramesCount, 8.6.5, 8.6.5.1, 12.31.2
	ieee8021PSFPNotPassingFramesCount	NotPassingFramesCount, 8.6.5, 8.6.5.1, 12.31.2
	ieee8021PSFPPassingSDUCount	PassingSDUCount, 8.6.5, 8.6.5.1, 12.31.2
	ieee8021PSFPNotPassingSDUCount	NotPassingSDUCount, 8.6.5, 8.6.5.1, 12.31.2
	ieee8021PSFPREDFramesCount	REDFramesCount, 8.6.5, 8.6.5.1, 12.31.2
	ieee8021PSFPStreamBlockedDueToOversizeFrameEnable	StreamBlockedDueToOversizeFrameEnable, 8.6.5.1, 8.6.5.1.1, 12.31.2
	ieee8021PSFPStreamBlockedDueToOversizeFrame	StreamBlockedDueToOversizeFrame, 8.6.5.1, 8.6.5.1.1, 12.31.2

Table 17-30—IEEE8021-PSFP-MIB Structure and relationship to this standard *(continued)*

MIB table	MIB object	Reference
ieee8021PSFPStreamGateParameters		
ieee8021PSFPStreamGateTable		Stream Gate Instance Table, 8.6.5, 8.6.5.1, 12.31.3
	ieee8021PSFPStreamGateInstance	StreamGateInstance, 8.6.5, 8.6.5.1, 12.31.3
	ieee8021PSFPGateEnabled	PSFPGateEnabled, 8.6.5, 8.6.5.1, 8.6.10, 12.31.3
	ieee8021PSFPAdminGateStates	PSFPAdminGateStates, 8.6.5, 8.6.5.1, 8.6.10, 12.31.3
	ieee8021PSFPOperGateStates	PSFPOperGateStates, 8.6.5, 8.6.5.1, 8.6.10, 12.31.3
	ieee8021PSFPAdminControlListLength	PSFPAdminControlListLength, 8.6.5, 8.6.5.1, 8.6.10, 12.31.3
	ieee8021PSFPOperControlListLength	PSFPOperControlListLength, 8.6.5, 8.6.10, 12.31.3
	ieee8021PSFPAdminControlList	PSFPAdminControlList, 8.6.5, 8.6.5.1, 8.6.10, 12.31.3
	ieee8021PSFPOperControlList	PSFPOperControlList, 8.6.5, 8.6.5.1, 8.6.10, 12.31.3
	ieee8021PSFPAdminCycleTimeNumerator	PSFPAdminCycleTime, 8.6.5, 8.6.5.1, 8.6.10, 12.31.3
	ieee8021PSFPAdminCycleTimeDenominator	PSFPAdminCycleTime, 8.6.5, 8.6.5.1, 8.6.10, 12.31.3
	ieee8021PSFPOperCycleTimeNumerator	PSFPOperCycleTime, 8.6.5, 8.6.5.1, 8.6.10, 12.31.3
	ieee8021PSFPOperCycleTimeDenominator	PSFPOperCycleTime, 8.6.5, 8.6.5.1, 8.6.10, 12.31.3
	ieee8021PSFPAdminCycleTimeExtension	PSFPAdminCycleTimeExtension, 8.6.5, 8.6.5.1, 8.6.10, 12.31.3
	ieee8021PSFPOperCycleTimeExtension	PSFPOperCycleTimeExtension, 8.6.5, 8.6.5.1, 8.6.10, 12.31.3
	ieee8021PSFPAdminBaseTime	PSFPAdminBaseTime, 8.6.5, 8.6.5.1, 8.6.10, 12.31.3
	ieee8021PSFPOperBaseTime	PSFPOperBaseTime, 8.6.5, 8.6.5.1, 8.6.10, 12.31.3
	ieee8021PSFPConfigChange	PSFPConfigChange, 8.6.5, 8.6.5.1, 8.6.10, 12.31.3
	ieee8021PSFPConfigChangeTime	PSFPConfigChangeTime,8.6.5, 8.6.5.1, 8.6.10, 12.31.3
	ieee8021PSFPTickGranularity	PSFPTickGranularity, 8.6.5, 8.6.5.1, 8.6.10, 12.31.3

Table 17-30—IEEE8021-PSFP-MIB Structure and relationship to this standard *(continued)*

MIB table	MIB object	Reference
	ieee8021PSFPCurrentTime	PSFPCurrentTime, 8.6.5, 8.6.5.1, 8.6.10, 12.31.3
	ieee8021PSFPConfigPending	PSFPConfigPending, 8.6.5, 8.6.5.1, 8.6.10, 12.31.3
	ieee8021PSFPConfigChangeError	PSFPConfigChangeError, 8.6.5, 8.6.5.1, 8.6.10, 12.31.3
	ieee8021PSFPAdminIPV	PSFPAdminIPV, 8.6.5, 8.6.5.1, 12.31.3
	ieee8021PSFPOperIPV	PSFPOperIPV, 8.6.5, 8.6.5.1, 12.31.3
	ieee8021PSFPGateClosedDueToInvalidRxEnable	PSFPGateClosedDueToInvalidRx-Enable, 8.6.5.1.2
	ieee8021PSFPGateClosedDueToInvalidRx	PSFPGateClosedDueToInvalidRx, 8.6.5.1.2
	ieee8021PSFPGateClosedDueToOctetsExceededEnable	PSFPGateClosedDueToOctets-ExceededEnable, 8.6.5.1.2
	ieee8021PSFPGateClosedDueToOctetsExceeded	PSFPGateClosedDueToOctets-Exceeded, 8.6.5.1.2
ieee8021PSFPFlowMeterParameters		
ieee8021PSFPFlowMeterTable		Flow Meter Instance Table, 8.6.5, 8.6.5.1, 12.31.4
	ieee8021PSFPFlowMeterInstance	FlowMeterInstanceID, 8.6.5, 8.6.5.1, 12.31.4
	ieee8021PSFPFlowMeterCIR	CIR, 8.6.5, 8.6.5.1, 12.31.4
	ieee8021PSFPFlowMeterCBS	CBS, 8.6.5, 8.6.5.1, 12.31.4
	ieee8021PSFPFlowMeterEIR	EIR, 8.6.5, 8.6.5.1, 12.31.4
	ieee8021PSFPFlowMeterCF	CF, 8.6.5, 8.6.5.1, 12.31.4
	ieee8021PSFPFlowMeterCM	CM, 8.6.5, 8.6.5.1, 12.31.4
	ieee8021PSFPFlowMeterDropOnYellow	DropOnYellow, 8.6.5, 8.6.5.1, 12.31.4
	ieee8021PSFPFlowMeterMarkAllFramesRedEnable	MarkAllFramesRedEnable, 8.6.5, 8.6.5.1, 12.31.4
	ieee8021PSFPFlowMeterMarkAllFramesRed	MarkAllFramesRed, 8.6.5, 8.6.5.1, 12.31.4

Table 17-30—IEEE8021-PSFP-MIB Structure and relationship to this standard *(continued)*

MIB table	MIB object	Reference
ieee8021PSFPStreamParameters		
ieee8021PSFPStreamParameterTable		StreamParameterTable, 8.6.5, 8.6.5.1, 12.31.1
	ieee8021PSFPMaxStreamFilterInstances	MaxStreamFilterInstances, 8.6.5, 8.6.5.1, 12.31.1
	ieee8021PSFPMaxStreamGateInstances	MaxStreamGateInstances, 8.6.5, 8.6.5.1, 12.31.1
	ieee8021PSFPMaxFlowMeterInstances	MaxFlowMeterInstances, 8.6.5, 8.6.5.1, 12.31.1
	ieee8021PSFPSupportedListMax	SupportedListMax, 8.6.5, 8.6.5.1, 12.31.1

17.3 Relationship to other MIBs

Insert new subclause 17.3.24 at the end of 17.3, as shown, renumbering as necessary.

17.3.24 Relationship of IEEE8021-PSFP-MIB to other MIBs

The IEEE8021-PSFP-MIB provides objects that extend the core management functionality of a Bridge, as defined by the IEEE8021-BRIDGE-MIB (17.7.2), in order to support the additional management functionality needed when the PSFP extensions, as defined in 8.6.5 and 8.6.10, are supported by the Bridge. As support of the objects defined in the IEEE8021-PSFP-MIB also requires support of the IEEE8021-BRIDGE-MIB, the provisions of 17.3.2 apply to implementations claiming support of the IEEE8021-PSFP-MIB.

17.4 Security considerations

Insert new subclause 17.4.24 at the end of 17.4, as shown, renumbering as necessary.

17.4.24 Security considerations of the IEEE8021-PSFP-MIB

There are a number of management objects defined in the IEEE8021-PSFP-MIB module that have a MAX-ACCESS clause of read-write or read-create. Such objects may be considered sensitive or vulnerable in some network environments. The support for SET operations in a nonsecure environment without proper protection can have a negative effect on network operations.

The following tables and objects in the IEEE8021-PSFP-MIB can be misconfigured to interfere with the operation of the forwarding and queuing mechanisms in a manner that would be detrimental to the operation of PSFP:

ieee8021PSFPStreamHandleSpec
ieee8021PSFPPrioritySpec
ieee8021PSFPStreamGateInstanceID
ieee8021PSFPFilterSpecificationList
ieee8021PSFPStreamBlockedDueToOversizeFrameEnable
ieee8021PSFPStreamBlockedDueToOversizeFrame

ieee8021PSFPGateEnabled
ieee8021PSFPAdminGateStates
ieee8021PSFPAdminControlListLength
ieee8021PSFPAdminControlList
ieee8021PSFPAdminCycleTimeNumerator
ieee8021PSFPAdminCycleTimeDenominator
ieee8021PSFPAdminCycleTimeExtension
ieee8021PSFPAdminBaseTime
ieee8021PSFPConfigChange
ieee8021PSFPConfigChangeTime
ieee8021PSFPAdminIPV
ieee8021PSFPOperIPV
ieee8021PSFPGateClosedDueToInvalidRxEnable
ieee8021PSFPGateClosedDueToInvalidRx
ieee8021PSFPGateClosedDueToOctetsExceededEnable
ieee8021PSFPGateClosedDueToOctetsExceeded
ieee8021PSFPFlowMeterInstance
ieee8021PSFPFlowMeterCIR
ieee8021PSFFlowMeterCBS
ieee8021PSFPFlowMeterEIR
ieee8021PSFPFlowMeterCF
ieee8021PSFPFlowMeterCM
ieee8021PSFPFlowMeterDropOnYellow
ieee8021PSFPFlowMeterMarkAllFramesRedEnable
ieee8021PSFPFlowMeterMarkAllFramesRed

a) ieee8021PSFPStreamHandleSpec, ieee8021PSFPPrioritySpec, ieee8021PSFPStreamGateInstanceID, and ieee8021PSFPFilterSpecificationList can be misconfigured to adversely affect the policing functions that are applied to received frames.
b) ieee8021PSFPGateEnabled can be misconfigured to enable/disable scheduled traffic processing.
c) ieee8021PSFPAdminGateStates can be misconfigured to affect the gate state of a stream filter on startup.
d) ieee8021PSFPAdminControlListLength, ieee8021PSFPAdminControlList, ieee8021PSFPAdminCycleTimeNumerator, ieee8021PSFPAdminCycleTimeDenominator, ieee8021PSFPAdminCycleTimeExtension, ieee8021PSFPAdminBaseTime, ieee8021PSFPConfigChange, and ieee8021PSFPConfigChangeTime can be misconfigured to affect the filter schedule for the Port.
ieee8021PSFPFlowMeterInstance, ieee8021PSFPFlowMeterCIR, ieee8021PSFFlowMeterCBS, ieee8021PSFPFlowMeterEIR, ieee8021PSFPFlowMeterCF, ieee8021PSFPFlowMeterCM, ieee8021PSFPFlowMeterDropOnYellow, ieee8021PSFPFlowMeterMarkAllFramesRedEnable, ieee8021PSFPFlowMeterMarkAllFramesRed, ieee8021PSFPGateClosedDueToInvalidRxEnable, ieee8021PSFPGateClosedDueToInvalidRx, ieee8021PSFPGateClosedDueToOctetsExceededEnable, ieee8021PSFPGateClosedDueToOctetsExceeded, ieee8021PSFPStreamBlockedDueToOversizeFrameEnable, and ieee8021PSFPStreamBlockedDueToOversizeFrame can be misconfigured to adversely affect the way that flow metering operates.

Some of the readable objects in this MIB module (i.e., objects with a MAX-ACCESS other than not accessible) can be considered sensitive or vulnerable in some network environments. It is thus important to control all types of access (including GET and/or NOTIFY) to these objects and possibly to encrypt the values of these objects when sending them over the network via SNMP.

17.7 MIB modules

Insert new subclause 17.7.24 at the end of 17.7, as shown, renumbering as necessary.

17.7.24Definitions for the IEEE8021-PSFP-MIB module

```
IEEE8021-PSFP-MIB DEFINITIONS ::= BEGIN

-- ============================================================
-- MIB for support of the Per-Stream Filtering and Policing
-- Enhancements for 802.1Q Bridges.
-- ============================================================

IMPORTS
    MODULE-IDENTITY,
    OBJECT-TYPE,
    Unsigned32,
    Integer32,
    Counter64
        FROM SNMPv2-SMI
    TruthValue, RowStatus
        FROM SNMPv2-TC
    MODULE-COMPLIANCE,
    OBJECT-GROUP
        FROM SNMPv2-CONF
    ieee802dot1mibs
        FROM IEEE8021-TC-MIB
    ieee8021BridgeBaseComponentId
        FROM IEEE8021-BRIDGE-MIB
    IEEE8021STPTPtimeValue
        FROM IEEE8021-ST-MIB
    ;

ieee8021PSFPMib MODULE-IDENTITY
    LAST-UPDATED "201709080000Z" -- October 6, 2017
    ORGANIZATION "IEEE 802.1 Working Group"
    CONTACT-INFO
        "  WG-URL: http://www.ieee802.org/1/
         WG-EMail: stds-802-1-L@ieee.org

           Contact: IEEE 802.1 Working Group Chair
            Postal: C/O IEEE 802.1 Working Group
                    IEEE Standards Association
                    445 Hoes Lane
                    P.O. Box 1331
                    Piscataway
                    NJ 08855-1331
                    USA
             E-mail: stds-802-1-L@ieee.org"
    DESCRIPTION
        "The Bridge MIB module for managing devices that support
        the Per-Stream Filtering and Policing enhancements
        for 802.1Q Bridges.

        Unless otherwise indicated, the references in this MIB
        module are to IEEE Std 802.1Q-2014.

        Copyright (C) IEEE (2017).
```

```
        This version of this MIB module is part of IEEE802.1Q;
        see the draft itself for full legal notices."

    REVISION "201709080000Z" -- October 6, 2017
    DESCRIPTION
            "Initial version published as part of IEEE Std 802.1Qci."

       ::= { ieee802dot1mibs 31 }


-- ================================================================
-- subtrees in the PSFP MIB
-- ================================================================

ieee8021PSFPNotifications
    OBJECT IDENTIFIER ::= { ieee8021PSFPMib 0 }

ieee8021PSFPObjects
    OBJECT IDENTIFIER ::= { ieee8021PSFPMib 1 }

ieee8021PSFPConformance
    OBJECT IDENTIFIER ::= { ieee8021PSFPMib 2 }

ieee8021PSFPStreamFilterParameters
    OBJECT IDENTIFIER ::= { ieee8021PSFPObjects 1 }

ieee8021PSFPStreamGateParameters
    OBJECT IDENTIFIER ::= { ieee8021PSFPObjects 2 }

ieee8021PSFPFlowMeterParameters
    OBJECT IDENTIFIER ::= { ieee8021PSFPObjects 3 }

ieee8021PSFPStreamParameters
    OBJECT IDENTIFIER ::= { ieee8021PSFPObjects 4 }


-- ================================================================
-- The ieee8021PSFPStreamFilterParameters subtree
-- This subtree defines the objects necessary for the management
-- of the stream filters for IEEE Std 802.1Q.
-- ================================================================

-- ================================================================
-- the ieee8021PSFPStreamFilterTable
-- ================================================================

ieee8021PSFPStreamFilterTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021PSFPStreamFilterEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table that contains the per-filter instance
        manageable parameters for stream filters.

        A row in the table exists for each stream filter instance.
        associated with a Bridge component.

        All writable objects in this table must be
        persistent over power up restart/reboot."
```

```
    REFERENCE   "8.6.5, 8.6.5.1 12.31.2"
    ::= { ieee8021PSFPStreamFilterParameters 1 }

ieee8021PSFPStreamFilterEntry OBJECT-TYPE
    SYNTAX      Ieee8021PSFPStreamFilterEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A list of objects that contains the manageable parameters for
        stream filters for a Bridge component."
    INDEX  { ieee8021BridgeBaseComponentId,
             ieee8021PSFPStreamFilterInstance
           }
    ::= { ieee8021PSFPStreamFilterTable 1 }

Ieee8021PSFPStreamFilterEntry ::=
    SEQUENCE {
        ieee8021PSFPStreamFilterInstance
            Unsigned32,
        ieee8021PSFPStreamHandleSpec
            Integer32,
        ieee8021PSFPPrioritySpec
            Integer32,
        ieee8021PSFPStreamGateInstanceID
            Unsigned32,
        ieee8021PSFPFilterSpecificationList
            OCTET STRING,
        ieee8021PSFPMatchingFramesCount
            Counter64,
        ieee8021PSFPPassingFramesCount
            Counter64,
        ieee8021PSFPNotPassingFramesCount
            Counter64,
        ieee8021PSFPPassingSDUCount
            Counter64,
        ieee8021PSFPNotPassingSDUCount
            Counter64,
        ieee8021PSFPREDFramesCount
            Counter64,
        ieee8021PSFPStreamBlockedDueToOversizeFrameEnable
            TruthValue,
        ieee8021PSFPStreamBlockedDueToOversizeFrame
            TruthValue,
        ieee8021PSFPStreamFilterEntryRowStatus
            RowStatus
          }

ieee8021PSFPStreamFilterInstance OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The StreamFilterInstance parameter is an index into the
        StreamFilterTable.

        The value of this object MUST be retained across
        reinitializations of the management system."
    REFERENCE   "8.6.5.1, 12.31.2"
    ::= { ieee8021PSFPStreamFilterEntry 1}
```

```
ieee8021PSFPStreamHandleSpec OBJECT-TYPE
    SYNTAX      Integer32 (-1..2147483647)
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The StreamHandleSpec parameter contains a stream identifier
        specification value. A value of -1 denotes the wild card value;
        all positive values denote stream identifier values.

        The value of this object MUST be retained across
        reinitializations of the management system."
    REFERENCE   "8.6.5.1, 12.31.2"
    ::= { ieee8021PSFPStreamFilterEntry 2}

ieee8021PSFPPrioritySpec OBJECT-TYPE
    SYNTAX      Integer32 (-1..2147483647)
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The PrioritySpec parameter contains a priority
        specification value. A value of -1 denotes the wild card value;
        zero or positive values denote priority values.

        The value of this object MUST be retained across
        reinitializations of the management system."
    REFERENCE   "8.6.5.1, 12.31.2"
    ::= { ieee8021PSFPStreamFilterEntry 3}

ieee8021PSFPStreamGateInstanceID OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The StreamGateInstance parameter contains the index of an
        entry in the Stream Gate Table.

        The value of this object MUST be retained across
        reinitializations of the management system."
    REFERENCE   "8.6.5.1, 12.31.2"
    ::= { ieee8021PSFPStreamFilterEntry 4}

ieee8021PSFPFilterSpecificationList OBJECT-TYPE
    SYNTAX      OCTET STRING
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The FilterSpecificationList parameter contains a list of
        filter specifications associated with this stream filter.

        The octet string value represents the contents of the list as
        an ordered list of entries, each encoded as a TLV, as follows.

        The first octet  of each TLV is interpreted as an
        unsigned integer representing a filter specification type:
            0: Maximum SDU Size.
            1: Flow meter instance identifier.
            2-255: Reserved for future gate operations
```

```
            The second and third octets of the TLV are the length field,
            interpreted as an unsigned integer, indicating the number of
            octets of the value that follows the length. A length of
            zero indicates that there is no value
            (i.e., the filter specification has no parameters).

            The fourth through (4 + length -1)th octets encode the
            parameters of the filter specification, as defined for each
            filter specification type.

            - Maximum SDU Size:
                A single SDU size parameter is encoded in four octets, and
                is interpreted as an unsigned integer value.

            - Flow meter instance identifier:
                A single flow meter instance identifier is encoded in
                four octets, and is interpreted as an unsigned integer value.

            The value of this object MUST be retained across
            reinitializations of the management system."
        REFERENCE   "8.6.5.1, 12.31.2"
        ::= { ieee8021PSFPStreamFilterEntry 5}

    ieee8021PSFPMatchingFramesCount OBJECT-TYPE
        SYNTAX      Counter64
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The MatchingFramesCount counter counts received frames that
            match this stream filter.
            "
        REFERENCE   "8.6.5.1, 12.31.2"
        ::= { ieee8021PSFPStreamFilterEntry 6}

    ieee8021PSFPPassingFramesCount OBJECT-TYPE
        SYNTAX      Counter64
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The PassingFramesCount counter counts received frames that
            pass the gate associated with this stream filter.
            "
        REFERENCE   "8.6.5.1, 12.31.2"
        ::= { ieee8021PSFPStreamFilterEntry 7}

    ieee8021PSFPNotPassingFramesCount OBJECT-TYPE
        SYNTAX      Counter64
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The NotPassingFramesCount counter counts received frames that
            do not pass the gate associated
            with this stream filter.
            "
        REFERENCE   "8.6.5.1, 12.31.2"
        ::= { ieee8021PSFPStreamFilterEntry 8}

    ieee8021PSFPPassingSDUCount OBJECT-TYPE
        SYNTAX      Counter64
```

```
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The  PassingSDUCount counter counts received frames that
            pass the SDU size filter specification associated
            with this stream filter.
            "
        REFERENCE   "8.6.5.1, 12.31.2"
        ::= { ieee8021PSFPStreamFilterEntry 9}

    ieee8021PSFPNotPassingSDUCount OBJECT-TYPE
        SYNTAX      Counter64
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The  NotPassingSDUCount counter counts received frames that
            do not pass the SDU size filter specification associated
            with this stream filter.
            "
        REFERENCE   "8.6.5.1, 12.31.2"
        ::= { ieee8021PSFPStreamFilterEntry 10}

    ieee8021PSFPREDFramesCount OBJECT-TYPE
        SYNTAX      Counter64
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The  REDFramesCount counter counts received
            random early detection (RED) frames associated
            with this stream filter.
            "
        REFERENCE   "8.6.5.1, 12.31.2"
        ::= { ieee8021PSFPStreamFilterEntry 11}

    ieee8021PSFPStreamBlockedDueToOversizeFrameEnable OBJECT-TYPE
        SYNTAX      TruthValue
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
            "The ieee8021PSFPStreamBlockedDueToOversizeFrameEnable object
            contains a Boolean value that indicates whether the
            ieee8021PSFPStreamBlockedDueToOversizeFrame function is
            enabled (TRUE) or disabled (FALSE).

            The value of this object MUST be retained across
            reinitializations of the management system."
        REFERENCE   "8.6.5.1, 8.6.5.1.1, 12.31.2"
        DEFVAL { false }
        ::= { ieee8021PSFPStreamFilterEntry 12 }

    ieee8021PSFPStreamBlockedDueToOversizeFrame OBJECT-TYPE
        SYNTAX      TruthValue
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
            "The ieee8021PSFPStreamBlockedDueToOversizeFrame object
            contains a Boolean value that indicates whether, if the
            ieee8021PSFPStreamBlockedDueToOversizeFrame function is
            enabled, all frames are to be discarded (TRUE)
```

```
        or not (FALSE).

        The value of this object MUST be retained across
        reinitializations of the management system."
    REFERENCE   "8.6.5.1,  8.6.5.1.1, 12.31.2"
    DEFVAL { false }
    ::= { ieee8021PSFPStreamFilterEntry 13 }

ieee8021PSFPStreamFilterEntryRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The status of the row.

         The writable columns in a row can not be changed if the row
         is active. All columns MUST have a valid value before a row
         can be activated.
        "
    ::= { ieee8021PSFPStreamFilterEntry 14 }


-- ================================================================
-- The ieee8021PSFPStreamGateParameters subtree
-- This subtree defines the objects necessary for the management
-- of the stream gate scheduling mechanism for IEEE Std 802.1Q.
-- ================================================================

-- ================================================================
-- the ieee8021PSFPStreamGateTable
-- ================================================================

ieee8021PSFPStreamGateTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021PSFPStreamGateEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table that contains the per-gate instance
        manageable parameters for stream gate scheduling.

        For a given Bridge component, a row in the table exists for
        each stream gate instance.

        All writable objects in this table must be
        persistent over power up restart/reboot."
    REFERENCE   "8.6.8.4, 8.6.9.4, 12.31.3"
    ::= { ieee8021PSFPStreamGateParameters 1 }

ieee8021PSFPStreamGateEntry OBJECT-TYPE
    SYNTAX      Ieee8021PSFPStreamGateEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A list of objects that contains the manageable parameters for
        stream gate scheduling for a Bridge component."
    INDEX  { ieee8021BridgeBaseComponentId,
             ieee8021PSFPStreamGateInstance
           }
    ::= { ieee8021PSFPStreamGateTable 1 }
```

```
Ieee8021PSFPStreamGateEntry ::=
    SEQUENCE {
        ieee8021PSFPStreamGateInstance
            Unsigned32,
        ieee8021PSFPGateEnabled
            TruthValue,
        ieee8021PSFPAdminGateStates
            INTEGER,
        ieee8021PSFPOperGateStates
            INTEGER,
        ieee8021PSFPAdminControlListLength
            Unsigned32,
        ieee8021PSFPOperControlListLength
            Unsigned32,
        ieee8021PSFPAdminControlList
            OCTET STRING,
        ieee8021PSFPOperControlList
            OCTET STRING,
        ieee8021PSFPAdminCycleTimeNumerator
            Unsigned32,
        ieee8021PSFPAdminCycleTimeDenominator
            Unsigned32,
        ieee8021PSFPOperCycleTimeNumerator
            Unsigned32,
        ieee8021PSFPOperCycleTimeDenominator
            Unsigned32,
        ieee8021PSFPAdminCycleTimeExtension
            Unsigned32,
        ieee8021PSFPOperCycleTimeExtension
            Unsigned32,
        ieee8021PSFPAdminBaseTime
            IEEE8021STPTPtimeValue,
        ieee8021PSFPOperBaseTime
            IEEE8021STPTPtimeValue,
        ieee8021PSFPConfigChange
            TruthValue,
        ieee8021PSFPConfigChangeTime
            IEEE8021STPTPtimeValue,
        ieee8021PSFPTickGranularity
            Unsigned32,
        ieee8021PSFPCurrentTime
            IEEE8021STPTPtimeValue,
        ieee8021PSFPConfigPending
            TruthValue,
        ieee8021PSFPConfigChangeError
            Counter64,
        ieee8021PSFPAdminIPV
            Integer32,
        ieee8021PSFPOperIPV
            Integer32,
        ieee8021PSFPGateClosedDueToInvalidRxEnable
            TruthValue,
        ieee8021PSFPGateClosedDueToInvalidRx
            TruthValue,
        ieee8021PSFPGateClosedDueToOctetsExceededEnable
            TruthValue,
        ieee8021PSFPGateClosedDueToOctetsExceeded
            TruthValue,
```

```
ieee8021PSFPStreamGateEntryRowStatus
            RowStatus
            }

ieee8021PSFPStreamGateInstance OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The StreamGateInstance parameter is an index into the
        StreamGateTable.

        The value of this object MUST be retained across
        reinitializations of the management system."
    REFERENCE   "8.6.5.1, 8.6.5.1.2, 12.31.3"
    ::= { ieee8021PSFPStreamGateEntry 1}

ieee8021PSFPGateEnabled OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The GateEnabled parameter determines whether the stream gate
        is active (true) or inactive (false).

        The value of this object MUST be retained across
        reinitializations of the management system."
    REFERENCE   "8.6.8.4, 8.6.9.4, 12.31.3"
    DEFVAL { false }
    ::= { ieee8021PSFPStreamGateEntry 2}

ieee8021PSFPAdminGateStates OBJECT-TYPE
    SYNTAX      INTEGER { open(1), closed(2) }
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The administrative value of the GateStates parameter for the
        stream gate.
        The open value indicates that the gate is open,
        the closed value indicates that the gate is closed.

        The value of this object MUST be retained across
        reinitializations of the management system."
    REFERENCE   "8.6.8.4, 8.6.9.4, 12.31.3"
    ::= { ieee8021PSFPStreamGateEntry 3 }

ieee8021PSFPOperGateStates OBJECT-TYPE
    SYNTAX      INTEGER { open(1), closed(2) }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The operational value of the GateStates parameter for the
        stream gate.
        The open value indicates that the gate is open,
        the closed value indicates that the gate is closed.
"
    REFERENCE   "8.6.8.4, 8.6.9.4, 12.31.3"
    ::= { ieee8021PSFPStreamGateEntry 4 }
```

```
ieee8021PSFPAdminControlListLength OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The administrative value of the ListMax parameter for the gate.
        The integer value indicates the number of entries (TLVs) in the
        AdminControlList.

        The value of this object MUST be retained across
        reinitializations of the management system."
    REFERENCE   "8.6.8.4, 8.6.9.4, 12.31.3"
    ::= { ieee8021PSFPStreamGateEntry 5 }

ieee8021PSFPOperControlListLength OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The operational value of the ListMax parameter for the gate.
        The integer value indicates the number of entries (TLVs) in the
        OperControlList."
    REFERENCE   "8.6.8.4, 8.6.9.4, 12.31.3"
    ::= { ieee8021PSFPStreamGateEntry 6 }

ieee8021PSFPAdminControlList OBJECT-TYPE
    SYNTAX      OCTET STRING
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The administrative value of the ControlList parameter for the gate.
        The octet string value represents the contents of the control list as
        an ordered list of entries, each encoded as a TLV, as follows.
        The first octet  of each TLV is interpreted as an
        unsigned integer representing a gate operation name:
            0: SetGateAndIPV
            1-255: Reserved for future gate operations

        The second octet of the TLV is the length field,
        interpreted as an unsigned integer, indicating the number of
        octets of the value that follows the length. A length of
        zero indicates that there is no value
        (i.e., the gate operation has no parameters).

        The third through (3 + length -1)th octets encode the
        parameters of the gate operation, in the order that they
        appear in the definition of the operation
        in Table 8-7. Three parameter types are defined:

        - StreamGateState:
            A GateState parameter is encoded in a single octet, and
            is interpreted as an integer value.
            The value 1 indicates open; the value 2 indicates closed.

        - IPV:
            An IPV is encoded in four octets as a 32-bit
            signed integer. A negative denotes the null value;
            zero or positive values denote internal priority values.
```

```
            - TimeInterval:
                A TimeInterval is encoded in 4 octets as a 32-bit
                unsigned integer, representing a number of nanoseconds.
                The first octet encodes the most significant 8 bits of the
                integer, and the fourth octet encodes the least
                significant 8 bits.
            - IntervalOctetMax:
                An integer representing the maximum number of MSDU octets
                that are permitted to pas the gate during the specified
                TimeInterval. If this parameter is omitted, there is
                no maximum.

            The value of this object MUST be retained across
            reinitializations of the management system."
        REFERENCE   "8.6.8.4, 8.6.9.4, 12.31.3"
        ::= { ieee8021PSFPStreamGateEntry 7 }

    ieee8021PSFPOperControlList OBJECT-TYPE
        SYNTAX      OCTET STRING
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The operational value of the ControlList parameter for the gate.
            The octet string value represents the contents of the control list as
            an ordered list of entries, each encoded as a TLV, as follows.
            The first octet  of each TLV is interpreted as an
            unsigned integer representing a gate operation name:
                0: SetGateAndIPV
                1-255: Reserved for future gate operations

            The second octet of the TLV is the length field,
            interpreted as an unsigned integer, indicating the number of
            octets of the value that follows the length. A length of
            zero indicates that there is no value
            (i.e., the gate operation has no parameters).

            The third through (3 + length -1)th octets encode the
            parameters of the gate operation, in the order that they
            appear in the definition of the operation
            in Table 8-7. Three parameter types are defined:

            - StreamGateState:
                A GateState parameter is encoded in a single octet, and
                is interpreted as an integer value.
                The value 1 indicates open; the value 2 indicates closed.

            - IPV:
                An IPV is encoded in four octets as a 32-bit
                signed integer. A negative value denotes the null value;
                zero and positive values denote internal priority values.

            - TimeInterval:
                A TimeInterval is encoded in 4 octets as a 32-bit
                unsigned integer, representing
                a number of nanoseconds. The first octet encodes the
                most significant 8 bits of the integer, and the fourth
                octet encodes the least significant 8 bits.
            - IntervalOctetMax:
                An integer representing the maximum number of MSDU octets
```

```
            that are permitted to pas the gate during the specified
            TimeInterval. If this parameter is omitted, there is
            no maximum.
        "
    REFERENCE   "8.6.8.4, 8.6.9.4, 12.31.3"
    ::= { ieee8021PSFPStreamGateEntry 8 }

ieee8021PSFPAdminCycleTimeNumerator OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The administrative value of the numerator of the CycleTime
        parameter for the gate.
        The numerator and denominator together represent the cycle time as
        a rational number of seconds.

        The value of this object MUST be retained across
        reinitializations of the management system."
    REFERENCE   "8.6.8.4, 8.6.9.4, 12.31.3"
    ::= { ieee8021PSFPStreamGateEntry 9 }

ieee8021PSFPAdminCycleTimeDenominator OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The administrative value of the denominator of the
        CycleTime parameter for the gate.
        The numerator and denominator together represent the cycle time as
        a rational number of seconds.

        The value of this object MUST be retained across
        reinitializations of the management system."
    REFERENCE   "8.6.8.4, 8.6.9.4, 12.31.3"
    ::= { ieee8021PSFPStreamGateEntry 10 }

ieee8021PSFPOperCycleTimeNumerator OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The operational value of the numerator of the
        CycleTime parameter for the gate.
        The numerator and denominator together represent the cycle
        time as a rational number of seconds."
    REFERENCE   "8.6.8.4, 8.6.9.4, 12.31.3"
    ::= { ieee8021PSFPStreamGateEntry 11 }

ieee8021PSFPOperCycleTimeDenominator OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The operational value of the denominator of the
        CycleTime parameter for the gate.
        The numerator and denominator together represent the
        cycle time as a rational number of seconds."
    REFERENCE   "8.6.8.4, 8.6.9.4, 12.31.3"
```

```
    ::= { ieee8021PSFPStreamGateEntry 12 }

ieee8021PSFPAdminCycleTimeExtension OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "nanoseconds"
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The administrative value of the CycleTimeExtension
        parameter for the gate.
        The value is an unsigned integer number of nanoseconds.

        The value of this object MUST be retained across
        reinitializations of the management system."
    REFERENCE   "8.6.8.4, 8.6.9.4, 12.31.3"
    ::= { ieee8021PSFPStreamGateEntry 13 }

ieee8021PSFPOperCycleTimeExtension OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "nanoseconds"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The operational value of the CycleTimeExtension
        parameter for the gate.
        The value is an unsigned integer number of nanoseconds."
    REFERENCE   "8.6.8.4, 8.6.9.4, 12.31.3"
    ::= { ieee8021PSFPStreamGateEntry 14 }

ieee8021PSFPAdminBaseTime OBJECT-TYPE
    SYNTAX      IEEE8021STPTPtimeValue
    UNITS       "PTP time"
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The administrative value of the BaseTime parameter for the gate.
        The value is a representation of a PTPtime value,
        consisting of a 48-bit integer
        number of seconds and a 32-bit integer number of nanoseconds.

        The value of this object MUST be retained across
        reinitializations of the management system."
    REFERENCE   "8.6.8.4, 8.6.9.4, 12.31.3"
    ::= { ieee8021PSFPStreamGateEntry 15 }

ieee8021PSFPOperBaseTime OBJECT-TYPE
    SYNTAX      IEEE8021STPTPtimeValue
    UNITS       "PTP time"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The operationsl value of the BaseTime parameter for the gate.
        The value is a representation of a PTPtime value,
        consisting of a 48-bit integer
        number of seconds and a 32-bit integer number of nanoseconds."
    REFERENCE   "8.6.8.4, 8.6.9.4, 12.31.3"
    ::= { ieee8021PSFPStreamGateEntry 16 }

ieee8021PSFPConfigChange OBJECT-TYPE
```

```
        SYNTAX      TruthValue
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
            "The ConfigChange parameter signals the start of a
            configuration change for the gate
            when it is set to TRUE. This should only be done
            when the various administrative parameters
            are all set to appropriate values."
        REFERENCE   "8.6.8.4, 8.6.9.4, 12.31.3"
        ::= { ieee8021PSFPStreamGateEntry 17 }

    ieee8021PSFPConfigChangeTime OBJECT-TYPE
        SYNTAX      IEEE8021STPTPtimeValue
        UNITS       "PTP time"
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The PTPtime at which the next config change is scheduled to occur.
            The value is a representation of a PTPtime value,
            consisting of a 48-bit integer
            number of seconds and a 32-bit integer number of nanoseconds.

            The value of this object MUST be retained across
            reinitializations of the management system."
        REFERENCE   "8.6.8.4, 8.6.9.4, 12.31.3"
        ::= { ieee8021PSFPStreamGateEntry 18 }

        ieee8021PSFPTickGranularity OBJECT-TYPE
        SYNTAX      Unsigned32
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The granularity of the cycle time clock, represented as an
            unsigned number of tenths of nanoseconds.

            The value of this object MUST be retained across
            reinitializations of the management system."
        REFERENCE   "8.6.8.4, 8.6.9.4, 12.31.3"
        ::= { ieee8021PSFPStreamGateEntry 19 }

     ieee8021PSFPCurrentTime OBJECT-TYPE
        SYNTAX      IEEE8021STPTPtimeValue
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The current time, in PTPtime, as maintained by the local system.
            The value is a representation of a PTPtime value,
            consisting of a 48-bit integer
            number of seconds and a 32-bit integer number of nanoseconds."
        REFERENCE   "8.6.8.4, 8.6.9.4, 12.31.3"
        ::= { ieee8021PSFPStreamGateEntry 20 }

     ieee8021PSFPConfigPending OBJECT-TYPE
        SYNTAX      TruthValue
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The value of the ConfigPending state machine variable.
```

```
        The value is TRUE if a configuration change is in progress
        but has not yet completed."
    REFERENCE   "8.6.8.4, 8.6.9.4, 12.31.3"
    ::= { ieee8021PSFPStreamGateEntry 21 }

 ieee8021PSFPConfigChangeError OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "A counter of the number of times that a re-configuration
        of the traffic schedule has been requested with the old
        schedule still running and the requested base time was
        in the past."
    REFERENCE   "8.6.8.4, 8.6.9.3, 8.6.9.1.1, 12.31.3"
    ::= { ieee8021PSFPStreamGateEntry 23 }

ieee8021PSFPAdminIPV OBJECT-TYPE
    SYNTAX      Integer32 (-1..2147483647)
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The administrative value of the IPV parameter for the gate.
        A value of -1 denotes the null value.
        "
    REFERENCE   "8.6.5.1.2, 8.6.10, 12.31.3"
    ::= { ieee8021PSFPStreamGateEntry 24 }

ieee8021PSFPOperIPV OBJECT-TYPE
    SYNTAX      Integer32 (-1..2147483647)
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The operational value of the IPV parameter for the gate.
        A value of -1 denotes the null value.
        "
    REFERENCE   "8.6.5.1.2, 8.6.10, 12.31.3"
    ::= { ieee8021PSFPStreamGateEntry 25 }

ieee8021PSFPGateClosedDueToInvalidRxEnable OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The PSFPGateClosedDueToInvalidRxEnable object contains
        a Boolean value that indicates whether the
        PSFPGateClosedDueToInvalidRx function is enabled (TRUE) or
        disabled (FALSE).

        The value of this object MUST be retained across
        reinitializations of the management system."
    REFERENCE   "8.6.5.1, 8.6.5.1.2, 12.31.3"
    DEFVAL { false }
    ::= { ieee8021PSFPStreamGateEntry 26}

ieee8021PSFPGateClosedDueToInvalidRx OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-create
    STATUS      current
```