IEEE SA
STANDARDS
ASSOCIATION

IEEE Standard for
Local and Metropolitan Area Networks—

Bridges and Bridged Networks

Amendment 34:
Asynchronous Traffic Shaping

STANDARDS

IEEE Computer Society

Developed by the
LAN/MAN Standards Committee

IEEE Std 802.1Qcr™-2020
(Amendment to IEEE Std 802.1Q™-2018 as amended by
IEEE Std 802.1Qcp™-2018, IEEE Std 802.1Qcc™-2018,
IEEE Std 802.1Qcy™-2019, and IEEE Std 802.1Qcx™-2020)

IEEE

IEEE Std 802.1Qcr™-2020
(Amendment to IEEE Std 802.1Q™-2018 as amended by
IEEE Std 802.1Qcp™-2018, IEEE Std 802.1Qcc™-2018,
IEEE Std 802.1Qcy™-2019, and IEEE Std 802.1Qcx™-2020)

IEEE Standard for
Local and Metropolitan Area Networks—

Bridges and Bridged Networks

Amendment 34:
Asynchronous Traffic Shaping

Developed by the

LAN/MAN Standards Committee
of the
IEEE Computer Society

Approved 24 September 2020

IEEE SA Standards Board

Abstract: Procedures and managed objects for Bridges and end stations to perform Asynchronous Traffic Shaping over full-duplex links with constant bit data rates are specified in this amendment to IEEE Std 802.1Q™-2018.

Keywords: amendment, Bridged Local Area Network, IEEE 802.1Q™, IEEE 802.1Qcr™, LAN, Local Area Network, MAC Bridge, Metropolitan Area Network, traffic shaping, Virtual Bridged Local Area Network, Virtual LAN

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

PDF: ISBN 978-1-5044-7100-8 STD24430
Print: ISBN 978-1-5044-7101-5 STDPD24430

Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE Standards documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page (https://standards.ieee.org/ipr/disclaimers.html), appear in all standards and may be found under the heading "Important Notices and Disclaimers Concerning IEEE Standards Documents."

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

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE SA) Standards Board. IEEE develops its standards through an accredited consensus development process, which brings together volunteers representing varied viewpoints and interests to achieve the final product. IEEE Standards are documents developed by volunteers with scientific, academic, and industry-based expertise in technical working groups. Volunteers are not necessarily members of IEEE or IEEE SA, 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 makes no warranties or representations concerning its standards, and expressly disclaims all warranties, express or implied, concerning this standard, including but not limited to the warranties of merchantability, fitness for a particular purpose and non-infringement. In addition, IEEE does not warrant or represent that the use of the material contained in its standards is free from patent infringement. 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: THE NEED TO PROCURE 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 is 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, nor 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 the presenter's views should be considered the personal views of that individual rather than the formal position of IEEE, IEEE SA, the Standards Committee, or the Working Group.

Comments on standards

Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE or IEEE SA. However, **IEEE does not provide interpretations, 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 evaluating comments or in revisions to an IEEE standard is welcome to join the relevant IEEE working group. You can indicate interest in a working group using the Interests tab in the Manage Profile & Interests area of the IEEE SA myProject system. An IEEE Account is needed to access the application.

Comments on standards should be submitted using the Contact Us form.

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 constitute 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.

Data privacy

Users of IEEE Standards documents should evaluate the standards for considerations of data privacy and data ownership in the context of assessing and using the standards in compliance with applicable laws and regulations.

Copyrights

IEEE draft and approved standards are copyrighted by IEEE under US 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 licensing fees, 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; https://www.copyright.com/. 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 10 years. When a document is more than 10 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 IEEE Xplore or contact IEEE. For more information about the IEEE SA or IEEE's standards development process, visit the IEEE SA Website.

Errata

Errata, if any, for all IEEE standards can be accessed on the IEEE SA Website. Search for standard number and year of approval to access the web page of the published standard. Errata links are located under the Additional Resources Details section. Errata are also available in IEEE Xplore. Users are encouraged to periodically check for errata.

Patents

IEEE Standards are developed in compliance with the IEEE SA Patent Policy.

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 https://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.

IMPORTANT NOTICE

IEEE Standards do not guarantee or ensure safety, security, health, or environmental protection, or ensure against interference with or from other devices or networks. IEEE Standards development activities consider research and information presented to the standards development group in developing any safety recommendations. Other information about safety practices, changes in technology or technology implementation, or impact by peripheral systems also may be pertinent to safety considerations during implementation of the standard. 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.

Participants

At the time this amendment was submitted to the IEEE SA Standards Board for approval, the IEEE 802.1 Working Group had the following membership:

Glenn Parsons, *Chair*
John Messenger, *Vice Chair*
Jessy V. Rouyer, *Secretary*
János Farkas*, Chair, Time-Sensitive Networking Task Group*
Craig Gunther, *Vice Chair, Time-Sensitive Networking Task Group*
Johannes Specht, *Editor*

Astrit Ademaj
Ralf Assmann
Christian Boiger
Paul Bottorff
Radhakrishna Canchi
Feng Chen
Weiying Cheng
Paul Congdon
Rodney Cummings
Josef Dorr
Hesham Elbakoury
Thomas Enzinger
Donald Fedyk
Norman Finn
Geoffrey Garner
Marina Gutierrez
Stephen Haddock
Mark Hantel
Jerome Henry
Marc Holness
Satoko Itaya
Yoshihiro Ito
Michael Karl
Stephan Kehrer
Randy Kelsey
Hajime Koto
James Lawlis
Christophe Mangin
Scott Mansfield
Kenichi Maruhashi
David McCall
Larry McMillan
Tero Mustala
Roy Myers
Hiroki Nakano
Bob Noseworthy
Tomoki Ohsawa
Hiroshi Ohue
Donald R. Pannell
Michael Potts
Dieter Proell
Wei Qiu
Karen Randall
Maximilian Riegel
Atsushi Sato
Frank Schewe
Mick Seaman
Maik Seewald
Ramesh Sivakolundu
Marius Stanica
Guenter Steindl
Karim Traore
Balazs Varga
Hao Wang
Tongtong Wang
Xinyuan Wang
Karl Weber
Brian Weis
Ludwig Winkel
Jordon Woods
Takahiro Yamaura
Xiang Yu
Nader Zein
William Zhao
Helge Zinner

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

Thomas Alexander
Butch Anton
Stefan Aust
Christian Boiger
Vern Brethour
Matthew Brown
Demetrio Bucaneg
William Byrd
Paul Cardinal
Steven Carlson
Juan Carreon
Pin Chang
Paul Congdon
Charles Cook
Todor Cooklev
Rodney Cummings
Janos Farkas
Avraham Freedman
Craig Gunther
Stephen Haddock
Mark Hantel
Marco Hernandez
Werner Hoelzl
Oliver Holland
Marc Holness
Russell Housley
Woojung Huh
Atsushi Ito
Raj Jain
Sangkwon Jeong
Pranav Jha
Lokesh Kabra
Stephan Kehrer
Randy Kelsey
Stuart Kerry
Evgeny Khorov
Yongbum Kim
Hyeong Ho Lee
Jennifer Lemon
James Lepp
Jon Lewis
Arthur Marris
Michael Maytum
Brett McClellan
Jonathon McLendon
Larry McMillan
Nick S. A. Nikjoo
Satoshi Obara
Bansi Patel
Walter Pienciak
Clinton Powell
Alon Regev
Maximilian Riegel
Robert Robinson
Jessy V. Rouyer
Chester Sandberg
Frank Schewe
Reinhard Schrage
Michael Seaman
Johannes Specht
Thomas Starai
Walter Struppler
Mark-Rene Uchida
James Van De Ligt
John Vergis
George Vlantis
Lisa Ward
Hung-Yu Wei
Scott Willy
Andreas Wolf
Chi Xu
Yu Yuan
Oren Yuen
Nader Zein

When the IEEE SA Standards Board approved this amendment on 24 September 2020, it had the following membership:

Introduction

This introduction is not part of IEEE Std 802.1Qcr-2020, IEEE Standard for Local and Metropolitan Area Networks—Bridges and Bridged Networks—Amendment 34: Asynchronous Traffic Shaping.

This amendment to IEEE Std 802.1Q-2018 specifies procedures and managed objects for Bridges and end stations to perform Asynchronous Traffic Shaping over full-duplex links with constant bit data rates. Asynchronous Traffic Shaping can be modeled as a layer of shaped first-in-first-out (FIFO) queues that are merged into per traffic class FIFO queues in transmission ports. The required minimum number of shaped FIFO queues is adjustable and at least the number of reception ports of a particular bridge.

The amendment specifies an information model for the capabilities of Asynchronous Traffic Shaping. It further specifies a YANG data model based on that information model to support configuration and status reporting. It further defines the relationship between the models introduced by this amendment and the models in the base standard (i.e., IEEE Std 802.1Q-2018 as amended specifically by IEEE Std 802.1Qcp-2018 and IEEE Std 802.1Qcx-2020).

Additionally, this amendment provides an informative framework for worst-case delay analysis in static networks with static configurations.

This amendment also addresses errors and omissions to existing features and includes syntax highlighting in the YANG module listings for easier human readability.

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
USA

Contents

Figures

Tables

IEEE Standard for
Local and Metropolitan Area Networks—

Bridges and Bridged Networks

Amendment 34: Asynchronous Traffic Shaping

(This amendment is based on IEEE Std 802.1Q™-2018 as modified by IEEE Std 802.1Qcp™-2018, IEEE Std 802.1Qcc™-2018, IEEE Std 802.1Qcy™-2019, and IEEE Std 802.1Qcx™-2020.)

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

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 either 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. 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. Overview

1.3 Introduction

Insert the following list item after item ck) in the lettered list of 1.3, and reletter the remaining list items accordingly:

cl) Allow for Asynchronous Traffic Shaping (ATS) over full-duplex links with constant bit data rates.

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

4. Abbreviations

Insert the following abbreviation into Clause 4 in alphabetic order:

ATS	Asynchronous Traffic Shaping

5. Conformance

5.4 VLAN Bridge component requirements

5.4.1 VLAN Bridge component options

5.4.1.8 Per-stream filtering and policing (PSFP) requirements

Change 5.4.1.8 as shown:

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~~8.6.5.2.1 and ~~8.6.6.1~~8.6.6 items d) and e).
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 subclause (5.4.1.10) after 5.4.1.9:

5.4.1.10 Asynchronous Traffic Shaping (ATS) requirements

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

a) Support per-stream classification and metering for ATS as specified in 8.6.5.2.2.
b) Support queuing with support for stream gates as specified in 8.6.6 items d) and e).
c) Support the ATS transmission selection algorithm as specified in 8.6.8.5.
d) Support the ATS scheduler state machines as specified in 8.6.11.
e) Support the management entities for ATS as specified in 12.31.

5.13 MAC Bridge component requirements

5.13.1 MAC Bridge component options

5.13.1.1 Per-stream filtering and policing (PSFP) requirements

Change 5.13.1.1 as shown:

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~~8.6.5.2.1 and ~~8.6.6.1~~8.6.6 items d) and e).
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 subclause (5.13.1.3) after 5.13.1.2:

5.13.1.3 Asynchronous Traffic Shaping (ATS) requirements

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

a) Support per-stream classification and metering for ATS as specified in 8.6.5.2.2.
b) Support queuing with support for stream gates as specified in 8.6.6 items d) and e).
c) Support the ATS transmission selection algorithm as specified in 8.6.8.5.

d) Support the ATS scheduler state machines as specified in 8.6.11.
e) Support the management entities for ATS as specified in 12.31.

Change the title and text of 5.27 as shown:

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~~8.6.5.2.1.
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 subclause (5.32) after 5.31:

5.32 End station requirements—ATS

An end station implementation that conforms to the provisions of this standard for ATS shall

a) Support a minimum of two traffic classes, of which
   1) One traffic class supports the strict priority transmission selection algorithm (8.6.8.1) and
   2) One traffic class supports the ATS transmission selection algorithm (8.6.8.5).
b) Support the ATS talker transmission behavior, as specified in 49.1.

8. Principles of Bridge operation

8.6 The Forwarding Process

8.6.5 Flow classification and metering

Delete the text of 8.6.5 (as published in IEEE Std 802.1Q-2018). This deletion includes the introductory text of 8.6.5, 8.6.5.1 through 8.6.5.3, Figure 8-13, and Table 8-4.

Insert the following material in place of the deleted material described above. This insertion includes new introductory text for 8.6.5, new 8.6.5.1 through 8.6.5.6, new Figure 8-13 through Figure 8-15, and a revised Table 8-4. Renumber the subsequent figures in Clause 8 accordingly.

The Forwarding Process can apply flow classification and metering to frames that are received on a Bridge Port and have one or more potential transmission ports. Bridge ports and end stations may support Per-Stream Filtering and Policing (PSFP), Asynchronous Traffic Shaping (ATS) filtering and eligibility time assignment, or the general flow classification rules specified in 8.6.5.1.

NOTE—The general flow classification and metering specification was added to this standard by IEEE Std 802.1Q-2005, PSFP by IEEE Std 802.1Qci-2017, and ATS by IEEE Std 802.1Qcr-2020.

PSFP and ATS share common per-stream classification and metering elements, as shown in Figure 8-13. The stream identification function specified in IEEE Std 802.1CB-2017 can be used to associate received frames with these elements.

8.6.5.1 General flow classification and metering

Bridges that implement general flow classification and metering can identify subsets of traffic (frames), each subject to the same flow metering and forwarding. Flow classification rules may be based on the following:

a) Destination MAC address
b) Source MAC address
c) VID
d) Priority

Item c), specifying a VID value, is not applicable to VLAN-unaware MAC Relays.

Frames classified using the same set of classification rules are subject to the same flow meter. The flow meter can change the drop_eligible parameter associated with each frame and can discard frames on the basis of the following parameters for each received frame and previously received frames, and the time elapsed since those frames were received:

e) The received value of the drop_eligible parameter
f) The mac_service_data_unit size

The flow meter shall not base its decision on the parameters of frames received on other Bridge Ports, or on any other parameters of those Ports. The metering algorithm described in the Metro Ethernet Forum (MEF) Technical Specification 10.3 (MEF 10.3) should be used.

NOTE 1—The flow meter described here can encompass a number of meters, each with a simpler specification. However, given the breadth of implementation choice permitted, further structuring to specify, for example, that frames can bypass a meter or are subject only to one of a number of meters provides no additional information.

NOTE 2—Although flow metering is applied after egress filtering (Figure 8-12), the meter(s) operate per reception Port, not per potential transmission Port(s).

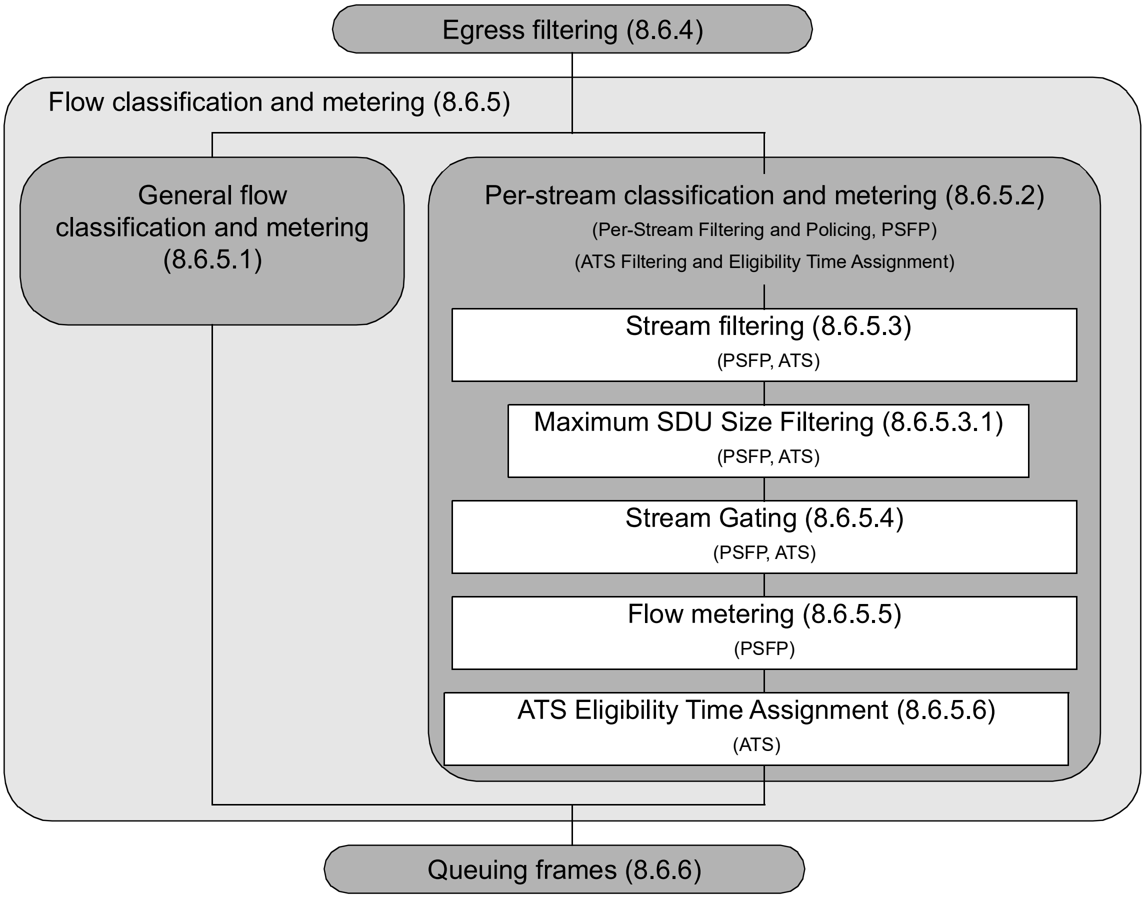


Figure 8-13—Flow classification and metering

8.6.5.2 Per-stream classification and metering

When Per-Stream Filtering and Policing (PSFP) or Asynchronous Traffic Shaping (ATS) is used, filtering and policing decisions for received frames are made, and subsequent queuing (8.6.6) and transmission selection decisions (8.6.8) supported, as follows:

a) Each received frame can be associated with a stream filter, as specified in 8.6.5.3. If a matching stream filter is specified (8.6.5.3), that is used to process the frame. Wildcard stream filters can be configured to match and discard frames not associated with a specified stream. If no matching stream filter is found, the frame is queued for transmission as specified in 8.6.6.
b) If the stream filter specifies maximum SDU size filtering (8.6.5.3.1), that is used to process the frame. The frame can be discarded if a maximum SDU size is exceed. The ATS scheduler state machine operation (8.6.11) assumes that the sizes of frames that it processes are less than or equal to the associated CommittedBurstSize parameter (8.6.11.3.5).
c) The stream filter specifies a stream gate (8.6.5.4), that is used to process the frame. The frame can be discarded if it is received outside of permitted reception intervals or a given data limit within a reception interval is exceeded. The frame’s priority can be mapped to an internal priority value (IPV) that can influence subsequent queuing decisions (8.6.6).
d) If the stream filter specifies a flow meter (8.6.5.5), that is used to process the frame. The frame can be discarded or marked as drop eligible if a traffic limit of a flow meter is exceeded. A given stream filter can be configured with flow meters and an ATS scheduler if both PSFP and ATS are supported.

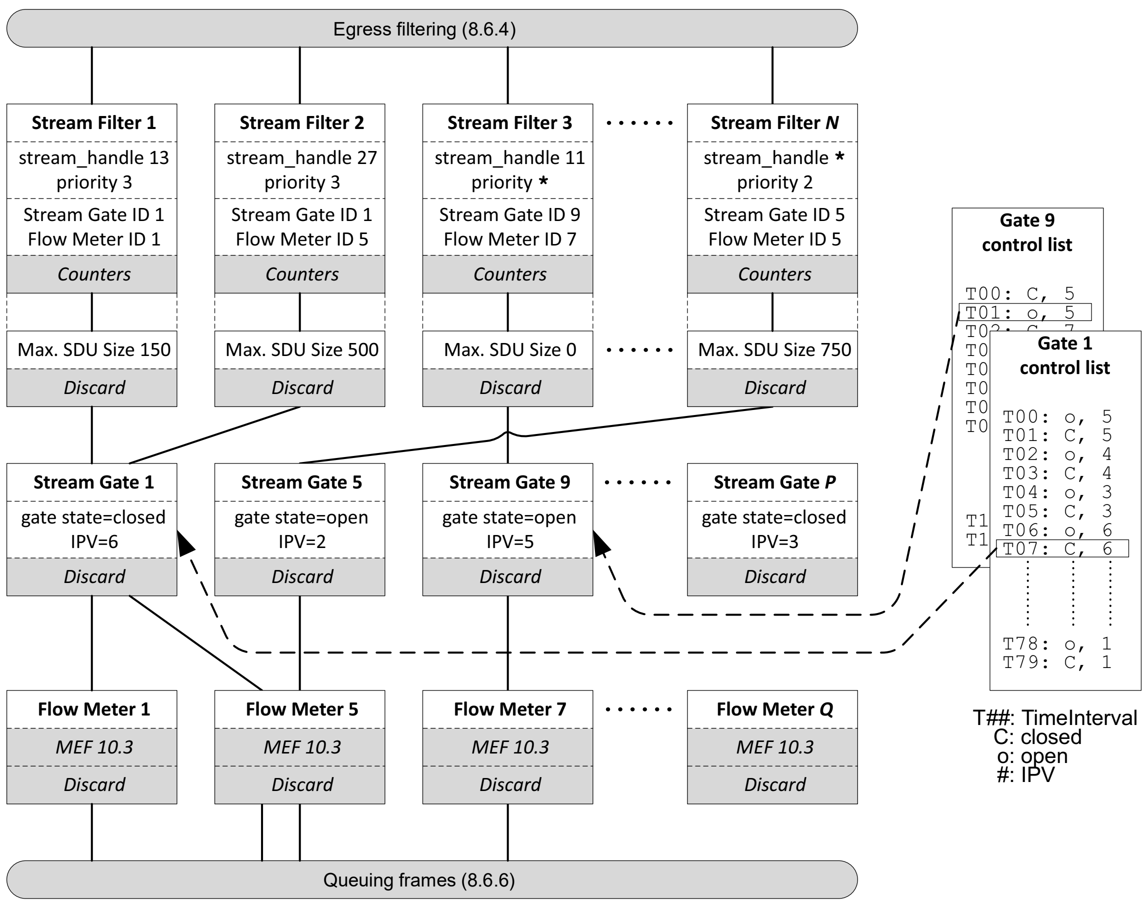
e) If the stream filter specifies an ATS scheduler (8.6.5.6), that is used to process the frame. It computes an eligibility time for the frame for subsequent use by the ATS transmission selection algorithm (8.6.8.5). The frame can be discarded if a maximum eligibility time is exceeded (8.6.11.3.13).

8.6.5.2.1 PSFP support

Each Bridge component or end station that implements PSFP shall support stream filtering, maximum SDU size filtering, stream gating, and flow metering, with the following:

a) A single *Stream Filter Instance Table* (8.6.5.3).
b) A single *Stream Gate Instance Table* (8.6.5.4).
c) A single *Flow Meter Instance Table* (8.6.5.5).

The relationship between stream filters, stream gates, flow meters for streams subject to PSFP processing (as identified by the stream filter) is illustrated by Figure 8-14 for a number of streams.



KEY
Counters: Matching, passing, and discarded frame counters (8.6.5.3).
Discard: Frame discarding abilities and parameters (8.6.5.3.1, 8.6.5.4, 8.6.5.5).
MEF 10.3: Flow metering based on MEF 10.3 Bandwidth Profile parameters and algorithm as specified in 8.6.5.5.

Figure 8-14—Per-stream classification for PSFP

8.6.5.2.2 ATS support

Each Bridge component that implements ATS shall support stream filtering, maximum SDU size filtering, stream gates supporting IPV assignment, ATS schedulers, and ATS scheduler groups, with the following:

a) A single *Stream Filter Instance Table* (8.6.5.3).
b) A single *Stream Gate Instance Table* (8.6.5.4).
If the Bridge component does not support PSFP in addition to ATS, each stream gate only supports IPV assignment (8.6.5.4). IPV can be used as part of adjusting per-hop delay bounds to meet specific networks' end-to-end delay requirements.
c) A single *ATS Scheduler Instance Table* (8.6.5.6).
d) A single *ATS Scheduler Group Instance Table* (8.6.5.6).
e) An *ATS Port Parameter Table* for each Bridge Port (8.6.5.6).

NOTE 1—The operation of ATS schedulers (8.6.11) provides policing capabilities similar to those provided by flow meters (8.6.5.5). This handling of improper traffic can be sufficient, such that no additional flow meters are required.

NOTE 2—For bridges with support for ATS, and without support for PSFP, stream gates of ATS traffic will never close. In this case, stream gates are only used for IPV assignment.

NOTE 3—In presence of state changes of stream gates associated with ATS traffic, the asynchronous nature of ATS traffic can require that state changes of multiple of these associated gates are executed simultaneously.

The relationship between stream filters, stream gates, and ATS schedulers for streams subject to ATS processing is illustrated by Figure 8-15.

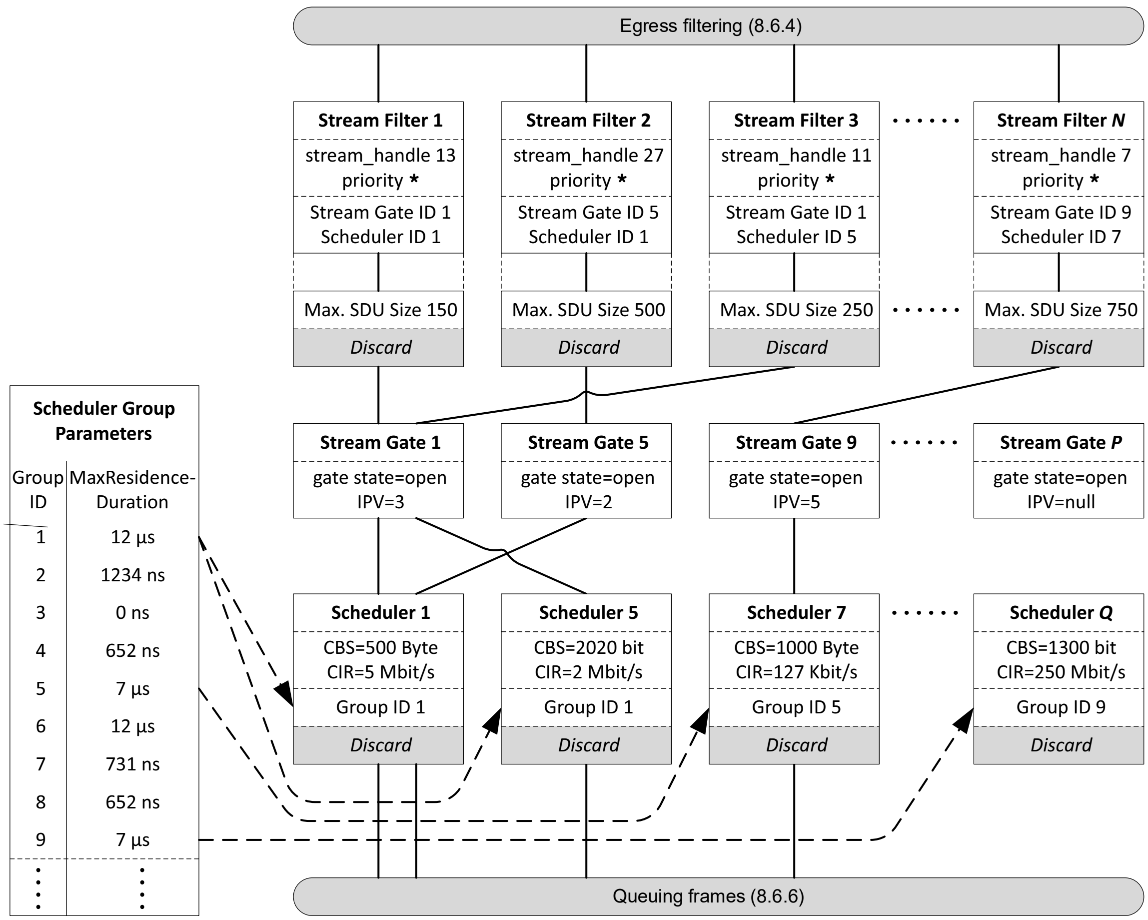
ATS support in end stations is provided by a modified variant of per-stream classification and metering for ATS, as specified in 49.1.

8.6.5.3 Stream filtering

A received frame can be associated with a stream filter using the frame's *stream_handle* and *priority* parameters. The *stream_handle* is a sub-parameter of the connection_identifier parameter of the ISS (6.6), provided by the stream identification function specified in Clause 6 of IEEE Std 802.1CB-2017 [B14].

Each stream filter comprises the following:

a) An integer *stream filter identifier*.
b) A *stream_handle* specification, either:
   1) A single value, as specified in IEEE Std 802.1CB.
   2) A wildcard, that matches any *stream_handle*.
c) A *priority* specification, either:
   1) A single priority value.
   2) A wildcard value that matches any priority value.
d) Maximum SDU size filtering (8.6.5.3.1) information, comprising:
   1) An integer *Maximum SDU size*, in octets. A value of 0 disables maximum SDU size filtering for this stream filter.
   2) A boolean *StreamBlockedDueToOversizeFrameEnable* parameter.
   3) A boolean *StreamBlockedDueToOversizeFrame* parameter.
e) An integer *stream gate identifier* (8.6.5.4).
f) An integer *flow meter instance identifier* (8.6.5.5).
If this parameter is absent, frames associated with the stream filter are not subject to flow metering.
g) An integer ATS *scheduler instance identifier* (8.6.5.6).
If this parameter is absent, frames associated with the stream filter are not subject to ATS scheduling and transmission selection.



KEY
Discard: Frame discarding abilities and parameters (8.6.5.3.1, 8.6.5.4, 8.6.5.6).
CBS: CommittedBurstSize parameter (8.6.5.6, 8.6.11.3.5).
CIR: CommittedInformationRate parameter (8.6.5.6, 8.6.11.3.6).

Figure 8-15—Per-stream classification and metering for ATS

Each stream filter also comprises the following counters for frames associated with the stream filter:

h) *MatchingFramesCount*: all frames associated with that stream filter.
i) *PassingSDUCount*: frames passing the maximum SDU size filter (8.6.5.3.1).
j) *NotPassingSDUCount*: frames not passing the maximum SDU size filter (8.6.5.3.1).
k) *PassingFrameCount*: frames passing the associated stream gate (8.6.5.4).
l) *NotPassingFrameCount*: frames not passing the stream gate (8.6.5.4).
m) *RedFramesCount*: frames discarded by the flow meter (8.6.5.5).

The *stream filter identifier* uniquely identifies the stream filter, indexing a *Stream Filter Instance Table* of up to *MaxStreamFilterInstances* stream filters. Each received frame is associated with the stream filter with the lowest *stream filter identifier* whose *stream_handle* and *priority* specification match the frame's parameters, and the *MatchingFramesCount* is incremented for that filter.

NOTE 1—The use of stream_handle and priority, along with the wild-carding rules previously stated, allow configuration possibilities that go beyond the selection of individual streams, for example, per-priority filtering and policing, or per-priority per-reception Port filtering and policing can be configured using these rules.

NOTE 2—If it is desired to discard frames that do not match any other stream filter, rather than such frames being processed without filtering, a stream filter can be placed at the end of the table with stream_handle and priority specifications both wild-carded and the stream gate identifier of a permanently closed stream gate.

NOTE 3—A discarded frame is not processed or counted by subsequent elements of the forwarding process. As a consequence, implementations can internally represent all counters [items h) through m) above] by the four counters MatchingFramesCount [item h) above], NotPassingSDUCount [item j) above], NotPassingFrameCount [item l) above], and RedFramesCount [item m) above].

8.6.5.3.1 Maximum SDU Size Filtering

If the SDU size of a frame exceeds the value of the associated stream filter's Maximum SDU size parameter, the frame is discarded and that stream filter's NotPassingSDUCount is incremented. If the stream filter's StreamBlockedDueToOversizeFrameEnable parameter is configured to be TRUE, the StreamBlockedDueToOversizeFrame parameter is set to TRUE and all subsequent frames will be discarded until StreamBlockedDueToOversizeFrame is administratively reset to FALSE.

Otherwise, the stream filter's PassingSDUCount is incremented (see 8.6.5.3). The default configuration of both StreamBlockedDueToOversizeFrameEnable and StreamBlockedDueToOversizeFrame is FALSE.

NOTE—The Maximum SDU size is defined per stream filter and can therefore differ from the queueMaxSDU specified in 8.6.8.4.As queueMaxSDU is applied after the flow classification and metering, it is possible that a frame that passes the Maximum SDU size filter will later be discarded because its SDU size exceeds queueMaxSDU.

8.6.5.4 Stream gating

Stream gates can discard frames whose reception times contradict a given time schedule. Stream gates can also map the frame's priority to an internal priority value (IPV) that is used to make subsequent queuing decisions (8.6.6), while retaining the frame's original priority for transmission.

NOTE 1—The IPV facilitates ATS per-hop delay bound adjustment to satisfy specific networks' end-to-end delay requirements. Annex (CQF) describes another IPV use case.

Each stream gate comprises the following:

a) An integer *stream gate instance identifier*.
b) An administrative and an operational *stream gate state* parameter. These parameters take the following values:
   1) Open: Frames are permitted to pass through the stream gate.
   2) Closed: Frames are not permitted to pass through the stream gate.
c) An administrative and an operational *internal priority value specification* parameter. These parameters take the following values:
   1) Null, in this case the received frame's priority parameter is used as the IPV.
   2) A specific IPV for the frame.

The *stream gate instance identifier* uniquely identifies the stream gate, indexing a *Stream Gate Instance Table* of up to *MaxStreamGateInstances* stream gates.

If PSFP are supported, each stream gate also includes the following:

d) An administrative and an operational *stream gate control list*.
e) A boolean *GateClosedDueToInvalidRxEnable* parameter.
f) A boolean *GateClosedDueToInvalidRx* parameter.
g) A boolean *GateClosedDueToOctetsExceededEnable* parameter.
h) A boolean *GateClosedDueToOctetsExceeded* parameter.

An instance of the stream gate control state machine (8.6.10) determines the operational values of the *stream gate state* and the *internal priority value specification* [items b) and c) above] by the cyclical execution of the control operations (see Table 8-4) specified in the stream gate's *stream gate control list* [item d) above]. The administrative *stream gate state* and *internal priority value specification* parameters are used to determine the initial values of the corresponding operational parameters, and the administrative *stream gate control list* parameter allows configuration of a new control list prior to enabling its use in a running system.

If a frame is passed by a stream gate, the *PassingFrameCount* of the stream filter (8.6.5.3) associated with that frame is incremented. The *NotPassingFrameCount* is incremented if the frame is discarded.

Table 8-4—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 value 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.

Stream gates are able to permanently discard frames and thus effectively override the operational gate state (i.e., the stream gate behaves as if the operational stream gate state is Closed). This capability is provided in case a frame is discarded due to a closed gate state [8.6.5.4 item b)], and in case a frame is discarded because there are insufficient octets left (i.e., the value of the IntervalOctetsLeft parameter, as specified in 8.6.10.8, is lower than the frame length).

i) Permanent frame discarding due to a frame received during a closed gate state is enabled if the GateClosedDueToInvalidRxEnable parameter is TRUE, and disabled if this parameter is FALSE. If enabled and any frame is discarded, then the GateClosedDueToInvalidRx parameter is set to TRUE, and all subsequent frames are discarded as long as the GateClosedDueToInvalidRxEnable and GateClosedDueToInvalidRx parameters are TRUE. Changes of the GateClosedDueToInvalidRxEnable parameter and transitions from TRUE to FALSE of the GateClosedDueToInvalidRx parameter are administrative actions.
j) Permanent frame discarding due to insufficient left octets is enabled if the GateClosedDueToOctetsExceededEnable parameter is TRUE, and disabled if this parameter is FALSE. If enabled and any frame is discarded because there are insufficient octets left, then the GateClosedDueToOctetsExceeded parameter is set to TRUE, and all subsequent frames are discarded as long as the GateClosedDueToOctetsExceededEnable and GateClosedDueToOctetsExceededEnable parameters are TRUE. Changes of the GateClosedDueToOctetsExceededEnable parameter and transitions from TRUE to FALSE of the GateClosedDueToOctetsExceeded parameter are administrative actions.

Per default, permanent frame discarding is disabled and all associated parameters have the default value FALSE.

NOTE 2—Permanent frame discarding allows the detection of incoming frames during time periods when the stream gate is in the closed state and exceptionally large ingress bursts to result in the stream gate behaving as it is in a permanently 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 with a limited overall amount of ingress octets. Hence, frames received by the stream gate when it is in the closed state and unexpected amounts of ingress octets represent invalid receive conditions.

8.6.5.5 Flow metering

The flow meters specified by this subclause (8.6.5.5) implement the parameters and algorithm specified in *Bandwidth Profile Parameters and Algorithm* in MEF 10.3 with the additions described in this subclause.

Each flow meter comprises the following:

a) An integer *flow meter identifier*.
b) An integer *Committed information rate (CIR)*, in bits per second (MEF 10.3).
c) An integer *Committed burst size (CBS)*, in octets (MEF 10.3).
d) An integer *Excess Information Rate (EIR)*, in bits per second (MEF 10.3).
e) An integer *Excess burst size (EBS) per bandwidth profile flow*, in octets (MEF 10.3).
f) A *Coupling flag (CF)*, which takes the value 0 or 1 (MEF 10.3).
g) A *Color mode (CM)*, which takes the value *color-blind* or *color-aware* (MEF 10.3).
h) A boolean *DropOnYellow* parameter.
i) A boolean *MarkAllFramesRedEnable* parameter.
j) A boolean *MarkAllFramesRed* parameter.

NOTE 1—This standard specifies the relationship among frame length, media-dependent overhead, ATS scheduler operation, and the associated parameters in 8.6.11. In contrast, the operation and parameters of flow meters in 8.6.5.5 are as specified by MEF 10.3.

NOTE 2—Envelope and Rank, as defined in MEF 10.3, are not used by the flow meters described in this subclause; i.e., the reduced functionality algorithm described in 12.2 of MEF 10.3 is used.

The *flow meter identifier* uniquely identifies the flow meter instance, indexing a *Flow Meter Instance Table* of up to *MaxFlowMeterInstances* flow meters.

The DropOnYellow parameter indicates whether frames marked yellow by the MEF 10.3 algorithm are discarded or marked as drop eligible:

k) A value of TRUE indicates that yellow frames are discarded.
l) A value of FALSE indicates that the drop_eligible parameter of yellow frames is set to TRUE.

NOTE 3—Changing the value of the drop_eligible parameter may change the contents of the frame, depending on how the frame is tagged when transmitted, which may then require updating the frame_check_sequence. Mechanisms for conveying information from ingress to egress that the frame_check_sequence may require updating are implementation dependent.

Flow meters can permanently discard all frames after an initial frame has been discarded. This capability is enabled if the MarkAllFramesRedEnable parameter is TRUE, and disabled if this parameter is FALSE. If enabled and the flow meter discards a frame, then the MarkAllFramesRed parameter is set to TRUE, and all subsequent frames are discarded as long as the MarkAllFramesRedEnable and MarkAllFramesRed parameters are TRUE.

Changes of the MarkAllFramesRedEnable parameter and transitions from TRUE to FALSE of the MarkAllFramesRed parameter are administrative actions. Per default, permanent frame discarding is disabled and both associated parameters have the default value FALSE.

Each time a flow meter discards a frame, the RedFramesCount counter of the originating stream filter (8.6.5.3) is increased.

8.6.5.6 ATS eligibility time assignment

Asynchronous Traffic Shaping (ATS) schedulers assign eligibility times to frames which are then used for traffic regulation by the ATS transmission selection algorithm (8.6.8.5).

NOTE 1—ATS schedulers, as defined in this subclause (8.6.5.6), realize the computational part of the overall traffic shaping operation of ATS. The complete operation is provided in combination with the ATS transmission selection algorithm (8.6.8.5), which uses the assigned eligibility times to regulate the traffic for transmission.

Each ATS scheduler comprises the following:

a) An integer *scheduler identifier*.
b) An integer *scheduler group identifier*.
c) An integer *CommittedBurstSizeParameter* parameter, in bits (8.6.11.3.5).
d) An integer *CommittedInformationRate* parameter, in bits per second (8.6.11.3.6).
e) An internal *bucket empty time* state variable, in seconds (8.6.11.3.3).

ATS schedulers are organized in *ATS scheduler groups*. There is one ATS scheduler group per reception Port per upstream traffic class, where the latter refers to the transmitting traffic class in the device connected to the given reception Port. All ATS schedulers that process frames from a particular reception Port and a particular upstream traffic class are in the respective ATS scheduler group.

Each ATS scheduler group comprises the following:

f) An integer *scheduler group identifier*.
g) An integer *MaximumResidenceTime* parameter, shared by all ATS schedulers in a scheduler group, in nanoseconds (8.6.11.3.13).
h) An internal *group eligibility time* state variable, shared by all ATS schedulers in a scheduler group, in seconds (8.6.11.3.10).

NOTE 2—The organization of ATS schedulers into groups results in a non-decreasing ordering of eligibility times of successive frames associated with a single group. This permits frames of one group to be queued in FIFO order.

Each ATS scheduler assigns eligibility times to the associated frames, and discards frames in exceptional situations. The underlying operations are performed by an ATS scheduler state machine (8.6.11) associated with an ATS scheduler. This state machine updates the associated bucket empty time and group eligibility time state variables based on the CommittedBurstSize, the CommittedInformationRate, the MaxResidenceTime, the frame arrival times, and the frame lengths (including media-specific overhead).

Each Port is associated with the following variable for ATS schedulers:

i) An integer *DiscardedFramesCount* of frames discarded by the ATS schedulers associated with that reception Port.

Each Bridge component provides an *ATS Scheduler Instance Table* of up to *MaxSchedulerInstances* ATS schedulers, an *ATS Scheduler Group Instance Table* of up to *MaxSchedulerGroupInstances* ATS scheduler groups, and a *ATS Scheduler Port Parameter Table* shared by all ATS schedulers associated with a reception Port.

NOTE 3—Whether ATS scheduler instances, ATS scheduler group instances, the scheduler instance table, and the scheduler group instance table are located in reception ports or in transmission ports is implementation specific.

8.6.6 Queuing frames

Change the fourth paragraph in 8.6.6 as shown (including splitting the paragraph into two paragraphs):

The Forwarding Process provides one or more queues for a given Bridge Port, each corresponding to a distinct traffic class. Each frame is mapped to a traffic class using the Traffic Class Table for the Port~~ and the frame's priority~~. The priority value used for this mapping is determined as follows:

c) If stream gates are unsupported (8.6.5.4), the frame's priority is used.
d) If stream gates are supported and the IPV specification assigned to the frame is the null value, the frame's priority is used.
e) If stream gates are supported and the IPV specification assigned to the frame is an IPV, this IPV is used.

Traffic class tables may be managed. Table 8-5 shows the recommended mapping for the number of classes implemented, in implementations that do not support the credit-based shaper transmission selection algorithm (8.6.8.2). The requirements for priority to traffic class mappings in implementations that support the credit-based shaper transmission selection algorithm are defined in 34.5. Up to eight traffic classes may be supported, allowing separate queues for each priority.

Change NOTE 3 in 8.6.6 as shown:

NOTE 3—A queue in this context is not necessarily a single FIFO data structure. A queue is a record of all frames of a given traffic class awaiting transmission on a given Bridge Port. ~~The structure of this record is not specified. ~~The transmission selection algorithm (8.6.8) determines which traffic class, among those classes with frames available for transmission, provides the next frame for transmission. ~~The method of determining which frame within a traffic class is the next available frame is not specified beyond conforming to the frame ordering requirements of this subclause. This allows a~~A variety of queue structures such as a single FIFO~~,~~ or a set of FIFOs with one for each ~~pairing of ingress and egress ports (i.e., Virtual Output Queuing), or a set of FIFOs with one for each ~~VLAN or priority~~, or hierarchical structures~~ is allowed.

Delete 8.6.6.1 ("PSFP queuing").

8.6.8 Transmission selection

Change the fourth paragraph of 8.6.8 as shown:

The strict priority transmission selection algorithm defined in 8.6.8.1 shall be supported by all Bridges as the default algorithm for selecting frames for transmission. The credit-based shaper transmission selection algorithm defined in 8.6.8.2, ~~and ~~the ETS algorithm defined in 8.6.8.3, and the ATS transmission selection algorithm defined in 8.6.8.5 may be supported in addition to the strict priority algorithm. Further transmission selection algorithms, selectable by management means, may be supported as an implementation option so long as the requirements of 8.6.6 are met.

Change Table 8-6 as shown:

Table 8-6—Transmission selection algorithm identifiers

Transmission selection algorithm	Identifier
Strict priority (8.6.8.1)	0
Credit-based shaper (8.6.8.2)	1
Enhanced Transmission Selection (ETS) (8.6.8.3)	2
ATS Transmission Selection (8.6.8.5)	3
Reserved for future standardization	~~3~~4–254
Vendor-specific Transmission Selection algorithm value for use with DCBX (D.2.8.8)	255
Vendor-specific	A four-octet integer, where the most significant 3 octets hold an OUI or CID value, and the least significant octet holds an integer value in the range 0–255 assigned by the owner of the OUI or CID.

8.6.8.4 Enhancements for scheduled traffic

Change the third paragraph of 8.6.8.4 as shown (including splitting the paragraph into two paragraphs and inserting a new Figure 8-17 and NOTE 1), and renumber the subsequent figures in Clause 8 accordingly:

In addition to the other checks carried out by the transmission selection algorithm, a frame on a traffic class queue is not available for transmission [as required for tests (a) and (b) in 8.6.8] if the transmission gate is in the closed state or if there is insufficient time available to transmit the entirety of that frame, plus the media-dependent overhead specified in 12.4.2.2, before the next gate-close event (3.97) associated with that queue (Figure 8-17).

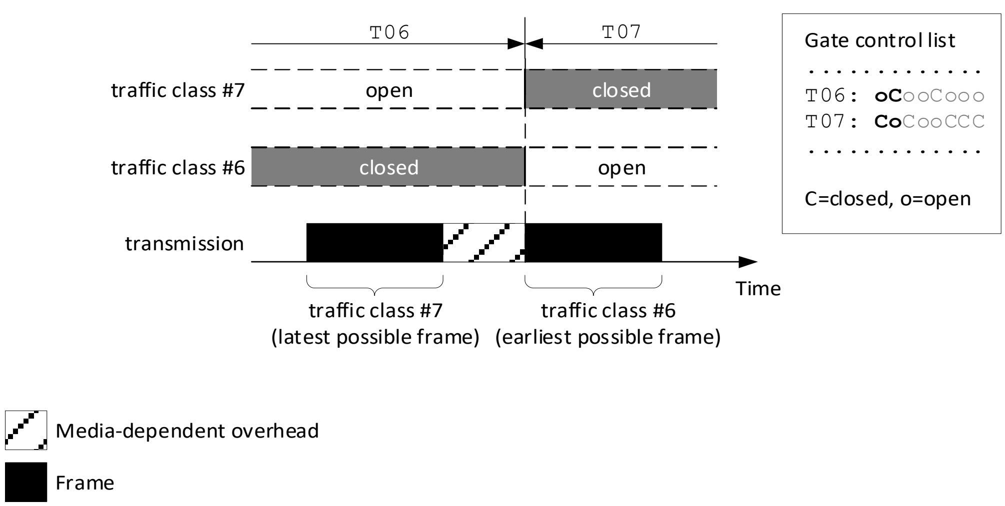


Figure 8-17—Frame timing at gate-close events

NOTE 1—For example, in the case of IEEE 802.3 media, the media-dependent overhead prior to the gate-close event includes the preamble of a potentially following frame from a different traffic class.

A per-traffic class counter, TransmissionOverrun (12.29.1.1.2), is incremented if the implementation detects that a frame from a given queue is still being transmitted by the MAC when the gate-close event for that queue occurs.

Change the former NOTE 1 of 8.6.8.4 as shown, and renumber the subsequent notes in this subclause accordingly:

NOTE ~~1~~2—~~It is assumed that the implementation has knowledge of the transmission overheads that are involved in transmitting a frame on a given Port and can therefore determine how long the transmission of a frame will take. However, t~~There can be reasons that the time required to complete the transmission of a frame is unknown. For example, where frame preemption is supported and there is no way of telling in advance how many times the given frame will be preempted before this transmission is complete. ~~why the frame size, and therefore the length of time needed for its transmission, is unknown; for example, where cut-through is supported, or where frame preemption is supported and there is no way of telling in advance how many times a given frame will be preempted before its transmission is complete.~~ It is desirable that the schedule for such traffic is designed to accommodate the intended pattern of transmission without overrunning the next gate-close event for the traffic classes concerned.

Insert the following subclause (8.6.8.5) after 8.6.8.4:

8.6.8.5 ATS transmission selection algorithm

For a given queue that supports ATS transmission selection, the algorithm determines that a frame is available for transmission if the queue contains one or more frames eligible for transmission. A frame is eligible for transmission if the assigned eligibility time (8.6.5.6, 8.6.11.3.2) is earlier than or at the current time.

The current time is determined by the TransmissionSelection Clock, which is an implementation specific local system clock function. The TransmissionSelection Clock determines the selectability time per frame, which is the time at which this frame is queued (8.6.6) and available for transmission selection. The selectability time is used as a reference to specify the handling of device-internal implementation specific timing properties (8.6.11.3.2).

All frames that reach their selectability time are selected for transmission in ascending order of the assigned eligibility times. Transmission selection of frames with identical assigned eligibility times shall maintain the ordering requirement specified in 8.6.6.

NOTE—In the case of frames with non-identical eligibility times, the ordering requirement from 8.6.6 is automatically satisfied due the operation of the ATS scheduler state machines (8.6.11), which assign eligibility times in a non-decreasing order.

8.6.10 Stream gate control state machines

Change the introductory text of 8.6.10 (including Table 8-8) as shown:

The execution of the gate operations in a stream gate control list (~~8.6.5.1.2~~8.6.5.4) is controlled by the three state machines specified in 8.6.9:

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~~stream gates. An overview of the operation of these state machines can be found in Figure 8-18.[2]

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~~stream gate versions of these procedures/variables.

Three additional variables needed by the Execute~~PSFP~~StreamGateOperation procedure are defined in 8.6.10.6 and 8.6.10.7.

Table 8-8—Scheduled Traffic and ~~PSFP~~Stream Gate procedures/variables

Procedure/variable name in 8.6.9	~~PSFP~~Stream Gate procedure/variable name
ExecuteOperation() (8.6.9.2.1)	Execute~~PSFP~~StreamGateOperation() (8.6.10.1)
SetGateStates() (8.6.9.2.2)	Set~~PSFP~~StreamGateStates() (8.6.10.2)
ListPointer (8.6.9.4.15)	~~PSFP~~StreamGateListPointer (8.6.9.2.2)
AdminGateStates (8.6.9.4.5)	~~PSFP~~StreamGateAdminGateStates (8.6.10.4)
OperGateStates (8.6.9.4.22)	~~PSFP~~StreamGateOperGateStates (8.6.10.5)

Change 8.6.10.1 through 8.6.10.8 as shown:

8.6.10.1 Execute~~PSFP~~StreamGateOperation()

The Execute~~PSFP~~StreamGateOperation() 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 ~~PSFP~~StreamGateListPointer variable (8.6.9.2.2) is used as an index into OperControlList. The procedure processes the operation according to its operation name (Table 8-4) as follows:

a) If the operation name is SetGateAndIPV, then the StreamGateState parameter value associated with the operation is assigned to the ~~PSFP~~StreamGateOperGateStates 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 ~~PSFP~~StreamGateListPointer 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.

[2] *Editor's note:* Figure 8-15 (Scheduled traffic state machines—overview and relationships) as published in IEEE Std 802.1Q-2018 has been renumbered to Figure 8-18 because this amendment inserted new figures earlier in this clause.

8.6.10.2 Set~~PSFP~~StreamGateStates()

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

8.6.10.3 ~~PSFP~~StreamGateListPointer

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 8-4). 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 ~~PSFP~~StreamGateAdminGateStates

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 ~~PSFP~~StreamGateAdminGateStates variable. The default value of ~~PSFP~~StreamGateAdminGateStates is open. The value of ~~PSFP~~StreamGateAdminGateStates can be changed by management.

8.6.10.5 ~~PSFP~~StreamGateOperGateStates

The current state of the gate associated with the stream gate. ~~PSFP~~StreamGateOperGateStates is set by the List Execute state machine (8.6.9.2), and its initial value is determined by the value of the ~~PSFP~~StreamGateAdminGateStates 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~~variable indicates how many more MSDU octets can be passed by the stream gate during the current TimeInterval. This variable is initialized by the Execute~~PSFP~~StreamGateOperation() 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.

Insert the following subclauses (8.6.11 through 8.6.11.3.14) after 8.6.10.8:

8.6.11 ATS Scheduler state machines

The ATS scheduler state machine operation is based on the ATS scheduler clocks (8.6.11.1), which is in relationship to transmission selection clocks (8.6.8.5). The state machine operation is specified by the ProcessFrame(frame) procedure (8.6.11.3) and the therein used state variables (8.6.11.3.3, 8.6.11.3.10). Parameters used by the ProcessFrame(frame) procedure are as defined in 8.6.5.6. Each arriving frame causes invocation of the ProcessFrame(frame) procedure.

The ATS scheduler state machine operation is based on a token bucket shaping algorithm as described in Tanenbaum and Wetherall [B86]. This subclause and its subclauses refer to the token bucket shaping algorithm, and the underlying terms, for informative explanation only. Specification of the ATS scheduler state machine operation does not depend on these explanations.

8.6.11.1 ATS Scheduler Clocks

An ATS scheduler clock is an implementation specific local system clock function. It is used to determine the arrival time of frames (8.6.11.3.1). A Bridge component may utilize one or more ATS scheduler clock instances. In the case of multiple scheduler clock instances, all ATS scheduler instances associated with the same reception Port share the same ATS scheduler clock instance (i.e., the arrival time of all frames received from a particular reception port is determined by the same ATS scheduler clock instance).

8.6.11.2 Relationship between ATS Scheduler Clocks and Transmission Selection Clocks

ATS scheduler clocks and transmission selection clocks (8.6.8.5) run at the same rate with bounded offset variation. A difference between an arbitrary instant of time t_{FA}, as recognized by an ATS scheduler clock instance, and the same instant of time t_{TS}, as recognized by a transmission selection clock instance, may be observed during the processing of a frame. The time difference may vary over a sequence of frames, which is characterized by the following equation:

$$\text{ClockOffsetMin} \le t_{TS} - t_{FA} \le \text{ClockOffsetMax}$$

where ClockOffsetMin and ClockOffsetMax are implementation specific constants that limit the variation. The range is characterized by ClockOffsetVariationMax (12.31.8.3) as follows:

$$\text{ClockOffsetVariationMax} = \text{ClockOffsetMax} - \text{ClockOffsetMin}$$

NOTE 1—ClockOffsetMin and ClockOffsetMax capture implementation specific properties such as the resolution of the associated clocks, associated rounding errors, constant offsets between clocks, Bridge-internal synchronization inaccuracies in presence of different underlying oscillators, and similar.

A pair of a scheduler clock instance and a transmission selection clock instance has an implementation specific nominal rate, and a maximal absolute deviation from this nominal rate during operation, as characterized by the ClockRateDeviationMax parameter (12.31.8.4).

NOTE 2—ClockRateDeviationMax captures implementation specific properties such as oscillator rate deviation, numeric resolution for the operations specified in 8.6.11.3, and similar.

NOTE 3—ATS scheduler clocks and transmission selection clocks provide a model to express different sources of delay, delay variation, and inaccuracy. It is not required to implement different multiple physical oscillators/clocks (i.e., ATS scheduler clocks and transmission selection clocks can actually be the same physical clock or can be generated from the same oscillator), but the model captures the properties of implementations with and without different physical oscillators/clocks in a unified manner.

8.6.11.3 ProcessFrame(frame)

This procedure computes eligibility time, assigns the eligibility times to frames, and updates the ATS scheduler state machine variables.

The procedure is described by the following pseudo-code in a neutral manner: The arithmetic precision, and the resolution of variables, are implementation specific, unless externally visible by management (12.31). The impact of the associated inaccuracies is discussed in Annex .

```
ProcessFrame(frame) {
       lengthRecoveryDuration     = length(frame)/
                                    CommittedInformationRate;
       emptyToFullDuration        = CommittedBurstSize/
                                    CommittedInformationRate;
       schedulerEligibilityTime   = BucketEmptyTime +
                                    lengthRecoveryDuration;
       bucketFullTime             = BucketEmptyTime +
                                    emptyToFullDuration;
       eligibilityTime            = max(arrivalTime(frame),
                                        GroupEligibilityTime,
                                        schedulerEligibilityTime);

       if (eligibilityTime <= (arrivalTime(frame) + MaxResidenceTime/1.0e9)){
              // The frame is valid
              GroupEligibilityTime = eligibilityTime;
              BucketEmptyTime      = (eligibilityTime < bucketFullTime) ?
                     schedulerEligibilityTime :
                     schedulerEligibilityTime + eligibilityTime - bucketFullTime;
              AssignAndProceed(frame,eligibilityTime);
       } else {
              // The frame is invalid
              Discard(frame);
       }
}
```

8.6.11.3.1 arrivalTime(frame)

The arrival time of the frame, in seconds. The arrival time refers to the instant of time at which the associated ATS scheduler clock instance recognizes the arrival of the entire frame.

The point at which this recognition happens on the path between reception Ports and transmission Ports is implementation dependent. The earliest option is when the frame passes the boundary between the network physical medium and a reception Port, the latest option is during invocation of the ProcessFrame procedure.

For all frames arriving at all reception Ports, the arrival time is determined relative to the frame end. The maximum delay between the time a frame passes the boundary between the network physical medium and its subsequent recognition by the associated ATS scheduler clock instance is implementation specific, and is characterized by the ArrivalRecognitionDelayMax parameter (12.31.8.5).

The order of all frames associated with one ATS scheduler group at the boundary between the network physical medium and a reception Port is the same as the order of the arrival times determined for these frames.

NOTE—For example, the arrival time of frames may be determined based on invocation of the M_UNITDATA.indication service primitive of the ISS (6.6), as specified in IEEE Std 802.1AC.

8.6.11.3.2 AssignAndProceed(frame,eligibilityTime)

This procedure assigns an eligibility time to a frame for further processing by the transmission selection (8.6.8.5).

The assigned eligibility time, as used for ATS transmission selection decisions (8.6.8.5), is derived from the eligibilityTime parameter. The calculation of the assigned eligibility time accounts for variations between the associated ATS scheduler clock instance and transmission selection clock instance (8.6.11.2), and processing delays through the forwarding process, as detailed in the following.

Any frame may experience an additional, non-negative, processing delay, between its arrival time recognition by the associated ATS scheduler clock instance and its selectability time (8.6.8.5). This delay may vary per frame, thus that there is a delay variation over a sequence of frames. The processing delay is characterized by the following equation:

$$\text{ProcessingDelayMin} \leq \text{processingDelay(frame)} \leq \text{ProcessingDelayMax}$$

where ProcessingDelayMin (12.31.8.6) and ProcessingDelayMax (12.31.8.7) are implementation specific parameters, and processingDelay(frame) denotes the processing delay of an actual frame, including any potential delay introduced by the operation of the associated ATS scheduler state machine instance.

The assigned eligibility time is calculated as follows:

$$\text{assignedEligibilityTime} = \text{eligibilityTime} + \text{ClockOffsetMin} + \text{ProcessingDelayMax}$$

8.6.11.3.3 BucketEmptyTime

A state variable that contains the most recent instant of time at which the token bucket of the ATS scheduler instance was empty, in seconds.

The BucketEmptyTime variable is initialized with a time earlier than CommittedBurstSize/ CommittedInformationRate in the past, as perceived by the ATS Scheduler Clock. After initialization, the number of tokens in the token bucket is equivalent to the CommittedBurstSize parameter.

8.6.11.3.4 bucketFullTime

The instant of time when the number of tokens in the token bucket is equivalent to the CommittedBurstSize parameter, in seconds.

8.6.11.3.5 CommittedBurstSize

The committed burst size of the ATS scheduler instance, in bits (8.6.5.6). The CommittedBurstSize parameter defines the maximum token capacity of the token bucket. In the token bucket model, the number of tokens removed from the bucket by a frame equals the length of the frame, as defined in 8.6.11.3.11.

8.6.11.3.6 CommittedInformationRate

The committed information rate of the ATS scheduler instance, in bits per second (8.6.5.6). The CommittedInformationRate parameter defines the rate at which the token bucket is refilled with tokens until the maximum token capacity of the token bucket is reached.

8.6.11.3.7 Discard(frame)

This procedure discards the frame and increases the DiscardedFramesCount counter of the associated reception port (8.6.5.6). The procedure is called in exceptional situations only (e.g., misbehavior of the connected upstream system).

8.6.11.3.8 eligibilityTime

The eligibility time of the frame, without taking the implementation specific device-internal timing properties of the forwarding process into account. These timing properties are taken into account by the AssignAndProceed(frame) procedure (8.6.11.3.2).

8.6.11.3.9 emptyToFullDuration

The duration required to accumulate a number of tokens equivalent to the CommittedBurstSize parameter, in seconds.

8.6.11.3.10 GroupEligibilityTime

A state variable that contains the most recent value of the eligibilityTime variable from the previous frame, as processed by any ATS scheduler instance in the same ATS scheduler group, in seconds.

The GroupEligibilityTime variable is initialized with a time earlier or equal to the current time, as perceived by the ATS scheduler clock.

8.6.11.3.11 length(frame)

The length of the frame, including all media-dependent overhead (12.4.2.2), in bits.

8.6.11.3.12 lengthRecoveryDuration

The duration required to accumulate a number of tokens equivalent to length(frame), in seconds.

8.6.11.3.13 MaxResidenceTime

The MaximumResidenceTime parameter of the ATS scheduler group instance associated with the ATS scheduler instance, in nanoseconds (8.6.5.6). The parameter limits the duration for which frames can reside in a Bridge.

NOTE—A consistent setup of MaxResidenceTime parameter can be determined by the per hop delay bound, which is a result of the timing analysis (Annex).

8.6.11.3.14 schedulerEligibilityTime

The instant of time when the number of tokens in the token bucket is at least equivalent to arrivalTime(frame), in seconds.

12. Bridge management

12.4 Bridge Management Entity

12.4.2 Port configuration

Change Table 12-2 as shown:

Table 12-2—Port table entry

Name	Data Type	Operations supported[a]	References
portComponentId	ComponentID	R	12.4.1.5
portPortNumber	Port Number	R	13.25
portMACAddress	MAC address	R	12.4.1.1.3 a)
portDelayExceededDiscards	counter	R	—
portMtuExceededDiscards	counter	R	—
portCapabilities	unsigned	R	—
portTypeCapabilities	unsigned	R	—
portType	enumerated	R	12.4.2.1
portExternal	Boolean	R	—
portAdminPointToPoint	unsigned	RW	IEEE Std 802.1AC
portOperPointToPoint	Boolean	R	IEEE Std 802.1AC
portName	Latin1 String (SIZE(0..32))	RW	—
portMediaDependentOverhead	unsigned	R	12.4.2.2

[a] R = Read-only access; RW = Read/Write access.

Insert the following subclause (12.4.2.2) after 12.4.2.1:

12.4.2.2 Media-dependent overhead

The portMediaDependentOverhead parameter provides the number of additional octets for media-dependent framing. The overhead includes

a) All octets prior to the first octet of the Destination Address field and
b) The minimum number of octets after the last octet of the frame check sequence before a subsequent frame, including its media-dependent overhead, can be transmitted.

NOTE—An example of media-dependent overhead is 20 octets to account preamble, start of frame delimiter, and a minimal inter-frame gap (IFG) of 12 octets for IEEE 802.3 point-to-point media.

Change 12.31 as shown. This change includes subclause titles, introductory text, previously existing and new subclauses, and previously existing and new tables; see also the subsequent instructions for the insertion of 12.31.5 through 12.31.8.

12.31 Managed objects for per-stream ~~filtering~~classification and ~~policing~~metering

The Bridge enhancements for support of per-stream ~~filtering,~~classification and ~~policing~~metering are defined in ~~8.6.5.1~~8.6.5.2, 8.6.5.3, 8.6.5.4, 8.6.5.5, and ~~the~~8.6.5.6. The associated state machines are defined in 8.6.10 and 8.6.11.

This managed resource comprises the following objects:

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)
e) The Scheduler Instance Table (12.31.5)
f) The Scheduler Group Instance Table (12.31.6)
g) The Scheduler Port Parameter Table (12.31.7)
h) The Scheduler Timing Characteristics Table (12.31.8)

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~~8.6.5.2.1) and ATS (8.6.5.2.2), as detailed in Table 12-34.[3] Tables can be created or removed dynamically in implementations that support dynamic configuration of Bridge components.

Table 12-34—The Stream Parameter Table

Name	Data type	Operations supported[a]	Conformance[b]	References
MaxStreamFilterInstances	integer	R	~~BE~~PSFP, ATS	~~8.6.5.1~~8.6.5.3, 12.31.2
MaxStreamGateInstances	integer	R	~~BE~~PSFP, ATS	~~8.6.5.1~~8.6.5.4, 12.31.3
MaxFlowMeterInstances	integer	R	~~BE~~PSFP, ats	~~8.6.5.1~~8.6.5.5, 12.31.4
SupportedListMax	integer	R	~~BE~~PSFP, ats	~~8.6.5.1~~8.6.5.4, 12.31.4
MaxSchedulerInstances	integer	R	psfp, ATS	8.6.5.6, 12.31.5
MaxSchedulerGroupInstances	integer	R	psfp, ATS	8.6.5.6, 12.31.6

[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.~~
PSFP = Required for Bridge, Bridge component, or end station support of PSFP.
psfp = Optional for Bridge, Bridge component, or end station support of PSFP.
ATS = Required for Bridge or Bridge component support of ATS.
ats = Optional for Bridge or Bridge component support of ATS.

[3] *Editor's note:* Table 12-31 through Table 12-34 as published in IEEE Std 802.1Q-2018 have been renumbered to Table 12-34 through Table 12-38, respectively, because IEEE Std 802.1Qcc-2018 inserted new tables earlier in this clause.

12.31.1.1 MaxStreamFilterInstances

The maximum number of Stream Filter instances supported by this Bridge component (8.6.5.3).

12.31.1.2 MaxStreamGateInstances

The maximum number of Stream Gate instances supported by this Bridge component (8.6.5.4).

12.31.1.3 MaxFlowMeterInstances

The maximum number of Flow Meter instances supported by this Bridge component (8.6.5.5).

12.31.1.4 SupportedListMax

The maximum value supported by this Bridge component of the AdminControlListLength (8.6.9.4.6) and OperControlListLength (8.6.9.4.23) parameters. It is available for use by schedule computation software to determine the Bridge component's control list capacity prior to computation.

12.31.1.5 MaxSchedulerInstances

The maximum number of ATS scheduler instances supported by this Bridge component (8.6.5.6).

12.31.1.6 MaxSchedulerGroupInstances

The maximum number of ATS scheduler group instances supported by this Bridge component (8.6.5.6).

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~~8.6.5.3), as detailed in Table 12-35. 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 12-35—Stream Filter Instance Table

Name	Data type	Operations supported[a]	Conformance[b]	References
StreamFilterInstance	integer	R	~~BE~~PSFP, ATS	~~8.6.5.1~~8.6.5.3
StreamHandleSpec	stream_handle specification	RW	~~BE~~PSFP, ATS	~~8.6.5.1~~8.6.5.3
PrioritySpec	priority specification	RW	~~BE~~PSFP, ATS	~~8.6.5.1~~8.6.5.3
MaximumSDUSize	integer	RW	PSFP, ATS	8.6.5.3.1, 12.31.2.4
StreamGateInstanceID	integer	RW	~~BE~~PSFP, ATS	~~8.6.5.1~~8.6.5.3, ~~8.6.5.1.2~~ 8.6.5.4
~~FilterSpecificationList~~	~~sequence of FilterSpecification~~	~~RW~~	~~BE~~	~~8.6.5.1, 8.6.5.1.3, 12.31.2.6~~

Table 12-35—Stream Filter Instance Table *(continued)*

Name	Data type	Operations supported[a]	Conformance[b]	References
FlowMeterInstanceID	integer	RW	PSFP, ats	8.6.5.5, 12.31.2.6
FlowMeterEnable	Boolean	RW	PSFP, ats	8.6.5.5, 12.31.2.6
SchedulerInstanceID	integer	RW	psfp, ATS	8.6.5.6, 12.31.2.7
SchedulerEnable	Boolean	RW	psfp, ATS	8.6.5.6, 12.31.2.7
MatchingFramesCount	counter	R	~~BE~~PSFP, ats	~~8.6.5.1~~8.6.5.3
PassingFramesCount	counter	R	~~BE~~PSFP, ats	~~8.6.5.1~~8.6.5.3, 8.6.5.4
NotPassingFramesCount	counter	R	~~BE~~PSFP, ats	~~8.6.5.1~~8.6.5.3, 8.6.5.4
PassingSDUCount	counter	R	~~BE~~PSFP, ats	~~8.6.5.1~~8.6.5.3, 8.6.5.3.1
NotPassingSDUCount	counter	R	~~BE~~PSFP, ats	~~8.6.5.1~~8.6.5.3, 8.6.5.3.1
REDFramesCount	counter	R	~~BE~~PSFP, ats	~~8.6.5.1~~8.6.5.3, 8.6.5.5
StreamBlockedDueToOversizeFrameEnable	Boolean	RW	~~BE~~PSFP, ATS	~~8.6.5.1~~8.6.5.3, 8.6.5.3.1
StreamBlockedDueToOversizeFrame	Boolean	RW	~~BE~~PSFP, ATS	~~8.6.5.1~~8.6.5.3, 8.6.5.3.1

[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.~~
PSFP = Required for Bridge, Bridge component, or end station support of PSFP.
psfp = Optional for Bridge, Bridge component, or end station support of PSFP.
ATS = Required for Bridge or Bridge component support of ATS.
ats = Optional for Bridge or Bridge component support of ATS.

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 MaxSDUSize

The MaxSDUSize parameter defines the maximum SDU size to be accepted by the stream filter (8.6.5.3). A value of 0 disables the maximum SDU size filtering for frames associated with the stream filter.

~~12.31.2.4 StreamGateInstance~~

12.31.2.5 StreamGateInstanceID

The StreamGateInstanceID parameter identifies the stream gate instance (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 Filter Specification 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).~~

12.31.2.6 FlowMeterInstanceID and FlowMeterEnable

If FlowMeterEnable is set to TRUE, the FlowMeterInstanceID parameter identifies the flow meter instance (12.31.4) that is associated with the stream filter. The relationship between stream filters and flow meters is many to one; a given stream filter can be associated with only one flow meter, but there can be multiple stream filters associated with a given flow meter. If FlowMeterEnable is set to FALSE, no flow meter instance is associated with the stream filter.

12.31.2.7 SchedulerInstanceID and SchedulerEnable

If SchedulerEnable is set to TRUE, the SchedulerInstanceID parameter identifies the ATS scheduler instance (12.31.5) that is associated with the stream filter. The relationship between stream filters and ATS schedulers is many to one; a given stream filter can be associated with only one ATS scheduler, but there can be multiple stream filters associated with a given ATS scheduler instance. If SchedulerEnable is set to FALSE, no ATS scheduler instance is associated with the stream filter.

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 (~~8.5.6.1.2~~8.6.5.4), as detailed in Table 12-36. 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 12-36—The Stream Gate Instance Table

Name	Data type	Operations supported[a]	Conformance[b]	References
StreamGateInstance	integer	R	~~BE~~PSFP, ATS	~~8.6.5.1,~~ ~~8.6.5.1.2~~ 8.6.5.4
~~PSFP~~StreamGateEnabled	Boolean	RW	~~BE~~PSFP, ATS	8.6.9.4.14
~~PSFP~~StreamGateAdminGateStates	~~PSFPg~~StreamGateStatesValue	RW	~~BE~~PSFP, ATS	8.6.10.4, 12.29.1.2.2
~~PSFP~~StreamGateOperGateStates	~~PSFPg~~StreamGateStatesValue	R	~~BE~~PSFP, ATS	8.6.10.5, 12.29.1.2.2
~~PSFP~~StreamGateAdminControlListLength	unsigned integer	RW	~~BE~~PSFP, ats	8.6.9.4.6, 12.31.3.2
~~PSFP~~StreamGateOperControlListLength	unsigned integer	R	~~BE~~PSFP, ats	8.6.9.4.23, 12.31.3.2
~~PSFP~~StreamGateAdminControlList	sequence of ~~PSFP~~StreamGateControlEntry	RW	~~BE~~PSFP, ats	8.6.9.4.2, 12.31.3.2, 12.31.3.2.2
~~PSFP~~StreamGateOperControlList	sequence of ~~PSFP~~StreamGateControlEntry	R	~~BE~~PSFP, ats	8.6.9.4.19, 12.31.3.2, 12.31.3.2.2
~~PSFP~~StreamGateAdminCycleTime	RationalNumber	RW	~~BE~~PSFP, ats	8.6.9.4.3, 12.29.1.3
~~PSFP~~StreamGateOperCycleTime	RationalNumber (seconds)	R	~~BE~~PSFP, ats	8.6.9.4.20, 12.29.1.3
~~PSFP~~StreamGateAdminCycleTimeExtension	Integer (nanoseconds)	RW	~~BE~~PSFP, ats	8.6.9.4.4
~~PSFP~~StreamGateOperCycleTimeExtension	Integer (nanoseconds)	R	~~BE~~PSFP, ats	8.6.9.4.21
~~PSFP~~StreamGateAdminBaseTime	PTPtime	RW	~~BE~~PSFP, ats	8.6.9.4.1, 12.29.1.4
~~PSFP~~StreamGateOperBaseTime	PTPtime	R	~~BE~~PSFP, ats	8.6.9.4.18, 12.29.1.4
~~PSFP~~StreamGateConfigChange	Boolean	RW	~~BE~~PSFP, ats	8.6.9.4.7
~~PSFP~~StreamGateConfigChangeTime	PTPtime	R	~~BE~~PSFP, ats	8.6.9.4.9, 12.29.1.4
~~PSFP~~StreamGateTickGranularity	Integer (tenths of nanoseconds)	R	~~BE~~PSFP, ats	8.6.9.4.16
~~PSFP~~StreamGateCurrentTime	PTPtime	R	~~BE~~PSFP, ats	8.6.9.4.10, 12.29.1.4
~~PSFP~~StreamGateConfigPending	Boolean	R	~~BE~~PSFP, ats	8.6.9.3, 8.6.9.4.8
~~PSFP~~StreamGateConfigChangeError	Integer	R	~~BE~~PSFP, ats	8.6.9.3.1

Table 12-36—The Stream Gate Instance Table *(continued)*

Name	Data type	Operations supported[a]	Conformance[b]	References
~~PSFP~~StreamGateAdminIPV	IPV	RW	~~BE~~PSFP, ATS	~~8.6.5.1.2~~ 8.6.5.4, 8.6.10.6, 12.31.3.3
~~PSFP~~StreamGateOperIPV	IPV	RW	~~BE~~PSFP, ats	~~8.6.5.1.2~~ 8.6.5.4, 8.6.10.7, 12.31.3.3
~~PSFP~~StreamGateClosedDueToInvalidRxEnable	Boolean	RW	~~BE~~PSFP, ats	~~8.6.5.1.2~~ 8.6.5.4
~~PSFP~~StreamGateClosedDueToInvalidRx	Boolean	RW	~~BE~~PSFP, ats	~~8.6.5.1.2~~ 8.6.5.4
~~PSFP~~StreamGateClosedDueToOctetsExceededEnable	Boolean	RW	~~BE~~PSFP, ats	~~8.6.5.1.2~~ 8.6.5.4
~~PSFP~~StreamGateClosedDueToOctetsExceeded	Boolean	RW	~~BE~~PSFP, ats	~~8.6.5.1.2~~ 8.6.5.4

[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.~~
PSFP = Required for Bridge, Bridge component, or end station support of PSFP.
psfp = Optional for Bridge, Bridge component, or end station support of PSFP.
ATS = Required for Bridge or Bridge component support of ATS.
ats = Optional for Bridge or Bridge component support of ATS.

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 8-4. Each entry in the list is structured as a GateControlEntry (12.31.3.2.2).

12.31.3.2.1 ~~PSFPg~~StreamGateStatesValue

The ~~PSFPg~~StreamGateStatesValue indicates the desired gate state, *open* or *closed*, for the stream gate.

12.31.3.2.2 ~~PSFP~~StreamGateControlEntry

A ~~PSFP~~StreamGateControlEntry consists of an operation name, followed by three parameters associated with the operation, as detailed in Table 8-4. The first parameter is a ~~PSFPg~~StreamGateStatesValue (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 8-4).

12.31.3.3 The Internal priority value (IPV) specification data type

The IPV data type represents an IPV value (~~8.6.5.1.2~~8.6.5.4); this is either the null value or an internal priority value.

12.31.3.4 Representation of times

Table 12-36 specifies times (e.g., ~~PSFP~~StreamGateAdminBaseTime) with reference to the on-the-wire timing point at which the start of a frame crosses the boundary between the physical network media and PHY. This is the message timestamp point specified by IEEE Std 802.1AS for various media.

Replace Figure 12-7 with the following figure:

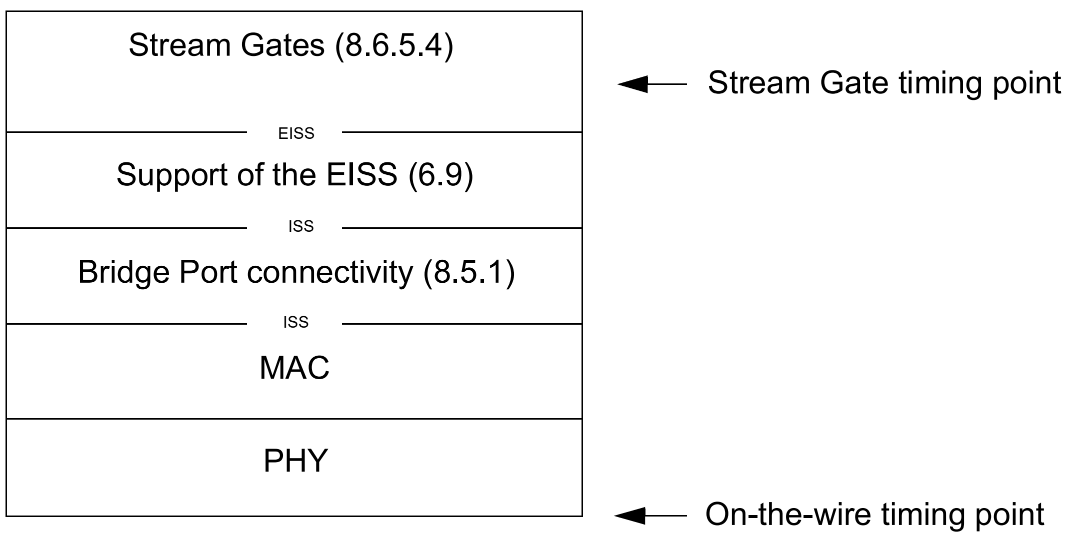


Figure 12-7—Timing points for Stream Gates

Figure 12-7 shows both the on-the-wire timing point and the ~~PSFP~~Stream Gate timing point within the interface stack (above MAC and PHY) that is used for gate open/close as described in ~~8.6.5.1~~8.6.5.4. Each timing point will have variance. A delay exists between the on-the-wire timing point and the ~~PSFP~~Stream Gate timing point. This delay will have variance that is bounded (minimum/maximum).

The Bridge contains the information needed in order to compute the minimum/maximum delay from the on-the-wire timing point to the ~~PSFP~~Stream Gate timing point. The Bridge shall adjust using the maximum delay, such that a frame's on-the-wire timing point occurs no later than the gate events represented in managed objects.

NOTE—Although the managed objects apply to a Bridge, the preceding specification of on-the-wire timing point can be applied to an end station.

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~~8.6.5.5), as detailed in Table 12-37. 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 12-37—The Flow Meter Instance Table

Name	Data type	Operations supported[a]	Conformance[b]	References
FlowMeterInstanceID	integer	R	~~BE~~PSFP, ats	~~8.6.5.1, 8.6.5.1.3~~ 8.6.5.2, 8.6.5.5
CIR	integer, bit/s	RW	~~BE~~PSFP, ats	~~8.6.5.1, 8.6.5.1.3~~ 8.6.5.5
CBS	integer, octets	RW	~~BE~~PSFP, ats	~~8.6.5.1, 8.6.5.1.3~~ 8.6.5.5
EIR	integer, bit/s	RW	~~BE~~PSFP, ats	~~8.6.5.1, 8.6.5.1.3~~ 8.6.5.5
EBS	integer, octets	RW	~~BE~~PSFP, ats	~~8.6.5.1, 8.6.5.1.3~~ 8.6.5.5
CF	integer, 0 or 1	RW	~~BE~~PSFP, ats	~~8.6.5.1, 8.6.5.1.3~~ 8.6.5.5
CM	enumerated, color-blind or color-aware	RW	~~BE~~PSFP, ats	~~8.6.5.1, 8.6.5.1.3~~ 8.6.5.5
DropOnYellow	Boolean	RW	~~BE~~PSFP, ats	~~8.6.5.1, 8.6.5.1.3~~ 8.6.5.5
MarkAllFramesRedEnable	Boolean	RW	~~BE~~PSFP, ats	~~8.6.5.1, 8.6.5.1.3~~ 8.6.5.5
MarkAllFramesRed	Boolean	RW	~~BE~~PSFP, ats	~~8.6.5.1, 8.6.5.1.3~~ 8.6.5.5

[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.~~
PSFP = Required for Bridge, Bridge component, or end station support of PSFP.
psfp = Optional for Bridge, Bridge component, or end station support of PSFP.
ATS = Required for Bridge or Bridge component support of ATS.
ats = Optional for Bridge or Bridge component support of ATS.

Insert the following subclauses (12.31.5 through 12.31.8.7, including Table 12-38 through Table 12-41) after 12.31.4, and renumber the subsequent tables in Clause 12 accordingly:

12.31.5 The Scheduler Instance Table

There is one Scheduler Instance Table per Bridge component. Each table row in the Scheduler Instance Table comprises a set of parameters that defines a single ATS scheduler instance, as detailed in Table 12-38.

Table 12-38—The Scheduler Instance Table

Name	Data type	Operations supported[a]	Conformance[b]	References
SchedulerInstanceID	integer	R	psfp, ATS	8.6.5.6, 12.31.5.1
CommittedBurstSize	integer, bits	RW	psfp, ATS	8.6.5.6, 8.6.11.3.5, 12.31.5.2
CommittedInformationRate	integer, bits/s	RW	psfp, ATS	8.6.5.6, 8.6.11.3.6, 12.31.5.3
SchedulerGroupInstanceID	integer	RW	psfp, ATS	8.6.5.6, 12.31.5.4

[a] R= Read only access; RW = Read/Write access.
[b] PSFP = Required for Bridge, Bridge component, or end station support of PSFP.
psfp = Optional for Bridge, Bridge component, or end station support of PSFP.
ATS = Required for Bridge or Bridge component support of ATS.
ats = Optional for Bridge or Bridge component support of ATS.

NOTE—ATS scheduler groups establish the relationship between ATS scheduler instances and the per port management variables (12.31.7), as described in 8.6.5.6. As a result, the scheduler instance table does not contain references to ports.

12.31.5.1 SchedulerInstanceID

An integer table index that allows the ATS scheduler instance to be referenced from Stream Filter Instance Table entries.

12.31.5.2 CommittedBurstSize

As specified in 8.6.11.3.5.

12.31.5.3 CommittedInformationRate

As specified in 8.6.11.3.6.

12.31.5.4 SchedulerGroupInstanceID

The SchedulerGroupInstanceID parameter identifies the ATS scheduler group (12.31.6) that is associated with the ATS scheduler instance. Multiple scheduler instance can be associated to one ATS scheduler group, as detailed in 8.6.5.6.

12.31.6 The Scheduler Group Instance Table

There is one Scheduler Group Instance Table per Bridge component. Each table row in the Scheduler Group Instance Table comprises a set of parameters that defines a single ATS scheduler group instance (8.6.5.6), as detailed in Table 12-39.

Table 12-39—The Scheduler Group Instance Table

Name	Data type	Operations supported[a]	Conformance[b]	References
SchedulerGroupInstanceID	integer	R	psfp, ATS	8.6.5.6, 12.31.6.1
MaxResidenceTime	integer, nanoseconds	RW	psfp, ATS	8.6.5.6, 12.31.6.2

[a] R = Read only access; RW = Read/Write access.
[b] PSFP = Required for Bridge, Bridge component, or end station support of PSFP.
psfp = Optional for Bridge, Bridge component, or end station support of PSFP.
ATS = Required for Bridge or Bridge component support of ATS.
ats = Optional for Bridge or Bridge component support of ATS.

12.31.6.1 SchedulerGroupInstanceID

An integer table index that allows the ATS scheduler group instance to be referenced from Scheduler Instance Table entries.

12.31.6.2 MaxResidenceTime

As specified in 8.6.11.3.13.

12.31.7 The Scheduler Port Parameter Table

There is one Scheduler Port Parameter Table per Bridge. Each table row in the Scheduler Port Parameter Table comprises a set of parameters shared by all ATS scheduler instance associated with a reception Port, as detailed in Table 12-40.

Table 12-40—The Scheduler Port Parameter Table

Name	Data type	Operations supported[a]	Conformance[b]	References
PortNumber	integer	R	psfp, ATS	12.31.7.1
DiscardedFramesCount	integer	R	psfp, ATS	8.6.5.6, 8.6.11.3.7, 12.31.7.2

[a] R = Read only access; RW = Read/Write access.
[b] PSFP = Required for Bridge, Bridge component, or end station support of PSFP.
psfp = Optional for Bridge, Bridge component, or end station support of PSFP.
ATS = Required for Bridge or Bridge component support of ATS.
ats = Optional for Bridge or Bridge component support of ATS.

12.31.7.1 PortNumber

An unique index of the associated Bridge Port (12.4.2).

12.31.7.2 DiscardedFramesCount

As specified in 8.6.5.6 and 8.6.11.3.7.

12.31.8 The Scheduler Timing Characteristics Table

There is one Scheduler Timing Characteristics Table per Bridge component. Each row in this table comprises the timing characteristics of a reception Port transmission Port pair, as detailed in Table 12-41.

Table 12-41—The Timing Characteristics Table

Name	Data type	Operations supported[a]	Conformance[b]	References
ReceptionPortNumber	integer	R	psfp, ATS	12.31.8.1
TransmissionPortNumber	integer	R	psfp, ATS	12.31.8.2
ClockOffsetVariationMax	integer	R	psfp, ATS	8.6.11.2, 12.31.8.3
ClockRateDeviationMax	integer	R	psfp, ATS	8.6.11.2, 12.31.8.4
ArrivalRecognitionDelayMax	integer	R	psfp, ATS	8.6.11.3.1, 12.31.8.5
ProcessingDelayMin	integer	R	psfp, ATS	8.6.11.3.2, 12.31.8.6
ProcessingDelayMax	integer	R	psfp, ATS	8.6.11.3.2, 12.31.8.7

[a] R = Read only access; RW = Read/Write access.
[b] PSFP = Required for Bridge, Bridge component, or end station support of PSFP.
psfp = Optional for Bridge, Bridge component, or end station support of PSFP.
ATS = Required for Bridge or Bridge component support of ATS.
ats = Optional for Bridge or Bridge component support of ATS.

12.31.8.1 ReceptionPortNumber

An unique index of the associated reception Port of the Bridge (12.4.2).

12.31.8.2 TransmissionPortNumber

An unique index of the associated transmission Port of the Bridge (12.4.2).

12.31.8.3 ClockOffsetVariationMax

As specified in 8.6.11.2, in nanoseconds, rounded to the next numerically higher representable value.

12.31.8.4 ClockRateDeviationMax

As specified in 8.6.11.2, in ppm, rounded to the next numerically higher representable value.

12.31.8.5 ArrivalRecognitionDelayMax

As specified in 8.6.11.3.1, in nanoseconds, rounded to the next numerically higher representable value.

12.31.8.6 ProcessingDelayMin

As specified in 8.6.11.3.2, in nanoseconds, rounded to the next numerically lower representable value.

12.31.8.7 ProcessingDelayMax

As specified in 8.6.11.3.2, in nanoseconds, rounded to the next numerically higher representable value.

17. Management Information Base (MIB)

17.2 Structure of the MIB

17.2.24 Structure of the IEEE8021-PSFP-MIB

Change 17.2.24 (including Table 17-30) as shown:

The IEEE8021-PSFP-MIB provides for configuration of PSFP (8.6.5, ~~8.6.5.1~~8.6.5.2.1, 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,~~ 8.6.5.2.1, 8.6.5.3, 12.31.2
	ieee8021PSFPStreamFilterInstance	StreamFilterInstance, ~~8.6.5, 8.6.5.1,~~ 8.6.5.2.1, 8.6.5.3, 12.31.2
	ieee8021PSFPStreamHandleSpec	StreamHandleSpec, ~~8.6.5, 8.6.5.1,~~ 8.6.5.2.1, 8.6.5.3, 12.31.2
	ieee8021PSFPPrioritySpec	PrioritySpec, ~~8.6.5, 8.6.5.1,~~ 8.6.5.2.1, 8.6.5.3, 12.31.2
	ieee8021PSFPStreamGateInstanceID	StreamGateInstanceID, ~~8.6.5, 8.6.5.1,~~ 8.6.5.2.1, 8.6.5.3, 12.31.2
	ieee8021PSFPFilterSpecificationList [a]	~~FilterSpecificationList~~MaxSDUSize, FlowMeterInstanceID, FlowMeterEnable, ~~8.6.5, 8.6.5.1,~~ 8.6.5.2.1, 8.6.5.3, 12.31.2, 12.31.2.4, 12.31.2.6
	ieee8021PSFPMatchingFramesCount	MatchingFramesCount, ~~8.6.5, 8.6.5.1,~~ 8.6.5.2.1, 8.6.5.3, 12.31.2
	ieee8021PSFPPassingFramesCount	PassingFramesCount, ~~8.6.5, 8.6.5.1,~~ 8.6.5.2.1, 8.6.5.3, 8.6.5.4, 12.31.2
	ieee8021PSFPNotPassingFramesCount	NotPassingFramesCount, ~~8.6.5, 8.6.5.1,~~ 8.6.5.2.1, 8.6.5.3, 8.6.5.4, 12.31.2
	ieee8021PSFPPassingSDUCount	PassingSDUCount, ~~8.6.5, 8.6.5.1,~~ 8.6.5.2.1, 8.6.5.3, 8.6.5.3.1, 12.31.2
	ieee8021PSFPNotPassingSDUCount	NotPassingSDUCount, ~~8.6.5, 8.6.5.1,~~ 8.6.5.2.1, 8.6.5.3, 8.6.5.3.1, 12.31.2
	ieee8021PSFPREDFramesCount	REDFramesCount, ~~8.6.5, 8.6.5.1,~~ 8.6.5.2.1, 8.6.5.3, 8.6.5.5, 12.31.2
	ieee8021PSFPStreamBlockedDueToOversizeFrameEnable	StreamBlockedDueToOversizeFrameEnable, ~~8.6.5.1, 8.6.5.1.1,~~ 8.6.5.2.1, 8.6.5.3.1, 12.31.2

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

MIB table	MIB object	Reference
	ieee8021PSFPStreamBlockedDueToOversizeFrame	StreamBlockedDueToOversizeFrame, ~~8.6.5.1, 8.6.5.1.1,~~ 8.6.5.2.1, 8.6.5.3.1, 12.31.2
ieee8021PSFPStreamGateParameters		
ieee8021PSFPStreamGateTable		Stream Gate Instance Table, ~~8.6.5, 8.6.5.1,~~ 8.6.5.2.1, 8.6.5.4, 12.31.3
	ieee8021PSFPStreamGateInstance	StreamGateInstance, ~~8.6.5, 8.6.5.1,~~ 8.6.5.2.1, 8.6.5.4, 12.31.3
	ieee8021PSFPGateEnabled	~~PSFP~~StreamGateEnabled, ~~8.6.5, 8.6.5.1,~~ 8.6.5.2.1, 8.6.5.4, 8.6.10, 12.31.3
	ieee8021PSFPAdminGateStates	~~PSFP~~StreamGateAdminGateStates, ~~8.6.5, 8.6.5.1,~~ 8.6.5.2.1, 8.6.5.4, 8.6.10, 12.31.3
	ieee8021PSFPOperGateStates	~~PSFP~~StreamGateOperGateStates, ~~8.6.5, 8.6.5.1,~~ 8.6.5.2.1, 8.6.5.4, 8.6.10, 12.31.3
	ieee8021PSFPAdminControlListLength	~~PSFP~~StreamGateAdminControlListLength, ~~8.6.5,~~ 8.6.5.2.1, 8.6.5.4, 8.6.10, 12.31.3
	ieee8021PSFPOperControlListLength	~~PSFP~~StreamGateOperControlListLength, ~~8.6.5, 8.6.5.1,~~ 8.6.5.2.1, 8.6.5.4, 8.6.10, 12.31.3
	ieee8021PSFPAdminControlList	~~PSFP~~StreamGateAdminControlList, ~~8.6.5, 8.6.5.1,~~ 8.6.5.2.1, 8.6.5.4, 8.6.10, 12.31.3
	ieee8021PSFPOperControlList	~~PSFP~~StreamGateOperControlList, ~~8.6.5, 8.6.5.1,~~ 8.6.5.2.1, 8.6.5.4, 8.6.10, 12.31.3
	ieee8021PSFPAdminCycleTimeNumerator	~~PSFP~~StreamGateAdminCycleTime, ~~8.6.5, 8.6.5.1,~~ 8.6.5.2.1, 8.6.5.4, 8.6.10, 12.31.3
	ieee8021PSFPAdminCycleTimeDenominator	~~PSFP~~StreamGateAdminCycleTime, ~~8.6.5, 8.6.5.1,~~ 8.6.5.2.1, 8.6.5.4, 8.6.10, 12.31.3
	ieee8021PSFPOperCycleTimeNumerator	~~PSFP~~StreamGateOperCycleTime, ~~8.6.5, 8.6.5.1,~~ 8.6.5.2.1, 8.6.5.4, 8.6.10, 12.31.3
	ieee8021PSFPOperCycleTimeDenominator	~~PSFP~~StreamGateOperCycleTime, ~~8.6.5, 8.6.5.1,~~ 8.6.5.2.1, 8.6.5.4, 8.6.10, 12.31.3
	ieee8021PSFPAdminCycleTimeExtension	~~PSFP~~StreamGateAdminCycleTimeExtension, ~~8.6.5, 8.6.5.1,~~ 8.6.5.2.1, 8.6.5.4, 8.6.10, 12.31.3

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

MIB table	MIB object	Reference
	ieee8021PSFPOperCycleTimeExtension	~~PSFP~~StreamGateOperCycleTimeExtension, ~~8.6.5, 8.6.5.1,~~ 8.6.5.2.1, 8.6.5.4, 8.6.10, 12.31.3
	ieee8021PSFPAdminBaseTime	~~PSFP~~StreamGateAdminBaseTime, ~~8.6.5, 8.6.5.1,~~ 8.6.5.2.1, 8.6.5.4, 8.6.10, 12.31.3
	ieee8021PSFPOperBaseTime	~~PSFP~~StreamGateOperBaseTime, ~~8.6.5, 8.6.5.1,~~ 8.6.5.2.1, 8.6.5.4, 8.6.10, 12.31.3
	ieee8021PSFPConfigChange	~~PSFP~~StreamGateConfigChange, ~~8.6.5, 8.6.5.1,~~ 8.6.5.2.1, 8.6.5.4, 8.6.10, 12.31.3
	ieee8021PSFPConfigChangeTime	~~PSFP~~StreamGateConfigChangeTime, ~~8.6.5, 8.6.5.1,~~ 8.6.5.2.1, 8.6.5.4, 8.6.10, 12.31.3
	ieee8021PSFPTickGranularity	~~PSFP~~StreamGateTickGranularity, ~~8.6.5, 8.6.5.1,~~ 8.6.5.2.1, 8.6.5.4, 8.6.10, 12.31.3
	ieee8021PSFPCurrentTime	~~PSFP~~StreamGateCurrentTime, ~~8.6.5, 8.6.5.1,~~ 8.6.5.2.1, 8.6.5.4, 8.6.10, 12.31.3
	ieee8021PSFPConfigPending	~~PSFP~~StreamGateConfigPending, ~~8.6.5, 8.6.5.1,~~ 8.6.5.2.1, 8.6.5.4, 8.6.10, 12.31.3
	ieee8021PSFPConfigChangeError	~~PSFP~~StreamGateConfigChangeError, ~~8.6.5, 8.6.5.1,~~ 8.6.5.2.1, 8.6.5.4, 8.6.10, 12.31.3
	ieee8021PSFPAdminIPV	~~PSFP~~StreamGateAdminIPV, ~~8.6.5, 8.6.5.1,~~ 8.6.5.2.1, 8.6.5.4, 12.31.3
	ieee8021PSFPOperIPV	~~PSFP~~StreamGateOperIPV, ~~8.6.5, 8.6.5.1,~~ 8.6.5.2.1, 8.6.5.4, 12.31.3
	ieee8021PSFPGateClosedDueToInvalidRxEnable	~~PSFP~~StreamGateGateClosedDueToInvalidRx-Enable, ~~8.6.5.1.2~~8.6.5.2.1, 8.6.5.4
	ieee8021PSFPGateClosedDueToInvalidRx	~~PSFP~~StreamGateGateClosedDueToInvalidRx, ~~8.6.5.1.2~~8.6.5.2.1, 8.6.5.4
	ieee8021PSFPGateClosedDueToOctetsExceededEnable	~~PSFP~~StreamGateGateClosedDueToOctetsExceededEnable, ~~8.6.5.1.2~~ 8.6.5.2.1, 8.6.5.4
	ieee8021PSFPGateClosedDueToOctetsExceeded	~~PSFP~~StreamGateGateClosedDueToOctetsExceeded, ~~8.6.5.1.2~~8.6.5.2.1, 8.6.5.4

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

MIB table	MIB object	Reference
ieee8021PSFPFlowMeterParameters		
ieee8021PSFPFlowMeterTable		Flow Meter Instance Table, ~~8.6.5, 8.6.5.1,~~ 8.6.5.2.1, 8.6.5.4, 12.31.4
	ieee8021PSFPFlowMeterInstance	FlowMeterInstanceID, ~~8.6.5, 8.6.5.1,~~ 8.6.5.2.1, 8.6.5.5, 12.31.4
	ieee8021PSFPFlowMeterCIR	CIR, ~~8.6.5, 8.6.5.1,~~ 8.6.5.2.1, 8.6.5.5, 12.31.4
	ieee8021PSFPFlowMeterCBS	CBS, ~~8.6.5, 8.6.5.1,~~ 8.6.5.2.1, 8.6.5.5, 12.31.4
	ieee8021PSFPFlowMeterEIR	EIR, ~~8.6.5, 8.6.5.1,~~ 8.6.5.2.1, 8.6.5.5, 12.31.4
	ieee8021PSFPFlowMeterCF	CF, ~~8.6.5, 8.6.5.1,~~ 8.6.5.2.1, 8.6.5.5, 12.31.4
	ieee8021PSFPFlowMeterCM	CM, ~~8.6.5, 8.6.5.1,~~ 8.6.5.2.1, 8.6.5.5, 12.31.4
	ieee8021PSFPFlowMeterDropOnYellow	DropOnYellow, ~~8.6.5, 8.6.5.1,~~ 8.6.5.2.1, 8.6.5.5, 12.31.4
	ieee8021PSFPFlowMeterMarkAllFramesRedEnable	MarkAllFramesRedEnable, ~~8.6.5, 8.6.5.1,~~ 8.6.5.2.1, 8.6.5.5, 12.31.4
	ieee8021PSFPFlowMeterMarkAllFramesRed	MarkAllFramesRed, ~~8.6.5, 8.6.5.1,~~ 8.6.5.2.1, 8.6.5.5, 12.31.4
ieee8021PSFPStreamParameters		
ieee8021PSFPStreamParameterTable		StreamParameterTable, ~~8.6.5, 8.6.5.1,~~ 8.6.5.2.1, 12.31.1
	ieee8021PSFPMaxStreamFilterInstances	MaxStreamFilterInstances, ~~8.6.5, 8.6.5.1,~~ 8.6.5.2.1, 8.6.5.3, 12.31.1
	ieee8021PSFPMaxStreamGateInstances	MaxStreamGateInstances, ~~8.6.5, 8.6.5.1,~~ 8.6.5.2.1, 8.6.5.4, 12.31.1
	ieee8021PSFPMaxFlowMeterInstances	MaxFlowMeterInstances, ~~8.6.5, 8.6.5.1,~~ 8.6.5.2.1, 8.6.5.5, 12.31.1
	ieee8021PSFPSupportedListMax	SupportedListMax, ~~8.6.5, 8.6.5.1,~~ 8.6.5.2.1, 8.6.5.4, 12.31.1

[a] To allow the PSFP MIB originally specified in IEEE Std 802.1Q-2018 to manage systems conformant to this amendment, the encoding of a Maximum SDU size and a flow meter identifier in an ieee8021PSFPFilterSpecificationList has been retained.

17.3 Relationship to other MIBs

17.3.25 Relationship of IEEE8021-PSFP-MIB to other MIBs

Change 17.3.25 as shown:

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~~8.6.5.2.1 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.

17.7 MIB modules[4,5]

17.7.24 Definitions for the IEEE8021-PSFP-MIB module

Change 17.7.24 as shown:

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

-- =============================================================
-- MIB for support of the Per-Stream Filtering and Policing
-- Enhancements for IEEE 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 "20~~180628~~2011060000Z" -- ~~June 28~~November 6, 20~~18~~20
    ORGANIZATION "IEEE 802.1 Working Group"
    CONTACT-INFO
        "~~  ~~WG-URL: http://~~www.~~ieee802.org/1/
```

[4] *Copyright release for MIB:* Users of this standard may freely reproduce the MIBs contained in this subclause so that they can be used for their intended purpose.

[5] An ASCII version of the revised MIB module is attached to the PDF of this amendment and can also be obtained from the IEEE 802.1 Website at https://1.ieee802.org/mib-modules/.

```
        WG-EMail: stds-802-1-Ll@ieee.org

         Contact: IEEE 802.1 Working Group Chair
          Postal: C/O IEEE 802.1 Working Group
                  IEEE Standards Association
                  445 Hoes Lane
                  Piscataway, NJ 08854
                  NJ 08854USA
                  USA
          E-mail: stds-802-1-Lchairs@ieee.org"
    DESCRIPTION
        "The Bridge MIB module for managing devices that support
        the Per-Stream Filtering and Policing enhancements
        for IEEE 802.1Q Bridges.

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

        Copyright (C) IEEE (201820).
        This version of this MIB module is part of IEEE Std 802.1Q;
        see the draft itself for full legal notices."

    REVISION "202011060000Z" -- November 6, 2020
    DESCRIPTION
            "Published as part of IEEE Std 802.1Qcr-2020.
            Cross references updated and corrected."

    REVISION "201806280000Z" -- June 28, 2018

    DESCRIPTION
            "Published as part of IEEE Std 802.1Q 20178 revision.
            Cross references updated and corrected."

    REVISION "201709080000Z" -- September 29, 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   "8.6.5.2.1, 8.6.5.13, 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~~2.1, 8.6.5.3, 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~~2.1, 8.6.5.3, 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~~2.1, 8.6.5.3, 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~~2.1, 8.6.5.3, 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 operation~~filter specification types

        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~~2.1, 8.6.5.3, 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~~2.1, 8.6.5.3, 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~~2.1, 8.6.5.3, 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~~2.1, 8.6.5.3, 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 maximum SDU size filter specification associated
        with this stream filter.
        "
    REFERENCE   "8.6.5.~~1~~2.1, 8.6.5.3, 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 maximum SDU size filter specification associated
        with this stream filter.
        "
    REFERENCE   "8.6.5.~~1~~2.1, 8.6.5.3, 12.31.2"
    ::= { ieee8021PSFPStreamFilterEntry 10}

ieee8021PSFPREDFramesCount OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The REDFramesCount counter counts received
         frames that were discarded as a result of the
```

```
         operation of the flow meter.
        "
    REFERENCE   "8.6.5.12.1, 8.6.5.3, 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   "8.6.5.2.1, 8.6.5.1.13, 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   "8.6.5.2.1, 8.6.5.1.13, 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 cannot 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   "8.6.5.2.1, 8.6.5.1.24, 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 8-4. 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 8-4. 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   "8.6.5.1.24, 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   "8.6.5.1.24, 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, 8.6.5.1.24, 12.31.3"
    DEFVAL { false }
    ::= { ieee8021PSFPStreamGateEntry 26}

ieee8021PSFPGateClosedDueToInvalidRx OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The PSFPGateClosedDueToInvalidRx object contains
        a Boolean value that indicates whether, if the
        PSFPGateClosedDueToInvalidRx 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, 8.6.5.1.24, 12.31.3"
    DEFVAL { false }
    ::= { ieee8021PSFPStreamGateEntry 27}

ieee8021PSFPGateClosedDueToOctetsExceededEnable OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The PSFPGateClosedDueToOctetsExceededEnable object contains
        a Boolean value that indicates whether the
        PSFPGateClosedDueToOctetsExceeded 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, 8.6.5.1.24, 12.31.3"
    DEFVAL { false }
    ::= { ieee8021PSFPStreamGateEntry 28}

ieee8021PSFPGateClosedDueToOctetsExceeded OBJECT-TYPE
```

```
    SYNTAX       TruthValue
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
        "The PSFPGateClosedDueToOctetsExceeded parameter contains
        a Boolean value that indicates whether, if the
        ~~PSFP~~GateClosedDueToOctetsExceeded 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, 8.6.5.1.2~~4, 12.31.3"
    DEFVAL { false }
    ::= { ieee8021PSFPStreamGateEntry 29}


ieee8021PSFPStreamGateEntryRowStatus OBJECT-TYPE
    SYNTAX       RowStatus
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
        "The status of the row.

         The writable columns in a row cannot be changed if the row
         is active. All columns MUST have a valid value before a row
         can be activated.
        "
    ::= { ieee8021PSFPStreamGateEntry 30 }



-- =============================================================
-- The ieee8021PSFPFlowMeterParameters subtree
-- This subtree defines the objects necessary for the management
-- of the flow meters for IEEE Std 802.1Q.
-- =============================================================

-- =============================================================
-- the ieee8021PSFPFlowMeterTable
-- =============================================================

ieee8021PSFPFlowMeterTable OBJECT-TYPE
    SYNTAX       SEQUENCE OF Ieee8021PSFPFlowMeterEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "A table that contains the per-meter instance
        manageable parameters for flow meters.

        For a given Bridge component, a row in the table exists for
        each flow meter instance.

        All writable objects in this table must be
        persistent over power up restart/reboot."
    REFERENCE    "8.6.5~~, 8.6.5.1~~.5, 12.31.4"
    ::= { ieee8021PSFPFlowMeterParameters 1 }

ieee8021PSFPFlowMeterEntry OBJECT-TYPE
    SYNTAX       Ieee8021PSFPFlowMeterEntry
```

```
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A list of objects that contains the manageable parameters for
        flow meters for a Bridge component."
    INDEX  { ieee8021BridgeBaseComponentId,
             ieee8021PSFPFlowMeterInstance
           }
    ::= { ieee8021PSFPFlowMeterTable 1 }

Ieee8021PSFPFlowMeterEntry ::=
    SEQUENCE {
        ieee8021PSFPFlowMeterInstance
            Unsigned32,
        ieee8021PSFPFlowMeterCIR
            Unsigned32,
        ieee8021PSFPFlowMeterCBS
            Unsigned32,
        ieee8021PSFPFlowMeterEIR
            Unsigned32,
        ieee8021PSFPFlowMeterEBS
            Unsigned32,
        ieee8021PSFPFlowMeterCF
            Integer32,
        ieee8021PSFPFlowMeterCM
            INTEGER,
        ieee8021PSFPFlowMeterDropOnYellow
            TruthValue,
        ieee8021PSFPFlowMeterMarkAllFramesRedEnable
            TruthValue,
        ieee8021PSFPFlowMeterMarkAllFramesRed
            TruthValue,
        ieee8021PSFPFlowMeterEntryRowStatus
            RowStatus
             }

ieee8021PSFPFlowMeterInstance OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The FlowMeterInstance parameter is an index into the
        FlowMeterTable.

        The value of this object MUST be retained across
        reinitializations of the management system."
    REFERENCE   "8.6.5.15, 12.31.4"
    ::= { ieee8021PSFPFlowMeterEntry 1}

ieee8021PSFPFlowMeterCIR OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The FlowMeterCIR parameter contains an integer value that
        represents the CIR value for the flow meter, in bit/second.

        The value of this object MUST be retained across
        reinitializations of the management system."
```

```
    REFERENCE   "8.6.5.~~1~~5, 12.31.4"
    ::= { ieee8021PSFPFlowMeterEntry 2}

ieee8021PSFPFlowMeterCBS OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The FlowMeterCBS parameter contains an integer value that
        represents the CBS value for the flow meter, in octets.

        The value of this object MUST be retained across
        reinitializations of the management system."
    REFERENCE   "8.6.5.~~1~~5, 12.31.4"
    ::= { ieee8021PSFPFlowMeterEntry 3}

ieee8021PSFPFlowMeterEIR OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The FlowMeterEIR parameter contains an integer value that
        represents the EIR value for the flow meter, in bit/second.

        The value of this object MUST be retained across
        reinitializations of the management system."
    REFERENCE   "8.6.5.~~1~~5, 12.31.4"
    ::= { ieee8021PSFPFlowMeterEntry 4}

ieee8021PSFPFlowMeterEBS OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The FlowMeterEBS parameter contains an integer value that
        represents the EBS value for the flow meter, in octets.

        The value of this object MUST be retained across
        reinitializations of the management system."
    REFERENCE   "8.6.5.~~1~~5, 12.31.4"
    ::= { ieee8021PSFPFlowMeterEntry 5}

ieee8021PSFPFlowMeterCF OBJECT-TYPE
    SYNTAX      Integer32 (0..1)
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The FlowMeterCF parameter contains an integer value that
        represents the CF value for the flow meter, as an integer
        value 0 or 1.

        The value of this object MUST be retained across
        reinitializations of the management system."
    REFERENCE   "8.6.5.~~1~~5, 12.31.4"
    ::= { ieee8021PSFPFlowMeterEntry 6}

ieee8021PSFPFlowMeterCM OBJECT-TYPE
    SYNTAX      INTEGER  {colorBlind(1), colorAware(2)}
    MAX-ACCESS  read-create
```

```
    STATUS      current
    DESCRIPTION
        "The FlowMeterCM parameter contains an integer value that
        represents the CM value for the flow meter, as an enumerated
        value indicating colorBlind(1) or colorAware(2).

        The value of this object MUST be retained across
        reinitializations of the management system."
    REFERENCE   "8.6.5.~~1~~5, 12.31.4"
    ::= { ieee8021PSFPFlowMeterEntry 7}

ieee8021PSFPFlowMeterDropOnYellow OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The FlowMeterDropOnYellow parameter contains a Boolean value that
        indicates whether yellow frames are dropped (TRUE) or
        have drop_eligible set to TRUE (FALSE).

        The value of this object MUST be retained across
        reinitializations of the management system."
    REFERENCE   "8.6.5.~~1~~5, 12.31.4"
    ::= { ieee8021PSFPFlowMeterEntry 8}

ieee8021PSFPFlowMeterMarkAllFramesRedEnable OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The FlowMeterMarkAllFramesRedEnable parameter contains
        a Boolean value that indicates whether the MarkAllFramesRed
        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, 8.6.5.1.3~~5, 12.31.4"
    DEFVAL { false }
    ::= { ieee8021PSFPFlowMeterEntry 9}

ieee8021PSFPFlowMeterMarkAllFramesRed OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The FlowMeterMarkAllFramesRed parameter contains
        a Boolean value that indicates whether, if the
        MarkAllFramesRed 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, 8.6.5.1.3~~5, 12.31.4"
    DEFVAL { false }
    ::= { ieee8021PSFPFlowMeterEntry 10}

ieee8021PSFPFlowMeterEntryRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
```

```
    STATUS      current
    DESCRIPTION
        "The status of the row.

         The writable columns in a row cannot be changed if the row
         is active. All columns MUST have a valid value before a row
         can be activated.
        "
    ::= { ieee8021PSFPFlowMeterEntry 11 }




-- ================================================================
-- The ieee8021PSFPStreamParameters subtree
-- This subtree defines the objects necessary for the management
-- of the flow meters for IEEE Std 802.1Q.
-- ================================================================

-- ================================================================
-- the ieee8021PSFPStreamParameterTable
-- ================================================================

ieee8021PSFPStreamParameterTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021PSFPStreamParameterEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table that contains per-Bridge component
        manageable parameters for PSFP.

        A row in the table exists for each Bridge component.

        All writable objects in this table must be
        persistent over power up restart/reboot."
    REFERENCE   "8.6.5, 8.6.5.1.2, 12.31.41"
    ::= { ieee8021PSFPStreamParameters 1 }

ieee8021PSFPStreamParameterEntry OBJECT-TYPE
    SYNTAX      Ieee8021PSFPStreamParameterEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A list of objects that contains the manageable parameters for
        flow meters for a Bridge component."
    INDEX  { ieee8021BridgeBaseComponentId
           }
    ::= { ieee8021PSFPStreamParameterTable 1 }

Ieee8021PSFPStreamParameterEntry ::=
    SEQUENCE {
        ieee8021PSFPMaxStreamFilterInstances
            Unsigned32,
        ieee8021PSFPMaxStreamGateInstances
            Unsigned32,
        ieee8021PSFPMaxFlowMeterInstances
            Unsigned32,
        ieee8021PSFPSupportedListMax
            Unsigned32
```

```
            }

ieee8021PSFPMaxStreamFilterInstances OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The MaxStreamFilterInstances parameter defines the
        maximum number of stream filter instances that are
        supported by this Bridge component."
    REFERENCE   "8.6.5.13, 12.31.2"
    ::= { ieee8021PSFPStreamParameterEntry 1}

ieee8021PSFPMaxStreamGateInstances OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The MaxStreamGateInstances parameter defines the
        maximum number of stream gate instances that are
        supported by this Bridge component."
    REFERENCE   "8.6.5.14, 12.31.3"
    ::= { ieee8021PSFPStreamParameterEntry 2}

ieee8021PSFPMaxFlowMeterInstances OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The MaxFlowMeterInstances parameter defines the
        maximum number of flow meter instances that are
        supported by this Bridge component."
    REFERENCE   "8.6.5.15, 12.31.4"
    ::= { ieee8021PSFPStreamParameterEntry 3}

ieee8021PSFPSupportedListMax OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The SupportedListMax parameter defines the
        The maximum value supported by this Bridge component of
        the AdminControlListLength and
        OperControlListLength parameters."
    REFERENCE   "8.6.5.14, 12.31.3"
    ::= { ieee8021PSFPStreamParameterEntry 4}


-- =============================================================
-- IEEE8021 PSFP MIB - Conformance Information
-- =============================================================

ieee8021PSFPCompliances
    OBJECT IDENTIFIER ::= { ieee8021PSFPConformance 1 }
ieee8021PSFPGroups
    OBJECT IDENTIFIER ::= { ieee8021PSFPConformance 2 }

-- =============================================================
-- units of conformance
```

```
-- =============================================================

-- =============================================================
-- the ieee8021PSFPObjectsGroup group
-- =============================================================

ieee8021PSFPObjectsGroup OBJECT-GROUP
    OBJECTS {
        ieee8021PSFPStreamHandleSpec,
        ieee8021PSFPPrioritySpec,
        ieee8021PSFPStreamGateInstanceID,
        ieee8021PSFPFilterSpecificationList,
        ieee8021PSFPMatchingFramesCount,
        ieee8021PSFPPassingFramesCount,
        ieee8021PSFPNotPassingFramesCount,
        ieee8021PSFPPassingSDUCount,
        ieee8021PSFPNotPassingSDUCount,
        ieee8021PSFPREDFramesCount,
        ieee8021PSFPStreamBlockedDueToOversizeFrameEnable,
        ieee8021PSFPStreamBlockedDueToOversizeFrame,
        ieee8021PSFPStreamFilterEntryRowStatus,
        ieee8021PSFPGateEnabled,
        ieee8021PSFPAdminGateStates,
        ieee8021PSFPOperGateStates,
        ieee8021PSFPAdminControlListLength,
        ieee8021PSFPOperControlListLength,
        ieee8021PSFPAdminControlList,
        ieee8021PSFPOperControlList,
        ieee8021PSFPAdminCycleTimeNumerator,
        ieee8021PSFPAdminCycleTimeDenominator,
        ieee8021PSFPOperCycleTimeNumerator,
        ieee8021PSFPOperCycleTimeDenominator,
        ieee8021PSFPAdminCycleTimeExtension,
        ieee8021PSFPOperCycleTimeExtension,
        ieee8021PSFPAdminBaseTime,
        ieee8021PSFPOperBaseTime,
        ieee8021PSFPConfigChange,
        ieee8021PSFPConfigChangeTime,
        ieee8021PSFPTickGranularity,
        ieee8021PSFPCurrentTime,
        ieee8021PSFPConfigPending,
        ieee8021PSFPConfigChangeError,
        ieee8021PSFPAdminIPV,
        ieee8021PSFPOperIPV,
        ieee8021PSFPGateClosedDueToInvalidRxEnable,
        ieee8021PSFPGateClosedDueToInvalidRx,
        ieee8021PSFPGateClosedDueToOctetsExceededEnable,
        ieee8021PSFPGateClosedDueToOctetsExceeded,
        ieee8021PSFPStreamGateEntryRowStatus,
        ieee8021PSFPFlowMeterCIR,
        ieee8021PSFPFlowMeterCBS,
        ieee8021PSFPFlowMeterEIR,
        ieee8021PSFPFlowMeterEBS,
        ieee8021PSFPFlowMeterCF,
        ieee8021PSFPFlowMeterCM,
        ieee8021PSFPFlowMeterDropOnYellow,
        ieee8021PSFPFlowMeterMarkAllFramesRedEnable,
        ieee8021PSFPFlowMeterMarkAllFramesRed,
        ieee8021PSFPFlowMeterEntryRowStatus,
```

```
        ieee8021PSFPMaxStreamFilterInstances,
        ieee8021PSFPMaxStreamGateInstances,
        ieee8021PSFPMaxFlowMeterInstances,
        ieee8021PSFPSupportedListMax
        }

    STATUS      current
    DESCRIPTION
        "Objects that allow management of PSFP."
    ::= { ieee8021PSFPGroups 1 }


-- ============================================================
-- compliance statements
-- ============================================================

ieee8021PSFPCompliance MODULE-COMPLIANCE
    STATUS      current
    DESCRIPTION
        "The compliance statement for devices supporting
        PSFP.

        Support of the objects defined in this MIB module
        also requires support of the IEEE8021-BRIDGE-MIB; the
        provisions of 17.3.2 apply to implementations claiming
        support of this MIB. "

    MODULE -- this module
        MANDATORY-GROUPS {
            ieee8021PSFPObjectsGroup
        }

    ::= { ieee8021PSFPCompliances 1 }

END
```

37. Enhanced Transmission Selection (ETS)

37.3 ETS algorithm

Change NOTE 1 in 37.3 as shown:

NOTE 1—~~While a traffic class is often referred to as a queue, it is not necessarily a single FIFO data structure.~~ The ETS algorithm is used to determine when an ETS traffic class has a frame available to transmit. ~~The ordering requirements in 8.6.6 are the only constraint of this standard on the order in which frames in a traffic class are transmitted.~~ The determination of which frame in the traffic class to send ~~may~~can be done in an implementation specific manner~~ as long as those ordering requirements are satisfied~~, subject to the ordering requirements in 8.6.6.~~ For example, an implementation might organize the traffic class as a set of FIFO queues with each FIFO queue containing the frames from a priority, VLAN or a source port. Such FIFO queues might be serviced with strict priority, weighted round robin or other bandwidth distribution algorithm.~~

48. YANG Data Model

48.3 IEEE 802.1Q YANG model

48.3.2 Generic IEEE 802.1Q-2018 bridge model

Replace Figure 48-5 with the figure on page 78.

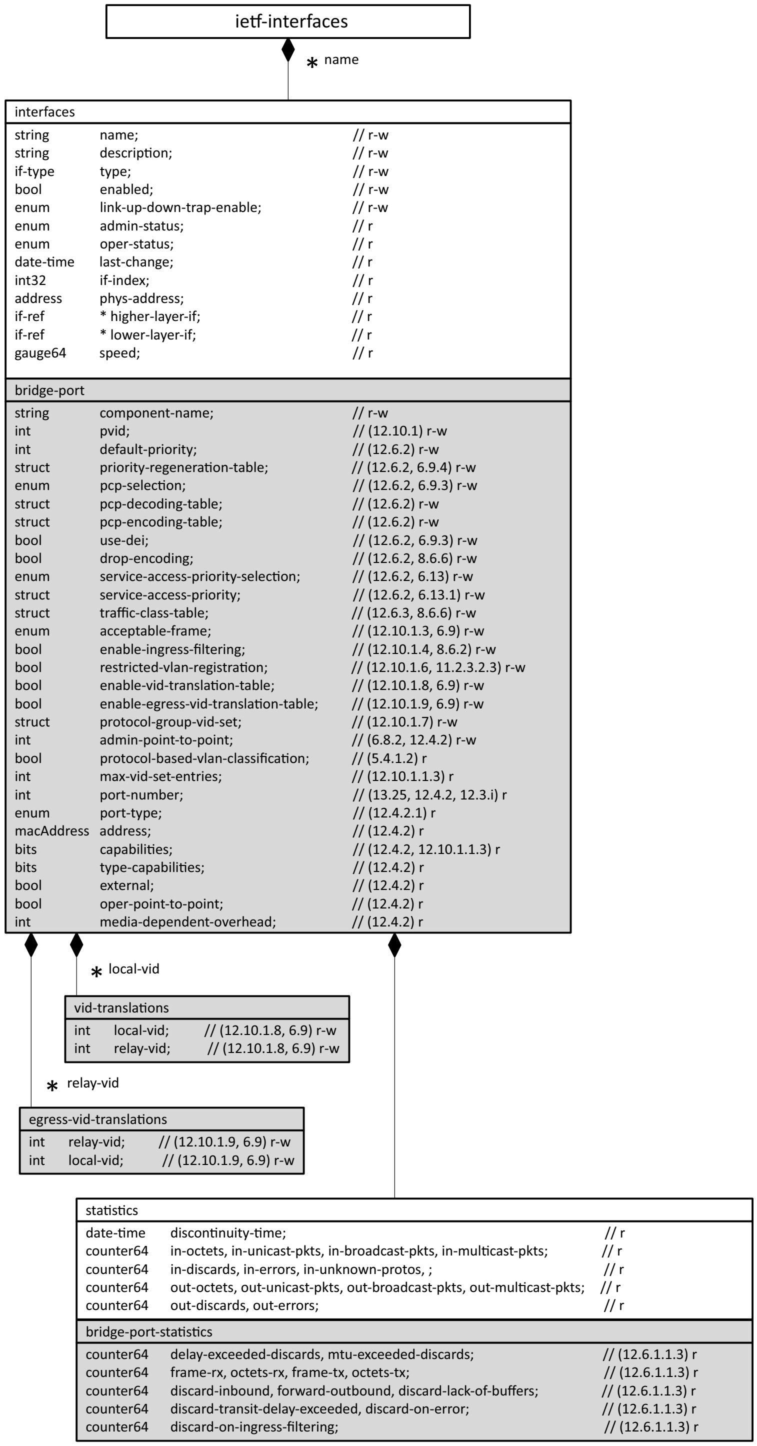
48.3.2.1 Two-Port MAC Relay model

Replace Figure 48-7 with the figure on page 79.

48.3.2.3 Provider Bridge model

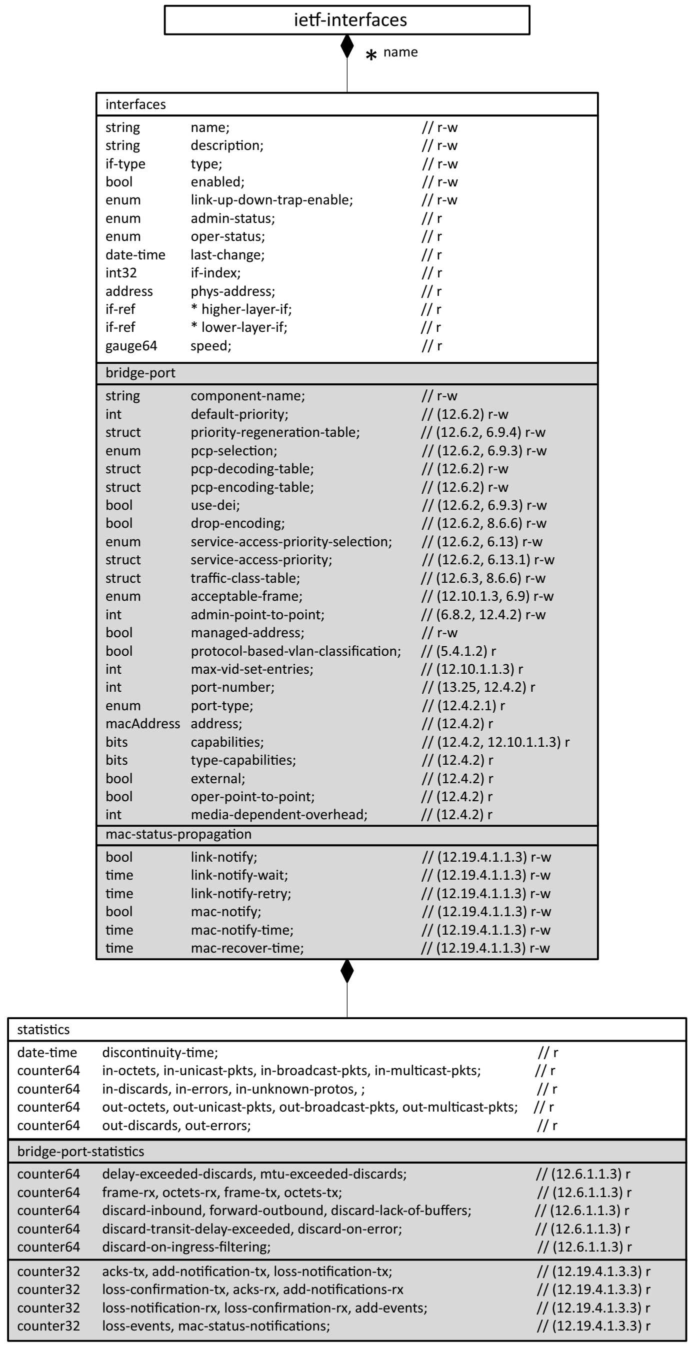
Replace Figure 48-9 with the figure on page 80.

Replace Figure 48-10 with the figure on page 81.



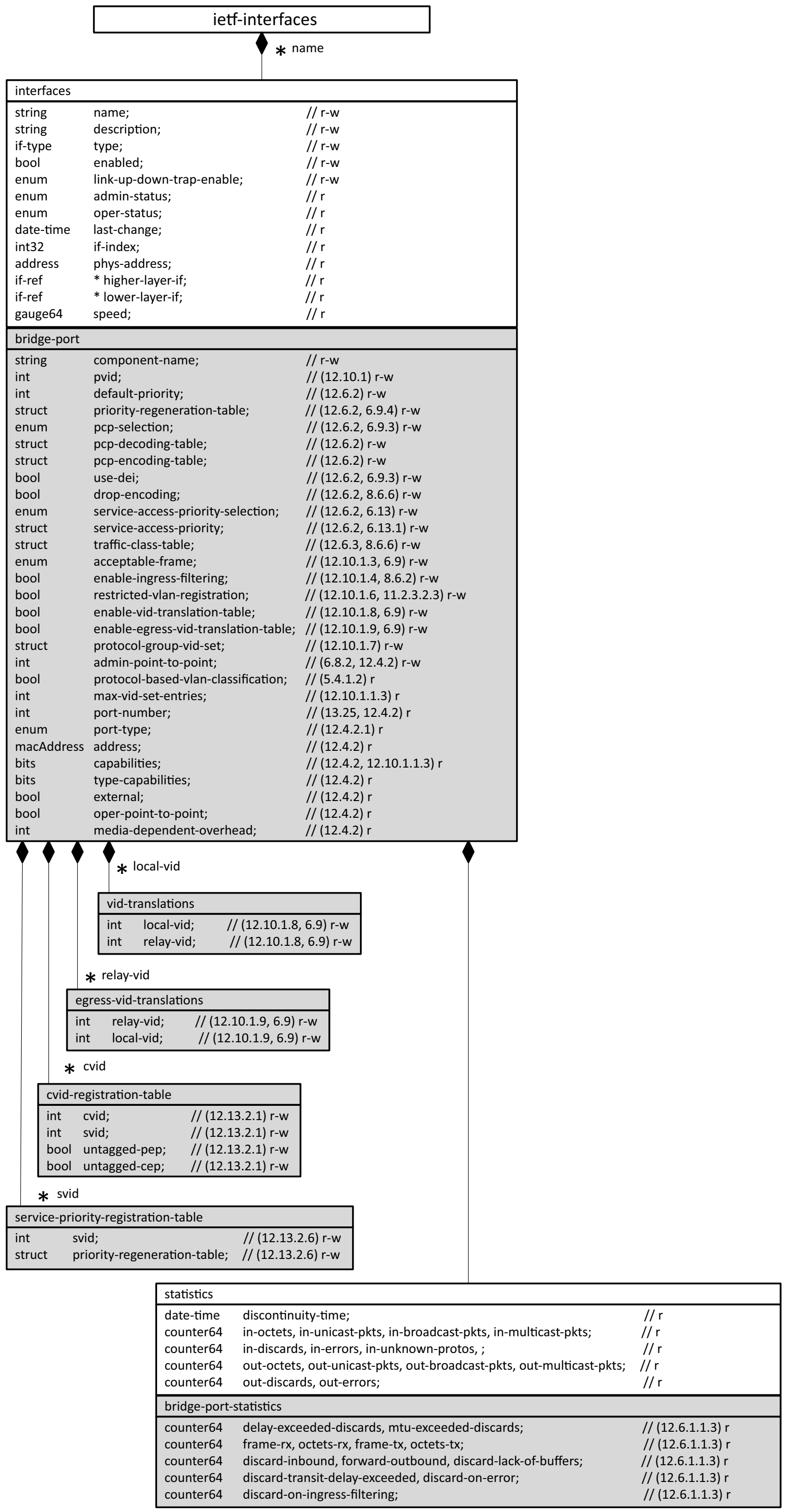
NOTE—Items shaded in gray indicate extension provided by the IEEE 802.1Q YANG models.

Figure 48-5—Bridge port model



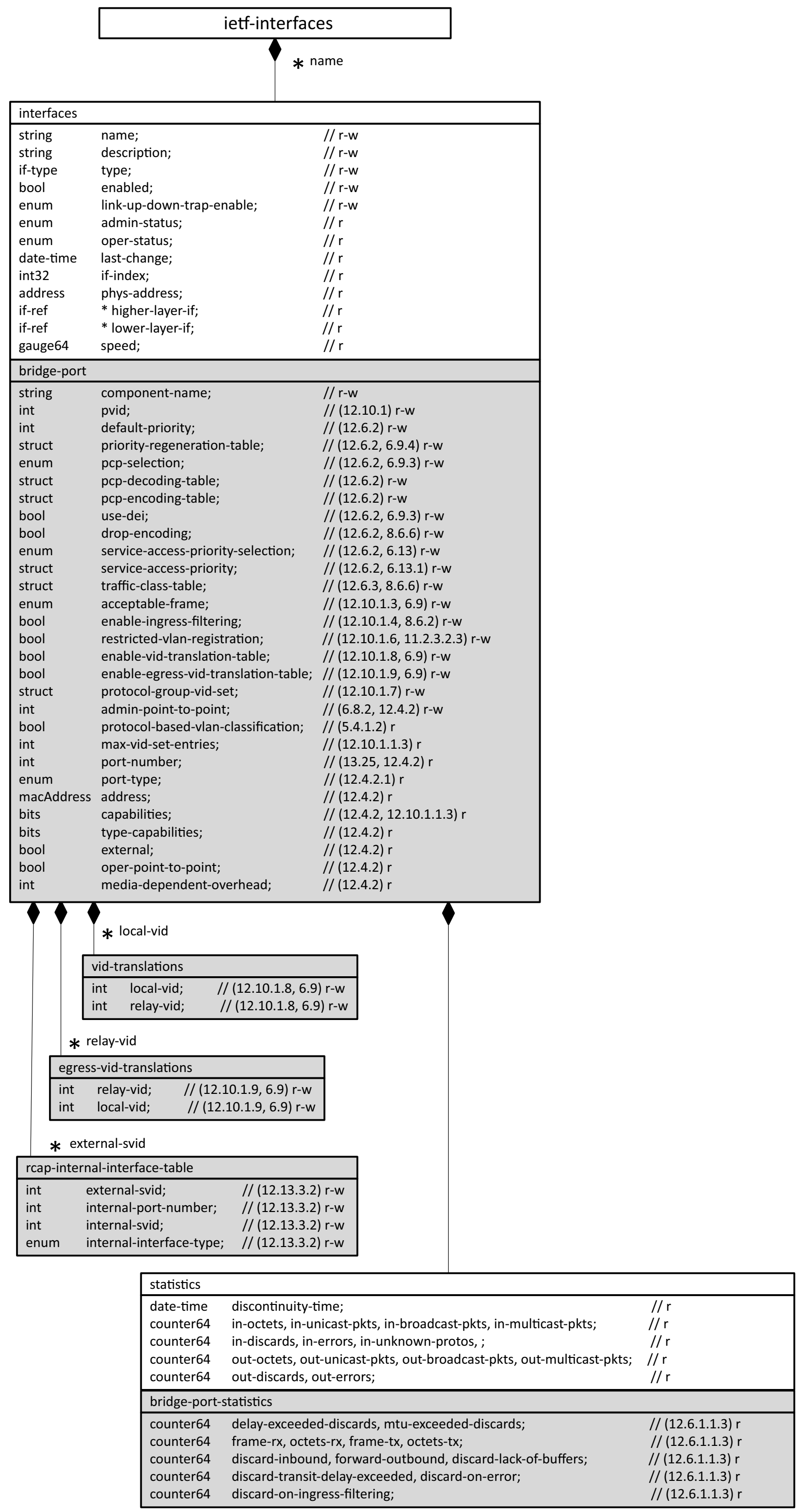
NOTE—Items shaded in gray indicate extensions provided by the IEEE 802.1Q YANG models.

Figure 48-7—TPMR port model



NOTE—Items shaded in gray indicate extensions provided by the IEEE 802.1Q YANG models.

Figure 48-9—Provider Edge Bridge C-VLAN Interface model



NOTE—Items shaded in gray indicate extensions provided by the IEEE 802.1Q YANG models.

Figure 48-10—Provider Edge Bridge S-VLAN interface model

Insert the following subclause (48.3.6, including Figure 48-17) after 48.3.3 (note that subclause numbers 48.3.4 and 48.3.5 and figure numbers 48-15 and 48-16 are reserved for a future amendment):

48.3.6 Stream filters and stream gates model

The stream filters and stream gates model augments the Bridge component model (48.3.1) by nodes which represent the following managed objects:

a) Stream Filter Instance Table (12.31.2)
b) Stream Gate Instance Table (12.31.3)

The UML representation of the stream filters and stream gates model is illustrated in Figure 48-17.

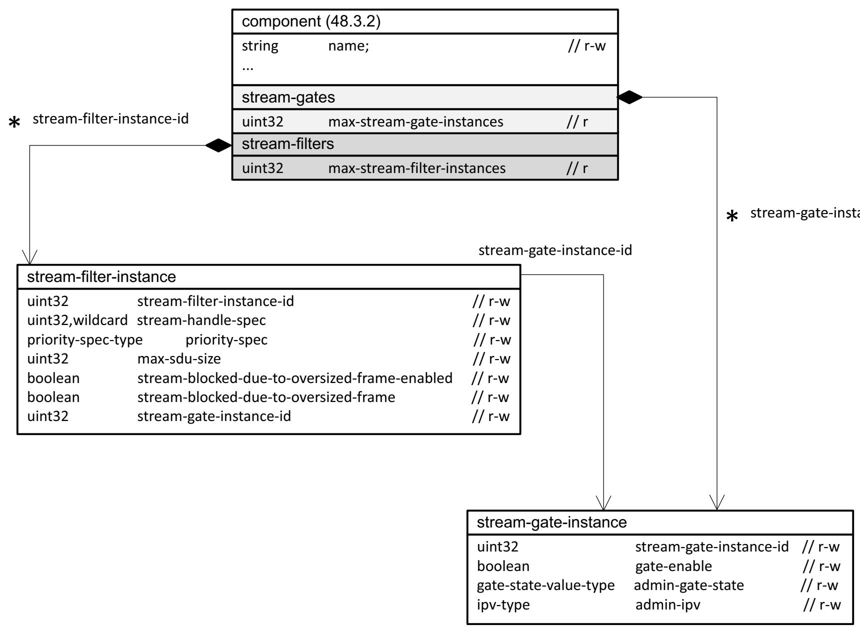


Figure 48-17—Stream filters and stream gates model

Insert the following subclause (48.3.8, including Figure 48-19) after 48.3.6 (note that subclause number 48.3.7 and figure number 48-18 are reserved for a future amendment):

48.3.8 Asynchronous Traffic Shaping (ATS) model

The ATS model augments the Bridge component model (48.3.1) and the stream filters and stream gates model (48.3.6) by nodes which represent to following managed objects:

a) The Scheduler Instance Table (12.31.5)
b) The Scheduler Group Instance Table (12.31.6)
c) The Scheduler Port Parameter Table (12.31.7)
d) The Scheduler Timing Characteristics Table (12.31.8)
e) A Stream Filter specification type representing an ATS scheduler instance identifier (12.31.2.5)

The UML representation of the ATS model is illustrated in Figure 48-19.

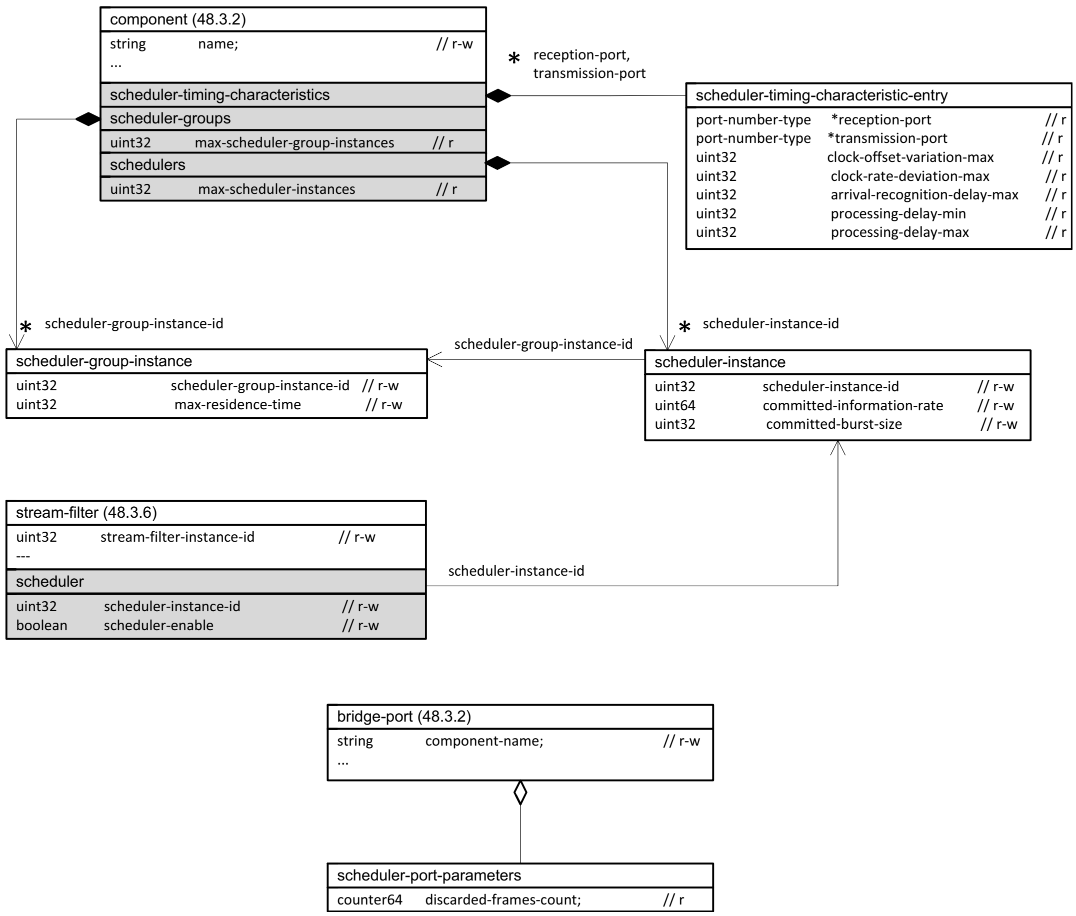


Figure 48-19—Asynchronous Traffic Shaping model

48.4 Structure of the YANG model

Insert the following rows at the end of Table 48-1:

Table 48-1—Summary of YANG modules

Module	References	Notes
ieee802-dot1q-stream-filters-gates	48.5.12, 48.6.12	Basic stream filtering (8.6.5.3) and stream gating (8.6.5.4) capabilities commonly used by PSFP (8.6.5.2.1) and ATS (8.6.5.2.2).
ieee802-dot1q-ats	48.5.14, 48.6.14	ATS-specific extensions to the ieee802-dot1q-stream-filters-gates and ieee802-dot1q-bridge modules.

Insert the following subclause (48.4.8, including Table 48-9) after 48.4.5 (note that subclause numbers 48.4.6 and 48.4.7 and table numbers 48-7 and 48-8 are reserved for a future amendment):

48.4.8 Stream filters and stream gates model

The stream filters and stream gates model (48.3.6) provides basic stream filter (8.6.5.3) and stream gate (8.6.5.4) capabilities and allows for augmentation by specific YANG models (e.g., ATS model).

A system implementing the stream filters and stream gates model implements the YANG modules as described in Table 48-9.

Table 48-9—YANG module dependencies for the stream filters and stream gates model

YANG module	Notes
ieee802-types	—
ieee802-dot1q-types	—
ieee802-dot1q-bridge	—
ieee802-dot1q-stream-filters-gates	—

Insert the following subclause (48.4.10, including Table 48-11) after 48.4.8 (note that subclause number 48.4.9 and table number 48-10 are reserved for a future amendment):

48.4.10 Asynchronous Traffic Shaping (ATS) model

A system implementing the ATS model (48.4.8) implements the YANG modules as described in Table 48-11.

Table 48-11—YANG module dependencies for the ATS model

YANG module	Notes
ieee802-types	—
ieee802-dot1q-types	—
ieee802-dot1q-bridge	—
ieee802-dot1q-stream-filters-gates	—
ieee802-dot1q-ats	—

48.5 YANG data scheme tree definitions

48.5.3 Data scheme definition for the ieee802-dot1q-bridge YANG module

Change 48.5.3 as shown:

```
module: ieee802-dot1q-bridge
    +--rw bridges
       +--rw bridge* [name]
          +--rw name              dot1qtypes:name-type
          +--rw address           ieee:mac-address
          +--rw bridge-type       identityref
          +--ro ports?            uint16
          +--ro up-time?          yang:zero-based-counter32
          +--ro components?       uint32
          +--rw component* [name]
             +--rw name                       string
             +--rw id?                        uint32
             +--rw type                       identityref
             +--rw address?                   ieee:mac-address
             +--rw traffic-class-enabled?     boolean
             +--ro ports?                     uint16
             +--ro bridge-port*               if:interface-ref
             +--ro capabilities
             |  +--ro extended-filtering?              boolean
             |  +--ro traffic-classes?                 boolean
             |  +--ro static-entry-individual-port?    boolean
             |  +--ro ivl-capable?                     boolean
             |  +--ro svl-capable?                     boolean
             |  +--ro hybrid-capable?                  boolean
             |  +--ro configurable-pvid-tagging?       boolean
             |  +--ro local-vlan-capable?              boolean
             +--rw filtering-database
             |  +--rw aging-time?                           uint32
             |  +--ro size?                                 yang:gauge32
             |  +--ro static-entries?                       yang:gauge32
             |  +--ro dynamic-entries?                      yang:gauge32
             |  +--ro static-vlan-registration-entries?     yang:gauge32
             |  +--ro dynamic-vlan-registration-entries?    yang:gauge32
            |  +--ro mac-address-registration-entries?    yang:gauge32 {extended-
filtering-services}?
             |  +--rw filtering-entry* [database-id vids address]
             |  |  +--rw database-id     uint32
             |  |  +--rw address         ieee:mac-address
             |  |  +--rw vids            dot1qtypes:vid-range-type
             |  |  +--rw entry-type?     enumeration
             |  |  +--rw port-map* [port-ref]
             |  |  |  +--rw port-ref                              port-number-type
             |  |  |  +--rw (map-type)?
             |  |  |     +--:(static-filtering-entries)
             |  |  |     |  +--rw static-filtering-entries
             |  |  |     |     +--rw control-element?          enumeration
             |  |  |     |     +--rw connection-identifier?    port-number-type
             |  |  |     +--:(static-vlan-registration-entries)
             |  |  |     |  +--rw static-vlan-registration-entries
             |  |  |     |     +--rw registrar-admin-control?    enumeration
             |  |  |     |     +--rw vlan-transmitted?           enumeration
             |  |  |     +--:(mac-address-registration-entries)
             |  |  |     |  +--rw mac-address-registration-entries
             |  |  |     |     +--rw control-element?    enumeration
             |  |  |     +--:(dynamic-vlan-registration-entries)
             |  |  |     |  +--rw dynamic-vlan-registration-entries
             |  |  |     |     +--rw control-element?    enumeration
             |  |  |     +--:(dynamic-reservation-entries)
             |  |  |     |  +--rw dynamic-reservation-entries
             |  |  |     |     +--rw control-element?    enumeration
             |  |  |     +--:(dynamic-filtering-entries)
             |  |  |        +--rw dynamic-filtering-entries
             |  |  |           +--rw control-element?    enumeration
```

```
   |  |  +--ro status?        enumeration
   |  +--rw vlan-registration-entry* [database-id vids]
   |     +--rw database-id    uint32
   |     +--rw vids           dot1qtypes:vid-range-type
   |     +--rw entry-type?    enumeration
   |     +--rw port-map* [port-ref]
   |        +--rw port-ref                              port-number-type
   |        +--rw (map-type)?
   |           +--:(static-filtering-entries)
   |           |  +--rw static-filtering-entries
   |           |     +--rw control-element?         enumeration
   |           |     +--rw connection-identifier?   port-number-type
   |           +--:(static-vlan-registration-entries)
   |           |  +--rw static-vlan-registration-entries
   |           |     +--rw registrar-admin-control?   enumeration
   |           |     +--rw vlan-transmitted?          enumeration
   |           +--:(mac-address-registration-entries)
   |           |  +--rw mac-address-registration-entries
   |           |     +--rw control-element?   enumeration
   |           +--:(dynamic-vlan-registration-entries)
   |           |  +--rw dynamic-vlan-registration-entries
   |           |     +--rw control-element?   enumeration
   |           +--:(dynamic-reservation-entries)
   |           |  +--rw dynamic-reservation-entries
   |           |     +--rw control-element?   enumeration
   |           +--:(dynamic-filtering-entries)
   |              +--rw dynamic-filtering-entries
   |                 +--rw control-element?   enumeration
   +--rw permanent-database
   |  +--ro size?                              yang:gauge32
   |  +--ro static-entries?                    yang:gauge32
   |  +--ro static-vlan-registration-entries?   yang:gauge32
   |  +--rw filtering-entry* [database-id vids address]
   |     +--rw database-id    uint32
   |     +--rw address        ieee:mac-address
   |     +--rw vids           dot1qtypes:vid-range-type
   |     +--ro status?        enumeration
   |     +--rw port-map* [port-ref]
   |        +--rw port-ref                              port-number-type
   |        +--rw (map-type)?
   |           +--:(static-filtering-entries)
   |           |  +--rw static-filtering-entries
   |           |     +--rw control-element?         enumeration
   |           |     +--rw connection-identifier?   port-number-type
   |           +--:(static-vlan-registration-entries)
   |           |  +--rw static-vlan-registration-entries
   |           |     +--rw registrar-admin-control?   enumeration
   |           |     +--rw vlan-transmitted?          enumeration
   |           +--:(mac-address-registration-entries)
   |           |  +--rw mac-address-registration-entries
   |           |     +--rw control-element?   enumeration
   |           +--:(dynamic-vlan-registration-entries)
   |           |  +--rw dynamic-vlan-registration-entries
   |           |     +--rw control-element?   enumeration
   |           +--:(dynamic-reservation-entries)
   |           |  +--rw dynamic-reservation-entries
   |           |     +--rw control-element?   enumeration
   |           +--:(dynamic-filtering-entries)
   |              +--rw dynamic-filtering-entries
   |                 +--rw control-element?   enumeration
   +--rw bridge-vlan
   |  +--ro version?                 uint16
   |  +--ro max-vids?                uint16
   |  +--ro override-default-pvid?   boolean
   |  +--ro protocol-template?        dot1qtypes:protocol-frame-format-
type {port-and-protocol-based-vlan}?
   |  +--ro max-msti?                uint16
   |  +--rw vlan* [vid]
   |  |  +--rw vid              dot1qtypes:vlan-index-type
   |  |  +--rw name?            dot1qtypes:name-type
   |  |  +--ro untagged-ports*   if:interface-ref
   |  |  +--ro egress-ports*     if:interface-ref
```

```
             |  +--rw protocol-group-database* [db-index] {port-and-protocol-
based-vlan}?
             |  |  +--rw db-index             uint16
             |  |  +--rw frame-format-type?   dot1qtypes:protocol-frame-format-
type
             |  |  +--rw (frame-format)?
             |  |  |  +--:(ethernet-rfc1042-snap8021H)
             |  |  |  |  +--rw ethertype?           dot1qtypes:ethertype-type
             |  |  |  +--:(snap-other)
             |  |  |  |  +--rw protocol-id?         string
             |  |  |  +--:(llc-other)
             |  |  |     +--rw dsap-ssap-pairs
             |  |  |        +--rw llc-address?   string
             |  |  +--rw group-id?            uint32
             |  +--rw vid-to-fid-allocation* [vids]
             |  |  +--rw vids               dot1qtypes:vid-range-type
             |  |  +--ro fid?               uint32
             |  |  +--ro allocation-type?   enumeration
             |  +--rw fid-to-vid-allocation* [fid]
             |  |  +--rw fid                uint32
             |  |  +--ro allocation-type?   enumeration
             |  |  +--ro vid*               dot1qtypes:vlan-index-type
             |  +--rw vid-to-fid* [vid]
             |     +--rw vid    dot1qtypes:vlan-index-type
             |     +--rw fid?   uint32
             +--rw bridge-mst
                +--rw mstid*                     dot1qtypes:mstid-type
                +--rw fid-to-mstid* [fid]
                |  +--rw fid      uint32
                |  +--rw mstid?   dot1qtypes:mstid-type
                +--rw fid-to-mstid-allocation* [fids]
                   +--rw fids     dot1qtypes:vid-range-type
                   +--rw mstid?   dot1qtypes:mstid-type
  augment /if:interfaces/if:interface:
    +--rw bridge-port
       +--rw component-name?                       string
       +--rw port-type?                            identityref
       +--rw pvid?                                 dot1qtypes:vlan-index-type
       +--rw default-priority?                     dot1qtypes:priority-type
       +--rw priority-regeneration
       |  +--rw priority0?   priority-type
       |  +--rw priority1?   priority-type
       |  +--rw priority2?   priority-type
       |  +--rw priority3?   priority-type
       |  +--rw priority4?   priority-type
       |  +--rw priority5?   priority-type
       |  +--rw priority6?   priority-type
       |  +--rw priority7?   priority-type
       +--rw pcp-selection?                       dot1qtypes:pcp-selection-type
       +--rw pcp-decoding-table
       |  +--rw pcp-decoding-map* [pcp]
       |     +--rw pcp             pcp-selection-type
       |     +--rw priority-map* [priority-code-point]
       |        +--rw priority-code-point    priority-type
       |        +--rw priority?              priority-type
       |        +--rw drop-eligible?         boolean
       +--rw pcp-encoding-table
       |  +--rw pcp-encoding-map* [pcp]
       |     +--rw pcp             pcp-selection-type
       |     +--rw priority-map* [priority dei]
       |        +--rw priority               priority-type
       |        +--rw dei                    boolean
       |        +--rw priority-code-point?   priority-type
       +--rw use-dei?                              boolean
       +--rw drop-encoding?                        boolean
       +--rw service-access-priority-selection?    boolean
       +--rw service-access-priority
       |  +--rw priority0?   priority-type
       |  +--rw priority1?   priority-type
       |  +--rw priority2?   priority-type
       |  +--rw priority3?   priority-type
       |  +--rw priority4?   priority-type
```

```
       |  +--rw priority5?   priority-type
       |  +--rw priority6?   priority-type
       |  +--rw priority7?   priority-type
       +--rw traffic-class
       |  +--rw traffic-class-map* [priority]
       |     +--rw priority                   priority-type
       |     +--rw available-traffic-class* [num-traffic-class]
       |        +--rw num-traffic-class    uint8
       |        +--rw traffic-class?       traffic-class-type
       +--rw acceptable-frame?                         enumeration
       +--rw enable-ingress-filtering?                 boolean
       +--rw enable-restricted-vlan-registration?   boolean
       +--rw enable-vid-translation-table?             boolean
       +--rw enable-egress-vid-translation-table?   boolean
       +--rw protocol-group-vid-set* [group-id] {port-and-protocol-based-vlan}?
       |  +--rw group-id    uint32
       |  +--rw vid*        dot1qtypes:vlanid
       +--rw admin-point-to-point?                     enumeration
       +--ro protocol-based-vlan-classification?    boolean {port-and-protocol-
based-vlan}?
       +--ro max-vid-set-entries?                      uint16 {port-and-protocol-
based-vlan}?
       +--ro port-number?                              dot1qtypes:port-number-type
       +--ro address?                                  ieee:mac-address
       +--ro capabilities?                             bits
       +--ro type-capabilties?                         bits
       +--ro external?                                 boolean
       +--ro oper-point-to-point?                      boolean
       +--ro media-dependent-overhead?                 uint8
       +--ro statistics
       |  +--ro delay-exceeded-discards?             yang:counter64
       |  +--ro mtu-exceeded-discards?               yang:counter64
       |  +--ro frame-rx?                            yang:counter64
       |  +--ro octets-rx?                           yang:counter64
       |  +--ro frame-tx?                            yang:counter64
       |  +--ro octets-tx?                           yang:counter64
       |  +--ro discard-inbound?                     yang:counter64
       |  +--ro forward-outbound?                    yang:counter64
       |  +--ro discard-lack-of-buffers?             yang:counter64
       |  +--ro discard-transit-delay-exceeded?   yang:counter64
       |  +--ro discard-on-error?                    yang:counter64
       |  +--ro discard-on-ingress-filtering?     yang:counter64 {ingress-
filtering}?
       +--rw vid-translations* [local-vid]
       |  +--rw local-vid    dot1qtypes:vlanid
       |  +--rw relay-vid?   dot1qtypes:vlanid
       +--rw egress-vid-translations* [relay-vid]
          +--rw relay-vid    dot1qtypes:vlanid
          +--rw local-vid?   dot1qtypes:vlanid
```

Insert the following subclause (48.5.12) after 48.5.9 (note that subclause numbers 48.5.10 and 48.5.11 are reserved for a future amendment):

48.5.12 Data scheme definition for the ieee802-dot1q-stream-filters-gates YANG module

```
module: ieee802-dot1q-stream-filters-gates
  augment /dot1q:bridges/dot1q:bridge/dot1q:component:
    +--rw stream-filters
    |  +--rw stream-filter-instance-table* [stream-filter-instance-id]
    |  |  +--rw stream-filter-instance-id                         uint32
    |  |  +--rw (stream-handle-spec)?
    |  |  |  +--:(wildcard)
    |  |  |  |  +--rw wildcard?                                   empty
    |  |  |  +--:(stream-handle)
    |  |  |     +--rw stream-handle                               uint32
```

```
    |  |  +--rw priority-spec                                   priority-spec-type
    |  |  +--rw max-sdu-size                                    uint32
    |  |  +--rw stream-blocked-due-to-oversize-frame-enabled?   boolean
    |  |  +--rw stream-blocked-due-to-oversize-frame?           boolean
    |  |  +--rw stream-gate-ref                                 stream-gate-ref
    |  +--ro max-stream-filter-instances?    uint32
    +--rw stream-gates
       +--rw stream-gate-instance-table* [stream-gate-instance-id]
       |  +--rw stream-gate-instance-id    uint32
       |  +--rw gate-enable?               boolean
       |  +--rw admin-gate-states?         gate-state-value-type
       |  +--rw admin-ipv?                 ipv-spec-type
       +--ro max-stream-gate-instances?    uint32
```

Insert the following subclause (48.5.14) after 48.5.12 (note that subclause number 48.5.13 is reserved for a future amendment):

48.5.14 Data scheme definition for the ieee802-dot1q-ats YANG module

```
module: ieee802-dot1q-ats
      augment   /dot1q:bridges/dot1q:bridge/dot1q:component/sfsg:stream-filters/
sfsg:stream-filter-instance-table:
    +--rw scheduler
       +--rw scheduler-ref?      ats:scheduler-ref-type
       +--rw scheduler-enable?   boolean
  augment /if:interfaces/if:interface/dot1q:bridge-port:
    +--rw ats-port-parameters
       +--ro discarded-frames-count?   yang:counter64
  augment /dot1q:bridges/dot1q:bridge/dot1q:component:
    +--rw schedulers
    |  +--rw scheduler-instance-table* [scheduler-instance-id]
    |  |  +--rw scheduler-instance-id         uint32
    |  |  +--rw committed-information-rate    uint64
    |  |  +--rw committed-burst-size          uint32
    |  |  +--rw scheduler-group-ref           ats:scheduler-group-ref-type
    |  +--ro max-scheduler-instances?    uint32
    +--rw scheduler-groups
       +--rw scheduler-group-instance-table* [scheduler-group-instance-id]
       |  +--rw scheduler-group-instance-id    uint32
       |  +--rw max-residence-time             uint32
       +--ro max-scheduler-group-instances?      uint32
       +--rw scheduler-timing-characteristics
                +--ro  scheduler-timing-characteristics-table*  [reception-port
transmission-port]
             +--ro reception-port                   dot1qtypes:port-number-type
             +--ro transmission-port                dot1qtypes:port-number-type
             +--ro clock-offset-variation-max       uint32
             +--ro clock-rate-deviation-max         uint32
             +--ro arrival-recognition-delay-max    uint32
             +--ro processing-delay-min             uint32
             +--ro processing-delay-max             uint32
```

48.6 YANG modules[6,7]

48.6.3 Definition for the ieee802-dot1q-bridge YANG module

Change 48.6.3 as shown:

```
module ieee802-dot1q-bridge {
  namespace urn:ieee:std:802.1Q:yang:ieee802-dot1q-bridge;
  prefix dot1q;
  import ieee802-types {
    prefix ieee;
  }
  import ietf-yang-types {
    prefix yang;
  }
  import ietf-interfaces {
    prefix if;
  }
  import iana-if-type {
    prefix ianaif;
  }
  import ieee802-dot1q-types {
    prefix dot1qtypes;
  }
  organization
    "IEEE 802.1 Working Group";
  contact
    "WG-URL: http://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
             Piscataway, NJ 08854
             USA

     E-mail: stds-802-1-chairs@ieee.org";
  description
    "This YANG module describes the bridge configuration model for the
    following IEEE 802.1Q Bridges:
      1) Two Port MAC Relays
      2) Customer VLAN Bridges
      3) Provider Bridges.";
  revision 2020-11-06 {
    description
      "Published as part of IEEE Std 802.1Qcr-2020.
      Third version.";
    reference
      "IEEE Std 802.1Qcr-2020, Bridges and Bridged Networks -
      Asynchronous Traffic Shaping.";
  }
  revision 2020-06-04 {
```

[6] *Copyright release for YANG:* Users of this standard may freely reproduce the YANG modules contained in this subclause so that they can be used for their intended purpose.
[7] An ASCII version of each YANG module is attached to the PDF of this amendment and can also be obtained from the IEEE 802.1 Website at https://1.ieee802.org/yang-modules/.

```
  description
    "Published as part of IEEE Std 802.1Qcx-2020.
    Second version.";
  reference
    "IEEE Std 802.1Qcx-2020, Bridges and Bridged Networks -
    YANG Data Model for Connectivity Fault Management.";
}
revision 2018-03-07 {
  description
    "Published as part of IEEE Std 802.1Q-2018.
    Initial version.";
  reference
    "IEEE Std 802.1Q-2018, Bridges and Bridged Networks.";
}

feature ingress-filtering {
  description
    "Each Port may support an Enable Ingress Filtering parameter. A
    frame received on a Port that is not in the member set (8.8.10)
    associated with the frames VID shall be discarded if this
    parameter is set. The default value for this parameter is reset,
    i.e., Disable Ingress Filtering, for all Ports. Any Port that
    supports setting this parameter shall also support resetting it.
    The parameter may be configured by the management operations
    defined in Clause 12.";
  reference
    "8.6.2 of IEEE Std 802.1Q-2018";
}
feature extended-filtering-services {
  description
    "Extended Filtering Services support the filtering behavior
    required for regions of a network in which potential recipients
    of multicast frames exist, and where both the potential
    recipients of frames and the Bridges are able to support dynamic
    configuration of filtering information for group MAC addresses.
    In order to integrate this extended filtering behavior with the
    needs of regions of the network that support only Basic
    Filtering Services, Bridges that support Extended Filtering
    Services can be statically and dynamically configured to modify
    their filtering behavior on a per-group MAC address basis, and
    also on the basis of the overall filtering service provided by
    each outbound Port with regard to multicast frames. The latter
    capability permits configuration of the Ports default forwarding
    or filtering behavior with regard to group MAC addresses for
    which no specific static or dynamic filtering information has
    been configured.";
  reference
    "8.8.4 of IEEE Std 802.1Q-2018
    Clause 10 of IEEE Std 802.1Q-2018";
}
feature port-and-protocol-based-vlan {
  description
    "A VLAN-aware bridge component implementation in conformance to
    the provisions of this standard for Port-and-Protocol-based VLAN
    classification (5.4.1) shall 1) Support one or more of the
    following Protocol Classifications and Protocol Template
    formats: Ethernet, RFC_1042, SNAP_8021H, SNAP_Other, or
    LLC_Other (6.12); and may 2) Support configuration of the
    contents of the Protocol Group Database.";
```

```
    reference
      "5.4.1.2 of IEEE Std 802.1Q-2018";
  }
  feature flow-filtering {
    description
      "Flow filtering support enables Bridges to distinguish frames
      belonging to different client flows and to use this information
      in the forwarding process. Information related to client flows
      may be used at the boundary of an SPT Domain to generate a flow
      hash value. The flow hash, carried in an F-TAG, serves to
      distinguish frames belonging to different flows and can be used
      in the forwarding process to distribute frames over equal cost
      paths. This provides for finer granularity load spreading while
      maintaining frame order for each client flow.";
    reference
      "44.2 of IEEE Std 802.1Q-2018";
  }
  feature simple-bridge-port {
    description
      "A simple bridge port allows underlying (MAC) layers to share
      the same Interface as the Bridge Port.";
  }
  feature flexible-bridge-port {
    description
      "A flexible bridge port supports an Interface that is a Bridge
      Port to be a separate Interface from the underlying (MAC) layer.";
  }

  identity type-of-bridge {
    description
      "Represents the configured Bridge type.";
  }
  identity customer-vlan-bridge {
    base type-of-bridge;
    description
      "Base identity for a Customer VLAN Bridge.";
  }
  identity provider-bridge {
    base type-of-bridge;
    description
      "Base identity for a Provider Bridge (PB).";
  }
  identity provider-edge-bridge {
    base type-of-bridge;
    description
      "Base identity for a Provider Edge Bridge (PEB).";
  }
  identity two-port-mac-relay-bridge {
    base type-of-bridge;
    description
      "Base identity for a Two Port MAC Relay (TPMR).";
  }
  identity type-of-component {
    description
      "Represents the type of Component.";
  }
  identity c-vlan-component {
    base type-of-component;
    description
```

```
    "Base identity for a C-VLAN component.";
}
identity s-vlan-component {
  base type-of-component;
  description
    "Base identity for a S-VLAN component.";
}
identity d-bridge-component {
  base type-of-component;
  description
    "Base identity for a VLAN unaware component.";
}
identity edge-relay-component {
  base type-of-component;
  description
    "Base identity for an EVB station ER component.";
}
identity type-of-port {
  description
    "Represents the type of Bridge port.";
}
identity c-vlan-bridge-port {
  base type-of-port;
  description
    "Indicates the port can be a C-TAG aware port of an enterprise
    VLAN aware Bridge.";
}
identity provider-network-port {
  base type-of-port;
  description
    "Indicates the port can be an S-TAG aware port of a Provider
    Bridge or Backbone Edge Bridge used for connections within a PBN
    (Provider Bridged Network) or PBBN (Provider Backbone Bridged
    Network).";
}
identity customer-network-port {
  base type-of-port;
  description
    "Indicates the port can be an S-TAG aware port of a Provider
    Bridge or Backbone Edge Bridge used for connections to the
    exterior of a PBN (Provider Bridged Network) or PBBN (Provider
    Backbone Bridged Network).";
}
identity customer-edge-port {
  base type-of-port;
  description
    "Indicates the port can be a C-TAG aware port of a Provider
    Bridge used for connections to the exterior of a PBN (Provider
    Bridged Network) or PBBN (Provider Backbone Bridged Network).";
}
identity d-bridge-port {
  base type-of-port;
  description
    "Indicates the port can be a VLAN-unaware member of an 802.1Q
    Bridge.";
}
identity remote-customer-access-port {
  base type-of-port;
  description
```

```
    "Indicates the port can be an S-TAG aware port of a Provider
    Bridge capable of providing Remote Customer Service Interfaces.";
}
identity bridge-interface {
  description
    "Generic interface property that represents any interface that
    can be associated with an IEEE 802.1Q compliant Bridge
    component. Any new Interface types would derive from this
    identity to automatically pick up Bridge related configuration
    or operational data.";
}

container bridges {
  description
    "Contains the Bridge(s) configuration information.";
  list bridge {
    key "name";
    unique "address";
    description
      "Provides configuration data in support of the Bridge
      Configuration resources. There is a single bridge data node
      per Bridge.";
    leaf name {
      type dot1qtypes:name-type;
      description
        "A text string associated with the Bridge, of locally
        determined significance.";
      reference
        "12.4 of IEEE Std 802.1Q-2018";
    }
    leaf address {
      type ieee:mac-address;
      mandatory true;
      description
        "The MAC address for the Bridge from which the Bridge
        Identifiers used by the STP, RSTP, and MSTP are derived.";
      reference
        "12.4 of IEEE Std 802.1Q-2018";
    }
    leaf bridge-type {
      type identityref {
        base type-of-bridge;
      }
      mandatory true;
      description
        "The type of Bridge.";
    }
    leaf ports {
      type uint16 {
        range "1..4095";
      }
      config false;
      description
        "The number of Bridge Ports (MAC Entities)";
      reference
        "12.4 of IEEE Std 802.1Q-2018";
    }
    leaf up-time {
      type yang:zero-based-counter32;
```

```
    units "seconds";
    config false;
    description
      "The count in seconds of the time elapsed since the Bridge
      was last reset or initialized.";
    reference
      "12.4 of IEEE Std 802.1Q-2018";
  }
  leaf components {
    type uint32;
    config false;
    description
      "The number of components associated with the Bridge.";
  }
  list component {
    key "name";
    description
      "The set of components associated with a given Bridge. For
      example, - A TPMR is associated with a single VLAN
      unaware component. - A Customer VLAN Bridge is associated
      with a single VLAN aware component. - A Provider Bridge is
      associated with a single S-VLAN component and zero or more
      C-VLAN components.";
    reference
      "12.3 of IEEE Std 802.1Q-2018";
    leaf name {
      type string;
      description
        "The name of the Component.";
    }
    leaf id {
      type uint32;
      description
        "Unique identifier for a particular Bridge component
        within the system.";
      reference
        "12.3, item l) of IEEE Std 802.1Q-2018";
    }
    leaf type {
      type identityref {
        base type-of-component;
      }
      mandatory true;
      description
        "The type of component used to classify a particular
        Bridge component within a Bridge system comprising
        multiple components.";
      reference
        "12.3, item m) of IEEE Std 802.1Q-2018";
    }
    leaf address {
      type ieee:mac-address;
      description
        "Unique EUI-48 Universally Administered MAC address
        assigned to a Bridge component.";
      reference
        "13.24 of IEEE Std 802.1Q-2018
        8.13.8 of IEEE Std 802.1Q-2018";
    }
```

```
leaf traffic-class-enabled {
  type boolean;
  default "true";
  description
    "Indication of Traffic Classes enablement associated with
    the Bridge Component. A value of True indicates that
    Traffic Classes are enabled on this Bridge Component. A
    value of False indicates that the Bridge Component
    operates with a single priority level for all traffic.";
  reference
    "12.4.1.5.1 of IEEE Std 802.1Q-2018";
}
leaf ports {
  type uint16 {
    range "1..4095";
  }
  config false;
  description
    "The number of Bridge Ports associated with the Bridge
    Component.";
  reference
    "12.4.1.1.3, item c) of IEEE Std 802.1Q-2018";
}
leaf-list bridge-port {
  type if:interface-ref;
  config false;
  description
    "List of bridge-port references.";
}
container capabilities {
  config false;
  description
    "Array of Boolean values of the feature capabilities
    associated with a given Bridge Component.";
  reference
    "12.10.1.1.3, item b) of IEEE Std 802.1Q-2018
    12.4.1.5.2 of IEEE Std 802.1Q-2018";
  leaf extended-filtering {
    type boolean;
    default "false";
    description
      "Can perform filtering on individual multicast addresses
      controlled by MMRP.";
    reference
      "12.4.1.5.2 of IEEE Std 802.1Q-2018";
  }
  leaf traffic-classes {
    type boolean;
    default "false";
    description
      "Can map priority to multiple traffic classes.";
    reference
      "12.4.1.5.2 of IEEE Std 802.1Q-2018";
  }
  leaf static-entry-individual-port {
    type boolean;
    default "false";
    description
      "Static entries per port.";
```

```
      reference
        "12.4.1.5.2 of IEEE Std 802.1Q-2018";
    }
    leaf ivl-capable {
      type boolean;
      default "true";
      description
        "Independent VLAN Learning (IVL).";
      reference
        "12.4.1.5.2 of IEEE Std 802.1Q-2018";
    }
    leaf svl-capable {
      type boolean;
      default "false";
      description
        "Shared VLAN Learning (SVL).";
      reference
        "12.4.1.5.2 of IEEE Std 802.1Q-2018";
    }
    leaf hybrid-capable {
      type boolean;
      default "false";
      description
        "Both IVL and SVL simultaneously.";
      reference
        "12.4.1.5.2 of IEEE Std 802.1Q-2018";
    }
    leaf configurable-pvid-tagging {
      type boolean;
      default "false";
      description
        "Whether the implementation supports the ability to
        override the default PVID setting and its egress status
        (VLAN-tagged or Untagged) on each port.";
      reference
        "12.4.1.5.2 of IEEE Std 802.1Q-2018";
    }
    leaf local-vlan-capable {
      type boolean;
      default "false";
      description
        "Can support multiple local Bridges, outside the scope
        of 802.1Q defined VLANs.";
      reference
        "12.4.1.5.2 of IEEE Std 802.1Q-2018";
    }
  }
  container filtering-database {
    when "../../bridge-type != 'two-port-mac-relay-bridge'" {
      description
        "Applies to non TPMRs.";
    }
    description
      "Contains filtering information used by the Forwarding
      Process in deciding through which Ports of the Bridge
      frames should be forwarded.";
    reference
      "12.7 of IEEE Std 802.1Q-2018";
    leaf aging-time {
```

```
    type uint32 {
      range "10..10000000";
    }
    units "seconds";
    default "300";
    description
      "The timeout period in seconds for aging out
      dynamically-learned forwarding information.";
    reference
      "12.7 of IEEE Std 802.1Q-2018
      8.8.3 of IEEE Std 802.1Q-2018";
  }
  leaf size {
    type yang:gauge32;
    config false;
    description
      "The maximum number of entries that can be held in the
      FDB.";
    reference
      "12.7 of IEEE Std 802.1Q-2018";
  }
  leaf static-entries {
    type yang:gauge32;
    config false;
    description
      "The number of Static Filtering entries currently in the
      FDB.";
    reference
      "12.7 of IEEE Std 802.1Q-2018
      8.8.1 of IEEE Std 802.1Q-2018";
  }
  leaf dynamic-entries {
    type yang:gauge32;
    config false;
    description
      "The number of Dynamic Filtering entries currently in
      the FDB.";
    reference
      "12.7 of IEEE Std 802.1Q-2018
      8.8.3 of IEEE Std 802.1Q-2018";
  }
  leaf static-vlan-registration-entries {
    type yang:gauge32;
    config false;
    description
      "The number of Static VLAN Registration entries
      currently in the FDB.";
    reference
      "12.7 of IEEE Std 802.1Q-2018
      8.8.2 of IEEE Std 802.1Q-2018";
  }
  leaf dynamic-vlan-registration-entries {
    type yang:gauge32;
    config false;
    description
      "The number of Dynamic VLAN Registration entries
      currently in the FDB.";
    reference
      "12.7 of IEEE Std 802.1Q-2018
```

```
    8.8.5 of IEEE Std 802.1Q-2018";
}
leaf mac-address-registration-entries {
  if-feature "extended-filtering-services";
  type yang:gauge32;
  config false;
  description
    "The number of MAC Address Registration entries
    currently in the FDB.";
  reference
    "12.7 of IEEE Std 802.1Q-2018
    8.8.4 of IEEE Std 802.1Q-2018";
}
list filtering-entry {
  key "database-id vids address";
  description
    "Information for the entries associated with the
    Permanent Database.";
  leaf database-id {
    type uint32;
    description
      "The identity of this Filtering Database.";
    reference
      "12.7.7 of IEEE Std 802.1Q-2018";
  }
  leaf address {
    type ieee:mac-address;
    description
      "A MAC address (unicast, multicast, broadcast) for
      which the device has forwarding and/or filtering
      information.";
    reference
      "12.7.7 of IEEE Std 802.1Q-2018";
  }
  leaf vids {
    type dot1qtypes:vid-range-type;
    description
      "The set of VLAN identifiers to which this entry
      applies.";
    reference
      "12.7.7 of IEEE Std 802.1Q-2018";
  }
  leaf entry-type {
    type enumeration {
      enum static {
        description
          "Static entry type";
      }
      enum dynamic {
        description
          "Dynamic/learnt entry type";
      }
    }
    description
      "The type of filtering entry. Whether static or
      dynamic. Static entries can be created, deleted, and
      retrieved. However, dynamic entries can only be
      deleted or retrieved by the management entity.
      Consequently, a Bridge is not required to accept a
```

```
        command that can alter the dynamic entries except
        delete a dynamic entry.";
      reference
        "12.7.7 of IEEE Std 802.1Q-2018";
    }
    uses dot1qtypes:port-map-grouping;
    leaf status {
      type enumeration {
        enum other {
          description
            "None of the following. This may include the case
            where some other object is being used to determine
            if and how frames addressed to the value of the
            corresponding instance of 'address' are being
            forwarded.";
        }
        enum invalid {
          description
            "This entry is no longer valid (e.g., it was
            learned but has since aged out), but has not yet
            been flushed from the table.";
        }
        enum learned {
          description
            "The value of the corresponding instance of the
            port node was learned and is being used.";
        }
        enum self {
          description
            "The value of the corresponding instance of the
            address node representing one of the devices
            address.";
        }
        enum mgmt {
          description
            "The value of the corresponding instance of
            address node that is also the value of an existing
            instance.";
        }
      }
      config false;
      description
        "The status of this entry.";
    }
  }
  list vlan-registration-entry {
    key "database-id vids";
    description
      "The VLAN Registration Entries models the operations
      that can be performed on a single VLAN Registration
      Entry in the FDB. The set of VLAN Registration Entries
      within the FDB changes under management control and also
      as a result of MVRP exchanges";
    reference
      "12.7.5 of IEEE Std 802.1Q-2018";
    leaf database-id {
      type uint32;
      description
        "The identity of this Filtering Database.";
```

```
          reference
            "12.7.7 of IEEE Std 802.1Q-2018";
        }
        leaf vids {
          type dot1qtypes:vid-range-type;
          description
            "The set of VLAN identifiers to which this entry
            applies.";
          reference
            "12.7.7 of IEEE Std 802.1Q-2018";
        }
        leaf entry-type {
          type enumeration {
            enum static {
              description
                "Static entry type";
            }
            enum dynamic {
              description
                "Dynamic/learnt entry type";
            }
          }
          description
            "The type of filtering entry. Whether static or
            dynamic. Static entries can be created, deleted, and
            retrieved. However, dynamic entries can only be
            deleted or retrieved by the management entity.
            Consequently, a Bridge is not required to accept a
            command that can alter the dynamic entries except
            delete a dynamic entry.";
          reference
            "12.7.7 of IEEE Std 802.1Q-2018";
        }
        uses dot1qtypes:port-map-grouping;
      }
    }
    container permanent-database {
      description
        "The Permanent Database container models the operations
        that can be performed on, or affect, the Permanent
        Database. There is a single Permanent Database per FDB.";
      leaf size {
        type yang:gauge32;
        config false;
        description
          "The maximum number of entries that can be held in the
          FDB.";
        reference
          "12.7.6 of IEEE Std 802.1Q-2018";
      }
      leaf static-entries {
        type yang:gauge32;
        config false;
        description
          "The number of Static Filtering entries currently in the
          FDB.";
        reference
          "12.7.6 of IEEE Std 802.1Q-2018";
      }
```

```
leaf static-vlan-registration-entries {
  type yang:gauge32;
  config false;
  description
    "The number of Static VLAN Registration entries
    currently in the FDB.";
  reference
    "12.7.6 of IEEE Std 802.1Q-2018";
}
list filtering-entry {
  key "database-id vids address";
  description
    "Information for the entries associated with the
    Permanent Database.";
  leaf database-id {
    type uint32;
    description
      "The identity of this Filtering Database.";
    reference
      "12.7.7 of IEEE Std 802.1Q-2018";
  }
  leaf address {
    type ieee:mac-address;
    description
      "A MAC address (unicast, multicast, broadcast) for
      which the device has forwarding and/or filtering
      information.";
    reference
      "12.7.7 of IEEE Std 802.1Q-2018";
  }
  leaf vids {
    type dot1qtypes:vid-range-type;
    description
      "The set of VLAN identifiers to which this entry
      applies.";
    reference
      "12.7.7 of IEEE Std 802.1Q-2018";
  }
  leaf status {
    type enumeration {
      enum other {
        description
          "None of the following. This may include the case
          where some other object is being used to determine
          if and how frames addressed to the value of the
          corresponding instance of 'address' are being
          forwarded.";
      }
      enum invalid {
        description
          "This entry is no longer valid (e.g., it was
          learned but has since aged out), but has not yet
          been flushed from the table.";
      }
      enum learned {
        description
          "The value of the corresponding instance of the
          port node was learned and is being used.";
      }
```

```
          enum self {
            description
              "The value of the corresponding instance of the
              address node representing one of the devices
              address.";
          }
          enum mgmt {
            description
              "The value of the corresponding instance of
              address node that is also the value of an existing
              instance.";
          }
        }
        config false;
        description
          "The status of this entry.";
      }
      uses dot1qtypes:port-map-grouping;
    }
  }
  container bridge-vlan {
    when "../../bridge-type != 'two-port-mac-relay-bridge'" {
      description
        "Applies to non TPMRs.";
    }
    description
      "The Bridge VLAN container models configuration
      information that modify, or inquire about, the overall
      configuration of the Bridges VLAN resources. There is a
      single Bridge VLAN Configuration managed object per
      Bridge.";
    reference
      "12.10 of IEEE Std 802.1Q-2018";
    leaf version {
      type uint16;
      config false;
      description
        "The version number supported.";
      reference
        "12.10.1.3 of IEEE Std 802.1Q-2018";
    }
    leaf max-vids {
      type uint16;
      config false;
      description
        "The maximum number of VIDs supported.";
      reference
        "12.10.1.3 of IEEE Std 802.1Q-2018";
    }
    leaf override-default-pvid {
      type boolean;
      default "false";
      config false;
      description
        "Indicates if the default PVID can be overridden, and
        its egress status (VLAN-tagged or untagged) on each
        port.";
      reference
        "12.10.1.3 of IEEE Std 802.1Q-2018";
```

```
}
leaf protocol-template {
  if-feature "port-and-protocol-based-vlan";
  type dot1qtypes:protocol-frame-format-type;
  config false;
  description
    "The data-link encapsulation format or the
    detagged_frame_type in a Protocol Template";
  reference
    "12.10.1.7 of IEEE Std 802.1Q-2018";
}
leaf max-msti {
  type uint16;
  config false;
  description
    "The maximum number of MSTIs supported within an MST
    region (i.e., the number of spanning tree instances that
    can be supported in addition to the CIST), for MST
    Bridges. For SST Bridges, this parameter may be either
    omitted or reported as 0.";
  reference
    "12.10.1.7 of IEEE Std 802.1Q-2018";
}
list vlan {
  key "vid";
  description
    "List of VLAN related configuration nodes associated
    with the Bridge.";
  reference
    "12.10.2 of IEEE Std 802.1Q-2018";
  leaf vid {
    type dot1qtypes:vlan-index-type;
    description
      "The VLAN identifier to which this entry applies.";
    reference
      "12.10.2 of IEEE Std 802.1Q-2018";
  }
  leaf name {
    type dot1qtypes:name-type;
    description
      "A text string of up to 32 characters of locally
      determined significance.";
    reference
      "12.10.2 of IEEE Std 802.1Q-2018";
  }
  leaf-list untagged-ports {
    type if:interface-ref;
    config false;
    description
      "The set of ports in the untagged set for this VID.";
    reference
      "12.10.2.1.3 of IEEE Std 802.1Q-2018
      8.8.2 of IEEE Std 802.1Q-2018";
  }
  leaf-list egress-ports {
    type if:interface-ref;
    config false;
    description
      "The set of egress ports in the member set for this
```

```
      VID.";
    reference
      "12.10.2.1.3 of IEEE Std 802.1Q-2018
      8.8.10 of IEEE Std 802.1Q-2018";
  }
}
list protocol-group-database {
  if-feature "port-and-protocol-based-vlan";
  key "db-index";
  description
    "List of the protocol group database entries.";
  reference
    "12.10.1.7 of IEEE Std 802.1Q-2018
    6.12.3 of IEEE Std 802.1Q-2018";
  leaf db-index {
    type uint16;
    description
      "The protocol group database index.";
  }
  leaf frame-format-type {
    type dot1qtypes:protocol-frame-format-type;
    description
      "The data-link encapsulation format or the
      detagged_frame_type in a Protocol Template";
    reference
      "12.10.1.7 of IEEE Std 802.1Q-2018";
  }
  choice frame-format {
    description
      "The identification of the protocol above the
      data-link layer in a Protocol Template. Depending on
      the frame type, the octet string will have one of the
      following values: - For ethernet, rfc1042 and
      snap8021H, this is the 16-bit (2-octet) IEEE 802
      Clause 9.3 EtherType field. - For snapOther, this is
      the 40-bit (5-octet) PID. - For llcOther, this is the
      2-octet IEEE 802.2 Link Service Access Point (LSAP)
      pair: first octet for Destination Service Access Point
      (DSAP) and second octet for Source Service Access
      Point (SSAP).";
    reference
      "12.10.1.7 of IEEE Std 802.1Q-2018";
    case ethernet-rfc1042-snap8021H {
      when
        "frame-format-type = 'Ethernet' or "+
        "frame-format-type = 'rfc1042' or "+
        "frame-format-type = 'snap8021H'" {
        description
          "Applies to Ethernet, RFC 1042, SNAP 8021H frame
          formats.";
      }
      description
        "Identifier used if Ethenet, RFC1042, or SNAP 8021H.";
      leaf ethertype {
        type dot1qtypes:ethertype-type;
        description
          "Format containing the 16-bit IEEE 802 EtherType
          field.";
        reference
```

```
          "9.3 of IEEE Std 802-2014";
      }
    }
    case snap-other {
      when "frame-format-type = 'snapOther'" {
        description
          "Applies to Snap Other frame formats.";
      }
      description
        "Identifier used if SNAP other.";
      leaf protocol-id {
        type string {
          pattern "[0-9a-fA-F]{2}(-[0-9a-fA-F]{2}){4}";
        }
        description
          "Format containing the 40-bit protocol identifier
          (PID). The canonical representation uses uppercase
          characters.";
        reference
          "12.10.1.7.1 of IEEE Std 802.1Q-2018";
      }
    }
    case llc-other {
      when "frame-format-type = 'llcOther'" {
        description
          "Applies to LLC Other frame formats";
      }
      description
        "Identifier used if LLC other.";
      container dsap-ssap-pairs {
        description
          "A pair of ISO/IEC 8802-2 DSAP and SSAP address
          field values, for matching frame formats of
          LLC_Other.";
        leaf llc-address {
          type string {
            pattern "[0-9a-fA-F]{2}-[0-9a-fA-F]{2}";
          }
          description
            "A pair of ISO/IEC 8802-2 DSAP and SSAP address
            field values, for matching frame formats of
            LLC_Other. The canonical representation uses
            uppercase characters.";
          reference
            "12.10.1.7.1 of IEEE Std 802.1Q-2018";
        }
      }
    }
  }
  leaf group-id {
    type uint32;
    description
      "Designates a group of protocols in the Protocol Group
      Database.";
    reference
      "6.12.2 of IEEE Std 802.1Q-2018";
  }
}
list vid-to-fid-allocation {
```

```
  key "vids";
  description
    "This list allows inquiries about VID to FID
    allocations.";
  leaf vids {
    type dot1qtypes:vid-range-type;
    description
      "Range of VLAN identifiers.";
    reference
      "12.10.3 of IEEE Std 802.1Q-2018";
  }
  leaf fid {
    type uint32;
    config false;
    description
      "The Filtering Database used by a set of VIDs.";
    reference
      "12.10.3 of IEEE Std 802.1Q-2018";
  }
  leaf allocation-type {
    type enumeration {
      enum undefined {
        description
          "No allocation defined.";
      }
      enum fixed {
        description
          "A fixed allocation to FID is defined.";
      }
      enum dynamic {
        description
          "A dynamic allocation to FID is defined.";
      }
    }
    config false;
    description
      "The type of allocation used";
    reference
      "12.10.3 of IEEE Std 802.1Q-2018";
  }
}
list fid-to-vid-allocation {
  key "fid";
  description
    "The FID to VID allocations managed object models
    operations that inquire about FID to VID allocations.";
  leaf fid {
    type uint32;
    description
      "The Filtering Database used by a set of VIDs.";
    reference
      "12.10.3 of IEEE Std 802.1Q-2018";
  }
  leaf allocation-type {
    type enumeration {
      enum undefined {
        description
          "No allocation defined.";
      }
```

```
          enum fixed {
            description
              "A fixed allocation to FID is defined.";
          }
          enum dynamic {
            description
              "A dynamic allocation to FID is defined.";
          }
        }
        config false;
        description
          "The type of allocation used";
        reference
          "12.10.3 of IEEE Std 802.1Q-2018";
      }
      leaf-list vid {
        type dot1qtypes:vlan-index-type;
        config false;
        description
          "The VLAN identifier to which this entry applies.";
        reference
          "12.7.7 of IEEE Std 802.1Q-2018";
      }
    }
    list vid-to-fid {
      key "vid";
      description
        "Fixed allocation of a VID to an FID. The underlying
        system will ensure that subsequent commands that make
        changes to the VID to FID mapping can override previous
        associations.";
      reference
        "12.10.3.4 of IEEE Std 802.1Q-2018
        12.10.3.5 of IEEE Std 802.1Q-2018";
      leaf vid {
        type dot1qtypes:vlan-index-type;
        description
          "A list of VLAN identifier associated with a given
          database identifier (i.e., FID).";
        reference
          "12.7.7 of IEEE Std 802.1Q-2018";
      }
      leaf fid {
        type uint32;
        description
          "The Filtering Database used by this VLAN";
        reference
          "12.10.3 of IEEE Std 802.1Q-2018";
      }
    }
  }
  container bridge-mst {
    when "../../bridge-type != 'two-port-mac-relay-bridge'" {
      description
        "Applies to non TPMRs.";
    }
    description
      "The Bridge MST container models configuration information
      that modify, or inquire about, the overall configuration
```

```
          of the Bridges MST resources.";
        reference
          "12.12 of IEEE Std 802.1Q-2018";
        leaf-list mstid {
          type dot1qtypes:mstid-type;
          description
            "The list of MSTID values that are currently supported
            by the Bridge";
        }
        list fid-to-mstid {
          key "fid";
          description
            "The FID to MSTID allocation table.";
          reference
            "12.12.2 of IEEE Std 802.1Q-2018";
          leaf fid {
            type uint32;
            description
              "The Filtering Database identifier.";
            reference
              "12.12.2 of IEEE Std 802.1Q-2018";
          }
          leaf mstid {
            type dot1qtypes:mstid-type;
            description
              "The MSTID to which the FID is to be allocated.";
            reference
              "12.12.2 of IEEE Std 802.1Q-2018";
          }
        }
        list fid-to-mstid-allocation {
          key "fids";
          description
            "The FID to MSTID allocation table";
          leaf fids {
            type dot1qtypes:vid-range-type;
            description
              "Range of FIDs.";
            reference
              "12.12.2 of IEEE Std 802.1Q-2018";
          }
          leaf mstid {
            type dot1qtypes:mstid-type;
            description
              "The MSTID to which the FID is allocated.";
            reference
              "12.12.2 of IEEE Std 802.1Q-2018";
          }
        }
      }
    }
  }
}
augment "/if:interfaces/if:interface" {
  when
    "if:type = 'ianaif:bridge' or if:type ="+
    "'ianaif:ethernetCsmacd' or if:type = 'ianaif:ieee8023adLag'"+
    "or if:type = 'ianaif:ilan'" {
    description
```

```
      "Applies when a Bridge interface.";
  }
  description
    "Augment the interface model with the Bridge Port";
  container bridge-port {
    description
      "Bridge Port is an extension of the IETF Interfaces model
      (RFC7223).";
    leaf component-name {
      type string;
      description
        "Used to reference configured Component node.";
    }
    leaf port-type {
      type identityref {
        base type-of-port;
      }
      description
        "The port type. Indicates the capabilities of this port.";
      reference
        "12.4.2.1 of IEEE Std 802.1Q-2018";
    }
    leaf pvid {
      when "../component-name != 'd-bridge-component'" {
        description
          "Applies to non TPMRs";
      }
      type dot1qtypes:vlan-index-type;
      default "1";
      description
        "The primary (default) VID assigned to a specific Bridge
        Port.";
      reference
        "12.10.1 of IEEE Std 802.1Q-2018
        5.4, item m) of IEEE Std 802.1Q-2018";
    }
    leaf default-priority {
      type dot1qtypes:priority-type;
      default "0";
      description
        "The default priority assigned to a specific Bridge Port.";
      reference
        "12.6.2 of IEEE Std 802.1Q-2018";
    }
    container priority-regeneration {
      description
        "The Priority Regeneration Table parameters associated with
        a specific Bridge Port. A list of Regenerated User
        Priorities for each received priority on each port of a
        Bridge. The regenerated priority value may be used to index
        the Traffic Class Table for each input port. This only has
        effect on media that support native priority. The default
        values for Regenerated User Priorities are the same as the
        User Priorities";
      reference
        "12.6.2 of IEEE Std 802.1Q-2018
        6.9.4 of IEEE Std 802.1Q-2018";
      uses dot1qtypes:priority-regeneration-table-grouping;
    }
```

```
leaf pcp-selection {
  type dot1qtypes:pcp-selection-type;
  default "8P0D";
  description
    "The Priority Code Point selection assigned to a specific
    Bridge Port. This object identifies the rows in the PCP
    encoding and decoding tables that are used to remark frames
    on this port if this remarking is enabled";
  reference
    "12.6.2 of IEEE Std 802.1Q-2018
    6.9.3 of IEEE Std 802.1Q-2018";
}
container pcp-decoding-table {
  description
    "The Priority Code Point Decoding Table parameters
    associated with a specific Bridge Port.";
  uses dot1qtypes:pcp-decoding-table-grouping;
}
container pcp-encoding-table {
  description
    "The Priority Code Point Encoding Table parameters
    associated with a specific Bridge Port.";
  uses dot1qtypes:pcp-encoding-table-grouping;
}
leaf use-dei {
  type boolean;
  default "false";
  description
    "The Drop Eligible Indicator. If it is set to True, then the
    drop_eligible parameter is encoded in the DEI of transmitted
    frames, and the drop_eligible parameter shall be true(1) for
    a received frame if the DEI is set in the VLAN tag or the
    Priority Code Point Decoding Table indicates drop_eligible
    True for the received PCP value. If this parameter is False,
    the DEI shall be transmitted as zero and ignored on receipt.";
  reference
    "12.6.2 of IEEE Std 802.1Q-2018
    6.9.3 of IEEE Std 802.1Q-2018";
}
leaf drop-encoding {
  type boolean;
  default "false";
  description
    "The Drop Encoding parameter. If a Bridge supports encoding
    or decoding of drop_eligible from the PCP field of a VLAN
    tag (6.7.3) on any of its Ports, then it shall implement a
    Boolean parameter Require Drop Encoding on each of its Ports
    with default value False. If Require Drop Encoding is True
    and the Bridge Port cannot encode particular priorities with
    drop_eligible, then frames queued with those priorities and
    drop_eligible True shall be discarded and not transmitted.";
  reference
    "12.6.2 of IEEE Std 802.1Q-2018
    8.6.6 of IEEE Std 802.1Q-2018";
}
leaf service-access-priority-selection {
  type boolean;
  default "false";
  description
```

```
    "The Service Access Priority selection. Indication of
    whether the Service Access Priority Selection function is
    supported on the Customer Bridge Port to request priority
    handling of the frame from a Port-based service interface.";
  reference
    "12.6.2 of IEEE Std 802.1Q-2018
    6.13 of IEEE Std 802.1Q-2018";
}
container service-access-priority {
  description
    "The Service Access Priority table parameters. A table that
    contains information about the Service Access Priority
    Selection function for a Provider Bridge. The use of this
    table enables a mechanism for a Customer Bridge attached to
    a Provider Bridged Network to request priority handling of
    frames.";
  reference
    "12.6.2 of IEEE Std 802.1Q-2018
    6.13.1 of IEEE Std 802.1Q-2018";
  uses dot1qtypes:service-access-priority-table-grouping;
}
container traffic-class {
  description
    "The Traffic Class table parameters. A table mapping
    evaluated priority to Traffic Class, for forwarding by the
    Bridge";
  reference
    "12.6.3 of IEEE Std 802.1Q-2018
    8.6.6 of IEEE Std 802.1Q-2018";
  uses dot1qtypes:traffic-class-table-grouping;
}
leaf acceptable-frame {
  when "../component-name != 'd-bridge-component'" {
    description
      "Applies to non TPMRs";
  }
  type enumeration {
    enum admit-only-VLAN-tagged-frames {
      description
        "Admit only VLAN-tagged frames.";
    }
    enum admit-only-untagged-and-priority-tagged {
      description
        "Admit only untagged and priority-tagged frames.";
    }
    enum admit-all-frames {
      description
        "Admit all frames.";
    }
  }
  default "admit-all-frames";
  description
    "To configure the Acceptable Frame Types parameter
    associated with one or more Ports";
  reference
    "12.10.1.3 of IEEE Std 802.1Q-2018
    6.9 of IEEE Std 802.1Q-2018";
}
leaf enable-ingress-filtering {
```

```
    when "../component-name != 'd-bridge-component'" {
      description
        "Applies to non TPMRs";
    }
    type boolean;
    default "false";
    description
      "To enable the Ingress Filtering feature associated with one
      or more Ports.";
    reference
      "12.10.1.4 of IEEE Std 802.1Q-2018
      8.6.2 of IEEE Std 802.1Q-2018";
  }
  leaf enable-restricted-vlan-registration {
    when "../component-name != 'd-bridge-component'" {
      description
        "Applies to non TPMRs";
    }
    type boolean;
    default "false";
    description
      "To enable the Restricted VLAN Registration associated with
      one or more Ports.";
    reference
      "11.2.3.2.3 of IEEE Std 802.1Q-2018
      12.10.1.6 of IEEE Std 802.1Q-2018";
  }
  leaf enable-vid-translation-table {
    when "../component-name != 'd-bridge-component'" {
      description
        "Applies to non TPMRs";
    }
    type boolean;
    default "false";
    description
      "To enable VID Translation table associated with a Bridge
      Port. This is not applicable to Bridge Ports that do no
      support a VID Translation Table.";
    reference
      "12.10.1.8 of IEEE Std 802.1Q-2018
      6.9 of IEEE Std 802.1Q-2018";
  }
  leaf enable-egress-vid-translation-table {
    when "../component-name != 'd-bridge-component'" {
      description
        "Applies to non TPMRs";
    }
    type boolean;
    default "false";
    description
      "To enable Egress VID Translation table associated with a
      Bridge Port. This is not applicable to Ports that do not
      support an Egress VID Translation table.";
    reference
      "12.10.1.9 of IEEE Std 802.1Q-2018
      6.9 of IEEE Std 802.1Q-2018";
  }
  list protocol-group-vid-set {
    when "../component-name != 'd-bridge-component'" {
```

```
        description
          "Applies to non TPMRs";
      }
      if-feature "port-and-protocol-based-vlan";
      key "group-id";
      description
        "The list of VID values associated with the Protocol Group
        Identifier for this port.";
      reference
        "12.10.1.1.3 of IEEE Std 802.1Q-2018";
      leaf group-id {
        type uint32;
        description
          "The protocol group identifier";
        reference
          "12.10.1.7 of IEEE Std 802.1Q-2018";
      }
      leaf-list vid {
        type dot1qtypes:vlanid;
        description
          "The VLAN identifier to which this entry applies.";
        reference
          "12.10.2 of IEEE Std 802.1Q-2018";
      }
    }
    leaf admin-point-to-point {
      type enumeration {
        enum force-true {
          value 1;
          description
            "Indicates that this port should always be treated as if
            it is connected to a point-to-point link.";
        }
        enum force-false {
          value 2;
          description
            "Indicates that this port should be treated as having a
            shared media connection.";
        }
        enum auto {
          value 3;
          description
            "Indicates that this port is considered to have a
            point-to-point link if it is an Aggregator and all of
            its members are aggregatable, or if the MAC entity is
            configured for full duplex operation, either through
            auto-negotiation or by management means.";
        }
      }
      description
        "For a port running spanning tree, this object represents
        the administrative point-to-point status of the LAN segment
        attached to this port, using the enumeration values of IEEE
        Std 802.1AC. A value of forceTrue(1) indicates that this
        port should always be treated as if it is connected to a
        point-to-point link. A value of forceFalse(2) indicates that
        this port should be treated as having a shared media
        connection. A value of auto(3) indicates that this port is
        considered to have a point-to-point link if it is an
```

```
    Aggregator and all of its members are aggregatable, or if
    the MAC entity is configured for full duplex operation,
    either through auto-negotiation or by management means.
    Manipulating this object changes the underlying
    adminPointToPointMAC.";
  reference
    "12.4.2 of IEEE Std 802.1Q-2018
    6.8.2 of IEEE Std 802.1Q-2018";
}
leaf protocol-based-vlan-classification {
  when "../component-name != 'd-bridge-component'" {
    description
      "Applies to non TPMRs";
  }
  if-feature "port-and-protocol-based-vlan";
  type boolean;
  config false;
  description
    "A boolean indication indicating if Port-and-Protocol-based
    VLAN classification is supported on a given Port.";
  reference
    "5.4.1.2 of IEEE Std 802.1Q-2018";
}
leaf max-vid-set-entries {
  when "../component-name != 'd-bridge-component'" {
    description
      "Applies to non TPMRs";
  }
  if-feature "port-and-protocol-based-vlan";
  type uint16;
  config false;
  description
    "The maximum number of entries supported in the VID set on a
    given Port.";
  reference
    "12.10.1.1.3 of IEEE Std 802.1Q-2018";
}
leaf port-number {
  type dot1qtypes:port-number-type;
  config false;
  description
    "An integer that uniquely identifies a Bridge Port.";
  reference
    "12.3, item i) of IEEE Std 802.1Q-2018
    17.3.2.2 of IEEE Std 802.1Q-2018";
}
leaf address {
  type ieee:mac-address;
  config false;
  description
    "The specific MAC address of the individual MAC Entity
    associated with the Port.";
  reference
    "12.4.2 of IEEE Std 802.1Q-2018
    12.4.2.1.1.3, item a) of IEEE Std 802.1Q-2018";
}
leaf capabilities {
  type bits {
    bit tagging {
```

```
        position "0";
        description
          "Supports 802.1Q VLAN tagging of frames and MVRP.";
      }
      bit configurable-acceptable-frame-type {
        position "1";
        description
          "Allows modified values of acceptable frame types";
      }
      bit ingress-filtering {
        position "2";
        description
          "Supports the discarding of any frame received on a Port
          whose VLAN classification does not include that Port in
          its member set.";
      }
    }
    config false;
    description
      "The feature capabilities associated with port. Indicates
      the parts of IEEE 802.1Q that are optional on a per-port
      basis, that are implemented by this device, and that are
      manageable.";
    reference
      "12.10.1.1.3, item c) of IEEE Std 802.1Q-2018
      12.4.2 of IEEE Std 802.1Q-2018";
  }
  leaf type-capabilties {
    type bits {
      bit customer-vlan-port {
        position "0";
        description
          "Indicates the port can be a C-TAG aware port of an
          enterprise VLAN aware Bridge";
      }
      bit provider-network-port {
        position "1";
        description
          "Indicates the port can be an S-TAG aware port of a
          Provider Bridge or Backbone Edge Bridge used for
          connections within a PBN or PBBN.";
      }
      bit customer-network-port {
        position "2";
        description
          "Indicates the port can be an S-TAG aware port of a
          Provider Bridge or Backbone Edge Bridge used for
          connections to the exterior of a PBN or PBBN.";
      }
      bit customer-edge-port {
        position "3";
        description
          "Indicates the port can be a C-TAG aware port of a
          Provider Bridge used for connections to the exterior of
          a PBN or PBBN.";
      }
      bit customer-backbone-port {
        position "4";
        description
```

```
          "Indicates the port can be a I-TAG aware port of a
          Backbone Edge Bridge's B-component.";
      }
      bit virtual-instance-port {
        position "5";
        description
          "Indicates the port can be a virtual S-TAG aware port
          within a Backbone Edge Bridge's I-component which is
          responsible for handling S-tagged traffic for a specific
          backbone service instance.";
      }
      bit d-bridge-port {
        position "6";
        description
          "Indicates the port can be a VLAN-unaware member of an
          802.1Q Bridge.";
      }
      bit remote-customer-access-port {
        position "7";
        description
          "Indicates the port can be an S-TAG aware port of a
          Provider Bridge capable of providing Remote Customer
          Service Interfaces.";
      }
      bit station-facing-bridge-port {
        position "8";
        description
          "Indicates the station-facing Bridge Port in a EVB
          Bridge.";
      }
      bit uplink-access-port {
        position "9";
        description
          "Indicates the uplink access port in an EVB Bridge or
          EVB station.";
      }
      bit uplink-relay-port {
        position "10";
        description
          "Indicates the uplink relay port in an EVB station.";
      }
    }
    config false;
    description
      "The type of feature capabilities supported with port.
      Indicates the capabilities of this port.";
    reference
      "12.4.2 of IEEE Std 802.1Q-2018";
  }
  leaf external {
    type boolean;
    config false;
    description
      "A boolean indicating whether the port is external. A value
      of True means the port is external. A value of False means
      the port is internal.";
    reference
      "12.4.2 of IEEE Std 802.1Q-2018";
  }
```

```
      leaf oper-point-to-point {
        type boolean;
        config false;
        description
          "For a port running spanning tree, this object represents
          the operational point-to-point status of the LAN segment
          attached to this port. It indicates whether a port is
          considered to have a point-to-point connection.

          If admin-point-to-point is set to auto(2), then the value of
          oper-point-to-point is determined in accordance with the
          specific procedures defined for the MAC entity concerned, as
          defined in IEEE Std 802.1AC.

          The value is determined dynamically; that is, it is
          re-evaluated whenever the value of admin-point-to-point
          changes, and whenever the specific procedures defined for
          the MAC entity evaluate a change in its point-to-point
          status.";
        reference
          "IEEE Std 802.1AC
          12.4.2 of IEEE Std 802.1Q-2018";
      }
      leaf media-dependent-overhead {
        type uint8;
        units "octets";
        config false;
        description
          "The portMediaDependentOverhead parameter provides the
          number of additional octets for media-dependent framing. The
          overhead includes all octets prior the first octet of the
          Destination Address field and all octets after the last octet
          of the frame check sequence.";
        reference
          "12.4.2 of IEEE Std 802.1Qcr-2020";
      }
      container statistics {
        config false;
        description
          "Container of operational state node information associated
          with the bridge port.";
        uses dot1qtypes:bridge-port-statistics-grouping;
        leaf discard-on-ingress-filtering {
          when "../../component-name != 'd-bridge-component'" {
            description
              "Applies to non TPMRs";
          }
          if-feature "ingress-filtering";
          type yang:counter64;
          description
            "The number of frames that were discarded as a result of
            Ingress Filtering being enabled.

            Discontinuities in the value of this counter can occur at
            re-initialization of the management system, and at other
            times as indicated by the value of 'discontinuity-time'.";
          reference
            "12.6.1.1.3 of IEEE Std 802.1Q-2018";
        }
```

```
}
list vid-translations {
  when "../component-name != 'd-bridge-component'" {
    description
      "Applies to non TPMRs";
  }
  key "local-vid";
  description
    "To configure the VID Translation Table (6.9) associated
    with a Port. This object is not applicable to Ports that do
    not support a VID Translation Table. The default
    configuration of the table has the value of the Relay VID
    equal to the value of the Local VID. If no local VID is
    configured, then it is assumed that the relay VID is the
    same value as the local VID.

    If the port supports an Egress VID translation table, the
    VID Translation Configuration object configures the Local
    VID to Relay VID mapping on ingress only. If an Egress VID
    translation is not supported, the VID Translation
    Configuration object defines a single bidirectional mapping.
    In this case, the Bridge should not allow multiple keys
    ('local-vid') mapped to the same 'relay-vid' value.";
  leaf local-vid {
    type dot1qtypes:vlanid;
    description
      "The Local VID after translation received at the ISS or
      EISS.";
    reference
      "12.10.1.8 of IEEE Std 802.1Q-2018
      6.9 of IEEE Std 802.1Q-2018";
  }
  leaf relay-vid {
    type dot1qtypes:vlanid;
    description
      "The Relay VID received before translation received at ISS
      or EISS.";
    reference
      "12.10.1.8 of IEEE Std 802.1Q-2018
      6.9 of IEEE Std 802.1Q-2018";
  }
}
list egress-vid-translations {
  when "../component-name != 'd-bridge-component'" {
    description
      "Applies to non TPMRs";
  }
  key "relay-vid";
  description
    "To configure the Egress VID Translation Table (6.9)
    associated with a Port. This object is not applicable to
    Ports that do not support an Egress VID Translation Table.
    The default configuration of the table has the value of the
    Local VID equal to the value of the Relay VID. If no Relay
    VID is configured, then it is assumed that the local VID is
    the same value as the relay VID.";
  leaf relay-vid {
    type dot1qtypes:vlanid;
    description
```

```
            "The Relay VID received before translation received at ISS
            or EISS.";
          reference
            "12.10.1.9 of IEEE Std 802.1Q-2018
            6.9 of IEEE Std 802.1Q-2018";
        }
        leaf local-vid {
          type dot1qtypes:vlanid;
          description
            "The Local VID after translation received at the ISS or
            EISS.";
          reference
            "12.10.1.9 of IEEE Std 802.1Q-2018
            6.9 of IEEE Std 802.1Q-2018";
        }
      }
    }
  }
}
```

Insert the following subclause (48.6.12) after 48.6.9 (note that subclause numbers 48.6.10 and 48.6.11 are reserved for a future amendment):

48.6.12 Definition for the ieee802-dot1q-stream-filters-gates YANG module

```
module ieee802-dot1q-stream-filters-gates {
  yang-version "1.1";
  namespace urn:ieee:std:802.1Q:yang:ieee802-dot1q-stream-filters-gates;
  prefix sfsg;
  import ieee802-dot1q-bridge {
    prefix dot1q;
  }
  organization
    "IEEE 802.1 Working Group";
  contact
    "WG-URL: http://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
             Piscataway, NJ 08854
             USA

     E-mail: stds-802-1-chairs@ieee.org";
  description
    "This module provides management of 802.1Q bridge components that support
    Stream Filters and Stream Gates.";
  revision 2020-11-06 {
    description
      "Published as part of IEEE Std 802.1Qcr-2020.
      Initial version.";
    reference
      "IEEE Std 802.1Qcr-2020, Bridges and Bridged Networks -
      Asynchronous Traffic Shaping.";
```

```
}

feature closed-gate-state {
  description
    "The bridge component supports gate state closed.";
  reference
    "IEEE Std 802.1Qcr-2020";
}

/* Types and groupings */
typedef priority-spec-type {
  type enumeration {
    enum zero {
      value 0;
      description
        "Priority 0";
    }
    enum one {
      value 1;
      description
        "Priority 1";
    }
    enum two {
      value 2;
      description
        "Priority 2";
    }
    enum three {
      value 3;
      description
        "Priority 3";
    }
    enum four {
      value 4;
      description
        "Priority 4";
    }
    enum five {
      value 5;
      description
        "Priority 5";
    }
    enum six {
      value 6;
      description
        "Priority 6";
    }
    enum seven {
      value 7;
      description
        "Priority 7";
    }
    enum wildcard {
      description
        "wildcard value";
    }
  }
}
typedef ipv-spec-type {
```

```
  type enumeration {
    enum zero {
      value 0;
      description
        "Priority 0";
    }
    enum one {
      value 1;
      description
        "Priority 1";
    }
    enum two {
      value 2;
      description
        "Priority 2";
    }
    enum three {
      value 3;
      description
        "Priority 3";
    }
    enum four {
      value 4;
      description
        "Priority 4";
    }
    enum five {
      value 5;
      description
        "Priority 5";
    }
    enum six {
      value 6;
      description
        "Priority 6";
    }
    enum seven {
      value 7;
      description
        "Priority 7";
    }
    enum null {
      description
        "null value";
    }
  }
  description
    "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.";
  reference
    "8.6.5.2 of IEEE Std 802.1Qcr-2020";
}
```

```
typedef gate-state-value-type {
  type enumeration {
    enum closed {
      description
        "Gate closed";
    }
    enum open {
      description
        "Gate open";
    }
  }
  description
    "The gate-state-value-type indicates a gate state, open or closed,
    for the stream gate.";
  reference
    "12.31.3.2.1 of IEEE Std 802.1Qcr-2020";
}
typedef stream-gate-ref {
  type leafref {
    path
      '/dot1q:bridges'+
      '/dot1q:bridge'+
      '/dot1q:component'+
      '/sfsg:stream-gates'+
      '/sfsg:stream-gate-instance-table'+
      '/sfsg:stream-gate-instance-id';
  }
  description
    "This type is used to refer to a stream gate instance.";
}
augment "/dot1q:bridges/dot1q:bridge/dot1q:component" {
  description
    "Augments the Bridge component with stream filters and stream gates.";
  container stream-filters {
    description
      "This container encapsulates all nodes related to stream bilters.";
    reference
      "12.31.1 of IEEE Std 802.1Qcr-2020
      12.31.2 of IEEE Std 802.1Qcr-2020
      12.31.3 of IEEE Std 802.1Qcr-2020";
    list stream-filter-instance-table {
      key "stream-filter-instance-id";
      description
        "Each list entry contains a set of parameters that defines a
        single stream filter (8.6.5.1) with associated maximum SDU size
        filtering (8.6.5.3.1), as detailed in Table 12-32. Entries can be
        created or removed dynamically in implementations that support
        dynamic configuration of stream filters. The value of the
        stream-handle-spec and priority-spec 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-spec and
        priority-spec 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-spec and priority-spec does not
        match any of the stream filters in the list, the frame is
        processed as if stream filters and stream gates would not be
        supported.";
```

```
reference
  "12.31.2 of IEEE Std 802.1Qcr-2020";
leaf stream-filter-instance-id {
  type uint32;
  mandatory true;
  description
    "An integer index value that determines the place of the stream
    filter in the ordered list of stream filter instances. The
    values are ordered according to their integer value; smaller
    values appear earlier in the ordered list.";
  reference
    "12.31.2.1 of IEEE Std 802.1Qcr-2020";
}
choice stream-handle-spec {
  description
    "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 wildcard value, which matches any frame";
  reference
    "12.31.2.2 of IEEE Std 802.1Qcr-2020";

  /* NOTE: The mapping of the wildcard literal is
   *       other than in the MIB definition, where
   *       the wildcard value is mapped to -1.
   */
  case wildcard {
    leaf wildcard {
      type empty;
      description
        "The stream handle specification represents a wildcard value.";
    }
  }
  case stream-handle {
    leaf stream-handle {
      type uint32;
      mandatory true;
      description
        "The stream handle specification refers to a stream_handle
        value.";
    }
  }
}
leaf priority-spec {
  type priority-spec-type;
  mandatory true;
  description
    "The priority specification data type allows either of the
    following to be represented:
     a) A priority value, represented as an integer.
     b) The wildcard value, which matches any priority.";
  reference
    "12.31.2.3 of IEEE Std 802.1Qcr-2020";
}
leaf max-sdu-size {
  type uint32;
  units "octets";
  mandatory true;
  description
```

```
          "The allowed maximum SDU size, in octets. If set to 0, any SDU
          size is accepted.";
        reference
          "8.6.5.3.1 of IEEE Std 802.1Qcr-2020";
      }
      leaf stream-blocked-due-to-oversize-frame-enabled {
        type boolean;
        default "false";
        description
          "A value of true indicates that
          stream-blocked-due-to-oversize-frame is set to true as soon as
          a frame exceeds max-sdu-size.";
        reference
          "8.6.5.3.1 of IEEE Std 802.1Qcr-2020";
      }
      leaf stream-blocked-due-to-oversize-frame {
        type boolean;
        default "false";
        description
          "Indicates by value true that frames are permanently discarded
          as a result of an initial frame exceeding max-sdu-size. The
          value of stream-blocked-due-to-oversize-frame can be
          administratively reset to false.";
        reference
          "8.6.5.3.1 of IEEE Std 802.1Qcr-2020";
      }
      leaf stream-gate-ref {
        type stream-gate-ref;
        mandatory true;
        description
          "This node refers to 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.";
        reference
          "12.31.2.4 of IEEE Std 802.1Qcr-2020";
      }
    }
    leaf max-stream-filter-instances {
      type uint32;
      config false;
      description
        "The maximum number of stream filter instances supported by this
        Bridge component.";
      reference
        "12.31.1.1 of IEEE Std 802.1Qcr-2020
        8.6.5.1 of IEEE Std 802.1Qcr-2020";
    }
  }
  container stream-gates {
    description
      "This container encapsulates all nodes related to Stream Gates.";
    list stream-gate-instance-table {
      key "stream-gate-instance-id";
      description
        "Each list entry contains a set of parameters that defines a
        single stream gate (8.6.5.2), as detailed in Table 12-33. Entries
```

```
          in the table can be created or removed dynamically in
          implementations that support dynamic configuration of stream
          gates.";
        reference
          "12.31.3 of IEEE Std 802.1Qcr-2020";
        leaf stream-gate-instance-id {
          type uint32;
          description
            "An integer table index that allows the stream gate to be
            referenced from Stream Filter Instance Table entries.";
          reference
            "12.31.2.4 of IEEE Std 802.1Qcr-2020
            8.6.5.3 of IEEE Std 802.1Qcr-2020
            8.6.5.4 of IEEE Std 802.1Qcr-2020";
        }
        leaf gate-enable {
          type boolean;
          default "false";
          description
            "A Boolean variable that indicates whether the operation of the
            state machines is enabled (TRUE) or disabled (FALSE). This
            variable is set by management. The default value of this
            variable is FALSE.";
          reference
            "8.6.9.4.14 of IEEE Std 802.1Q-2018";
        }
        leaf admin-gate-states {
          type gate-state-value-type;
          default "open";
          description
            "The administratively set gate state of this gate.";
          reference
            "12.31.3.2.1 of IEEE Std 802.1Qcr-2020
            8.6.10.4 of IEEE Std 802.1Qcr-2020";
        }
        leaf admin-ipv {
          type ipv-spec-type;
          default "null";
          description
            "The administratively set internal priority value
            specification.";
          reference
            "12.31.3.3 of IEEE Std 802.1Qcr-2020
            8.6.10.6 of IEEE Std 802.1Qcr-2020
            8.6.5.4 of IEEE Std 802.1Qcr-2020";
        }
      }
      leaf max-stream-gate-instances {
        type uint32;
        config false;
        description
          "The maximum number of Stream Gate instances supported by this
          Bridge component.";
        reference
          "12.31.1.2 of IEEE Std 802.1Qcr-2020";
      }
    }
  }
}
```

Insert the following subclause (48.6.14) after 48.6.12 (note that subclause number 48.6.13 is reserved for a future amendment):

48.6.14 Definition for the ieee802-dot1q-ats YANG module

```
module ieee802-dot1q-ats {
  yang-version "1.1";
  namespace urn:ieee:std:802.1Q:yang:ieee802-dot1q-ats;
  prefix ats;
  import ietf-yang-types {
    prefix yang;
  }
  import ietf-interfaces {
    prefix if;
  }
  import ieee802-dot1q-types {
    prefix dot1qtypes;
  }
  import ieee802-dot1q-bridge {
    prefix dot1q;
  }
  import ieee802-dot1q-stream-filters-gates {
    prefix sfsg;
  }
  organization
    "IEEE 802.1 Working Group";
  contact
    "WG-URL: http://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
             Piscataway, NJ 08854
             USA

     E-mail: stds-802-1-chairs@ieee.org";
  description
    "This module provides management of 802.1Q bridge components that support
    Asynchronous Traffic Shaping (ATS).";
  revision 2020-11-06 {
    description
      "Published as part of IEEE Std 802.1Qcr-2020.
      Initial version.";
    reference
      "IEEE Std 802.1Qcr-2020, Bridges and Bridged Networks -
      Asynchronous Traffic Shaping.";
  }
  typedef scheduler-ref-type {
    type leafref {
      path
        '/dot1q:bridges'+
        '/dot1q:bridge'+
        '/dot1q:component'+
        '/ats:schedulers'+
        '/ats:scheduler-instance-table'+
        '/ats:scheduler-instance-id';
```

```
    }
    description
      "This type is used to refer to an ATS scheduler instance.";
  }
  typedef scheduler-group-ref-type {
    type leafref {
      path
        '/dot1q:bridges'+
        '/dot1q:bridge'+
        '/dot1q:component'+
        '/ats:scheduler-groups'+
        '/ats:scheduler-group-instance-table'+
        '/ats:scheduler-group-instance-id';
    }
    description
      "This type is used to refer to an ATS scheduler group instance.";
  }
  augment
    "/dot1q:bridges"+
    "/dot1q:bridge"+
    "/dot1q:component"+
    "/sfsg:stream-filters"+
    "/sfsg:stream-filter-instance-table" {
    description
      "Augments the Bridge component stream filter for ATS schedulers.";
    container scheduler {
      description
        "Enapsulates ATS scheduler nodes.";
      leaf scheduler-ref {
        type ats:scheduler-ref-type;
        description
          "A reference to the ATS scheduler associated with this stream
          filter.";
      }
      leaf scheduler-enable {
        type boolean;
        default "false";
        description
          "If TRUE, this stream filter has an associated ATS scheduler
          referenced by scheduler-ref. If FALSE, no ATS scheduler is
          associated with this stream filter (scheduler-ref is ignored).";
      }
    }
  }
  augment "/if:interfaces/if:interface/dot1q:bridge-port" {
    description
      "Augments Bridge Ports by ATS per-Port parameters.";
    container ats-port-parameters {
      description
        "This container comprises all ATS per-Port parameters.";
      leaf discarded-frames-count {
        type yang:counter64;
        config false;
        description
          "A counter of frames discarded by ATS scheduler instances
          associated with the Bridge Port.";
        reference
          "12.31.7.3 of IEEE Std 802.1Qcr-2020";
      }
```

```
    }
  }
  augment "/dot1q:bridges/dot1q:bridge/dot1q:component" {
    description
      "Augments the Bridge component by
       a) ATS schedulers
       b) ATS scheduler groups";
    container schedulers {
      description
        "This container comprises all nodes related to an ATS schedulers.";
      list scheduler-instance-table {
        key "scheduler-instance-id";
        description
          "Each list entry comprises a set of parameters that defines a
          single ATS scheduler instance, as detailed in Table 12-33.";
        reference
          "12.31.5 of IEEE Std 802.1Qcr-2020";
        leaf scheduler-instance-id {
          type uint32;
          mandatory true;
          description
            "A unique index identifying this ATS scheduler instance.";
          reference
            "12.31.5.1 of IEEE Std 802.1Qcr-2020
            8.6.5.6 of IEEE Std 802.1Qcr-2020";
        }
        leaf committed-information-rate {
          type uint64;
          units "bits/second";
          mandatory true;
          description
            "The committed information rate parameter of this ATS scheduler
            instance.";
          reference
            "12.31.5.3 of IEEE Std 802.1Qcr-2020
            8.6.5.6 of IEEE Std 802.1Qcr-2020";
        }
        leaf committed-burst-size {
          type uint32;
          units "bits";
          mandatory true;
          description
            "The committed burst size parameter of this ATS scheduler
            instance.";
          reference
            "12.31.5.2 of IEEE Std 802.1Qcr-2020
            8.6.5.6 of IEEE Std 802.1Qcr-2020";
        }
        leaf scheduler-group-ref {
          type ats:scheduler-group-ref-type;
          mandatory true;
          description
            "A reference to the scheduler group (12.32.5) associated with
            this ATS scheduler instance. Multiple ATS scheduler instances
            can be associated to one scheduler group, as detailed in
            8.6.5.6.";
          reference
            "12.31.6 of IEEE Std 802.1Qcr-2020";
        }
```

```
    }
    leaf max-scheduler-instances {
      type uint32;
      config false;
      description
        "The maximum number of ATS scheduler instances supported by this
        Bridge component.";
      reference
        "12.31.1.5 of IEEE Std 802.1Qcr-2020";
    }
  }
  container scheduler-groups {
    description
      "This container comprises all ATS scheduler group related nodes.";
    list scheduler-group-instance-table {
      key "scheduler-group-instance-id";
      description
        "Each list entry comprises a set of parameters that defines a
        single ATS scheduler group instance.";
      reference
        "12.31.6 of IEEE Std 802.1Qcr-2020
        8.6.5.6 of IEEE Std 802.1Qcr-2020";
      leaf scheduler-group-instance-id {
        type uint32;
        description
          "A unique index identifying this ATS scheduler group instance.";
        reference
          "12.31.6.1 of IEEE Std 802.1Qcr-2020
          8.6.5.6 of IEEE Std 802.1Qcr-2020";
      }
      leaf max-residence-time {
        type uint32;
        units "nanoseconds";
        mandatory true;
        description
          "The maximum residence time parameter of the ATS scheduler
          group.";
        reference
          "8.6.11.2.13 of IEEE Std 802.1Qcr-2020
          8.6.5.6 of IEEE Std 802.1Qcr-2020";
      }
    }
    leaf max-scheduler-group-instances {
      type uint32;
      config false;
      description
        "The maximum number of ATS scheduler group instances supported by
        this Bridge component.";
      reference
        "12.31.1.6 of IEEE Std 802.1Qcr-2020
        8.6.5.6 of IEEE Std 802.1Qcr-2020";
    }
    container scheduler-timing-characteristics {
      description
        "This container comprises all ATS scheduler timing
        characteristics related nodes.";
      list scheduler-timing-characteristics-table {
        key "reception-port transmission-port";
        config false;
```

```
description
  "Each list entry comprises the timing characteristics of a
  reception Port transmission Port pair, as detailed in Table
  12-36.";
reference
  "12.31.8 of IEEE Std 802.1Qcr-2020
  8.6.11 of IEEE Std 802.1Qcr-2020";
leaf reception-port {
  type dot1qtypes:port-number-type;
  config false;
  mandatory true;
  description
    "A reference to the associated reception Port.";
  reference
    "12.31.8.1 of IEEE Std 802.1Qcr-2020";
}
leaf transmission-port {
  type dot1qtypes:port-number-type;
  config false;
  mandatory true;
  description
    "A reference to the associated transmission Port.";
  reference
    "12.31.8.2 of IEEE Std 802.1Qcr-2020";
}
leaf clock-offset-variation-max {
  type uint32;
  units "nanoseconds";
  config false;
  mandatory true;
  description
    "The maximum clock offset variation associated with the
    reception Port transmission Port pair.";
  reference
    "12.31.8.3 of IEEE Std 802.1Qcr-2020";
}
leaf clock-rate-deviation-max {
  type uint32;
  units "ppm";
  config false;
  mandatory true;
  description
    "The maximum clock rate deviation associated with the
    reception Port transmission Port pair.";
  reference
    "12.31.8.4 of IEEE Std 802.1Qcr-2020";
}
leaf arrival-recognition-delay-max {
  type uint32;
  units "nanoseconds";
  config false;
  mandatory true;
  description
    "The maximum arrival time recognition delay associated with
    the reception Port transmission Port pair.";
  reference
    "12.31.8.5 of IEEE Std 802.1Qcr-2020";
}
leaf processing-delay-min {
```

```
              type uint32;
              units "nanoseconds";
              config false;
              mandatory true;
              description
                "The minimum processing delay associated with the reception
                Port transmission Port pair.";
              reference
                "12.31.8.6 of IEEE Std 802.1Qcr-2020";
            }
            leaf processing-delay-max {
              type uint32;
              units "nanoseconds";
              config false;
              mandatory true;
              description
                "The maximum processing delay associated with the reception
                Port transmission Port pair.";
              reference
                "12.31.8.7 of IEEE Std 802.1Qcr-2020";
            }
          }
        }
      }
    }
  }
```

Insert the following text (Clause 49) after Clause 48:

49. Asynchronous Traffic Shaping (ATS) in end stations

49.1 Talker transmission behavior

The operation of ATS traffic classes in Bridges is specified as a combination of per-stream classification and metering for ATS (8.6.5.2.2) and the ATS transmission selection algorithm (8.6.8.5). In order for end station originated data streams to make use of ATS traffic classes in Bridges, it is required that Talkers emit data streams in a compliant manner, and that the parameters of ATS scheduler state machines and ATS scheduler groups in Bridges are consistently set.

49.1.1 ATS traffic class model in Talkers

End stations that are Talkers have to emit streams consistent with the operation of the ATS scheduler state machines in Bridges (8.6.11). This subclause specifies a model of ATS traffic classes that satisfies this purpose. End stations shall exhibit the transmission behavior of this model at the egress ports.

In this model, Talker ports are composed by ATS scheduler instances, which assign eligibility times to the frames of all emitted streams at the egress port, followed by a queue per traffic class, for which the ATS transmission selection algorithm is executed. The relationship between ATS scheduler instances and streams is one-to-one, and each scheduler state machine is associated with a dedicated scheduler group. This effectively eliminates the association of multiple streams to an ATS scheduler instance, as well as the coupling of multiple ATS scheduler instances by ATS scheduler groups.

49.1.2 Simplified ProcessFrame(frame) procedure

Due to the one-to-one relationship between streams and ATS scheduler instances, as well as ATS scheduler instances and scheduler groups, the ProcessFrame(frame) procedure of ATS scheduler state machines, as specified in 8.6.11.3, can be simplified. The simplified procedure is described by the following pseudo-code.

```
ProcessFrame(frame) {
       lengthRecoveryDuration     = length(frame)/
                                    CommittedInformationRate;
       emptyToFullDuration        = CommittedBurstSize/
                                    CommittedInformationRate;
       schedulerEligibilityTime   = BucketEmptyTime +
                                    lengthRecoveryDuration;
       bucketFullTime             = BucketEmptyTime +
                                    emptyToFullDuration;
       eligibilityTime            = max(arrivalTime(frame),
                                        schedulerEligibilityTime);

       BucketEmptyTime       = (eligibilityTime < bucketFullTime) ?
              schedulerEligibilityTime :
              schedulerEligibilityTime + eligibilityTime - bucketFullTime;
       AssignAndProceed(frame,eligibilityTime);
}
```

49.1.3 System clock functions and processing delays

This model has a single implementation specific local system clock function, which is used as

a) ATS scheduler clock (8.6.11.1) for all ATS scheduler instances and
b) Transmission selection clock (8.6.8.5) of all ATS traffic classes.

The processing delay from the ATS scheduler instances to the transmission selection is zero in this model (8.6.11.3.2):

$$0 = \text{ProcessingDelayMin} = \text{ProcessingDelayMax}$$

49.2 Scheduler parameter consistency

The ATS scheduler parameters CommittedBurstSize (8.6.11.3.5) and CommittedInformationRate (8.6.11.3.6) of Talker streams have to be consistently set to the respective parameters in the Bridge component connected to the Talker. In the Bridge component, a single ATS scheduler instance can be associated with multiple streams of this Talker.

For *n* streams associated with a single ATS scheduler instance in Bridge component, Equation (49-1) and Equation (49-2) can be used for scheduler parameter consistency.

$$\text{CBS}_{\text{Bridge}} \geq \sum_{i=1}^{n} \text{CBS}_i \tag{49-1}$$

$$\text{CIR}_{\text{Bridge}} \geq \left(\frac{1 + 10^{-6}\text{ClkDev}_{\text{Talker}}}{1 - 10^{-6}\text{ClkDev}_{\text{Bridge}}}\right) \sum_{i=1}^{n} \text{CIR}_i \tag{49-2}$$

where

$\text{CIR}_{\text{Bridge}}$	is the CommittedInformationRate parameter of the scheduler instance in the Bridge
CIR_i	is the CommittedInformationRate parameter of the *i*th stream in the Talker
$\text{CBS}_{\text{Bridge}}$	is the CommittedBurstSize parameter of the scheduler instance in the Bridge
CBS_i	is the CommittedBurstSize parameter of the *i*th stream in the Talker
$\text{ClkDev}_{\text{Bridge}}$	is the maximum deviation of the associated ATS scheduler clock (8.6.11.1) in the Bridge from its nominal frequency, in ppm
$\text{ClkDev}_{\text{Talker}}$	is the maximum deviation of the transmission selection clock (8.6.8.5 and 49.1.3) in the Talker from its nominal frequency, in ppm

Annex A

(normative)

PICS proforma—Bridge implementations[8]

A.5 Major capabilities

Change the following row in the table in A.5 as shown:

Item	Feature	Status	References	Support
PSFP	Does the implementation support PSFP?	O	~~8.6.5.1~~8.6.5.2.1, ~~8.6.6.1~~8.6.6 items d) and e), 8.6.10, 12.31	Yes [ ] No [ ]

Insert the following row at the end of the table in A.5:

Item	Feature	Status	References	Support
ATS	Does the implementation support Asynchronous Traffic Shaping?	O	5.4.1.10, 5.13.1.3, 8.6.5.2.2, 8.6.6 items d) and e), 8.6.8, 8.6.8.5, 8.6.11, 12.31	Yes [ ] No [ ]

A.14 Bridge Management

Change the following row in the table in A.14 as shown:

Item	Feature	Status	References	Support
MGT-250	Does the implementation support the management entities defined in 12.31 for PSFP?	PSFP OR CQF:M	5.4.1.9 item e), 5.13.1.2 item e), ~~8.6.5.1~~8.6.5.2.1, 8.6.10, 12.31	Yes [ ]

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

Insert the following row at the end of the table in A.14:

Item	Feature	Status	References	Support
MGT-251	Does the implementation support the management entities defined in 12.31 for ATS?	ATS:M	5.4.1.10, 5.13.1.3, 8.6.5.2.2, 8.6.6 items d) and e), 8.6.8, 8.6.8.5, 8.6.11, 12.31	Yes [ ] N/A [ ]

A.46 Per-stream filtering and policing

Change the table in A.46 as shown:

Item	Feature	Status	References	Support
	If neither per-stream filtering and policing (PSFP in Table A.5) nor cyclic queuing and forwarding (CQF in Table A.5) are supported, mark N/A and ignore the remainder of this table.		5.4.1.9, 5.13.1.2, ~~8.6.5.1~~8.6.5.2.1, ~~8.6.6.1~~8.6.6 items d) and e), 8.6.10, 12.31, 17.7.24	N/A[ ]
PSFP1	Does the implementation support the state machines and associated definitions as specified in 8.6.10?	PSFP OR CQF:M	5.4.1.9 item b), 5.13.1.2 item b), ~~8.6.5~~8.6.5.2.1, 8.6.10	Yes [ ] N/A [ ]
PSFP2	Does the implementation support the management entities defined in 12.31 for PSFP?	PSFP OR CQF:M	5.4.1.9 item e), 5.13.1.2 item e), ~~8.6.5.1~~8.6.5.2.1, ~~8.6.6.1~~8.6.6 items d) and e), 8.6.10, 12.31	Yes [ ] N/A [ ]
PSFP3	Is the IEEE8021-PSFP-MIB module fully supported (per its MODULE-COMPLIANCE)?	MIB AND (PSFP OR CQF):O	5.4.1.9 item e), 5.13.1.2 item e), 12.31, 17.7.24	Yes [ ] N/A [ ] No [ ]

A.48 YANG

Insert the following rows at the end of the table in A.48:

Item	Feature	Status	References	Support
YANG-10	Reserved for future use.			
YANG-11	Reserved for future use.			
YANG-12	Is the *ieee802-dot1q-stream-filters-gates* module supported?	ATS AND YANG:O	48.3.6	Yes [ ] No [ ] N/A [ ]
YANG-13	Reserved for future use.			
YANG-14	Is the *ieee802-dot1q-ats* module supported?	ATS AND YANG:O	48.3.8	Yes [ ] No [ ] N/A [ ]

Insert the following subclause (A.49) after A.48:

A.49 Asynchronous Traffic Shaping

Item	Feature	Status	References	Support
	If Asynchronous Traffic Shaping (ATS in Table A.5) is not supported, mark N/A and ignore the remainder of this table.		5.4.1.10, 5.13.1.3, 8.6.5.2.2, 8.6.6 items d) and e), 8.6.8.5, 8.6.8, 8.6.8.5, 8.6.11, 12.31	N/A [ ]
ATS1	Does the implementation support the ATS per-stream classification and metering for ATS as specified in 8.6.5.2.2?	ATS:M	5.4.1.10, 5.13.1.3, 8.6.5.2.2	Yes [ ] N/A [ ]
ATS2	Does the implementation support the ATS transmission selection algorithm as specified in 8.6.8.5?	ATS:M	5.4.1.10, 5.13.1.3, 8.6.8.5	Yes [ ] N/A [ ]
ATS3	Does the implementation support the ATS scheduler state machines as specified in 8.6.11?	ATS:M	5.4.1.10, 5.13.1.3, 8.6.11	Yes [ ] N/A [ ]
ATS4	Does the implementation support the management entities defined in 12.31 for ATS?	ATS:M	5.4.1.10, 5.13.1.3, 12.31	Yes [ ] N/A [ ]

Annex B

(normative)

PICS proforma—End station implementations[9]

B.5 Major capabilities

Change the following row in the table in B.5 as shown:

Item	Feature	Status	References	Support
PSFP	Does the implementation support PSFP?	O	~~8.6.5.1~~8.6.5.2.1, ~~8.6.6.1,~~ 8.6.10, 12.31	Yes [ ] No [ ]

Insert the following row at the end of the table in B.5:

Item	Feature	Status	References	Support
ATS	Does the implementation support Asynchronous Traffic Shaping?	O	8.6.5.2.2, 8.6.8, 8.6.8.5, 8.6.11, 12.31	Yes [ ] No [ ]

B.17 Per-stream filtering and policing

Change the table in B.17 as shown:

Item	Feature	Status	References	Support
	If neither per-stream filtering and policing (PSFP in Table B.5) nor cyclic queuing and forwarding (CQF in Table B.5) are supported, mark N/A and ignore the remainder of this table.		5.28 items d) and e), ~~8.6.5.1~~8.6.5.2.1, ~~8.6.6.1,~~ 8.6.10, 12.31, 17.7.24	N/A[ ]
PSFP1	Does the implementation support the state machines and associated definitions as specified in 8.6.10?	PSFP OR CQF:M	5.28 items b) and d), ~~8.6.5~~8.6.5.4, 8.6.10	Yes [ ] N/A [ ]
PSFP2	Does the implementation support the management entities defined in 12.31 for PSFP?	PSFP OR CQF:M	5.28 item e), ~~8.6.5.1~~8.6.5.2.1, ~~8.6.6.1,~~ 8.6.10, 12.31	Yes [ ] N/A [ ]
PSFP3	Is the IEEE8021-PSFP-MIB module fully supported (per its MODULE-COMPLIANCE)?	MIB AND (PSFP OR CQF):O	12.31, 17.7.24	Yes [ ] N/A [ ] No [ ]

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

Insert the following subclause (B.18) after B.17:

B.18 Asynchronous Traffic Shaping

Item	Feature	Status	References	Support
	If Asynchronous Traffic Shaping (ATS in Table B.5) is not supported, mark N/A and ignore the remainder of this table.		5.32, 8.6.5.2.2, 8.6.8, 8.6.8.5, 8.6.11, 49.1.2	N/A [ ]
ATS1	Does the implementation support at least one traffic class that supports the strict priority transmission selection algorithm (8.6.8.1) and another traffic class that supports the ATS transmission selection algorithm (8.6.8.5)?	ATS:M	5.32, 8.6.8, 8.6.8.5	Yes [ ] N/A [ ]
ATS2	Does the implementation support ATS schedulers as specified in 8.6.5.6?	ATS:M	5.32, 8.6.5.6, 49.1	Yes [ ] N/A [ ]
ATS3	Does the implementation support the ATS scheduler state machines as specified in 49.1?	ATS:M	5.32, 49.1	Yes [ ] N/A [ ]

Annex T

(informative)

Cyclic queuing and forwarding[10]

T.2 An approach to CQF implementation

Change the second paragraph of T.2 as shown:

CQF is implemented by configuring a combination of the stream gate control mechanisms defined ~~for per-stream filtering and policing (PSFP, 8.6.5.1)~~in 8.6.5.4 and the traffic scheduling mechanisms defined in 8.6.8.4 and 8.6.9. Per-stream filtering is used to direct received frames to one of a pair of outbound queues on a timed basis, determined by the cycle time of the per-stream filter, and traffic scheduling is used to ensure that frames are transmitted from the appropriate queue using the same cycle time, as described in the rest of this annex.

T.3 Use of per-stream filtering and policing for CQF

Change the introductory text of T.3 as shown:

The first step in establishing the filtering and queuing structures needed for CQF is to set up one or more stream filters (~~8.6.5.1.1~~8.6.5.3) and a stream gate instance (~~8.6.5.1.2~~8.6.5.4) that will be receiving incoming time-sensitive frames. The stream filter(s) are configured so that all time-sensitive frames received on a given Port are directed to the same stream gate instance; in turn, the stream gate instance is configured so that the internal priority value (IPV) associated with the time-sensitive frames will direct them to one of two outbound queues on a timed basis. The use of the IPV allows this direction of frames to outbound queues to be independent of the received priority, and also does not affect the priority associated with the frame on transmission.

Change the text of T.3.1 and T.3.2 and the title of Figure T-1 as shown:

T.3.1 Stream filter configuration

The simplest stream filter configuration would be achieved where the same priority is used for all time-sensitive frames (and this priority is not used for any other frames); for example, the default priority assigned to SR class A (see Clause 34) could be used, in which case, the priority associated with the time-sensitive frames would be 3. The parameters that would define the stream filter for the time-sensitive frames would then be as follows:

a) The *stream_handle~~identifier~~ specification* would take the wild-card value.
b) The *priority specification* would take the priority value 3.
c) The *stream gate instance identifier* would take the value of the instance identifier for the stream gate (T.3.2).
d) In the simplest case, there would be no ~~filter specifications~~ further per-stream classification and metering operations (8.6.5.2); however, these could be added as appropriate, for example if the maximum SDU size (8.6.5.3.1) for the time-sensitive traffic is bounded at a value less than the maximum SDU size for the medium.

[10] In early discussions, CQF was known as the “Peristaltic Shaper” [B53].

This stream filter ~~specification~~configuration results in all frames that carry a priority value of 3 being submitted to the stream gate. As the operation of ~~PSFP~~stream filters is such that received frames that do not match a stream filter are handled as if ~~PSFP~~subsequent per-stream classification amd metering operations were~~is~~ not implemented, there is no need for further stream filter configuration~~specifications~~ to handle frames that carry priorities other than 3 unless there are other filtering or gating decisions that need to be taken for such frames.

T.3.2 Stream gate configuration

The *stream gate instance* (~~8.6.5.1.1~~8.6.5.4) needed to support the stream filter described in T.3.1 has a *stream gate control list* that contains two entries, each containing a SetGateAndIP~~S~~V operation, with parameters as follows:

1) StreamGateState = *open*, IPV = 7, TimeInterval = T
2) StreamGateState = *open*, IPV = 6, TimeInterval = T

This control list has the effect of directing any traffic that passes the stream filter specified in T.3.1 to one of two different outbound queues (assuming that the outbound Ports support 8 queues, and that the default assignments for priorities to traffic classes follows the recommendation shown in Table 34-1); in the first time interval T, traffic is directed to queue 7, in the second time interval T, to queue 6, in the third time interval to queue 7, in the fourth time interval, to queue 6, and so on. The choice of time interval T is discussed in T.5; the cycle time (OperCycleTime, see 8.6.9.4.20) for the stream gate state machines would need to be set to 2T in order to accommodate the sum of the time intervals for the two gate operations. See Figure T-1.

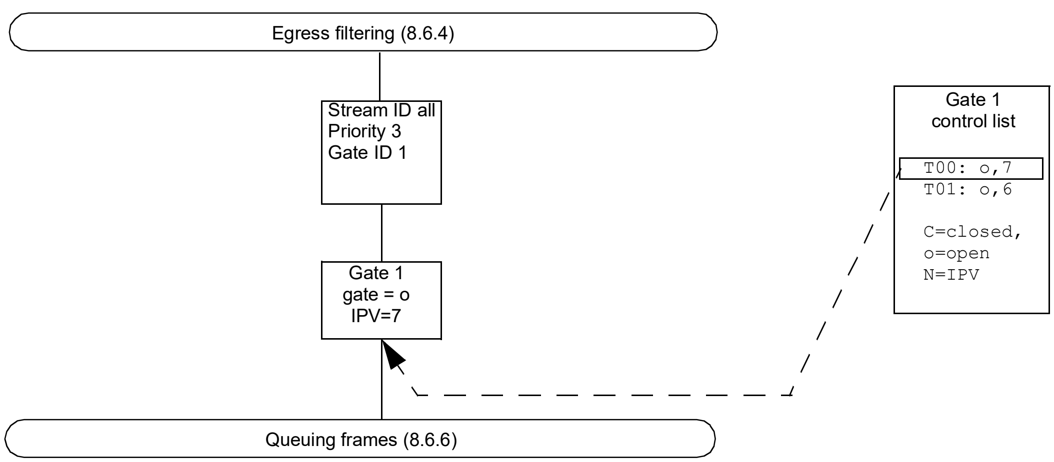


Figure T-1—Example ~~PSFP~~Stream Filter and Stream Gate configuration for ~~Q~~CQF

T.5 Timing considerations

T.5.1 Choice of Time Interval T

Change the second paragraph of T.5.1 as shown:

If streams associated with two different observation intervals are being handled, for example if streams that use SR classes A and B pass through the Bridge, then the OperCycleTime used for the transmit traffic scheduling has to be a common multiple of the two class measurement intervals that are in use in order to make it possible for the transmission cycles to properly match the two values of T that are chosen. Figure T-2 and Figure T-3 illustrate how the ~~PSFP~~Stream Filters, Stream Gates, and traffic scheduling could be configured in the case where SR classes A and B are active; in Figure T-2, incoming frames that carry SR Class A (priority 3) are handled using Gate 1, and the cycle time for the stream gate control list is twice the class measurement interval for SR Class A, which is 2 × 125 µs. Gate 1 alternately tags these frames with an IPV of 7 or 6. Incoming frames that carry SR Class B (priority 2) are handled using Gate 2, and the cycle time for the stream gate control list is twice the class measurement interval for SR Class B, which is 2 × 250 µs. Gate 2 alternately tags these frames with an IPV of 5 or 4.

Change the title of Figure T-2 as shown:

Figure T-2—Example ~~PSFP~~Stream Filter and Stream Gate configuration with two values of T

Insert the following text (Annex V) after Annex U [see subsequent instructions for the Bibliography annex (now Annex W)]:

Annex V

(informative)

Asynchronous Traffic Shaping delay analysis framework

The framework in this annex provides methods for worst-case delay analysis in static networks with static configurations. General assumptions of this framework are listed in V.1. The end-to-end delay modeling approach is described in V.2. An upper bound on buffering delays is described in V.3. Subsequent clauses (V.4 through V.8) cover additional sources of delays. The combined delay bounds are shown in V.9.

V.1 General assumptions

The following assumptions are made throughout the remainder of this framework:

a) The Transmission Selection Table (8.6.8) of all transmission Ports under consideration assigns the ATS Transmission Selection Algorithm (Table 8-6, 8.6.8.5) to one or more numerically highest traffic classes (i.e., no other algorithm than the ATS Transmission Selection Algorithm is assigned to a numerically higher traffic class than the traffic classes to which the ATS Transmission Selection Algorithm is assigned).
b) The transmission gates (8.6.8.4) associated with all traffic classes using the ATS Transmission Selection Algorithm of all transmission Ports under consideration reside permanently in state Open.
c) All streams are associated with ATS traffic classes in all transmission Ports.
d) If frame preemption (6.7.2) is supported by a transmission Port under consideration, it is assumed that the frame preemption status in the frame preemption status table associated with the transmission Port in the network is either express, or preemptible for all traffic classes using the ATS Transmission Selection Algorithm.
e) The underlying MAC Service of all transmission Ports under consideration transmits at a constant data rate.
f) The committed information rate parameters in Bridges under consideration are set consistently with the constraints in V.8.
g) The data rate of a Port under consideration is greater than the sum of associated committed information rates.
h) Frames at all transmission Ports have the same media-dependent overhead (12.4.2.2). Likewise, there is no variation in frame lengths (i.e., tag addition or removal along a path is not considered).

V.2 End-to-end delay modeling approach

The path from the Talker station to a single Listener station is subdivided into the contained n hops, where the 1st hop is the hop from the Talker to the first Bridge on the path, and the nth hop is the hop from the last Bridge on the path to the Listener.

The end-to-end delay bound $d_{max}(f)$ of a stream of interest is given by per hop delay bounds $d_{max}(k, f)$ of the n subsequent hops along the path of stream f as shown in Equation (V-1).

$$d_{max}(f) = \sum_{k=1}^{n} d_{max}(k, f) \qquad \text{(V-1)}$$

where

f	is the stream of interest
k	is the hop index at the path from the Talker to the Listener ($1 \le k \le n$)
$d_{max}(k, f)$	is the per hop delay bound of stream f at the kth hop

Figure V-1 illustrates the path of stream *f* across the relevant mechanisms and interfaces of two subsequent bridges along the path, and the associated delay bounds [e.g., $d_{AT,\,max}(k)$, introduced in V.5] described in the following subclauses.

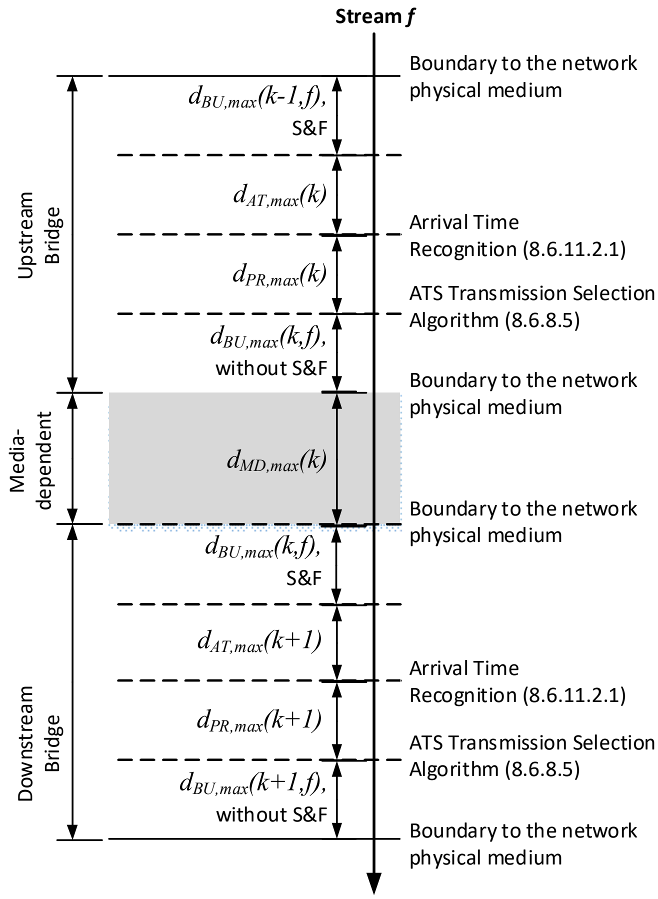


Figure V-1—Path of frames along a single hop with index *k* with two Bridges

V.3 Buffering delays

The buffering delays comprise all delays on frames of a stream *f* resulting from their residence in queues of associated traffic classes in the transmission Ports along the path. For a transmission Port on the path, these delays are caused by

a) Competing frames in queues of traffic classes different than the traffic class of stream *f*.
b) Competing frames in the queue of the traffic class of stream *f* competing for transmission.
c) The enforcement of token-bucket traffic envelopes by the operation of the ATS scheduler state machines (8.6.11).
d) Store and forward operation in reception Ports.

A delay bound $d_{BU,\,max}(k,f)$ on the buffering delay for a single hop in absence of the effects covered in subsequent clauses is known (see Specht and Samii [B87]). For $k < n$ (i.e., all but the last hop to the Listener), the delay bound $d_{BU,\,max}(k,f)$ is given by Equation (V-2).

$$d_{BU,\,max}(k,f) = \max_{h \in F_S(k,f)} \left\{ \frac{\sum_{g \in F_H(k,h) \cup F_S(k,h)} b_{max}(k,g) - l_{min}(h) + l_{LP,\,max}(k,h)}{R(k) - \sum_{g \in F_H(k,h)} r_{max}(k,g)} + \frac{l_{min}(h)}{R(k)} \right\} \quad \text{(V-2)}$$

For $k = n$ (i.e., the last hop to the Listener), this bound is given by Equation (V-3).

$$d_{BU,\,max}(n,f) = \frac{\sum_{g \in F_H(n,f) \cup F_S(n,f)} b_{max}(n,g) - l_{min}(f) + l_{LP,\,max}(n,f)}{R(n) - \sum_{g \in F_H(n,f)} r_{max}(n,g)} + \frac{l_{min}(f)}{R(n)} \quad \text{(V-3)}$$

where

$F_H(k,f)$ and $F_H(k,h)$	denote the set of streams transmitted in a numerically higher traffic class (8.6.8) than stream *f* and a stream *h*, respectively, at the upstream transmission Port of the *k*th hop
$F_S(k,f)$ and $F_S(k,h)$	denote the set of streams transmitted in the same traffic class as stream *f*, including stream *f* and stream *h*, respectively, at the upstream transmission Port of the *k*th hop
$l_{LP,\,max}(k,f)$ and $l_{LP,\,max}(k,h)$	denote the maximum interference length, in bits, by any numerically lower traffic class than the class of stream *f* and a stream *h*, respectively, at the upstream transmission Port of the *k*th hop
$l_{min}(f)$ and $l_{min}(h)$	denote the minimum frame length of stream *f* and a stream *h*, respectively, in bits, including all media-dependent overhead (8.6.11.3.11, 12.4.2.2)
$b_{max}(k,g)$	is the maximum burst size associated with a stream *g* at the *k*th hop, in bits
$r_{max}(k,g)$	is the committed information rate of stream *g* at in the upstream device of the *k*th hop, in bits per second
$R(k)$	is the transmission rate, in bits per second, that the underlying MAC Service that supports transmission through the upstream transmission Port of the *k*th hop provides

NOTE 1—The relationship between $b_{max}(k,g)$ and the committed burst size at the Talker station is further discussed in V.7.

NOTE 2—The relationship between $r_{max}(k,g)$ and the committed information rate at the Talker station (49.2) is further discussed in V.8.

NOTE 3—If frame preemption is not supported, $l_{LP,\,max}(k,f)$ is at most one maximum frame length, including all media-dependent overhead, supported by the upstream transmission Port of the *k*th hop. If frame preemption is supported, and all classes with the ATS transmission selection algorithm can preempt all numerically lower traffic classes, $l_{LP,\,max}(k,f)$ is at most a maximum fragment length (S.2), including all media-dependent overhead.

V.4 Media-dependent delays

Frames of stream *f* experience a media-dependent delay between the upstream transmission Port of this hop and the downstream reception Port of a hop. This delay is measured between

a) The time at which a particular octet of a frame passed the boundary from the upstream transmission Port to the network physical medium and
b) The time at which this particular octet passed the boundary from the network physical medium to the downstream Port.

The maximum media-dependent delay of the *k*th hop is denoted as $d_{MD,\,max}(k)$ and assumed to be known for analysis.

Variations in media-dependent delays do not affect the combined delay bounds in V.9. Such variations happen before the arrival time recognition by the associated ATS scheduler clock and the processing by the associated ATS scheduler instance. A frame that experiences a lower media-dependent delay than an earlier frame processed by the same scheduler instance would be delayed by scheduler, if required by token bucket envelope enforcement.

V.5 Bridge—Internal arrival time recognition delays

The maximum arrival time recognition delay in the upstream Bridge of the *k*th hop ($k > 1$) for any frame of a stream is given by the associated ArrivalRecognitionDelayMax parameter (8.6.11.3.1, 12.31.8.5). For compact representation and notational consistency, ArrivalRecognitionDelayMax is subsequently denoted as $d_{AT,\,max}(k)$.

Variations in arrival time recognition delays do not affect the combined delay bounds in V.9, similar to variations in media-dependent delays (V.4).

V.6 Bridge—Internal processing delays

The bounds on the processing delays in the upstream Bridge of the *k*th hop ($k > 1$) for any frame of a stream are given by given by the associated ProcessingDelayMin and ProcessingDelayMax parameters (8.6.11.3.2, 12.31.8.6, 12.31.8.7). The associated ProcessingDelayMin and ProcessingDelayMax parameters are subsequently denoted as $d_{PR,\,min}(k)$ and $d_{PR,\,max}(k)$, respectively.

Variations in Bridge-internal processing delays can increase the burst sizes in buffering delays (V.3). Based on the specified computation of the assigned eligibility time (8.6.11.3.2), these variations reside within the associated clock offset variations (V.7).

NOTE—The total of the maximum arrival time recognition delay and the maximum processing delay is different from the associated maximum independent delay found in the Bridge Delay attributes (12.32.1.1). This total can be larger.

V.7 Bridge—Internal clock offset variations

The maximum clock offset variation in the upstream Bridge of the *k*th hop ($k > 1$) for any frame of a stream is given by the associated ClockOffsetVariationMax parameter (8.6.11.2, 12.31.8.3). The associated ClockOffsetVariationMax is subsequently denoted as $\Delta_{CO,\,max}(k, g)$.

Clock offset variations of up to $d_{AT,\,max}(k)$ between ATS scheduler clock and transmission selection clock instances (8.6.11.2) associated with a stream *g* can increase the burst size of streams subsequently competing in the transmission Port.

This impact can be taken into account by definition of $b_{max}(k, g)$ for $k > 1$, as shown in Equation (V-4).

$$b_{max}(k, g) = r_{max}(k, g)\Delta_{CO,\,max}(k, g) + b_{max}(1, g) \qquad \text{(V-4)}$$

where $b_{max}(1, g)$ is the committed burst size of stream *g* at the Talker station (49.2).

V.8 Inter-device clock rate deviations

The clock constraints in 8.6.11 limit ATS scheduler clock instances and transmission selection clock instances in a Bridge to effectively operate at the same rate, although differences within [ClockOffsetMin,ClockOffsetMax] are permitted (8.6.11.2). As a basic property of asynchronous mechanisms such as ATS, no such limitation exists between different devices (i.e., clocks in different devices are not synchronized).

Deviations of clocks from their nominal rates (e.g., within oscillator tolerances) affect the spacing between successive frames according to the assigned eligibility times (8.6.11.3.2, 8.6.8.5). If the upstream device at a hop runs faster than nominal (e.g., +100 ppm), and a connected downstream Bridge at this hop runs slower than nominal (e.g., –100 ppm), the backlog as well as the per hop delay in the downstream Bridge could grow under peak load conditions.

The situation can be prevented by management constraints (12.31.5) on $r_{max}(k, g)$ for $1 < k < n$, such that Equation (V-5) holds for any stream *g*.

$$r_{max}(k, g) \geq \frac{1 + 10^{-6}\Delta_{CR,\,max}(k-1)}{1 - 10^{-6}\Delta_{CR,\,max}(k)} r_{max}(k-1, g) \qquad \text{(V-5)}$$

where

$\Delta_{CR,\,max}(k)$	is the upper bound over all absolute rate deviations of all ATS scheduler clocks and transmission selection clocks from their nominal rate in the upstream device of the *k*th hop available via the associated ClockRateDeviationMax parameter (12.31.8.4), in ppm (e.g., $\Delta_{CR,\,max}(k) = 100$)
$r_{max}(1, g)$	is the committed information rate of stream *g* at the Talker station (49.2)

V.9 Combined delay bounds

For a path from a Talker station to a Listener station with at least one Bridge ($n > 1$), a combined end-to-end delay bound $d_{max}(f)$ of a stream of interest *f* can be summarized by Equation (V-6)

$$d_{max}(f) = \sum_{k=1}^{n} d_{BU,\,max}(k, f) + \sum_{k=1}^{n} d_{MD,\,max}(k) + \sum_{k=2}^{n} d_{AT,\,max}(k) + \sum_{k=2}^{n} d_{PR,\,max}(k) \qquad \text{(V-6)}$$

with Equation (V-7), Equation (V-8), and Equation (V-9) for $1 < k < n$.

$$d_{BU,max}(1,f) = \max_{h \in F_S(1,f)} \left\{ \frac{\sum_{g \in F_H(1,h) \cup F_S(1,h)} b_{max}(1,g) - l_{min}(h) + l_{LP,max}(1,h)}{R(1) - \sum_{g \in F_H(1,h)} r_{max}(1,g)} + \frac{l_{min}(h)}{R(1)} \right\} \quad \text{(V-7)}$$

$$d_{BU,max}(n,f) = \frac{\sum_{g \in F_H(n,f) \cup F_S(n,f)} (r_{max}(n,g)\Delta_{CO,max}(n,g) + b_{max}(1,g)) - l_{min}(f) + l_{LP,max}(n,f)}{R(n) - \sum_{g \in F_H(n,f)} r_{max}(n,g)} + \frac{l_{min}(f)}{R(n)} \quad \text{(V-8)}$$

$$d_{BU,max}(k,f) = \max_{h \in F_S(k,f)} \left\{ \frac{\sum_{g \in F_H(k,h) \cup F_S(k,h)} (r_{max}(k,g)\Delta_{CO,max}(k,g) + b_{max}(1,g)) - l_{min}(h) + l_{LP,max}(k,h)}{R(k) - \sum_{g \in F_H(k,h)} r_{max}(k,g)} + \frac{l_{min}(h)}{R(k)} \right\} \quad \text{(V-9)}$$

In absence of Bridges on the path from the Talker station to the Listener station (i.e., $n = 1$), the end-to-end delay bound can be summarized by Equation (V-10).

$$d_{max}(1,f) = \frac{\sum_{g \in F_H(1,f) \cup F_S(1,f)} b_{max}(1,g) - l_{min}(f) + l_{LP,max}(1,f)}{R(1) - \sum_{g \in F_H(1,f)} r_{max}(1,g)} + \frac{l_{min}(f)}{R(1)} + d_{MD,max}(1) \quad \text{(V-10)}$$

Change the designation for the Bibliography annex as shown:

~~Annex V~~

Annex W

(informative)

Bibliography

Insert the following bibliographic references into Annex W in alphanumeric order:

[B86] Tanenbaum, A. S., and D. J. Wetherall, *Computer Networks*, 5th ed. New Jersey: Prentice Hall, 2010, pp. 407–411.

[B87] Specht, J., and S. Samii, "Urgency-Based Scheduler for Time-Sensitive Switched Ethernet Networks," *28th Euromicro Conference on Real-Time Systems (ECRTS)*, pp. 75–85, 2016.

IEEE SA
STANDARDS
ASSOCIATION

RAISING THE WORLD'S STANDARDS

Connect with us on:

Twitter: twitter.com/ieeesa

Facebook: facebook.com/ieeesa

LinkedIn: linkedin.com/groups/1791118

Beyond Standards blog: beyondstandards.ieee.org

YouTube: youtube.com/ieeesa

standards.ieee.org
Phone: +1 732 981 0060

IEEE