IEEE STANDARDS ASSOCIATION

IEEE Standard for
Local and metropolitan area networks—

Bridges and Bridged Networks—

Amendment 29: Cyclic Queuing and Forwarding

IEEE Computer Society

Sponsored by the
LAN/MAN Standards Committee

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

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

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

IEEE Standard for
Local and metropolitan area networks—

Bridges and Bridged Networks—

Amendment 29: Cyclic Queuing and Forwarding

Sponsor

LAN/MAN Standards Committee
of the
IEEE Computer Society

Approved 18 May 2017

IEEE-SA Standards Board

Abstract: The use of traffic scheduling and per-stream filtering and policing to support cyclic queuing and forwarding are described in this amendment to IEEE Std 802.1Q-2014.

Keywords: Bridged Local Area Networks, cyclic queuing and forwarding (CQF), IEEE 802®, IEEE 802.1Q™, IEEE Std 802.1Qbu™, IEEE 802.1Qbv™, IEEE 802.1Qch™, IEEE 802.1Qci™, local area networks (LANs), MAC Bridges, metropolitan area networks, scheduled traffic, Virtual Bridged Local Area Networks (virtual LANs)

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

Print:	ISBN 978-1-5044-4095-0	STD22629
PDF:	ISBN 978-1-5044-4096-7	STDPD22629

Important Notices and Disclaimers Concerning IEEE Standards Documents

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

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

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

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

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

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

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

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

Translations

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

Official statements

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

Comments on standards

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

Comments on standards should be submitted to the following address:

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

Laws and regulations

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

Copyrights

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

Photocopies

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

Updating of IEEE Standards documents

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

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

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

Errata

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

Patents

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

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

Participants

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

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

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

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

Thomas Alexander
Butch Anton
Stefan Aust
Gordon Bechtel
Steinar Bjornstad
Christian Boiger
Nancy Bravin
William Byrd
Juan Carreon
Keith Chow
Charles Cook
Rodney Cummings
Eugene Dai
Patrick Diamond
Sourav Dutta
Marc Emmelmann
János Farkas
Norman Finn
Michael Fischer
Yukihiro Fujimoto
Devon Gayle
Eric W. Gray
Randall Groves
Craig Gunther
Stephen Haddock
Marek Hajduczenia
Mark Hantel
Marco Hernandez
Guido Hiertz
Werner Hoelzl
Rita Horner
Noriyuki Ikeuchi
Osamu Ishida
Atsushi Ito
Raj Jain
Tony Jeffree
SangKwon Jeong
Peter Jones
Jeritt Kent
Stuart Kerry
Yongbum Kim
Jeff Koftinoff
Jouni Korhonen
Bruce Kraemer
John Lemon
Joseph Levy
Jon Lewis
Arthur H. Light
Elvis Maculuba
Arthur Marris
Charles Moorwood
Matthew Mora
Henry Muyshondt
Michael Newman
Charles Ngethe
Satoshi Obara
Glenn Parsons
Bansi Patel
Adee Ran
Alon Regev
Robert Robinson
Benjamin Rolfe
Dan Romascanu
Osman Sakr
Frank Schewe
Michael Seaman
Thomas Starai
Walter Struppler
Patrik Sundstrom
Steven Tilden
Mark-Rene Uchida
Prabodh Varshney
George Vlantis
Karl Weber
Andreas Wolf
Oren Yuen
Zhen Zhou

When the IEEE-SA Standards Board approved this standard on 18 May 2017, it had the following membership:

Introduction

This introduction is not part of IEEE Std 802.1Qch-2017, IEEE Standard for Local and metropolitan area networks—Bridges and Bridged Networks—Amendment 29: Cyclic Queuing and Forwarding.

This amendment to IEEE Std 802.1Q-2014 describes the use of traffic scheduling and per-stream filtering and policing to support cyclic queuing and forwarding.

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

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

Contents

List of figures

IEEE Standard for Local and metropolitan area networks—

Bridges and Bridged Networks—

Amendment 29: Cyclic Queuing and Forwarding

(This amendment is based on IEEE Std 802.1Q™-2014 as amended by IEEE Std 802.1Qca™-2015, IEEE Std 802.1Qcd™-2015, IEEE Std 802.1Q-2014/Cor 1-2015, IEEE Std 802.1Qbv™-2015, IEEE Std 802.1Qbu™-2016, IEEE Std 802.1Qbz™-2016, and IEEE Std 802.1Qbci™-2017)

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

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

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

2. Normative references

Insert the following references in alphanumeric order:

IEEE Std 802.1Qbv™, IEEE Standard for Local and metropolitan area networks—Bridges and Bridged Networks—Amendment 25: Enhancements for Scheduled Traffic.

IEEE Std 802.1Qci™, IEEE Standard for Local and metropolitan area networks—Bridges and Bridged Networks—Amendment 28: Per-Stream Filtering and Policing.

NOTE—IEEE Std 802.1Qbv and IEEE Std 802.1Qci are both amendments to IEEE Std 802.1Q-2014, as is this standard. This NOTE and the references to IEEE Std 802.1Qbv and IEEE P802.1Qci will be removed when this standard is incorporated into the IEEE 802.1Q base standard in the next revision.

4. Abbreviations

Insert the following abbreviation in alphanumeric order:

CQF cyclic queuing and forwarding

5. Conformance

5.4 VLAN Bridge component requirements

5.4.1 VLAN Bridge component options

Insert new subclause 5.4.1.9 as shown, renumbering as necessary:

5.4.1.9 Cyclic queuing and forwarding (CQF) requirements

A VLAN Bridge component implementation that conforms to the provisions of this standard for CQF (see Annex T) shall

a) Support the enhancements for scheduled traffic as specified in 8.6.8.4.
b) Support the state machines for scheduled traffic as specified in 8.6.9.
c) Support the state machines for stream gate control as specified in 8.6.10.
d) Support the management entities for scheduled traffic as specified in 12.29.
e) Support the requirements for per-stream filtering and policing (PSFP) as stated in 5.4.1.8.
f) Support the management entities for PSFP as specified in 12.31.

NOTE—The enhancements for scheduled traffic are defined in IEEE Std 802.1Qbv, and PSFP is defined in IEEE Std 802.1Qci, both of which are amendments to IEEE Std 802.1Q-2014, as is this standard. This NOTE will be removed when this standard is incorporated into the IEEE 802.1Q base standard in the next revision.

5.13 MAC Bridge component requirements

5.13.1 MAC Bridge component options

Insert new subclause 5.13.1.2 as shown, renumbering as necessary:

5.13.1.2 Cyclic queuing and forwarding requirements

A MAC Bridge component implementation that conforms to the provisions of this standard for CQF (see Annex T) shall

a) Support the enhancements for scheduled traffic as specified in 8.6.8.4.
b) Support the state machines for scheduled traffic as specified in 8.6.9.
c) Support the state machines for stream gate control as specified in 8.6.10.
d) Support the management entities for scheduled traffic as specified in 12.29.
e) Support the requirements for PSFP as stated in 5.13.1.1.
f) Support the management entities for PSFP as specified in 12.31.

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

5.28 End station requirements—Cyclic queuing and forwarding

An end station implementation that conforms to the provisions of this standard for CQF (see Annex T) shall

a) Support the enhancements for scheduled traffic as specified in 8.6.8.4.
b) Support the state machines for scheduled traffic as specified in 8.6.9.
c) Support the state machines for stream gate control as specified in 8.6.10.

d) Support the management entities for scheduled traffic as specified in 12.29.
e) Support the requirements for PSFP as stated in 5.27.
f) Support the management entities for PSFP as specified in 12.31.

Annex A

(normative)

PICS proforma—Bridge implementations[2]

A.5 Major capabilities

Insert the following row at the end of Table A.5:

CQF	Does the implementation support cyclic queuing and forwarding?	O	5.4.1.9, 5.13.1.2	Yes [ ]	No [ ]

A.14 Bridge management

Change MGT-248 and MGT-249 in Table A.14 as shown:

Item	Feature	Status	References	Support
MGT-248	Does the implementation support the management entities defined in 12.29?	SCHED OR CQF: M	5.4.1 item ad), 5.4.1.9 item c), 5.13.1.2 item c), 12.29	Yes [ ] N/A [ ]
MGT-249	Does the implementation support the management entities defined in 12.31?	PSFP OR CQF: M	5.4.1.9 item e), 5.13.1.2 item e), 8.6.5.1, 8.6.6.1, 8.6.10, 12.31	Yes [ ] N/A [ ]

A.24 Management Information Base (MIB)

Change MIB-42 and MIB-43 in Table A.24 as shown:

Item	Feature	Status	References	Support
MIB-42	Is the IEEE8021-ST-MIB module fully supported (per its MODULE-COMPLIANCE)?	MIB AND (SCHED OR CQF): O	5.4.1 item ad), 5.4.1.9 item c), 5.13.1.2 item c), 12.29, 17.7.22	Yes [ ] N/A [ ] No [ ]
MIB-43	Is the IEEE8021-PSFP-MIB module fully supported (per its MODULE-COMPLIANCE)?	PSFP OR CQF: O	5.4.1.9 item e), 5.13.1.2 item e), 8.6.5.1, 8.6.6.1, 8.6.10, 12.31, 17.7.24	Yes [ ] N/A [ ]

[2]*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.

Change Table A.44 as shown:

A.44 Scheduled traffic

Item	Feature	Status	References	Support
	If neither scheduled traffic (SCHED in Table A.5) nor cyclic queuing and forwarding (CQF in Table A.5) ~~is not~~ are supported, mark N/A and ignore the remainder of this table.		5.4.1, 5.13.1, 8.6.8, 8.6.9, 12.29, 17.7.22	N/A[ ]
SCHED1	Support the state machines and associated definitions as specified in 8.6.9	SCHED OR CQF:M	5.4.1, 5.13.1, 8.6.8, 8.6.9	Yes [ ] N/A [ ]
SCHED2	Does the implementation support the management entities defined in 12.29?	SCHED OR CQF:M	5.4.1 item ad), 5.4.1.9 item c), 5.13.1.2 item c), 12.29	Yes [ ] N/A [ ]
SCHED3	Is the IEEE8021-ST-MIB module fully supported (per its MODULE-COMPLIANCE)?	MIB AND (SCHED OR CQF):O	5.4.1 item ad), 5.4.1.9 item c), 5.13.1.2 item c), 12.29, 17.7.22	Yes [ ] N/A [ ] No [ ]

Change Table A.45 as shown:

A.45 Per-stream filtering and policing

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) ~~is not~~ are supported, mark N/A and ignore the remainder of this table.		5.4.1.9, 5.13.1.2, 8.6.5.1, 8.6.6.1, 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	5.4.1.9 item b), 5.13.1.2 item b), 8.6.5, 8.6.10	Yes [ ] N/A [ ]
PSFP2	Does the implementation support the management entities defined in 12.31?	PSFP OR CQF:M	5.4.1.9 item e), 5.13.1.2 item e), 8.6.5.1, 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	5.4.1.9 item e), 5.13.1.2 item e), 12.31, 17.7.24	Yes [ ] N/A [ ] No [ ]

Annex B

(normative)

PICS proforma—End station implementations[3]

B.5 Major capabilities

Insert the following row at the end of Table B.5:

CQF	Does the implementation support cyclic queuing and forwarding?	O	5.25, 5.28	Yes [ ]	No [ ]

Change Table B.15 as shown:

B.15 Scheduled traffic

Item	Feature	Status	References	Support
	If neither scheduled traffic (SCHED in Table B.5) nor cyclic queuing and forwarding (CQF in Table B.5) ~~is not~~ are supported, mark N/A and ignore the remainder of this table.		~~5.4.1, 5.13.1,~~5.25, 5.28, 8.6.8, 8.6.9, 12.29, 17.7.22	N/A[ ]
SCHED1	Support the state machines and associated definitions as specified in 8.6.9	SCHED OR CQF:M	~~5.4.1, 5.13.1,~~5.28 item b),~~,~~ 8.6.8, 8.6.9	Yes [ ] N/A [ ]
SCHED2	Does the implementation support the management entities defined in 12.29?	SCHED OR CQF:M	~~5.4.1 item ad)~~5.28 item c), 12.29	Yes [ ] N/A [ ]
SCHED3	Is the IEEE8021-ST-MIB module fully supported (per its MODULE-COMPLIANCE)?	MIB AND (SCHED OR CQF):O	~~5.4.1 item ad)~~5.28 item c), 12.29, 17.7.22	Yes [ ] N/A [ ] No [ ]

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

Change Table B.16 as shown:

B.16 Per-stream filtering and policing

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) ~~is not~~ are supported, mark N/A and ignore the remainder of this table.		5.28 items d) and e), 8.6.5.1, 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, 8.6.10	Yes [ ] N/A [ ]
PSFP2	Does the implementation support the management entities defined in 12.31?	PSFP OR CQF:M	5.28 item e), 8.6.5.1, 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 [ ]

Annex T

(informative)

Cyclic queuing and forwarding[4]

T.1 Overview of CQF

Cyclic queuing and forwarding (CQF) is a method of traffic shaping that can deliver deterministic, and easily calculated, latency for time-sensitive traffic streams. As the name implies, the principle underlying CQF is that stream traffic is transmitted and queued for transmission along a network path in a cyclic manner. Time is divided into numbered time intervals i, $i+1$, $i+2$, ... $i+N$, each of duration d. Frames transmitted by a Bridge, *Alice*, during time interval i are received by a downstream Bridge, *Bob*, during time interval i and are transmitted onwards by *Bob* towards Bridge *Charlie* during time interval $i+1$, and so on. A starting assumption is that, for a given traffic class, all Bridges and all end stations connected to a given bridge have a common understanding (to a known accuracy) of the start time of cycle i, and the cycle duration, d.

Frames transmitted by *Alice* during interval i are transmitted by *Bob* in interval $i+1$; the maximum possible delay experienced by a given frame is from the beginning of i to the end of $i+1$, or twice d. Similarly, the minimum possible delay experienced is from the end of i to the beginning of $i+1$, which is zero. More generally, the maximum delay experienced by a given frame is

$$(h+1) \times d$$

and the minimum delay experienced by a given frame is

$$(h-1) \times d$$

where h is the number of hops.

This illustrates the attraction of CQF as a technique for handling time-sensitive traffic; the latency introduced as a frame transits the network is completely described by the cycle time and the number of hops, and is unaffected by any other topology considerations, including interference from other non time-sensitive traffic. This only holds, however, if frames are kept to their allotted cycles; if, for example, some of the frames that were expected to be received by *Bob* during cycle i do not appear until cycle $i+1$ has started, then the stated assumptions about maximum latency calculation no longer hold. Careful choice of cycle times, alignment of cycle times among the Bridges in the network, and the timing of first and last transmissions within a cycle are required in order to ensure that the desired latency bounds are achieved.

Any delays through a particular intermediate relay (for example, *Bob*) do not affect the end-to-end delay so long as *Bob's* performance does not affect the correct assignment of frames to time intervals.

Since one of the goals for the handling of time-sensitive streams is zero frame loss (assuming that no unrecognizable non-conformant traffic is present), it is prudent to assume that reception is continuous—a frame received by a downstream system will always be assigned to one interval or another. This places most of the burden of correct interval assignment on the transmitting system; frames should not be transmitted if incorrect interval assignment is possible upon reception. It is therefore necessary to define the anticipated (and accommodated) errors in reception assignment with respect to the point in interval time, t,

[4]In early discussions, CQF was known as the "Peristaltic Shaper" [B52].

where interval i–1 becomes interval i. A relay (such as *Bob*) can of course choose when to start reception assignment to i in relation to t; it is assumed that *Bob's* intent is that the earliest frame to be assigned to i is the first whose very last octet (or other frame transmission encoding symbol) is still on the transmission medium (or other definable external event to what is considered to be *Bob* reference point) at t, thus placing any accommodation of known implementation dependent delays within *Bob* under *Bob's* control.

While *Bob* attempts to start i reception with a frame coming off the medium at i, and may factor known and repeatable internal delays into the way he goes about that intent, his actual start time depends on:

a) The error in *Bob's* time sync (i.e., the error in his determination as to when t actually occurs).
b) The maximum deviation (jitter) in *Bob's* use of that time.
c) Additional delays that *Bob* does not account for, such as delays in selecting the output queue to be used for i.

Alice has to stop transmitting frames for i–1 before t, by a time that is the sum of *Bob's* possible early start of i as a consequence of a) through c), and the following:

d) The error in *Alice's* time sync (i.e., the error in her determination as to when t occurs).
e) The maximum deviation (jitter) in *Alice's* use of that time.
f) The time between *Alice* deciding to commit a frame for transmission and the appearance of the last octet/symbol "on the medium" at *Alice's* end.
g) The length of "the medium" in transmission time, i.e., the time for the last octet/symbol to leave *Alice* and reach *Bob*, including any consideration of the effect of interfering frames or fragments.

The description of CQF in terms of a number of consecutive intervals (as opposed to their support by "odd/even" queues, as discussed in T.2 onwards) gives easy answers to what to do with traffic still queued when its selected transmission interval has expired—discard it, or mark it down (discard eligible or priority change) and generate an alarm. In an environment where the stream bandwidth is allocated appropriately (i.e., the bandwidth allocated per time interval is less than can be received/transmitted in the chosen interval duration), this will be a rare occurrence, the traffic that follows will be conformant, and the overall system performance will be recoverable.

The discussion so far has assumed that all link speeds are the same; however, the situation becomes more complicated when links of different speeds are considered. One typical arrangement might comprise low speed links at the start and end of the path (network periphery to periphery), another with the high speed towards one end (periphery to core or vice versa). Taking the first of these, and placing *Alice* at the first transition from slow to fast, *Bob* as her fast neighbor, *Charlie* as his fast neighbor, and *Donald* at the transition from fast to slow, the important thing (treating the fast core of the network as a CQF black box) is that all conformant traffic received by *Alice* in interval i (say) is transmitted by *Donald* in a later interval $i+n$. A number of internal arrangements might be made between *Alice*, *Bob*, *Charlie*, and *Donald* to make this happen and would be valid from an external CQF perspective. It is also possible to consider fractional n, where n is still > 1, as *Alice* may need to collect the entirety of any slow cycle before transmitting that in a more compressed burst into the rest of the fast network. More complex possibilities are equivalent to redefining the slow cycle time. Some of the less elaborate possibilities for the use of links of different speeds are discussed in T.5.

T.2 An approach to CQF implementation

In essence, the approach involves the use of two transmission queues and a cycle timer. During even numbered cycles (intervals), queue 1 accumulates received frames from the Bridge's reception Ports (and does not transmit them), while queue 2 transmits any queued frames from the previous odd-numbered cycle (and does not receive any frames). During odd-numbered cycles, queue 2 accumulates received frames from the Bridge's reception Ports (and does not transmit them), while queue 1 transmits any queued frames from the previous even-numbered cycle (and does not receive any frames). With appropriate choice of receive and transmit cycle times (see T.5), such that, for any given stream, the cycle is at least long enough to accommodate all of the time-sensitive traffic that will need to be transmitted on the Bridge Port during the class measurement interval for that stream (see 34.6.1, also known as the observation interval in IEEE Std 802.1BA™ [B51]), plus a maximum-sized interfering frame (or frame fragment, if preemption is supported), then all of the stream's traffic will be accumulated during the cycle time in queues that are in receive mode, and it will all be transmitted during the cycle time when the queues switch to transmit mode.

CQF is implemented by configuring a combination of the stream gate control mechanisms defined for per-stream filtering and policing (PSFP, IEEE Std 802.1Qci) and the traffic scheduling mechanisms defined in 8.6.8.4 and 8.6.9 of IEEE Std 802.1Qbv. 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

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) and a stream gate instance (8.6.5.1.2) 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.

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_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; however, these could be added as appropriate, for example if the maximum SDU size for the time-sensitive traffic is bounded at a value less than the maximum SDU size for the medium.

This stream filter specification results in all frames that carry a priority value of 3 being submitted to the stream gate. As the operation of PSFP is such that received frames that do not match a stream filter are handled as if PSFP is not implemented, there is no need for further stream filter 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) needed to support the stream filter described in T.3.1 has a *stream gate control list* that contains two entries, each containing a SetGateAndIPS 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

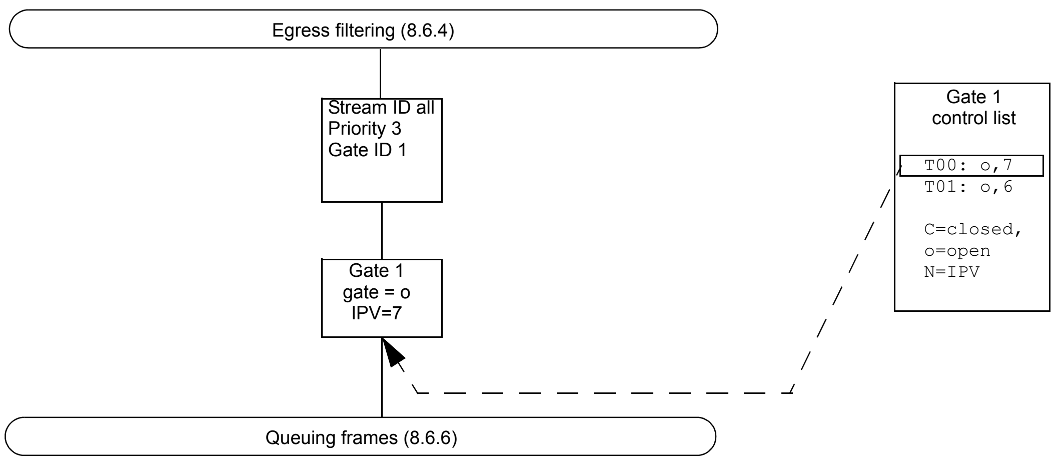

Figure T-1—Example PSFP configuration for QCF

T.4 Use of traffic scheduling for CQF

The traffic scheduling support needed on each outbound Port in order to support the PSFP configuration described in T.3 is to execute a gate control list that will set the GateState to *open* for queue 6 and *closed* for queue 7 for a TimeInterval of T, and then set the GateState to *open* for queue 7 and *closed* for queue 6 for a TimeInterval of T, repeating ad infinitum. If there are no other traffic scheduling considerations, this can be achieved with a gate control list that contains just two SetGateStates gate operations, with parameters as follows:

1) GateState: 0, 1, 2, 3, 4, 5, 6 *open*, 7 *closed*, TimeInterval = T
2) GateState: 0, 1, 2, 3, 4, 5, 7 *open*, 6 *closed*, TimeInterval = T

This sequence of gate operations has the effect that during the initial time period T, the GateState for queue 7 is closed while queue 7 is being filled, and queue 6 is open to allow any queued frames to be transmitted; during the second time period T, the GateState for queue 6 is closed while queue 6 is being filled, and queue 7 is open to allow any queued frames to be transmitted. The gates for all other queues are open. The choice of time interval T is discussed in T.5; the cycle time (OperCycleTime; see 8.6.9.4.20) for the scheduled traffic state machines would be set to 2T in order to accommodate the sum of the time intervals for the two gate operations.

If there are traffic scheduling requirements for any of the other queues, then the gate control list could be extended to accommodate those requirements; however, the time interval between the changes of state of the gates for queues 6 and 7 has to be T, and consequently, OperCycleTime has to be a multiple of 2T, in order for the CQF requirements to be met. Figure T-2 illustrates the simplest possible traffic scheduling configuration for the case that traffic scheduling is only needed to support CQF.

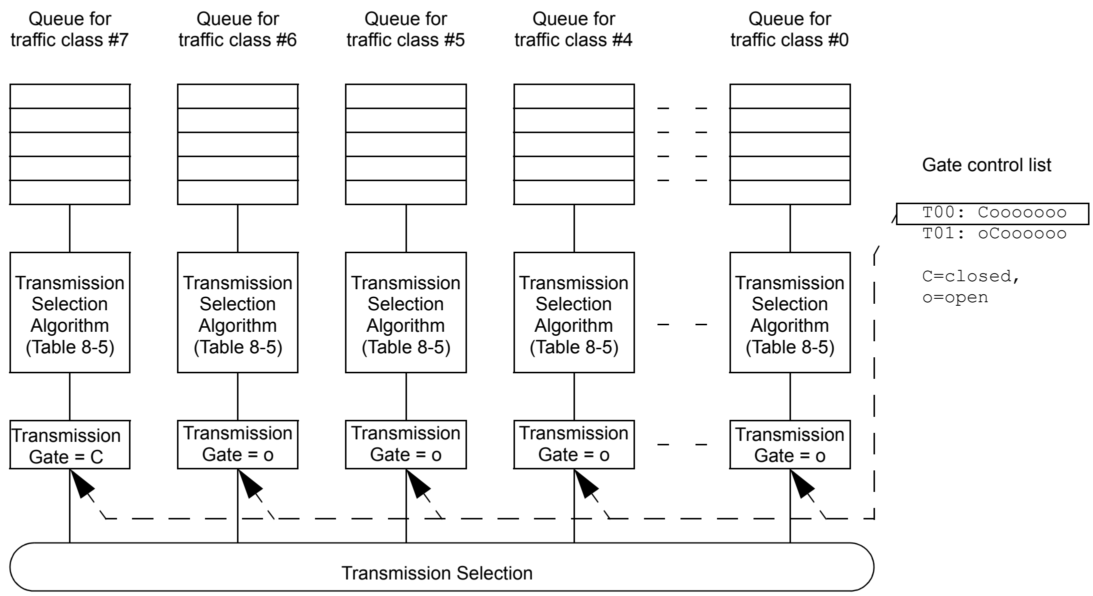

Figure T-2—Traffic scheduling example for CQF

T.5 Timing considerations

T.5.1 Choice of T

T should be chosen such that it is large enough to accommodate the stream data that can be received during the class measurement interval for the stream(s) concerned, plus at least one maximal interfering frame or frame fragment. This is important in order to ensure the key performance aspect associated with the class measurement interval; namely, that if a stream or set of streams is observed over time, the reserved data rate for that stream or set of streams will not be exceeded during any observed time period equal to the class measurement interval associated with those streams. This effectively places a lower bound on the choice of T, that it should not be smaller than the class measurement interval, and also places a restriction on larger values of T, that they should be integer multiples of the class measurement interval.

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-3 and Figure T-4 illustrate how the PSFP and traffic scheduling could be configured in the case where SR classes A and B are active; in Figure T-3, 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.

The traffic schedule is based on the smaller of the two class measurement intervals, 125 µs, but now has four entries in the gate control list (as opposed to 2 entries in Figure T-2), giving an overall cycle time of 500 µs. The gate control list switches the gate states for traffic classes 7 and 6 every 125 µs, and switches the gate states for traffic classes 5 and 4 every 250 µs.

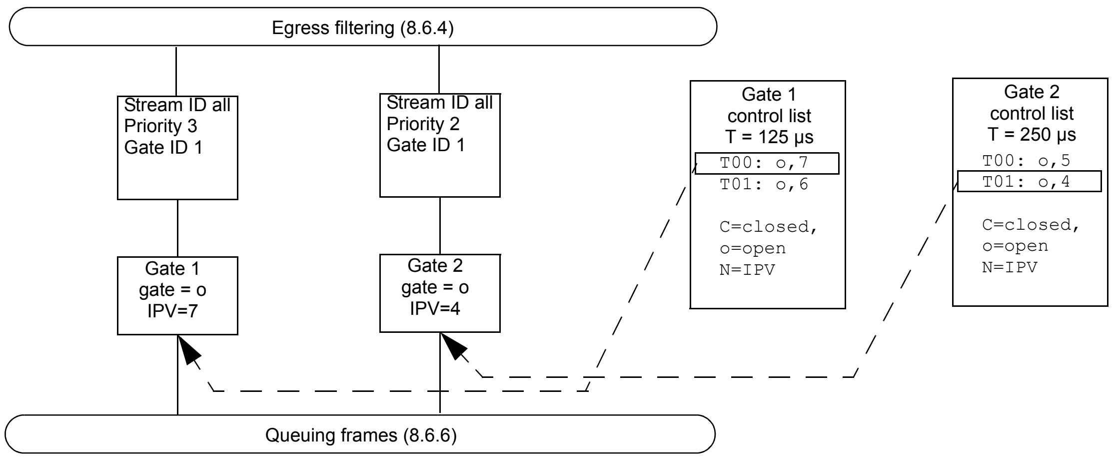

Figure T-3—Example PSFP configuration with two values of T

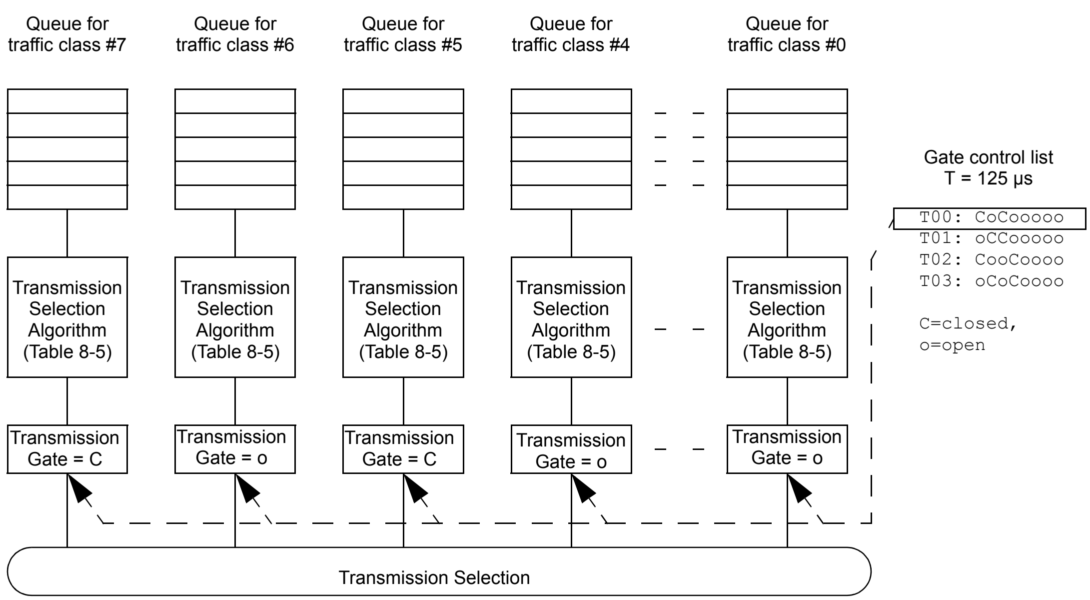

Figure T-4—Traffic scheduling example with two values of T

T.5.2 Cycle interleaving

In some circumstances, particularly where the data rates differ between reception and transmission Ports, it can be desirable to interleave cycles on the faster Port so that the best use is made of the higher bandwidth available, and also to reduce the latency that is added as a stream passes through faster parts of the transmission path. Because there is a delay imposed on the transmission of received frames, caused by the cyclic switching of reception and transmission between a pair of queues, when a queue is allowed to transmit, all of its received frames will have been enqueued, and all of them will therefore be transmitted in a burst, assuming that priorities permit, and that the transmission queue uses the strict priority transmission selection algorithm (8.6.8.1). Hence, if the received traffic from a given Port was spread out over the time interval T, and it is all sent to the same queue, the transmitted traffic will be compressed into a burst. If the transmission data rate is, say, ten times the reception data rate, then the maximum length of that burst is T/10, so there is the potential to fit 9 more such bursts into the bandwidth available on that transmission Port. With appropriate timing on reception Ports and transmission Ports, the reception and transmission cycles can be interleaved such that those additional transmission bursts can occur. In the example illustrated in Figure T-5, it is assumed that:

a) There are two Ports on which stream data is being received, Rx1 and Rx2.
b) There is a single Port on which stream data is being transmitted, Tx1.
c) Rx1 and Rx2 operate at half of the data rate of Tx1 (or less).
d) All stream traffic is SR Class A, and is received with priority 3.

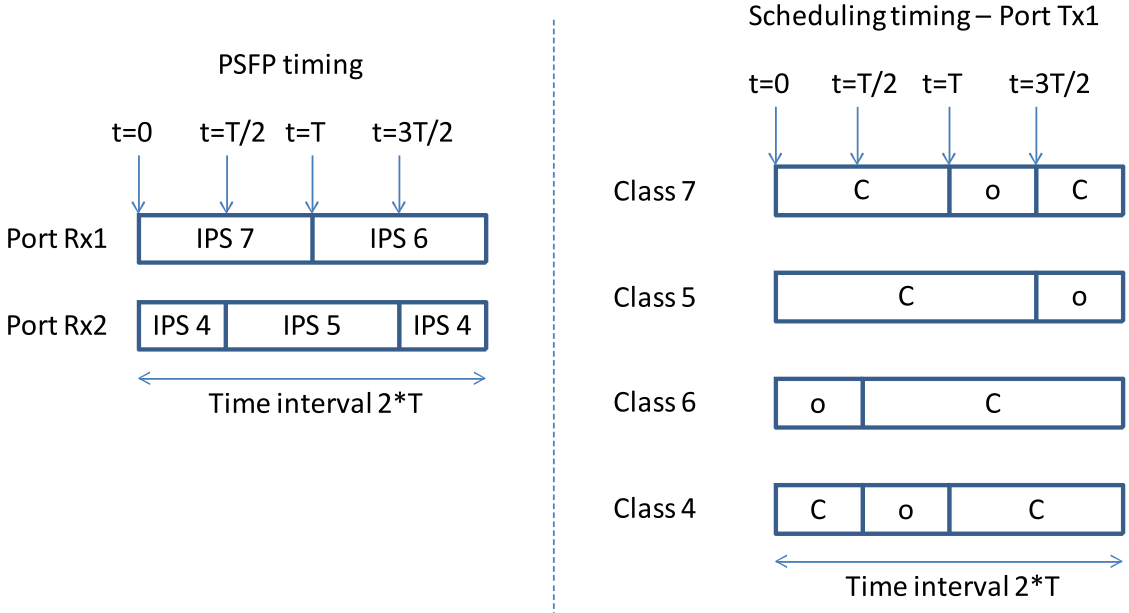

Figure T-5—Interleaving example—factor of 2

On the reception side, traffic received on Rx1 with priority 3 is sent to traffic class queue 7 during the odd cycles and to traffic class queue 6 during the even cycles. Similarly, traffic received on Rx2 with priority 3 is sent to traffic class queue 5 during the odd cycles and to traffic class queue 4 during the even cycles; however, these cycles are offset with respect to the cycles for Rx1 by T/2.

On the transmission side, each queue is in the open (transmitting) state for T/2, and in the closed (receiving) state for 3T/2. The timings of the gate open/gate closed events are arranged so that only one queue is transmitting at any one time.

It should be noted that with a scheme like this, although the transmit side appears to be operating on the basis of a value of T that is half that used on the Rx side, the effect from the point of view of the streams originating from a given reception Port is that the Rx and Tx timing is the same, as frames received on that Port are transmitted once in every time period T, and as assumed in the preceding point c), the transmit rate is twice the receive rate (or more), so the available transmit bandwidth is the same as or more than the receive bandwidth. The interleaving therefore meets the requirements stated in T.5.1. The reception Port of the Bridge downstream of Tx1 can operate using T/2, and if it is possible to carry this through to the transmission Ports as well, the contribution to the latency for streams passing through this Bridge will be T/2, rather than the T contributed by the first Bridge.

More complex schemes can be envisaged; for example, using more than two Rx Ports, and these can be made to work as long as the received bandwidth can be shared equally between the pairs of traffic classes that are used. Interleaving of this kind can also be defined for larger interleave factors; the only limitation is the number of available outbound queues. It is also possible to define interleaving schemes where the received bandwidth is not shared equally among the pairs of traffic classes, as long as the bandwidth allocated to a given pair of traffic classes does not exceed the bandwidth available during the time intervals when those traffic classes are able to transmit.

NOTE—Although the number of traffic classes described in this standard is limited to 8, the value of the IPV does not have such a limitation placed upon it. If a system supported more than 8 traffic classes, it would therefore be possible to define interleaving factors greater than 4.

T.5.3 Cycle alignment between adjacent Ports

The examples so far assume a perfect world where the transmission from transmission Port to reception Port is instantaneous and the internal timings of the transmitting and receiving systems are perfectly synchronized. In reality, transmission takes time, and synchronization is not perfect; therefore, it would be possible that a transmission Port launches the last frame(s) of one transmission burst just after the reception Port downstream has switched reception and transmission queues, which would mean that those last frames are placed in the wrong queue. Similarly, if the timing misalignment worked the other way, it would be possible for the transmission Port to finish transmitting its burst early, and switch to transmitting the next burst before the downstream Port had changed state.

In order to avoid this problem, the timings must be adjusted such that there is a very high degree of probability that when a reception Port changes the state of the stream filters to direct incoming frames to a different outbound queue, there are no frames still to be transmitted, or in flight, from the upstream transmission Port. This can be achieved by slightly delaying the start of the transmission window, and slightly advancing the end of the transmission window. The value of “slightly” depends on a number of factors, including the following:

a) Any error in the time synchronization between the adjacent systems.
b) Jitter in the propagation time of a frame from starting to leave the transmit queue in the upstream system to being presented to the downstream policing function.
c) Jitter in the propagation time of a frame between the downstream policing function and the appropriate transmission queue in the downstream system.
d) The size of any potential interfering frame or frame fragment.
e) Difference in the resolution of the clocks that are maintained by adjacent systems.

The effect of this adjustment factor, S, on the timings shown on the transmission Port in the earlier examples would be that the time slots where the gate is in the “open” state would be shorter by a factor of 2S, and would start S later. Hence, the transmission phase for traffic class 7 in Figure T-5 would start at T+S, and would end at (3T/2)–S. S should be set to the sum of the errors or jitter values from all sources given in the preceding list.

Annex U

(informative)

Bibliography

Insert the following in alphanumeric sequence, renumbering as appropriate:

[B51] IEEE Std 802.1BA™, IEEE Standard for Local and metropolitan area networks—Audio Video Bridging (AVB) Systems.

[B52] Peristaltic Shaper: updates, multiple speeds, Michael Johas Teener, 23 January 2014. http://www.ieee802.org/1/files/public/docs2014/new-tsn-mjt-peristaltic-shaper-0114.pdf.

Consensus

WE BUILD IT.

IEEE
standards.ieee.org
Phone: +1 732 981 0060 Fax: +1 732 562 1571