IEEE STANDARDS ASSOCIATION IEEE

IEEE Standard for
Local and metropolitan area networks—

Bridges and Bridged Networks—
Amendment 26: Frame Preemption

IEEE Computer Society

Sponsored by the
LAN/MAN Standards Committee

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

IEEE Std 802.1Qbu™-2016
(Amendment to
IEEE Std 802.1Q™-2014)

IEEE Std 802.1Qbu™-2016
(Amendment to
IEEE Std 802.1Q™-2014)

IEEE Standard for
Local and metropolitan area networks—

Bridges and Bridged Networks—
Amendment 26: Frame Preemption

Sponsor

LAN/MAN Standards Committee
of the
IEEE Computer Society

Approved 30 June 2016
IEEE-SA Standards Board

Abstract: Enhancements to the forwarding process that support frame preemption are provided in this amendment to IEEE Std 802.1Q-2014.

Keywords: Bridged Local Area Networks, IEEE 802.1Q™, LANs, local area networks, MAC Bridges, metropolitan area networks, preemption, Virtual Bridged Local Area Networks, virtual LANs

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

PDF: ISBN 978-1-5044-2257-4 STD21074
Print: ISBN 978-1-5044-2258-1 STDPD21074

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 Notice" or "Important Notices and Disclaimers Concerning IEEE Standards Documents."

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. Volunteers 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 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/browse/standards/collection/ieee 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 this standard was completed, the IEEE 802.1 working group had the following membership:

Glenn Parsons, *Chair*
John Messenger, *Vice-Chair*
Michael Johas Teener, *Chair, Time Sensitive Networking Task Group*
Tony Jeffree, *Editor*

Christian Boiger
Paul Bottorff
David Chen
Feng Chen
Weiying Cheng
Rodney Cummings
Janos Farkas
Norman Finn
Geoffrey Garner
Eric Gray
Craig Gunther
Stephen Haddock
Mark Hantel
Marc Holness
Hal Keen
Stephan Kehrer
Marcel Kiessling
Philippe Klein
Jouni Korhonen
Yizhou Li
Christophe Mangin
Tom McBeath
James McIntosh
Hiroki Nakano
Bob Noseworthy
Donald R. Pannell
Walter Pienciak
Karen Randall
Maximilian Riegel
Dan Romascanu
Jessy Rouyer
Panagiotis Saltsidis
Michael Seaman
Daniel Sexton
Johannes Specht
Wilfried Steiner
Patricia Thaler
David Thornburg
Jeremy Touve
Paul Unbehagen
Karl Weber
Brian Weis
Jordon Woods
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
Lee Armstrong
Stefan Aust
Christian Boiger
Nancy Bravin
William Byrd
Juan Carreon
Rodney Cummings
Janos Farkas
Yukihiro Fujimoto
David Gregson
Randall Groves
Stephen Haddock
Marek Hajduczenia
Jerome Henry
Marco Hernandez
Guido Hiertz
Werner Hoelzl
C. Huntley
Noriyuki Ikeuchi
Atsushi Ito
Tony Jeffree
Michael Johas Teener
Adri Jovin
Shinkyo Kaku
Piotr Karocki
Stuart Kerry
Yongbum Kim
Robert Landman
Mark Laubach
David Lewis
Arthur H. Light
William Lumpkins
Michael Lynch
Elvis Maculuba
Arthur Marris
Jonathon Mclendon
Richard Mellitz
Charles Moorwood
Michael Newman
Nick S.A. Nikjoo
Satoshi Obara
Alon Regev
Robert Robinson
Benjamin Rolfe
Dan Romascanu
Jessy Rouyer
Larry Samberg
Bartien Sayogo
Michael Seaman
David Solomon
Kevin Stanton
Thomas Starai
Eugene Stoudenmire
Walter Struppler
Michael Swearingen
Patricia Thaler
Mark-Rene Uchida
Lorenzo Vangelista
Dmitri Varsanofiev
George Vlantis
Khurram Waheed
Stephen Webb
Karl Weber
Hung-Yu Wei
Natalie Wienckowski
Oren Yuen
Zhen Zhou

When the IEEE-SA Standards Board approved this standard on 30 June 2016, it had the following membership:

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

Chuck Adams
Masayuki Ariyoshi
Stephen Dukes
Jianbin Fan
J. Travis Griffith
Gary Hoffman
Ronald W. Hotchkiss
Michael Janezic
Joseph L. Koepfinger*
Hung Ling
Kevin Lu
Annette D. Reilly
Gary Robinson
Mehmet Ulema
Yingli Wen
Howard Wolfman
Don Wright
Yu Yuan
Daidi Zhong

*Member Emeritus

Introduction

This introduction is not part of IEEE Std 802.1Qbu-2016, IEEE Standard for Local and metropolitan area networks—Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks—Amendment 26: Frame Preemption.

This amendment to IEEE Std 802.1Q-2014 provides enhancements to the forwarding process that support frame preemption.

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

Tables

IEEE Standard for
Local and metropolitan area networks—

Bridges and Bridged Networks—

Amendment 26: Frame Preemption

(This amendment is based on IEEE Std 802.1Q™-2014 as amended by IEEE Std 802.1Qbv-2015.)

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]

IMPORTANT NOTICE: IEEE Standards documents are not intended to ensure safety, health, or environmental protection, or ensure against interference with or from other devices or networks. Implementers 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.

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

[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 reference in the appropriate collating sequence:

IEEE Std 802.3br™, IEEE Approved Draft Standard for Ethernet—Amendment 5: Specification and Management Parameters for Interspersing Express Traffic.

3. Definitions

Insert the following definition in the appropriate collating sequence and renumber appropriately:

3.x preemption: The suspension of the transmission of a preemptable frame to allow one or more express frames to be transmitted before transmission of the preemptable frame is resumed.

4. Abbreviations

Insert the following definitions in the appropriate collating sequence and renumber appropriately:

eMAC	express Media Access Control
pMAC	preemptable Media Access Control

5. Conformance

5.4 VLAN Bridge component requirements

5.4.1 VLAN Bridge component options

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

ae) Support frame preemption as specified in 6.7.1, 6.7.2, and 8.6.8.

5.13 MAC Bridge component requirements

5.13.1 MAC Bridge component options

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

n) Support frame preemption as specified in 6.7.1, 6.7.2, and 8.6.8.

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

5.26 End station requirements—enhancements for frame preemption

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

a) Support the provisions of 6.7.1, 6.7.2, and 8.6.8.

6. Support of the MAC Service

6.7 Support of the Internal Sublayer Service by specific MAC procedures

6.7.1 Support of the Internal Sublayer Service by IEEE Std 802.3 (Ethernet)

Change the first paragraph of 6.7.1 as follows:

In addition to the provisions of 12.1.1 of IEEE Std 802.1AC-2012, an M_CONTROL.request primitive is mapped to an IEEE 802.3 MA_CONTROL.request primitive having the same parameters. If the MAC supports IEEE 802.3br Interspersing Express Traffic, then PFC M_CONTROL.requests are mapped onto the MAC control interface associated with the express MAC (eMAC). An IEEE 802.3 MA_CONTROL.indication primitive is mapped to an M_CONTROL.indication primitive having the same parameters.

Insert new text at the end of 6.7.1 as follows:

If frame preemption (6.7.2) is supported on a Port, then the IEEE 802.3 MAC provides the following two MAC service interfaces (99.4 of IEEE Std 802.3br):

a) A preemptable MAC (pMAC) service interface, and
b) An express MAC (eMAC) service interface.

For priority values that are identified in the frame preemption status table (6.7.2) as *preemptable*, frames that are selected for transmission shall be transmitted using the pMAC service instance, and for priority values that are identified in the frame preemption status table as *express*, frames that are selected for transmission shall be transmitted using the eMAC service instance.

In all other respects, the Port behaves as if it is supported by a single MAC service interface. In particular, all frames received by the Port are treated as if they were received on a single MAC service interface regardless of whether they were received on the eMAC service interface or the pMAC service interface, except with respect to frame preemption.

If the value of the holdRequest managed object (12.30.1.5) transitions from FALSE to TRUE, a MM_CTL.request(hold_req) primitive is issued to the underlying IEEE 802.3 MAC, with a hold_req parameter value of HOLD, as described in Clause 99 of IEEE Std 802.3br. If the value of the holdRequest managed object (12.30.1.5) transitions from TRUE to FALSE, a MM_CTL.request(hold_req) primitive is issued to the underlying IEEE 802.3 MAC, with a hold_req parameter value of RELEASE.

NOTE—This additional material will be moved to IEEE Std 802.1AC in a future revision.

Insert new subclause 6.7.2 as follows:

6.7.2 Frame preemption

If the Port supports frame preemption, then a value of frame preemption status is assigned to each value of priority via a *frame preemption status table*. The possible values of frame preemption status are *express* or *preemptable*.

The frame preemption status table can be changed by management as described in 12.30.1.1. The default value of frame preemption status is *express* for all priority values.

8. Principles of bridge operation

8.6 The Forwarding Process

8.6.8 Transmission selection

Insert the following immediately after NOTE 2:

In a port of a Bridge or station that supports frame preemption, a frame of priority *n* is not available for transmission if that priority is identified in the frame preemption status table (6.7.2) as preemptable and either the holdRequest object (12.30.1.5) is set to the value *hold*, or the transmission of a prior preemptable frame has yet to complete because it has been interrupted to allow the transmission of an express frame.

8.6.8.2 Credit-based shaper algorithm

Change the text of list items e) and f), and insert new NOTEs, renumbering subsequent NOTEs, as follows:

e) ***transmit***. Takes the value TRUE for the duration of a frame transmission from the queue; FALSE when any frame transmission from the queue has completed. If the credit-based shaper algorithm is used in combination with frame preemption (6.7.2), transmit only takes the value TRUE while the frame is actually being transmitted by the MAC. If the frame transmission is delayed or interrupted (e.g., the frame is a preemptable frame and its transmission is interrupted to allow the transmission of an express frame from a different queue, or the frame is an express frame and there is a delay before transmission can start because a preemptable frame was being transmitted) transmit takes the value FALSE until transmission of the frame commences or is resumed. Transmit also takes the value FALSE during the transmission of any overhead that is a consequence of frame preemption; i.e., any additional frame overhead that is added to the preemptable frame when preemption occurs.

NOTE 1—The consequence of this is that any overhead associated with preemption does not come out of the reserved bandwidth for the credit-based shaper.

f) ***credit.*** The transmission credit, in bits, that is currently available to the queue. If, at any time, there are no frames in the queue, and the *transmit* parameter is FALSE, and the transmission gate for the queue is open (8.6.8.4), and *credit* is positive, and there is no preemptable frame from this queue for which transmission is in progress but has been interrupted, then *credit* is set to zero.

NOTE 2—The condition that the *transmit* parameter is FALSE and a preemptable frame or part-frame is waiting in the MAC for transmission can only occur if the credit-based shaper algorithm is used in combination with Preemption.

8.6.8.4 Enhancements for scheduled traffic

Change the second paragraph of 8.6.8.4 as follows:[2]

A *gate control list* associated with each Port contains an ordered list of gate operations. Each gate operation changes the transmission gate state for the gate associated with each of the Port's traffic class queues and allows associated control operations to be scheduled. In an implementation that does not support enhancements for scheduled traffic, all gates are assumed to be permanently in the *open* state. Table 8-6 identifies the gate operation types, their parameters, and the actions that result from their execution. The state machines that control the execution of the gate control list, along with their variables and procedures, are specified in 8.6.9.

[2] The base text of 8.6.8.4 is to be found in IEEE Std 802.1Qbv-2016.

Insert two new rows at the end of Table 8-6 as follows:

Table 8-6—Gate operations

Operation name	Parameter(s)	Action
Set-And-Hold-MAC	GateState, TimeInterval	Performs all of the actions defined for the SetGateStates operation;[1] in addition, the start of this operation marks the point in the sequence of gate operations at which the MAC associated with the port is to have stopped transmitting preemptable frames. This is achieved by setting the holdRequest managed object to the value *hold (1)*, at holdAdvance (Table 12-29) nanoseconds in advance of this point for the hold to have taken effect at this point. If frame preemption is not supported or not enabled (preemptionActive is FALSE), this operation behaves the same as SetGateStates.
Set-And-Release-MAC	GateState, TimeInterval	Performs all of the actions defined for the SetGateStates operation; in addition, the start of this operation marks the point in the sequence of gate operations at which the MAC associated with the port is permitted to resume transmitting preemptable frames; if an express frame is currently being transmitted by the MAC, the release takes effect at the end of that transmission. This is achieved by setting the holdRequest managed object to the value *release (2)*, at releaseAdvance (Table 12-29) nanoseconds in advance of this point for the release to have taken effect at this point.[2] If frame preemption is not supported or not enabled (preemptionActive is FALSE), this operation behaves the same as SetGateStates.

[1] The SetGateStates operation is defined in IEEE Std 802.1Qbv-2016.

[2] The releaseAdvance parameter allows the timing of when the release command is issued to vary depending upon the constraints of a particular implementation. Its value should be less than the minimum frame size so that release does not occur too early and interfere with transmission of the last express frame.

12. Bridge management

Insert new subclause 12.30 and its subclauses and tables, as follows, renumbering as necessary:

12.30 Managed objects for frame preemption

The Bridge enhancements for support of frame preemption are defined in 8.6.8, 8.6.8.4, and 6.7.2. The objects that comprise this managed resource are as follows:

a) Frame Preemption Parameter Table (12.30.1)

12.30.1 Frame Preemption Parameter table

There is one Frame Preemption Parameter table per Port of a Bridge component or end station. Each table row contains a set of parameters that supports the enhancements for frame preemption (6.7.2), as detailed in Table 12-29. Rows in the table can be created or removed dynamically in implementations that support dynamic configuration of ports and components.

Table 12-29—Frame Preemption Parameter table

Name	Data type	Operations supported[a]	Conformance[b]	References
framePreemptionStatusTable	sequence of framePreemptionAdminStatus values	RW	BE	6.7.2, 12.30.1.1, 12.30.1.1.1.
holdAdvance	Integer, nanoseconds	R	BE	Table 8-6, 12.30.1.2
releaseAdvance	Integer, nanoseconds	R	BE	Table 8-6, 12.30.1.3
preemptionActive	Boolean	R	BE	12.30.1.4
holdRequest	Interger {hold (1), release (2)}	R	BE	Table 8-6, 12.30.1.5

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

[b] B = Required for Bridge or Bridge component support of enhancements for frame preemption.
E = Required for end station support of enhancements for frame preemption.

12.30.1.1 framePreemptionStatusTable structure and data types

The framePreemptionStatusTable (6.7.2) consists of 8 framePreemptionAdminStatus values (12.30.1.1.1), one per priority.

12.30.1.1.1 framePreemptionAdminStatus

This parameter is the administrative value of the preemption status for the priority. It takes value *express* if frames queued for the priority are to be transmitted using the express service for the Port, or *preemptable* if frames queued for the priority are to be transmitted using the preemptable service for the Port and preemption is enabled for the Port.

Priorities that all map to the same traffic class should be constrained to use the same value of preemption status.

12.30.1.2 holdAdvance object

The holdAdvance object contains an integer value representing the maximum number of nanoseconds that can elapse between issuing a HOLD (12.30.1.5) to the MAC and the MAC ceasing to transmit any preemptable frame that is in the process of transmission or any preemptable frames that are queued for transmission, including any MAC-specific delay before transmission of an express frame could start once preemptable frame transmission has ceased. This object exists per Port, and is a characteristic of the underlying MAC.

12.30.1.3 releaseAdvance object

The releaseAdvance object contains an integer value representing the maximum number of nanoseconds that can elapse between issuing a RELEASE (12.30.1.5) to the MAC and the MAC being ready to resume transmission of preemptable frames, in the absence of there being any express frames available for transmission. This object exists per Port, and is a characteristic of the underlying MAC.

12.30.1.4 preemptionActive object

TRUE if preemption is both supported by the MAC and currently active.

12.30.1.5 holdRequest object

The holdRequest object contains an enumerated integer value, with *hold (1)* indicating that a Set-And-Hold-MAC gate operation (8.6.8.4, Table 8-6) has been executed, and *release (2)* indicating that a Set-And-Release-MAC gate operation has been executed. This object exists per Port.

NOTE—In order to determine support of frame preemption by the Bridge, a network management application could attempt to read the preemptionActive object. If the read operation returns an error, frame preemption is not supported. Determining whether the MAC supports preemption could similarly be achieved by examining the objects in the MAC MIB.

17. Management Information Base (MIB)

17.2 Structure of the MIB

Insert the following new subclause 17.2.23 and Table 17-29 at the end of 17.2, renumbering as necessary:

17.2.23 Structure of the IEEE8021-Preemption-MIB

The IEEE8021-Preemption-MIB provides for configuration of frame preemption (6.7.2, 8.6.8) on ports. Table 17-29 indicates the relationship between the SMIv2 objects defined in the MIB module (17.7.23) and managed objects defined in 12.30.

Table 17-29—IEEE8021-Preemption-MIB Structure and relationship to this standard

MIB table	MIB object	Reference
ieee8021PreemptionParameterTable subtree		
ieee8021PreemptionParameterTable		Frame Preemption parameter table, 6.7.2, 12.30.1
	ieee8021PreemptionPriority	Priority (Table index)
	ieee8021FramePreemptionAdminStatus	framePreemptionAdminStatus, 6.7.2, 12.30.1
ieee8021PreemptionConfigTable		Frame Preemption configuration table, 6.7.2, 12.30.1
	ieee8021FramePreemptionHoldAdvance	framePreemptionHoldAdvance, 6.7.2, 12.30.1, 12.30.1.2
	ieee8021FramePreemptionReleaseAdvance	framePreemptionReleaseAdvance, 6.7.2, 12.30.1, 12.30.1.3
	ieee8021FramePreemptionActive	framePreemptionActive, 6.7.2, 12.30.1, 12.30.1.4
	ieee8021FramePreemptionHoldRequest	framePreemptionHoldRequest, 6.7.2, 12.30.1, 12.30.1.5

17.3 Relationship to other MIBs

Insert the following new subclause 17.3.23 at the end of 17.3, renumbering as necessary:

17.3.23 Relationship of the IEEE8021-Preemption-MIB to other MIBs

The IEEE8021-Preemption-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 frame preemption extensions are supported by the Bridge. As support of the objects defined in the IEEE8021-Preemption-MIB also requires support of the IEEE8021-BRIDGE-MIB, the provisions of 17.3.2 apply to implementations claiming support of the IEEE8021-Preemption-MIB.

17.4 Security considerations

Insert the following new subclause 17.4.23 at the end of 17.4, renumbering as necessary:

17.4.23 Security considerations of the IEEE8021-ST-MIB

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

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

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

ieee8021FramePreemptionAdminStatus

a) Misconfiguration of the ieee8021FramePreemptionAdminStatus object can lead to the degradation of the quality of service for the application on the port at the respective traffic class that is wrongly preempted, or to the saturation of the priority queues on the port because of preemptable traffic classes receiving express processing.

17.7 MIB modules

Replace subclause 17.7.22 with the following:

17.7.22 Definitions for the IEEE8021-ST-MIB module

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

-- ================================================================
-- MIB for support of the Scheduled Traffic Enhancements
-- for IEEE 802.1Q Bridges.
-- ================================================================


IMPORTS
    MODULE-IDENTITY,
    OBJECT-TYPE,
    Unsigned32,
    Counter64
        FROM SNMPv2-SMI
    TEXTUAL-CONVENTION,
    TruthValue
        FROM SNMPv2-TC
    MODULE-COMPLIANCE,
    OBJECT-GROUP
        FROM SNMPv2-CONF
    ieee802dot1mibs
        FROM IEEE8021-TC-MIB
    ieee8021BridgeBaseComponentId,
    ieee8021BridgeBasePort
        FROM IEEE8021-BRIDGE-MIB
    ;

ieee8021STMib MODULE-IDENTITY
    LAST-UPDATED "201608150000Z" -- August 15, 2016
    ORGANIZATION "IEEE 802.1 Working Group"
    CONTACT-INFO
        "  WG-URL: www.ieee802.org/1
         WG-EMail: STDS-802-1-L@listserv.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-L@listserv.ieee.org"
    DESCRIPTION
        "The Bridge MIB module for managing devices that support
        the Scheduled Traffic Enhancements
        for 802.1Q Bridges.

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

        Copyright (C) IEEE (2014).
        This version of this MIB module is part of IEEE802.1Q;
        see the draft itself for full legal notices."
    REVISION "201608150000Z" -- August 15, 2016
    DESCRIPTION
            "Revised to include Set-And-Hold-MAC and
            Set-And-Release-MAC in the description of
            ieee8021STAdminControlList and
            ieee8021STOperControlList.
            Published as part of IEEE Std 802.1Qbu."

    REVISION "201509250000Z" -- September 25, 2015
    DESCRIPTION
            "Initial version published as part of IEEE Std 802.1Qbv."


       ::= { ieee802dot1mibs 30 }

-- =============================================================
-- Textual Conventions
-- =============================================================

IEEE8021STTrafficClassValue ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "d"
    STATUS      current
    DESCRIPTION
        "A traffic class value.
        This is the numerical value associated with a traffic
        class in a Bridge. Larger values are associated with
        higher priority traffic classes."
    REFERENCE   "12.29.1"
    SYNTAX      Unsigned32 (0..7)

IEEE8021STPTPtimeValue ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
        "A PTPtime value, represented as a 48-bit unsigned integer
        number of seconds and a 32-bit unsigned integer number of
        nanoseconds.
        The first 6 octets represent the number of seconds: the
        first octet is the most significant
        octet of the 48-bit seconds value and the sixth octet
        is the least significant octet of the seconds value.
        The remaining octets, 7 through 10,  represent the
        number of nanoseconds: the seventh octet
        is the most significant octet of the 32-bit nanoseconds
        value and the tenth octet is the
        least significant octet of the nanoseconds value."
    REFERENCE   "8.6.8.4, 8.6.9.4, 12.29.1"
    SYNTAX      OCTET STRING (SIZE(10))


-- =============================================================
-- subtrees in the ST MIB
-- =============================================================

ieee8021STNotifications
    OBJECT IDENTIFIER ::= { ieee8021STMib 0 }

ieee8021STObjects
    OBJECT IDENTIFIER ::= { ieee8021STMib 1 }
```

```
ieee8021STConformance
    OBJECT IDENTIFIER ::= { ieee8021STMib 2 }

ieee8021STMaxSDUSubtree
    OBJECT IDENTIFIER ::= { ieee8021STObjects 1 }

ieee8021STParameters
    OBJECT IDENTIFIER ::= { ieee8021STObjects 2 }

-- =============================================================
-- The ieee8021STMaxSDUSubtree subtree
-- This subtree defines the objects necessary for the management
-- of the max SDU size parameters for each traffic class on a Port.
-- =============================================================

-- =============================================================
-- the ieee8021STMaxSDUTable
-- =============================================================

ieee8021STMaxSDUTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021STMaxSDUEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table containing a set of max SDU
        parameters, one for each traffic class.
        All writeable objects in this table must be
        persistent over power up restart/reboot."
    REFERENCE   "8.6.8.4, 8.6.9.4, 12.29.1"
    ::= { ieee8021STMaxSDUSubtree 1 }

ieee8021STMaxSDUEntry OBJECT-TYPE
    SYNTAX      Ieee8021STMaxSDUEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A list of objects containing Max SDU size
        for each traffic class supported by the Port."
    INDEX  { ieee8021BridgeBaseComponentId,
             ieee8021BridgeBasePort,
             ieee8021STTrafficClass  }
    ::= { ieee8021STMaxSDUTable 1 }

Ieee8021STMaxSDUEntry ::=
    SEQUENCE {
        ieee8021STTrafficClass
            IEEE8021STTrafficClassValue,
        ieee8021STMaxSDU
            Unsigned32,
        iee8021TransmissionOverrun
            Counter64
        }

ieee8021STTrafficClass OBJECT-TYPE
    SYNTAX      IEEE8021STTrafficClassValue
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The traffic class number associated with the row of
        the table.

        A row in this table is created for each traffic class
        that is supported by the Port"
    REFERENCE   "8.6.8.4, 8.6.9.4, 12.29.1"
    ::= { ieee8021STMaxSDUEntry 1 }


ieee8021STMaxSDU OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "octets"
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The value of the MaxSDU parameter for the traffic class.
        This value is represented as an unsigned integer. A value
        of 0 is interpreted as the max SDU size supported by
        the underlying MAC.

        The default value of the MaxSDU parameter is 0.

        The value of this object MUST be retained across
        reinitializations of the management system."
    REFERENCE   "8.6.8.4, 8.6.9.4, 12.29.1"
    DEFVAL { 0 }
    ::= { ieee8021STMaxSDUEntry 2}


ieee8021TransmissionOverrun OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "A counter of transmission overrun events, where
        a PDU is still being transmitted by a MAC at the
        time when the transmission gate for the queue closed."
    REFERENCE   "8.6.8.4, 8.6.9.4, 12.29.1, 12.29.1.1.2"
    DEFVAL { 0 }
    ::= { ieee8021STMaxSDUEntry 2}


-- ================================================================
-- The ieee8021STParameters subtree
-- This subtree defines the objects necessary for the management
-- of the traffic scheduling mechanism for IEEE Std 802.1Q.
-- ================================================================

-- ================================================================
-- the ieee8021STParametersTable
-- ================================================================

ieee8021STParametersTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021STParametersEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table that contains the per-port manageable parameters for
        traffic scheduling.
        For a given Port, a row in the table exists.

        All writable objects in this table must be
        persistent over power up restart/reboot."
    REFERENCE   "8.6.8.4, 8.6.9.4, 12.29.1"
    ::= { ieee8021STParameters 1 }

ieee8021STParametersEntry OBJECT-TYPE
    SYNTAX      Ieee8021STParametersEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A list of objects that contains the manageable parameters for
        traffic scheduling for a port."
    INDEX  { ieee8021BridgeBaseComponentId,
             ieee8021BridgeBasePort
           }
    ::= { ieee8021STParametersTable 1 }

Ieee8021STParametersEntry ::=
    SEQUENCE {
        ieee8021STGateEnabled
            TruthValue,
        ieee8021STAdminGateStates
            OCTET STRING,
        ieee8021STOperGateStates
            OCTET STRING,
        ieee8021STAdminControlListLength
            Unsigned32,
        ieee8021STOperControlListLength
            Unsigned32,
        ieee8021STAdminControlList
            OCTET STRING,
        ieee8021STOperControlList
            OCTET STRING,
        ieee8021STAdminCycleTimeNumerator
            Unsigned32,
        ieee8021STAdminCycleTimeDenominator
            Unsigned32,
        ieee8021STOperCycleTimeNumerator
            Unsigned32,
        ieee8021STOperCycleTimeDenominator
            Unsigned32,
        ieee8021STAdminCycleTimeExtension
            Unsigned32,
        ieee8021STOperCycleTimeExtension
            Unsigned32,
        ieee8021STAdminBaseTime
            IEEE8021STPTPtimeValue,
        ieee8021STOperBaseTime
            IEEE8021STPTPtimeValue,
        ieee8021STConfigChange
            TruthValue,
        ieee8021STConfigChangeTime
            IEEE8021STPTPtimeValue,
        ieee8021STTickGranularity
            Unsigned32,
        ieee8021STCurrentTime
            IEEE8021STPTPtimeValue,
        ieee8021STConfigPending
            TruthValue,
        ieee8021STConfigChangeError
            Counter64,
        ieee8021STSupportedListMax
            Unsigned32
            }

ieee8021STGateEnabled OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The GateEnabled parameter determines whether traffic scheduling
        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, 12.29.1"
    DEFVAL { false }
    ::= { ieee8021STParametersEntry 1 }

ieee8021STAdminGateStates OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE(1))
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The administrative value of the GateStates parameter for the Port.
        The bits of the octet represent the gate states for the
       corresponding traffic classes; the MS bit corresponds to traffic class 7,
        the LS bit to traffic class 0. A bit value of 0 indicates closed; a
        bit value of 1 indicates open.

        The value of this object MUST be retained across
        reinitializations of the management system."
    REFERENCE   "8.6.8.4, 8.6.9.4, 12.29.1"
    ::= { ieee8021STParametersEntry 2 }

ieee8021STOperGateStates OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE(1))
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The operational value of the GateStates parameter for the Port.
        The bits of the octet represent the gate states for the
       corresponding traffic classes; the MS bit corresponds to traffic class 7,
        the LS bit to traffic class 0. A bit value of 0 indicates closed; a
        bit value of 1 indicates open."
    REFERENCE   "8.6.8.4, 8.6.9.4, 12.29.1"
    ::= { ieee8021STParametersEntry 3 }

ieee8021STAdminControlListLength OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The administrative value of the ListMax parameter for the Port.
        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, 12.29.1"
    ::= { ieee8021STParametersEntry 4 }

ieee8021STOperControlListLength OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The operational value of the ListMax parameter for the Port.
        The integer value indicates the number of entries (TLVs) in the
        OperControlList."
    REFERENCE   "8.6.8.4, 8.6.9.4, 12.29.1"
    ::= { ieee8021STParametersEntry 5 }

ieee8021STAdminControlList OBJECT-TYPE
    SYNTAX      OCTET STRING
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The administrative value of the ControlList parameter for the Port.
        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: SetGateStates
            1: Set-And-Hold-MAC
            2: Set-And-Release-MAC
            3-255: 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 8-6. Two parameter types are currently defined:

        - GateState:
            A GateState parameter is encoded in a single octet.
            The bits of the octet represent the gate states for the
            corresponding traffic classes; the MS bit corresponds
            to traffic class 7,
            the LS bit to traffic class 0. A bit value of 0 indicates
            closed; a bit value of 1 indicates open.

        - 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.

        The value of this object MUST be retained across
        reinitializations of the management system."
    REFERENCE   "8.6.8.4, 8.6.9.4, 12.29.1"
    ::= { ieee8021STParametersEntry 6 }

ieee8021STOperControlList OBJECT-TYPE
    SYNTAX      OCTET STRING
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The operational value of the ListMax parameter for the Port.
        The octet string value represents the contents of the control list as
        an ordered list of TLVs, as follows.
        The first octet  of each TLV is interpreted as a gate operation name:
            0: SetGateStates
            1: Set-And-Hold-MAC
            2: Set-And-Release-MAC
            3-255: 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 8-6. Two parameter types are currently defined:

        - GateState:
            A GateState parameter is encoded in a single octet.
            The bits of the octet represent the gate states for the
            corresponding traffic classes; the MS bit corresponds to
            traffic class 7, the LS bit to traffic class 0.
            A bit value of 0 indicates closed; a
            bit value of 1 indicates open.

        - 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."
    REFERENCE   "8.6.8.4, 8.6.9.4, 12.29.1"
    ::= { ieee8021STParametersEntry 7 }

ieee8021STAdminCycleTimeNumerator OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The administrative value of the numerator of the CycleTime
        parameter for the Port.
        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, 12.29.1"
    ::= { ieee8021STParametersEntry 8 }

ieee8021STAdminCycleTimeDenominator OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The administrative value of the denominator of the
        CycleTime parameter for the Port.
        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, 12.29.1"
    ::= { ieee8021STParametersEntry 9 }

ieee8021STOperCycleTimeNumerator OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The operational value of the numerator of the
        CycleTime parameter for the Port.
        The numerator and denominator together represent the cycle
        time as a rational number of seconds."
    REFERENCE   "8.6.8.4, 8.6.9.4, 12.29.1"
    ::= { ieee8021STParametersEntry 10 }

ieee8021STOperCycleTimeDenominator OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The operational value of the denominator of the
        CycleTime parameter for the Port.
        The numerator and denominator together represent the
        cycle time as a rational number of seconds."
    REFERENCE   "8.6.8.4, 8.6.9.4, 12.29.1"
    ::= { ieee8021STParametersEntry 11 }

ieee8021STAdminCycleTimeExtension OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "nanoseconds"
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The administrative value of the CycleTimeExtension
        parameter for the Port.
        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, 12.29.1"
    ::= { ieee8021STParametersEntry 12 }

ieee8021STOperCycleTimeExtension OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "nanoseconds"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The operational value of the CycleTimeExtension parameter for the Port.
        The value is an unsigned integer number of nanoseconds."
    REFERENCE   "8.6.8.4, 8.6.9.4, 12.29.1"
    ::= { ieee8021STParametersEntry 13 }

ieee8021STAdminBaseTime OBJECT-TYPE
    SYNTAX      IEEE8021STPTPtimeValue
    UNITS       "PTP time"
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The administrative value of the BaseTime parameter for the Port.
        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, 12.29.1"
    ::= { ieee8021STParametersEntry 14 }

ieee8021STOperBaseTime OBJECT-TYPE
    SYNTAX      IEEE8021STPTPtimeValue
    UNITS       "PTP time"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The operationsl value of the BaseTime parameter for the Port.
        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, 12.29.1"
    ::= { ieee8021STParametersEntry 15 }

ieee8021STConfigChange OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The ConfigChange parameter signals the start of a
        configuration change
        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, 12.29.1"
    ::= { ieee8021STParametersEntry 16 }

ieee8021STConfigChangeTime 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, 12.29.1"
    ::= { ieee8021STParametersEntry 17 }

    ieee8021STTickGranularity 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, 12.29.1"
    ::= { ieee8021STParametersEntry 18 }

 ieee8021STCurrentTime 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, 12.29.1"
    ::= { ieee8021STParametersEntry 19 }

 ieee8021STConfigPending 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, 12.29.1"
    ::= { ieee8021STParametersEntry 20 }

 ieee8021STConfigChangeError 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, 12.29.1"
    ::= { ieee8021STParametersEntry 21 }

ieee8021STSupportedListMax OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The maximum value supported by this Port of the
        AdminControlListLength and OperControlListLength
        parameters."
    REFERENCE   "12.29.1.5"
    ::= { ieee8021STParametersEntry 22 }


-- =============================================================
-- IEEE8021 FQTSS MIB - Conformance Information
-- =============================================================

ieee8021STCompliances
    OBJECT IDENTIFIER ::= { ieee8021STConformance 1 }
ieee8021STGroups
    OBJECT IDENTIFIER ::= { ieee8021STConformance 2 }

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

-- =============================================================
-- the ieee8021STObjectsGroup group
-- =============================================================

ieee8021STObjectsGroup OBJECT-GROUP
    OBJECTS {
        ieee8021STMaxSDU,
        ieee8021TransmissionOverrun,
        ieee8021STGateEnabled,
        ieee8021STAdminGateStates,
        ieee8021STOperGateStates,
        ieee8021STAdminControlListLength,
        ieee8021STOperControlListLength,
        ieee8021STAdminControlList,
        ieee8021STOperControlList,
        ieee8021STAdminCycleTimeNumerator,
        ieee8021STAdminCycleTimeDenominator,
        ieee8021STOperCycleTimeNumerator,
        ieee8021STOperCycleTimeDenominator,
        ieee8021STAdminCycleTimeExtension,
        ieee8021STOperCycleTimeExtension,
        ieee8021STAdminBaseTime,
        ieee8021STOperBaseTime,
        ieee8021STConfigChange,
        ieee8021STConfigChangeTime,
        ieee8021STTickGranularity,
        ieee8021STCurrentTime,
        ieee8021STConfigPending,
        ieee8021STConfigChangeError,
        ieee8021STSupportedListMax
    }
    STATUS      current
    DESCRIPTION
        "Objects that allow management of scheduled traffic."
    ::= { ieee8021STGroups 1 }


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

ieee8021STCompliance MODULE-COMPLIANCE
    STATUS      current
    DESCRIPTION
        "The compliance statement for devices supporting
        scheduled traffic.

        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 {
            ieee8021STObjectsGroup
        }

    ::= { ieee8021STCompliances 1 }

END
```

Insert the following new subclause 17.7.23 at the end of 17.7, renumbering as necessary:

17.7.23 Definitions for the IEEE8021-Preemption-MIB module

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

-- ===============================================================
-- MIB for support of the frame preemption enhancements
-- for IEEE 802.1Q Bridges.
-- ===============================================================

IMPORTS
    MODULE-IDENTITY,
    OBJECT-TYPE,
    Unsigned32
        FROM SNMPv2-SMI
    MODULE-COMPLIANCE,
    OBJECT-GROUP
        FROM SNMPv2-CONF
    ieee802dot1mibs,
    IEEE8021PriorityValue
        FROM IEEE8021-TC-MIB
    ieee8021BridgeBaseComponentId,
    ieee8021BridgeBasePort
        FROM IEEE8021-BRIDGE-MIB
    ;

ieee8021PreemptionMib MODULE-IDENTITY
    LAST-UPDATED "201608150000Z" -- August 15, 2016
    ORGANIZATION "IEEE 802.1 Working Group"
    CONTACT-INFO
        "  WG-URL: www.ieee802.org/1
         WG-EMail: STDS-802-1-L@listserv.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-L@listserv.ieee.org"
    DESCRIPTION
        "The Bridge MIB module for managing devices that support
        the frame preemption enhancements
        for 802.1Q Bridges.

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

        Copyright (C) IEEE (2014).
        This version of this MIB module is part of IEEE802.1Q;
        see the draft itself for full legal notices."

    REVISION "201608150000Z" -- August 15, 2016
    DESCRIPTION
            "Initial version published as part of IEEE Std 802.1Qbu."
        ::= { ieee802dot1mibs 29 }


-- =============================================================
-- subtrees in the Preemption MIB
-- =============================================================

ieee8021PreemptionNotifications
    OBJECT IDENTIFIER ::= { ieee8021PreemptionMib 0 }

ieee8021PreemptionObjects
    OBJECT IDENTIFIER ::= { ieee8021PreemptionMib 1 }

ieee8021PreemptionConformance
    OBJECT IDENTIFIER ::= { ieee8021PreemptionMib 2 }

ieee8021PreemptionParameters
    OBJECT IDENTIFIER ::= { ieee8021PreemptionObjects 1 }

-- =============================================================
-- The ieee8021PreemptionNotifications subtree
-- This subtree defines any notifications necessary for the management
-- of frame preemption on a Port. This subtree is currently unused,
-- but is retained as a place-holder for future standardisation.
-- =============================================================


-- =============================================================
-- The ieee8021PreemptionParameters subtree
-- This subtree defines the objects necessary for the management
-- of the frame preemption parameters for each priority value
-- on a Port.
-- =============================================================

-- =============================================================
-- the ieee8021PreemptionParameterTable
-- =============================================================

ieee8021PreemptionParameterTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021PreemptionParameterEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table containing a set of frame preemption
        parameters, one for each priority value.
        All writeable objects in this table must be
        persistent over power up restart/reboot."
    REFERENCE   "6.7.2, 12.30.1"
    ::= { ieee8021PreemptionParameters 1 }

ieee8021PreemptionParameterEntry OBJECT-TYPE
    SYNTAX      Ieee8021PreemptionParameterEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A list of objects containing preemption parameters
        for each  priority value."
    INDEX  { ieee8021BridgeBaseComponentId,
             ieee8021BridgeBasePort,
             ieee8021PreemptionPriority  }
    ::= { ieee8021PreemptionParameterTable 1 }

Ieee8021PreemptionParameterEntry ::=
    SEQUENCE {
        ieee8021PreemptionPriority
            IEEE8021PriorityValue,
        ieee8021FramePreemptionAdminStatus
            INTEGER
        }

ieee8021PreemptionPriority OBJECT-TYPE
    SYNTAX      IEEE8021PriorityValue
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The priority number associated with the row of
        the table.

        A row in this table is created for each priority value."
    REFERENCE   "6.7.2, 12.30.1"
    ::= { ieee8021PreemptionParameterEntry 1 }


ieee8021FramePreemptionAdminStatus OBJECT-TYPE
    SYNTAX      INTEGER {express (1), preemptable (2)}
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The value of the framePreemptionAdminStatus parameter
        for the traffic class.

        The default value of the framePreemptionAdminStatus parameter
        is express (1).

        The value of this object MUST be retained across
        reinitializations of the management system."
    REFERENCE   "6.7.2, 12.30.1"
    ::= { ieee8021PreemptionParameterEntry 2}


-- =============================================================
-- the ieee8021PreemptionConfigTable
-- =============================================================

ieee8021PreemptionConfigTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021PreemptionConfigEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table containing a set of frame preemption
        parameters, one for each Port.
        All writeable objects in this table must be
        persistent over power up restart/reboot."
    REFERENCE   "6.7.2, 12.30.1"
    ::= { ieee8021PreemptionParameters 2 }

ieee8021PreemptionConfigEntry OBJECT-TYPE
    SYNTAX      Ieee8021PreemptionConfigEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A list of objects containing preemption parameters
        for each Port."
    INDEX  { ieee8021BridgeBaseComponentId,
             ieee8021BridgeBasePort }
    ::= { ieee8021PreemptionConfigTable 1 }

Ieee8021PreemptionConfigEntry ::=
    SEQUENCE {
        ieee8021FramePreemptionHoldAdvance
            Unsigned32,
        ieee8021FramePreemptionReleaseAdvance
            Unsigned32,
        ieee8021FramePreemptionActive
            INTEGER,
        ieee8021FramePreemptionHoldRequest
            INTEGER
        }

ieee8021FramePreemptionHoldAdvance OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The value of the holdAdvance parameter
        for the Port in nanoseconds.

        There is no default value; the holdAdvance is
        a property of the underlying MAC."
    REFERENCE   "6.7.2, 12.30.1.2"
    ::= { ieee8021PreemptionConfigEntry 1 }


ieee8021FramePreemptionReleaseAdvance OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The value of the releaseAdvance parameter
        for the Port in nanoseconds.

        There is no default value; the releaseAdvance is
        a property of the underlying MAC."
    REFERENCE   "6.7.2, 12.30.1.3"
    ::= { ieee8021PreemptionConfigEntry 2}

ieee8021FramePreemptionActive OBJECT-TYPE
    SYNTAX      INTEGER {idle (1), active (2)}
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The value is active (2) when preemption is operationally
        active for the Port, and idle (1) otherwise."
    REFERENCE   "6.7.2, 12.30.1.4"
    ::= { ieee8021PreemptionConfigEntry 3}

ieee8021FramePreemptionHoldRequest OBJECT-TYPE
```

```
    SYNTAX      INTEGER {hold (1), release (2)}
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The value is hold (1) when the sequence of gate operations
        for the Port has executed a Set-And-Hold-MAC operation,
        and release (2) when the sequence of gate operations has
        executed a Set-And-Release-MAC operation. The
        value of this object is release (2) on system
        initialization."
    REFERENCE   "6.7.2, Table 8-6, 12.30.1.5"
    ::= { ieee8021PreemptionConfigEntry 4}


-- =============================================================
-- IEEE8021 Preemption MIB - Conformance Information
-- =============================================================

ieee8021PreemptionCompliances
    OBJECT IDENTIFIER ::= { ieee8021PreemptionConformance 1 }
ieee8021PreemptionGroups
    OBJECT IDENTIFIER ::= { ieee8021PreemptionConformance 2 }

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

-- =============================================================
-- the ieee8021PreemptionGroup group
-- =============================================================

ieee8021PreemptionGroup OBJECT-GROUP
    OBJECTS {
        ieee8021FramePreemptionAdminStatus,
        ieee8021FramePreemptionHoldAdvance,
        ieee8021FramePreemptionReleaseAdvance,
        ieee8021FramePreemptionActive,
        ieee8021FramePreemptionHoldRequest
    }
    STATUS      current
    DESCRIPTION
        "Objects that allow management of frame preemption."
    ::= { ieee8021PreemptionGroups 1 }


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

ieee8021PreemptionCompliance MODULE-COMPLIANCE
    STATUS      current
    DESCRIPTION
        "The compliance statement for devices supporting
        frame preemption.

        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 {
            ieee8021PreemptionGroup
        }

    ::= { ieee8021PreemptionCompliances 1 }

END
```

Annex A

(normative)

PICS proforma—Bridge implementations[4]

A.5 Major capabilities

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

PRE	Does the implementation support frame preemption?	O	5.4.1, 5.13.1, 6.7.2, 8.6.8, 12.30, 17.7.23	Yes [ ]	No [ ]

A.14 Bridge management

Insert the following row at the end of Table A.14, renumbering if necessary:

Item	Feature	Status	References	Support	
MGT-221	Does the implementation support the management entities defined in 12.30?	PRE: M	5.4.1 item ae), 12.30	Yes [ ]	N/A [ ]

A.24 Management Information Base (MIB)

Insert the following row at the end of Table A.24, renumbering if necessary:

Item	Feature	Status	References	Support	
MIB-42	Is the IEEE8021-Preemption-MIB module fully supported (per its MODULE-COMPLIANCE)?	PRE: O	5.4.1 item ae), 12.30, 17.7.23	Yes [ ]	N/A [ ]

[4] *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 new Table A.44 at the end of Annex A, renumbering as necessary:

A.44 Frame preemption

Item	Feature	Status	References	Support
	If frame preemption (PRE in Table A.5) is not supported, mark N/A and ignore the remainder of this table.		5.4.1, 5.13.1, 6.7.2, 8.6.8, 12.30, 17.7.23	N/A[ ]
PRE1	Does the implementation support the functionality of frame preemption as specified in 6.7.2 and 8.6.8?	PRE: M	5.4.1, 5.13.1, 6.7.2, 8.6.8	Yes [ ] N/A [ ]

Annex B

(normative)

PICS proforma—End station implementations[5]

B.5 Major capabilities

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

PRE	Does the implementation support frame preemption?	O	5.4.1, 5.13.1, 6.7.2, 8.6.8, 12.30, 17.7.23	Yes [ ]	No [ ]

Insert the following new Table B.15 at the end of Annex B, renumbering as necessary:

B.15 Scheduled traffic

Item	Feature	Status	References	Support
	If frame preemption (PRE in Table B.5) is not supported, mark N/A and ignore the remainder of this table.		5.4.1, 5.13.1, 6.7.2, 8.6.8, 12.30, 17.7.23	N/A[ ]
PRE1	Does the implementation support the functionality of frame preemption as specified in 6.7.2 and 8.6.8?	PRE: M	5.4.1, 5.13.1, 6.7.2, 8.6.8	Yes [ ] N/A [ ]
PRE2	Does the implementation support the management entities defined in 12.30?	PRE: M	5.4.1 item ae), 12.30	Yes [ ] N/A [ ]
PRE3	Is the IEEE8021-Preemption-MIB module fully supported (per its MODULE-COMPLIANCE)?	PRE: O	5.4.1 item ae), 12.30, 17.7.23	Yes [ ] N/A [ ]

[5] *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 new Annex Q, as follows, renumbering as appropriate:

Annex Q

(informative)

Preemption and IEEE Std 802.1AE MAC Security

This annex answers questions about the use of MAC Security (MACsec) that arose in the development of the IEEE Std 802.1Qbu (Frame Preemption) amendment to this standard. It provides background for readers not familiar with MACsec. For the normative specification of MACsec and its supporting authentication and key management framework refer to IEEE Std 802.1AE and IEEE Std 802.1X.

NOTE—While these issues arose during development of preemption, they can also apply to other situations, such as a customer link running MACSec over a provider bridged network.

Preemption allows one or more higher priority (express) frames to interrupt the transmission of a lower priority (preemptable) frame, the preemptable frame transmission being resumed and completed once the express frame(s) have been transmitted. A preemptable frame can thus be broken into two or more fragments. A MACsec Cipher Suite implementation can protect each frame in its entirety before requesting its transmission, or can transmit the octets of the protected frame as soon as they are made available by the operation of the block by block symmetric cryptography. The latter is often preferred as it reduces the delay due to protection, and some implementations impose a constant delay in support of time-sensitive networking objectives. Similarly validation of a received frame can be delayed until its reception is complete, or can proceed as each block of octets becomes available. These implementation details lie outside the normative scope of IEEE Std 802.1AE, but naturally give rise to questions as to how frames should be protected when they are fragmented, particularly when the symmetric keys used to protect frames are being changed.

Each time a Cipher Suite's symmetric cryptography is used to protect a frame with a given key, a unique value, known as a 'nonce' or IV (initial value), has to be included in the computation. A PN (packet number) that is incremented for each frame transmitted forms part of the MACsec IV. A replayProtect control with a configurable replay window allows the receiver to use the PN to protect against misordered, replayed, and much delayed frames. A frame whose PN value is less than the highest received with a successfully validated frame (from the same source, protected by the same key) minus the replay window size is discarded. Setting the replay window size to zero and enabling replayProtect enforces strict replay protection, no out of order or replayed frames are accepted. The transmitter of each MACsec frame is identified by a Secure Channel Identifier (SCI), and the Secure Channel (SC) that allows it to communicate securely with its peers in a secure Connectivity Association (CA) is supported by a sequence of Secure Associations (SAs). Each SA is replaced by its successor when its associated secret Secure Association Key (SAK) can no longer be used, either because all or most of the PN space has been used with one of the SAs in the CA, or because the network administrator has a security policy that mandates regular key changes (often once a week). New SAs and SAKs are also required after reauthentication and reauthorization, when the (possibly changed) members of a CA acquire a fresh secure Connectivity Association Key (CAK), the shared secret key whose possession permits participation in the CA and that is used by the MAC Security Key Agreement protocol (MKA) to identify the CA members to each other and to distribute SAKs.

MACsec connectivity is not interrupted or delayed by SA and SAK changes. The SCI and the Association Number (AN), a two-bit circular sequence number that identifies the SA within the SC, are communicated by the MAC Security TAG (SecTAG) and form part of the IV. Their values thus have to be decided and encoded in the frame to be transmitted prior to encipherment. The MACsec operational and management models reflect the fact that, at any instant, the frame whose SecTAG is currently being encoded can be using a different SA and SAK than the frame currently being cryptographically processed. This accommodates implementations that buffer user transmit requests for performance or implementation reasons (e.g., to allow parallel processing). A transmit SA/SAK is not deleted until sometime after it has ceased to be used. Similarly an implementation that is enciphering a preemptable frame block by block can suspend its cryptographic operation, encipher an express frame that has been assigned to a subsequent SA, and resume processing the preempted frame using the IV already encoded, the SAK identified by the SCI and AN for that IV, and the intermediate results produced by the Cipher Suite up to the point that the frame was preempted. The Integrity Check Value (ICV) generated, and appended to the MACsec protected frame, protects the entire frame. An attacker cannot omit a fragment or substitute one for another. The ICV also has exactly the same value that would be obtained by protecting the entire frame prior to transmitting any of its octets, which is important for reasons that go beyond facilitating interoperability between different styles of implementation. While MACsec protects communication between neighboring peer users of the MAC Service, those service instances can be supported by bridging at a lower layer. Communication between two C-VLAN Bridges can be supported by individual LANs connected by TPMRs, or by S-VLAN Bridges, for example. The receiving MAC Security Entity (SecY) that validates a preemptable frame can receive it in fragments that differ from those transmitted by the port and SecY that originally protected the frame, and is not necessarily aware that the received frame was fragmented en-route. The first of two TPMRs supporting a link between two C-VLAN Bridges (for example) might reassemble all received frames prior to forwarding, thus allowing express frames to overtake preemptable frames on the link connecting the TPMRs without fragmenting the latter.

Just as the use of an SA can be overlapped with the use of its successor on transmission, each receive SA and its successor are enabled for reception for 3 seconds after an MKA Key Server tells the other member(s) of the CA to start transmitting using a new SAK and SA(s). Thus the receipt of an express frame protected with a new SAK does not preclude the receipt and validation of a preempted frame, protected with the prior SAK, after its final fragment (including the protected frame's ICV) has been received.

IEEE Std 802.1AE describes the use of MACsec in a number of different systems, interface stacks, and interworking scenarios. When MACsec is used to secure connectivity across a provider network, the protected frame can be priority-tagged (6.13) so the provider can forward frames of different priorities appropriately, even if the contents of those frames is otherwise confidential, and they can be received and validated out of order just as they might be if some frames had been preempted en-route. The receiving SecY's replay window size can be managed to accommodate these frames, though it is not always easy to determine or predict an optimal value for the size of replay window that is needed. MACsec's strict replay protection can also be retained by assigning transmitted frames for different service access priorities (or for express and preemptable frames) to different SCs, each supported by its own sequence of SAs. From the point of view of a receiving SecY, the separate SCs behave just as if they had been transmitted by separate SecYs. The IEEE P802.1AEcg[6] amendment allows the choice of SCI and access priority[7] to be determined by the frame's user priority.[8] (See Seaman [B1].)

[6] See http://standards.ieee.org/develop/project/802.1AEcg.html.

[7] The priority value accompanying the service request made at the SecY's lower Common Port to transmit the protected frame.

[8] The priority value provided by the service request, to protect and transmit the frame, made at the SecY's upper Controlled Port.

Insert the following new Annex R, renumbering as appropriate:

Annex R

(informative)

Preemption and scheduled traffic

The enhancements for scheduled traffic specified in 8.6.8.4 and the support for frame preemption specified in 6.7.2 and 8.6.8 can be used individually or in combination in order to provide different degrees of protection to "time-sensitive traffic" from interference caused by traffic with less stringent timing requirements. The choice of which features to use, and in what combination, will depend upon the latency and jitter requirements of the application concerned. There are four permutations of interest. They are as follows:

a) Scheduling used in isolation (R.1)
b) Preemption used in isolation (R.2)
c) Scheduling and preemption used in combination, but without use of HOLD and RELEASE (R.3)
d) Scheduling and preemption used in combination, with use of HOLD and RELEASE (R.4)

NOTE—As the only MAC standard for preemption at the time of writing is IEEE Std 802.3br, this annex has been written to reflect the constraints that exist in that particular approach to preemption.

R.1 Scheduling used in isolation

This subject is dealt with in Annex P.

R.2 Preemption used in isolation

Preemption allows one or more higher priority (express) frames to interrupt the transmission of a lower priority (preemptable) frame, the preemptable frame transmission being resumed and completed once the express frame(s) have been transmitted.

The ability to designate one or more priority as express (and the remaining priorities as preemptable) means that, when an express frame is ready for transmission and a preemptable frame is already in the process of being transmitted, the delay before the express frame transmission starts is at worst 123 octet times and will often be 64 octet times or less. Preemption, used without scheduling, is therefore a useful tool for reducing the jitter experienced by time-sensitive frames. These frames could be transmitted using any of the available transmission selection algorithms (Table 8-5), but the likely choice would be an algorithm that is intended for use with time-sensitive data, such as the credit-based shaper (8.6.8.2). In common use cases, it would be expected that express traffic would have a higher priority than preemptable traffic.

NOTE 1—IEEE Std 802.3br defines a minimum final fragment size of 64 octets and allows the receiver to request that any non-final fragments are larger than this by 0, 64, 128, or 192 octets. Therefore, the minimum non-final fragment size is 64, 128, 192, or 256 octets, depending upon the non-final fragment size stipulated by the receiving MAC. For the purposes of the discussion in this annex, it is assumed that the minimum non-final fragment size selected is 64 octets.

NOTE 2—The 123 octet times worst-case delay is because non-final fragments have to be 64 octets in length, including the mCRC that is added by the MAC Merge sublayer, so a fragment that is 124 (60 + 64) octets in length can be preempted, but a fragment that is 123 or fewer octets in length cannot be preempted. This occurs when transmission has just started for a fragment of a frame with less than 124 octets left to transmit, or for a preemptable frame of less than

124 octets in total. The more likely case is that the delay would be between 0 and 64 octet times. The actual delay experienced also has to take account of the extra overheads imposed by the need for preamble, start of frame delimiter (SFD), inter-frame gap (IFG), and the latency of the MAC itself.

It should be noted that, other things being equal, designating a priority as "express" effectively increases its priority above that of any priority designated as "preemptable". Also, in cases where the number of traffic classes supported on a port is less than 8, all priority values that map to the same traffic class should have the same designation (all express or all preemptable).

If preemptable and express priorities are assigned to the same traffic class, an express frame might not be able to preempt a preemptable frame that is ahead of it in the queue.

R.3 Scheduling and preemption used in combination, no HOLD/RELEASE

The discussion of scheduling in Annex P points out that there is no need for an explicitly scheduled "guard band" when using the scheduling mechanism in isolation, because the specification of the operation of the transmission gates is such that, in a conformant implementation, a frame cannot start transmission if the transmission gate for its traffic class is due to close before transmission of the frame would complete. However, the requirement not to transmit a frame if transmission cannot complete before the gate closes means that, depending upon the size of frames in the queue, there would be a significant length of time during which frames cannot be transmitted.

If the schedule that applies to the preemptable traffic classes was designed to allow preemptable traffic to be transmitted at any time (i.e., there is no "gate close" event for the preemptable traffic), then preemptable frame transmission can occur at any time up to the start of an express transmission, and depending upon the timings, can continue up to 123 octet times after the express traffic gate opens. This dramatically reduces the delay experienced by preemptable traffic, at the expense of increasing the delay experienced by express traffic. While that may be acceptable for some applications, it is clearly not acceptable for applications where it is desired to reduce or remove any delay experienced by express frames.

R.4 Scheduling and preemption used in combination with HOLD/RELEASE

The HOLD and RELEASE mechanism provided by preemption (6.7.2, 12.30.1) allows an explicit "guard band" to be implemented around a protected transmission window, but one that is rather smaller than would otherwise be needed. As with R.3, this scenario assumes that the preemptable traffic's gate is scheduled to be open at all times. If a M_P_HOLD.request(HOLD) primitive is scheduled 124 octet times before the express traffic window opens, then preemptable frame transmission can start up to 124 octet times before the start of the protected window, and can potentially continue up to (but not beyond) the start of the window.

The result of using scheduling, HOLD/RELEASE and preemption in combination is that the express traffic's protected window can be completely protected from interference from preemptable traffic, while at the same time reducing the impact of the protected window on the amount of bandwidth that is available to preemptable traffic.

A further reason to use HOLD and RELEASE is that a schedule will often be intended for a string of express fames to be sent within an open window. If HOLD/RELEASE is not used and there is a greater than IPG gap between transmitting express frames (e.g., because of minor jitter in there arrival time), a preemptable frame or fragment could be started between express and cause up to 124 octets of latency in the middle of the string of express packets. Using HOLD ensures that this will not happen.

R.5 Bandwidth allocation and express traffic

The implementation of preemption in an IEEE 802.3 MAC (IEEE Std 802.3br) does not enforce a limit on how long a preemptable frame can be preempted by express traffic. This is at odds with the following:

a) Transit delay requirements of a bridge, if there is any preempted traffic present
b) The impossibility, as shown by queuing theory, of arranging express traffic to occupy exactly 100% of the available bandwidth in the absence of unbounded queues.

For effective use of preemption, system-wide mechanisms, such as bandwidth allocation, must operate so as to keep the bandwidth allocated to express traffic below 100% as judged on timescales comparable to the delays tolerated by time-sensitive traffic.

The fact that preemption can add delay that is invisible to transit delay protection could be addressed by schedules being set such that express traffic windows are short enough and gaps between them are short enough that the worst case time to send a preeptable packet is much less than transit delay requirements. For example, if transit delay is one second, the time between express windows is greater than a max packet time, and express window size is small compared to one second (e.g., <50 ms), then preemption will not add significantly to worst-case transit delay.

For preemption without scheduling, there is no guarantee on the added latency, but that alternative should be used when the volume of express traffic is expected to be small. If the express traffic is using a shaper, such as the credit-based shaper, that could put a similar reasonable bound on transit delay.

Annex S

(informative)

Bibliography

Insert the following item in the bibliography, renumbering as appropriate:

[B1] Seaman, Mick, *Preemption and MACsec replay protection,* November 2014. http://www.ieee802.org/1/files/public/docs2014/ae-seaman-preemption-1114-v04.pdf.

Consensus

WE BUILD IT.

Connect with us on:

Facebook: https://www.facebook.com/ieeesa

Twitter: @ieeesa

LinkedIn: http://www.linkedin.com/groups/IEEESA-Official-IEEE-Standards-Association-1791118

IEEE-SA Standards Insight blog: http://standardsinsight.com

YouTube: IEEE-SA Channel

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