IEEE P802.1Qdj/D1.1
June 2, 2023

(Amendment to IEEE Std 802.1Q™-2022,
as amended by IEEE Std 802.1Qcz™-2023,
IEEE Std 802.1Qcj™-2023,
IEEE Std 802.1Qcw™-2023)

IEEE P802.1Qdj/D1.1

Draft Standard for
Local and metropolitan area networks—

Bridges and Bridged Networks—
Amendment XX: Configuration Enhancements for Time-Sensitive Networking

Sponsor

LAN/MAN Standards Committee of the IEEE Computer Society

Prepared by the Time Sensitive Networking Task Group of IEEE 802.1

DRAFT STATUS:

Draft for second Working Group ballot.

Abstract: This amendment specifies procedures, interfaces, and managed objects to enhance the three models of 'Time-Sensitive Networking (TSN) configuration'. It specifies enhancements to the User/Network Interface (UNI) to include new capabilities to support bridges and end stations in order to extend the configuration capability.

Keywords: amendment, Bridged Local Area Networks, IEEE 802, IEEE 802.1Q™, IEEE 802.1Qdj™, Time-Sensitive Networking, TSN, Time-Sensitive Networking configuration, TSN configuration

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 on 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

Photocopies

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/xpl/standards.jsp 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 submitted to the IEEE-SA Standards Board for approval, the IEEE 802.1 Working Group had the following membership:

Glenn Parsons, *Chair*
Jessy Rouyer, *Vice Chair*
János Farkas, *Chair Time-Sensitive Networking Task Group*
Stephan Kehrer, *Editor*

<<TBA>>

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

<<TBA>>

When the IEEE-SA Standards Board approved this standard on XX Month 20xx, it had the following membership:

<<TBA>>

<<TBA>>

*Member Emeritus

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

<<TBA>>

Introduction

This introduction is not part of IEEE P802.1Qdj/D1.1, Draft Standard for Local and metropolitan area networks—Bridges and Bridged Networks— Amendment XX: Configuration Enhancements for Time-Sensitive Networking.

This amendment specifies procedures, interfaces, and managed objects to enhance the three models of 'Time-Sensitive Networking (TSN) configuration'. It specifies enhancements to the User/Network Interface (UNI) to include new capabilities to support bridges and end stations in order to extend the configuration capability. This amendment preserves the existing separation between configuration models and protocol specifications.

This amendment also addresses errors and omissions in the description of existing functionality.

<<Editor's Note: The PAR for this project is available at: https://development.standards.ieee.org/myproject-web/public/view.html#pardetail/7088. The CSD responses for this project are available at https://mentor.ieee.org/802-ec/dcn/19/ec-19-0139-00-ACSD-p802-1qdj.pdf.>>

<<Editor's Note: The following maintenance change request items have been actioned in this revised text:

0312: https://www.802-1.org/items/423.

>>

<<Editor's Introduction to historical draft changes

P802.1Qdj/D1.1: This draft was prepared by Stephan Kehrer for second Working Group ballot as a result of comment resolution on Draft D1.0, finalized during the meeting of the TSN Task Group on March 17, 2023.

Revision bars in D1.1 are relative to Draft 1.0.

Some comments in the comment resolution provided different wording for the same parts of the document in their respective responses. The current wording in D1.1 is an attempt of the editor to satisfy the spirit of all of the comments relating to the same text. This is the case for the following comments:

— C**omment #9** and **comment #39**
— C**omment #20** and **comment #64**

The responses of **comment #10**, **comment #125**, **comment #126**, **comment #138**, and **comment #24** (make changes to the editing instructions for 48.6.3, adding the clause title, and providing a diff-marked version of the YANG module showing the changes made) have been made obsolete by the response to **comment #143** (remove the YANG module in 48.6.3 due to the fact that the intended change to the module is not being implemented in this amendment). The response to **comment #143** is the one being implemented.

In deviation from the response to **comment #136** the first occurrence of "Centralized User Configuration (CUC)" has been kept in 1.3. This is in line with the introduction of other abbreviations in this clause. For the CNC only the abbreviated form is used because it has been introduced in IEEE Std 802.1Q-2022 in the bullet point cq) already.

For **comment #137** only four of the occurrences of “TSN” have been removed from the draft. For the other occurrences the editor feels that removing them would cause inconsistencies with already existing text in IEEE Std 802.1Q-2022.

>>

<<Editor’s Introduction to historical draft changes

P802.1Qdj/D1.0: This draft was prepared by Stephan Kehrer for first Working Group ballot as a result of comment resolution on Draft D0.3, finalized during the meeting of the TSN Task Group on September 12, 2022.

D1.0 has been rebased on P802.1Q-2022.

The compact YANG data scheme definition for ieee802-dot1q-tsn-config-uni.yang in 48.5.13 has been removed, as indicated by the editor’s note.

>>

<<Editor’s Introduction to historical draft changes

P802.1Qdj/D0.3: This draft was prepared by Stephan Kehrer for third Task Group ballot as a result of comment resolution on Draft D0.2, finalized during the electronic meeting of the TSN Task Group on May 9, 2022.

Revision bars in D0.3 are relative to Draft 0.2.

The following comments have not or only partially been partially implemented in D0.3:

Comment #59: The editor decided that this standard will be rebased on P802.1Q-Rev in a subsequent draft, as discussed during comment resolution.

Comment #83: After review of the terms listed in the comment and their used in this standard, the editor is of the opinion that the terms are used correctly. The commenter is asked to review D0.3 and if he still has concerns, make a comment against D0.3 that provides additional detail on where the terms are used in an inconsistent way.

>>

<<Editor’s Introduction to historical draft changes

P802.1Qdj/D0.2: This draft was prepared by Stephan Kehrer for second Task Group ballot as a result of comment resolution on Draft D0.1, finalized during the electronic meeting of the TSN Task Group on March 15, 2021.

Revision bars in D0.2 are relative to Draft 0.1.

The YANG module in Clause 48 has not been updated for D0.2. It will be updated in a future version, once additional elements that need to be implemented in the YANG module, e.g., YANG actions or YANG notifications, have been sufficiently discussed in the WG and have stabilized.

The following comments have not or only partially been partially implemented in D0.2:

Comment #4: Only deleted the first bullet item and rephrased the second one.

Comment #6: In the understanding of the editor and after double checking against IEEE Std 802.1Qcc-2018, the current YANG model matches the text in IEEE Std 802.1Qcc-2018, 46.2. StreamID, in the current model, is a leaf of the stream-id-type. This type consists of a MAC-Address and a unique StreamID. For this reason no changes have been made to the document because no further alignment was necessary. If the commenter feels that the understanding of the editor is incorrect, a comment to this effect is encouraged.

Comment #8: The configuration group in the YANG module provided in this document is a container for StatusInfo and FailedInterfaces as well as for AccumulatedLatency and InterfaceConfiguration. AccumulatedLatency and InterfaceConfiguration are distinct for each Talker or Listener and are therefore grouped under separate containers in the model. Naming the group "configuration" is indeed confusing. It would most likely be more adequate to name the group "status" to be in line with the naming provided in IEEE Std 802.1Qcc-2018. Since the YANG module has not been edited in this draft of P802.1Qdj an item in Annex Z has been created to address the topic with the next revision of the YANG module.

Comment #18: An item for this topic has been added to Annex Z. No text has been added to the main part of the document at this time because it is unclear to the editor how to best integrate the requested features into the new clause 46.1.7.1. The editor will work together with the original contributor to add the item in a future draft. Additional contributions on the topic are welcome.

Comments #23 and #24: No actions and notifications have been added to the YANG model because the YANG model was not updated for this draft. A specification of actions and notifications is not only required in the YANG model but also in clause 46.2. As there have been discussions and contributions to the topic after comment resolution on D0.1 was finished the editor feels it would be beneficial to work together further with the contributors before adding the content to a draft. Two items for these comments have been created in Annex Z. The editor intends to add content on actions and notifications in the next draft.

Comments #43 and #51: After reviewing clauses 46.1.3.2 and 46.1.3.3 in IEEE Std 802.1Qcc-2018 the editor is of the opinion that the information in these clauses and the additional information and clarification in 46.1.6 and 46.1.5 are not in conflict with each other. The already existing clauses in IEEE Std 802.1Qcc-2018 provide an overview over the different management models, the added information in this document describe that actual tasks assigned to the CUC and CNC entities in more detail. Removing the information in 46.1.3.2 and 46.1.3.3 of IEEE Std 802.1Qcc-2018 and moving them into the new clauses of this document would make it very hard to understand the management models since information would be missing in their description. On the other hand moving the information from the new clauses to 46.1.3.2 and 46.1.3.3 of IEEE Std 802.1Qcc-2018 would overload the description of the model with a more detailed description of the responsibilities of CUC and CNC. For this reason the editor did not try to consolidate the text at this time. If the commenter still thinks this needs to be done the editor would like to ask for another comment against this draft so the topic can be further discussed in the WG.

Comment #48: After a review of the base standard IEEE P802.1Q-Rev the editor decided nothing needs to be done for this comment. At the beginning of Clause 3 of IEEE P802.1Q-Rev it is clearly stated that the standard makes use of the term "station", as defined in IEEE Std 802. Since this standard is an amendment to IEEE 802.1Q the use of "station" as stand alone term seems justified to the editor.

>>

<<Editor's Introduction to historical draft changes

P802.1Qdj/D0.1: This draft was prepared by Stephan Kehrer for a first Task Group ballot. Everything in this draft can be considered a contribution to the Time-Sensitive Networking Task Group by the editor; nothing has been approved by the Task Group or Working Group.

>>

<<Editor's Introduction to historical draft changes

P802.1Qdj/D0.0: This draft was prepared by Stephan Kehrer as the first draft. Everything in this draft can be considered a contribution to the Time-Sensitive Networking Task Group by the editor; nothing has been approved by the Task Group or Working Group.

>>

IEEE P802.1Qdj/D1.1

Draft Standard for Local and metropolitan area networks—

Bridges and Bridged Networks—Amendment XX: Configuration Enhancements for Time-Sensitive Networking

This amendment is based on IEEE Std 802.1Q™-2022, as amended by IEEE Std 802.1Qcz™-2023, IEEE Std 802.1Qcj™-2023, IEEE Std 802.1Qcw™-2023.

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.[b]

1. Overview

1.3 Introduction

Insert the following items after item cq) in 1.3 and renumber the items in the lettered list, as necessary:

a) Defines the Centralized User Configuration (CUC) (46.1.5), and the CNC (46.1.6).
b) Specifies a Configuration Domain (46.1.7).
c) Defines YANG configuration and operational state models (48.6.23) in support of the UNI (Clause 46).

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

3. Definitions

Insert the following definitions in the appropriate collating sequence, renumbering accordingly:

3.1 Configuration Domain: A set of stations that are under a common configuration, management scheme, and responsibility.

3.2 TSN features: The protocols and mechanisms that define the set of tools available for building a time-sensitive network.

NOTE—See Annex X for more information.

5. Conformance

5.29 TSN CNC station requirements

Change item d) in 5.29, as follows

d) If a YANG-based protocol is supported by the TSN CNC for the User/network configuration information, that protocol shall use the YANG modules specified in 46.3.

46. Time-Sensitive Networking (TSN) configuration

46.1 Overview of TSN configuration

46.1.3 TSN configuration models

46.1.3.2 Centralized network/distributed user model

Change the third paragraph in 46.1.3.2, as follows.

The centralized network/distributed user model is similar to the fully distributed model in that end stations communicate their Talker/Listener requirements directly over the ~~TSN~~ UNI. In contrast, in the centralized network/distributed user model, the configuration information is directed to/from a ~~Centralized Network Configuration (CNC)~~CNC (46.1.6) entity. All configuration of Bridges for TSN Streams is performed by this CNC using a remote network management protocol.

46.1.3.3 Fully centralized model

Change the second paragraph in 46.1.3.3, as follows.

In order to accommodate this sort of ~~TSN~~ use case, the fully centralized model enables a ~~Centralized User Configuration (CUC)~~CUC (46.1.5) entity to discover end stations, retrieve end station capabilities and user requirements, and configure TSN features in end stations. The protocols that the CUC uses for this purpose are specific to the user application and outside the scope of this standard.

Replace Figure 46-3 with the following figure.

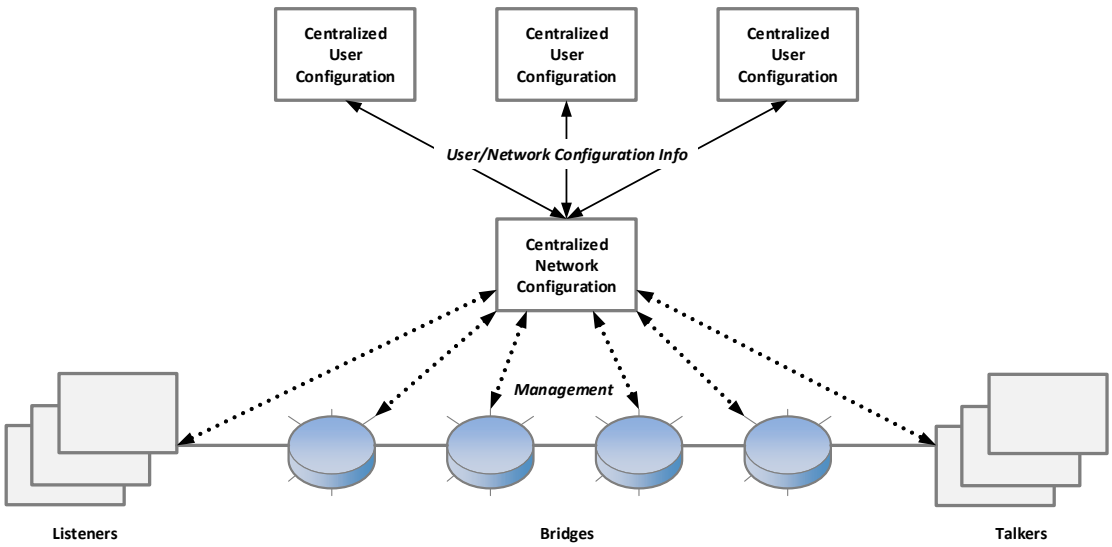

Figure 46-3—Fully centralized model

Insert the following subclauses after 46.1.4

46.1.5 Centralized User Configuration

The Centralized User Configuration (CUC) delivers user requirements to the CNC. The CUC delivers information for configuring TSN features to end stations. It is a logical entity that can be located in any station of a network.

The CUC is responsible for:

a) Reconciling the requirements from Talkers and Listeners to Stream requirements, if necessary.
b) Recommending a StreamID group (46.2.3.1) to the Stream requirements.
c) Sending the Stream requirements to the CNC.
d) Receiving the end station communication-configuration from the CNC.
e) Distributing the end station communication-configuration to Talkers and Listeners.

NOTE–It is the responsibility of the CNC to ensure that Streams are assigned a unique StreamID group. For this an RPC RequestFreeStreamId (46.2.7.5) is available so the CUC can request a free StreamID from the CNC.

Stream requirements, in the context of the CUC, result from combining the Stream requirements of one Talker with the Stream requirements of one or multiple Listeners that, together, apply to form a Stream. Reconciling the requirements for the Stream does not change the parameters in the Stream request originating from the Talker or the Listener(s).

The end station communication-configuration that is received by the CUC from the CNC and then distributed to the Talkers and Listeners does not directly configure features on the end stations. It consists of configuration information that a CUC can provide for a Talker and Listeners to configure the Stream. An end station could, for example, make use of the information it receives in the communication-configuration from the CUC to configure an application in a way that ensures different TSN Streams are sent by the application in a specific order that correlates with the expected Stream's transmission on the network.

A CUC affects only one Configuration Domain. Talkers and Listeners can only make use of the CUC to reconcile their Stream requirements into a Stream request, if they are part of the same Configuration Domain. If a Talker wants to communicate with one or more Listeners in a different Configuration Domain, this needs to be done through dedicated inter-domain communication mechanisms. Such inter-domain communication mechanisms are not specified by this standard.

The protocols that the CUC uses for communication with end stations are not specified by this standard. A CUC exchanges information with a CNC in order to configure TSN features on behalf of its end stations. It communicates with the CNC through the CUC-CNC interface specified in 46.2. The CUC can request computation of paths and configurations for Streams in the following ways:

f) Request computation of the paths and configurations for a set of Streams, using the protocol operation described in 46.2.7.1. The computation is performed by the CNC on the complete set of Streams of this request. This allows for optimized scheduling of Streams in the network.
g) Request computation of the paths and configurations for new or modified Streams, using the protocol operation described in 46.2.7.2. The computation is performed by the CNC on all Streams in a Configuration Domain that have a StreamStatus (46.2.3.8) of either planned or modified.
h) Request the joining of a set of Listeners to an already existing Stream. The paths are extended to allow forwarding of the Stream to the new Listeners.
i) Request the removal of an existing Stream, using the protocol operation described in 46.2.8.1.
j) Request the removal of one or more Listeners from an existing Stream.

A CUC can be present for initial configuration, to manage changes to a running network, or both. Multiple CUCs can co-exist and operate in parallel in the same Configuration Domain as shown in Figure 46-3.

46.1.6 Centralized Network Configuration

The Centralized Network Configuration (CNC) is a logical entity that configures network resources on behalf of ~~TSN~~ applications (users) and can be located in any station of a network.

The CNC is responsible for:

a) Receiving the Stream requirements for one or more Streams from the corresponding CUC.
b) Providing a way to the CUC to request a free StreamID.
c) Assigning a unique destination MAC address in the Configuration Domain it is responsible for to each of the requested Streams.
d) Computing paths for requested Streams.
e) Performing computation of scheduling and/or shaping configuration for the requested Streams.
f) Configuring the network devices to provide the required resources for the Streams (e.g. FDB entries, configuration of transmission gates, etc.), using remote management.
g) Providing the end station communication-configuration for the Streams to the corresponding CUC. If the paths for the Streams impact existing Streams the CNC is also responsible for providing that information to the CUCs that originally requested the impacted Streams.
h) Removing of Streams as requested by a CUC.
i) Discovering physical topology, using remote management.
j) Retrieving Station capabilities, using remote management.

The CNC communicates with a CUC through the CUC-CNC interface specified in 46.2. It communicates with the stations using the managed objects defined in IEEE Std 802.1Q-2022 and other IEEE 802.1 standards. There can only be one active CNC per Configuration Domain.

46.1.7 Configuration Domain

A Configuration Domain provides boundary information for the common management scheme and responsibility of Streams. Whether a CNC and one or more CUCs are present in a Configuration Domain depends on the TSN configuration model (46.1.3) that is used in the domain (e.g., whether the fully centralized model or a different configuration model is used). The CNC and the CUCs required for the configuration of a Configuration Domain affect only one Configuration Domain.

Change clause 46.2.2, as follows:

46.2.2 Protocol integration

Change the third paragraph in clause 46.2.2, as follows:

Each TSN configuration protocol shall use the StreamID of ~~this~~ clause ~~(~~46.2.3.1~~)~~ as the unique identifier of each Stream's configuration. The StreamID identifies configuration, not data, so it has no formal relation to the data frame encoding for the Stream.

Add the following note below the bullet list in the fourth paragraph in clause 46.2.2, as follows:

TSN configuration can be viewed conceptually as a request/response exchange:

— Request: End station or CUC transmits a protocol message that contains a Talker or Listener group.
— Response: Bridge or CNC transmits a protocol message that contains a Status group.

Note–The Response can be unsolicited in order to update configuration, e.g., to address a change in the network.

Change the last paragraph in clause 46.2.2, as follows:

The protocol message(s) that invoke the join or leave operation are not required to coincide with the protocol message(s) that contain the associated groups (Talker, Listener, or Status). Nevertheless, the groups specify elements that are required for a subsequent join or leave operation to be valid. For example, for the fully centralized model (46.1.3.3), the CUC can transfer a list of Talker/Listener groups to the CNC, followed by a separate protocol message with a join request that applies to the entire list. For the join request to succeed, each of the Talker/Listener groups ~~must contain~~contains the required elements. At a later time, the CUC can read the resulting list of Status groups from the CNC, which provides the response to the join.

Insert the following subclauses after 46.2.2.

46.2.2.1 DomainID

DomainID is a unique identifier that specifies the Configuration Domain of a CUC, and the Streams associated with that CUC. DomainID is only used if the centralized network/distributed user model (46.1.3.2) or the fully centralized model (46.1.3.3) is used.

46.2.2.2 CucID

CucID uniquely identifies a CUC within a Configuration Domain. It is used along with the DomainID to associate Streams with a CUC. CucID is only used if the centralized network/distributed user model (46.1.3.2) or the fully centralized model (46.1.3.3) is used.

46.2.2.3 CncEnabled

CncEnabled is used to enable or disable the CNC functionality of a station capable of acting as a CNC. If CncEnabled is set to TRUE the CNC functionality is enabled. If it is set to FALSE the CNC functionality is disabled. The default value for CncEnabled is FALSE.

46.2.3 Talker

Change the third paragraph in clause 46.2.3, as follows:

The Talker group contains the following groups:

- StreamID (46.2.3.1)
- StreamRank (46.2.3.2)
- EndStationInterfaces (46.2.3.3)
- DataFrameSpecification (46.2.3.4)
- TrafficSpecification (46.2.3.5)
- UserToNetworkRequirements (46.2.3.6)
- InterfaceCapabilities (46.2.3.7)
- StreamStatus (46.2.3.8)

Insert the following sentence at the end of clause 46.2.3, as new paragraph:

For the join and leave operation, StreamStatus shall be included.

Insert the following subclause after 46.2.3.7. Insert Table 46-12 in subclause 46.2.3.8, and number the table appropriately, renumbering subsequent tables as required.

46.2.3.8 StreamStatus

StreamStatus is an enumeration specified in Table 46-12 that indicates the status of a Stream. The status is maintained by the CNC and is used to determine which Streams are computed by calling the RPC ComputePlannedAndModifiedStreams (46.2.7.2).

Table 46-12—StreamStatus enumeration

Name	Value	Description
Planned	0	Stream has been requested but has not yet been configured.
Configured	1	Stream has been computed and configured.
Modified	2	Stream has been configured but Stream parameters have been modified after configuration..

Insert the following subclauses after 46.2.5

46.2.6 Protocol operations

The TSN user/network configuration makes use of protocol operations to request specific actions and to receive notifications. The following operations are supported:

— **Remote Procedure Calls (RPC):** this protocol operation allows requesting an action for the complete YANG data model.
— **Actions:** this protocol operation allows requesting an action on a specific part of the YANG data model.
— **Notifications:** this protocol operation provides information, e.g., it allows the CNC to inform the CUC that computing the configuration has finished.

46.2.7 Remote Procedure Calls

The TSN user/network configuration provides the following RPCs:

— ComputeStreams (46.2.7.1)
— ComputePlannedAndModifiedStreams (46.2.7.2)
— ComputeAllStreams (46.2.7.3)
— RequestDomainId (46.2.7.4)
— RequestFreeStreamId (46.2.7.5)

46.2.7.1 ComputeStreams

This RPC starts the computation of path and resource allocation for one or more Streams. The Streams that are to be included in the computation are specified by providing their associated DomainID (46.2.2.1), CucID (46.2.2.2), and StreamID (46.2.3.1). This RPC can be applied to compute new Streams as well as recompute already configured Streams.

The RPC returns information that indicates the state of stream computations. It does not return information on the Stream configuration status because computation and configuration can take an arbitrary amount of

time. The notification ConfigureStreamsCompleted (46.2.9.1) is available to the CNC to return information on success or failure of the Stream computation, after the computation has finished.

46.2.7.2 ComputePlannedAndModifiedStreams

This RPC starts the computation of path and resource allocation for Streams that have not been configured or that have been configured and have been modified since configuration. The Streams that are to be included in the computation are specified by providing their associated DomainID (46.2.2.1) and CucID (46.2.2.2). The object StreamStatus (46.2.3.8) is used to determine if a Stream is included in the computation initiated by this RPC.

The RPC returns information that indicates only if the Stream computation has been started successfully or not. It does not return information on whether the Stream configuration itself has been successful or not, because computation can take an arbitrary amount of time. The notification ConfigureStreamsCompleted (46.2.9.1) is available to the CNC to return information on success or failure of the Stream computation, after the computation has finished.

46.2.7.3 ComputeAllStreams

This RPC starts the computation of path and resource allocation for all Streams in a Configuration Domain and that are belonging to a specified CUC. The Streams that are to be included in the computation are specified by providing their associated DomainID (46.2.2.1) and CucID (46.2.2.2).

The RPC returns information that indicates only if the Stream computation has been started successfully or not. It does not return information on whether the Stream configuration itself has been successful or not, because computation can take an arbitrary amount of time. The notification ConfigureStreamsCompleted (46.2.9.1) is available to the CNC to return information on success or failure of the Stream computation, after the computation has finished.

46.2.7.4 RequestDomainId

This RPC allows a CUC to request the DomainID (46.2.2.1) of the Configuration Domain that the CUC belongs to from the CNC. If a CUC already knows the Configuration Domain it belongs to, this RPC can be used to verify that the information the CUC has is correct.

46.2.7.5 RequestFreeStreamId

This RPC allows a CUC to request a free StreamID group (46.2.3.1) from a CNC. Requesting a free StreamID group allows a CUC to provide an unused, i.e., unique, StreamId group for a Stream when requesting that Stream from the CNC.

46.2.8 Actions

The TSN user/network configuration provides the following actions:

— RemoveStreams (46.2.8.1)

46.2.8.1 RemoveStreams

This action starts the removal of one or more Streams. The Streams that are to be removed are specified by providing their associated StreamIDs (46.2.3.1). This action returns information that indicates only if the Stream removal has been started successfully or not. It does not return information on whether the Stream removal itself has been successful or not, because execution can take an arbitrary amount of time. When a

Stream is succesfully removed, the StreamId associated with that Stream can be used as a free StreamId by the RPC RequestFreeStreamId (46.2.7.5) again.

The notification RemoveStreamsCompleted (46.2.9.3) is available to the CNC to return information on success or failure of the Stream removal.

46.2.9 Notifications

The TSN user/network configuration provides the following notfications:

— CoputeStreamsCompleted (46.2.9.1)
— ConfigureStreamsCompleted (46.2.9.2)
— RemoveStreamsCompleted (46.2.9.3)

46.2.9.1 CoputeStreamsCompleted

This notification is used by the CNC to inform a CUC that has requested the computation of Streams, that the computation for these Streams has finished. If the computation of these Streams impacts other Streams that are already configured in the network, it can also be used to notify the CUCs that originally requested the impacted Streams about the modification.

NOTE—ComputeStreamsCompleted returns only information on the computation of Streams. This does not provide any information on whether the configuration of these Streams has been performed successfully or not.

It returns a list of Domains, identified by their DomainIDs (46.2.2.1), CUCs in that domain, identified by their CucIDs (46.2.2.2) and Streams associated with a CUC, identified by their StreamIDs (46.2.3.1). For each Stream it also returns either 0, if the Stream computation was successful, or a FailureCode (46.2.5.1.3), if it was not.

46.2.9.2 ConfigureStreamsCompleted

This notification is used by the CNC to inform a CUC that has requested the computation of Streams, that the computation and configuration for these Streams has finished. If the computation or configuration of these Streams impacts other Streams that are already configured in the network, it can also be used to notify the CUCs that originally requested the impacted Streams about the modification.

It returns a list of Domains, identified by their DomainIDs (46.2.2.1), CUCs in that domain, identified by their CucIDs (46.2.2.2) and Streams associated with a CUC, identified by their StreamIDs (46.2.3.1). For each Stream it also returns either 0, if the Stream computation and configuration was successful, or a FailureCode (46.2.5.1.3), if it was not.

46.2.9.3 RemoveStreamsCompleted

This notification is used by the CNC to inform a CUC that has requested the removal of Streams, that the removal of these Streams has finished. It returns a list of Domains, identified by their DomainIDs (46.2.2.1), CUCs in that domain, identified by their CucIDs (46.2.2.2) and Streams associated with a CUC, identified by their StreamIDs (46.2.3.1). For each Stream it also returns either 0, if the Stream computation and configuration was successful, or 1, if it was not.

46.3 YANG for TSN user/network configuration

Change 46.3, as follows:

In order to support the use of YANG-based protocols for the fully centralized model (46.1.3.3), 48.6.3 and 48.6.23 ~~specifies a~~specify YANG module~~s~~.

If a YANG-based protocol is specified by another standard for the TSN user/network configuration information (46.2), that specification shall use the YANG module~~s~~ specified in 48.6.3 and 48.6.23 [see item d) in 5.29].

The YANG module of 48.6.3 provides YANG text for each group of elements in 46.2. Each element is specified using a YANG leaf. Each group is specified as a YANG typedef or grouping. The YANG module for user/network configuration (48.6.23) imports the YANG module of 48.6.3 and uses the typedef and grouping nodes in order to specify the schema tree used for communication between CUC and CNC.

YANG identifiers use a naming convention of hyphens between lowercase names (e.g., “mac-address”). Identifiers for elements and groups in 46.2 use a naming convention of camel case (e.g., “MacAddress”). The specifications for an identifier in 48.6.3 and 48.6.23 shall be interpreted as applying to the corresponding identifier in 46.2 regardless of differences in naming convention (e.g., requirements for “MacAddress” in 46.2 apply to “mac-address” in 48.6.3).

In the YANG module definitions of 48.6.3 and 48.6.23, if any discrepancy between the “description” text and the corresponding specifications in 46.2 occurs, the specifications in 46.2 take precedence.

48. YANG Data Models

Insert the following subclause (48.2.12) after 48.2.11

48.2.12 User/Network Interface model

The UNI allows communication between a CUC and a CNC and can be implemented in an end station or Bridge.

The UNI consists of three high-level groups, Talker (46.2.3), Listener (46.2.4), and Status (46.2.5) and is modeled as illustrated in Figure 48-21.

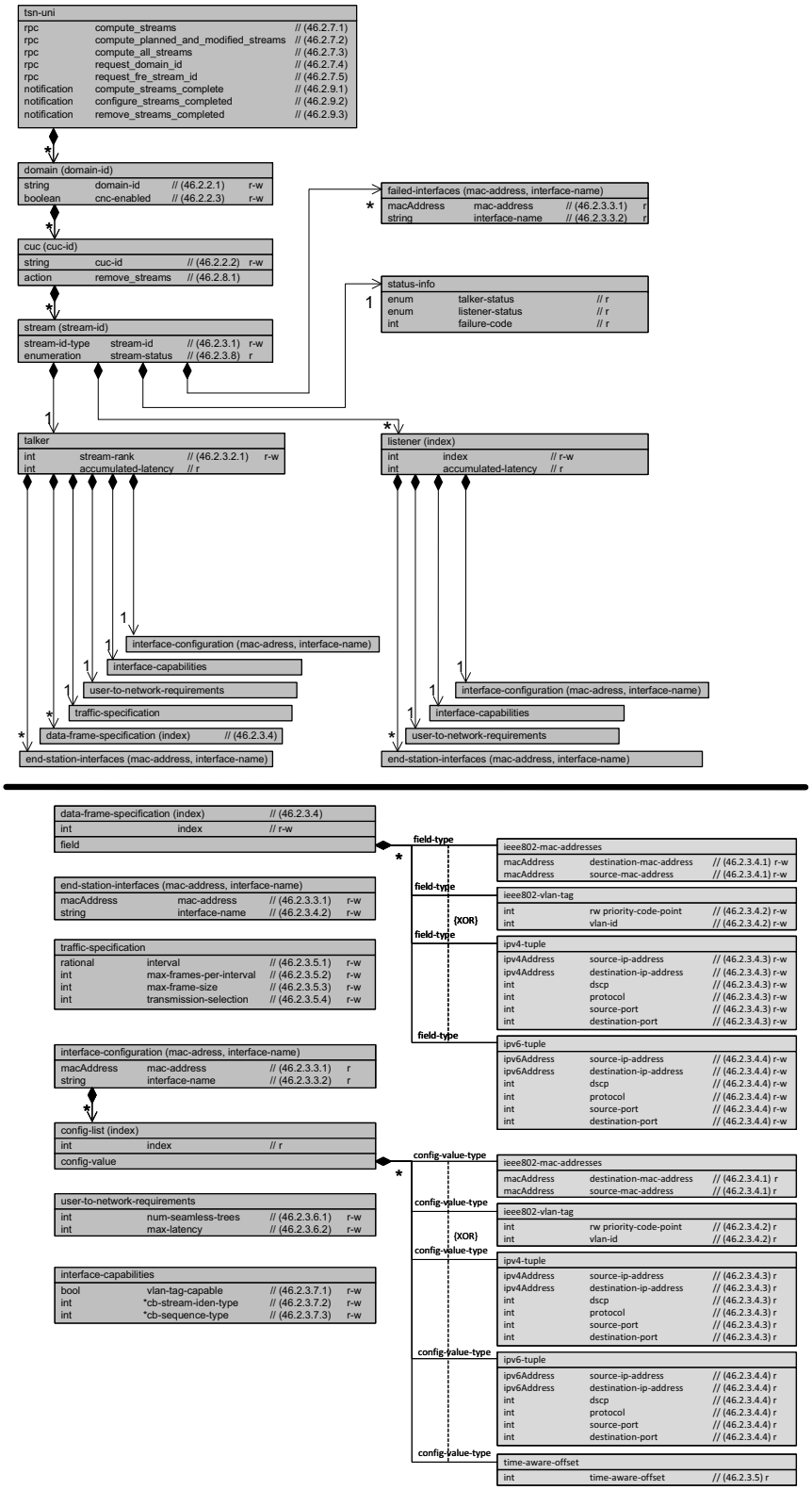

Figure 48-21—User/Network Interface model

48.3 Structure of the YANG models

Insert the following line at the end of Table 48-1 in 48.3

Table 48-1—Summary of the YANG modules

Module	References	Managed functionality	Initial YANG specification Notes
ieee802-dot1q-tsn-config-uni	48.5.23, 48.6.23	46.1.5, 46.1.6, 46.2	IEEE Std 802.1Qdj Time-Sensitive Networking configuration UNI

Insert the following subclause (48.3.12) after 48.3.11

48.3.12 User/Network Interface model

A station implementing the User/Network Interface model (48.2.12) implements the YANG modules in Table 48-13.

Table 48-13—User/Network Interface model YANG modules

YANG module
ieee802-dot1q-tsn-types
ieee802-dot1q-tsn-config-uni

48.4 Security Conisderations

Insert the following subclause (48.4.12) after 48.4.11

48.4.12 Security considerations of the User/Network Interface model

There are a number of management objects defined in the ieee802-dot1q-tsn-config-uni YANG module that are configurable (i.e., read-write) and/or operational (i.e., read-only). Such objects may be considered sensitive or vulnerable in some network environments. A network configuration protocol, such as NETCONF (IETF RFC 6241 [B39]), can support protocol operations that can edit or delete YANG module configuration data (i.e., edit-config, delete-config, copy-config). If this is done in a non-secure environment without proper protection, then negative effects on the network operation are possible.

tsn-config-uniconfiguration of the UNI

tsn-uni/domain/cuc/stream

tsn-uni/domain/cuc/remove_stream

48.5 YANG schema tree definitions

Insert the following subclause (48.5.23) after 48.5.22

48.5.23 Schema for the ieee802-dot1q-tsn-config-uni YANG module

```
module: ieee802-dot1q-tsn-config-uni
  +--rw tsn-uni
     +--rw domain* [domain-id]
        +--rw domain-id      string
        +--rw cnc-enabled?   boolean
        +--rw cuc* [cuc-id]
           +--rw cuc-id            string
           +--rw stream* [stream-id]
           |  +--rw stream-id            tsn:stream-id-type
           |  +--ro stream-status?       enumeration
           |  +--rw talker
           |  |  +--rw stream-rank
           |  |  |  +--rw rank?   uint8
           |  |  +--rw end-station-interfaces* [mac-address interface-name]
           |  |  |  +--rw mac-address       string
           |  |  |  +--rw interface-name    string
           |  |  +--rw data-frame-specification* [index]
           |  |  |  +--rw index                          uint8
           |  |  |  +--rw (field)?
           |  |  |     +--:(ieee802-mac-addresses)
           |  |  |     |  +--rw ieee802-mac-addresses
           |  |  |     |     +--rw destination-mac-address?   string
           |  |  |     |     +--rw source-mac-address?        string
           |  |  |     +--:(ieee802-vlan-tag)
           |  |  |     |  +--rw ieee802-vlan-tag
           |  |  |     |     +--rw priority-code-point?   uint8
           |  |  |     |     +--rw vlan-id?               uint16
           |  |  |     +--:(ipv4-tuple)
           |  |  |     |  +--rw ipv4-tuple
           |  |  |     |     +--rw source-ip-address?        inet:ipv4-address
           |  |  |     |     +--rw destination-ip-address?   inet:ipv4-address
           |  |  |     |     +--rw dscp?                     uint8
           |  |  |     |     +--rw protocol?                 uint16
           |  |  |     |     +--rw source-port?              uint16
           |  |  |     |     +--rw destination-port?         uint16
           |  |  |     +--:(ipv6-tuple)
           |  |  |        +--rw ipv6-tuple
           |  |  |           +--rw source-ip-address?        inet:ipv6-address
           |  |  |           +--rw destination-ip-address?   inet:ipv6-address
           |  |  |           +--rw dscp?                     uint8
           |  |  |           +--rw protocol?                 uint16
           |  |  |           +--rw source-port?              uint16
           |  |  |           +--rw destination-port?         uint16
           |  |  +--rw traffic-specification
           |  |  |  +--rw interval
           |  |  |  |  +--rw numerator?     uint32
           |  |  |  |  +--rw denominator?   uint32
           |  |  |  +--rw max-frames-per-interval?   uint16
           |  |  |  +--rw max-frame-size?            uint16
           |  |  |  +--rw transmission-selection?    uint8
           |  |  |  +--rw time-aware!
           |  |  |     +--rw earliest-transmit-offset?   uint32
           |  |  |     +--rw latest-transmit-offset?     uint32
           |  |  |     +--rw jitter?                     uint32
           |  |  +--rw user-to-network-requirements
           |  |  |  +--rw num-seamless-trees?   uint8
           |  |  |  +--rw max-latency?          uint32
           |  |  +--rw interface-capabilities
           |  |  |  +--rw vlan-tag-capable?           boolean
```

```
|  |  |  +--rw cb-stream-iden-type-list*   uint32
|  |  |  +--rw cb-sequence-type-list*      uint32
|  |  +--ro accumulated-latency?           uint32
|  |  +--ro interface-configuration
|  |     +--ro interface-list* [mac-address interface-name]
|  |        +--ro mac-address       string
|  |        +--ro interface-name    string
|  |        +--ro config-list* [index]
|  |           +--ro index                          uint8
|  |           +--ro (config-value)?
|  |              +--:(ieee802-mac-addresses)
|  |              |  +--ro ieee802-mac-addresses
|  |              |     +--ro destination-mac-address?   string
|  |              |     +--ro source-mac-address?        string
|  |              +--:(ieee802-vlan-tag)
|  |              |  +--ro ieee802-vlan-tag
|  |              |     +--ro priority-code-point?   uint8
|  |              |     +--ro vlan-id?               uint16
|  |              +--:(ipv4-tuple)
|  |              |  +--ro ipv4-tuple
|  |              |     +--ro source-ip-address?
|  |              |     |       inet:ipv4-address
|  |              |     +--ro destination-ip-address?
|  |              |     |       inet:ipv4-address
|  |              |     +--ro dscp?                     uint8
|  |              |     +--ro protocol?                 uint16
|  |              |     +--ro source-port?              uint16
|  |              |     +--ro destination-port?         uint16
|  |              +--:(ipv6-tuple)
|  |              |  +--ro ipv6-tuple
|  |              |     +--ro source-ip-address?
|  |              |     |       inet:ipv6-address
|  |              |     +--ro destination-ip-address?
|  |              |     |       inet:ipv6-address
|  |              |     +--ro dscp?                     uint8
|  |              |     +--ro protocol?                 uint16
|  |              |     +--ro source-port?              uint16
|  |              |     +--ro destination-port?         uint16
|  |              +--:(time-aware-offset)
|  |                 +--ro time-aware-offset?       uint32
|  +--rw listener* [index]
|  |  +--rw index                            uint32
|  |  +--rw end-station-interfaces* [mac-address interface-name]
|  |  |  +--rw mac-address       string
|  |  |  +--rw interface-name    string
|  |  +--rw user-to-network-requirements
|  |  |  +--rw num-seamless-trees?   uint8
|  |  |  +--rw max-latency?          uint32
|  |  +--rw interface-capabilities
|  |  |  +--rw vlan-tag-capable?           boolean
|  |  |  +--rw cb-stream-iden-type-list*   uint32
|  |  |  +--rw cb-sequence-type-list*      uint32
|  |  +--ro accumulated-latency?           uint32
|  |  +--ro interface-configuration
|  |     +--ro interface-list* [mac-address interface-name]
|  |        +--ro mac-address       string
|  |        +--ro interface-name    string
|  |        +--ro config-list* [index]
|  |           +--ro index                          uint8
```

```
          |  |           +--ro (config-value)?
          |  |              +--:(ieee802-mac-addresses)
          |  |              |  +--ro ieee802-mac-addresses
          |  |              |     +--ro destination-mac-address?   string
          |  |              |     +--ro source-mac-address?        string
          |  |              +--:(ieee802-vlan-tag)
          |  |              |  +--ro ieee802-vlan-tag
          |  |              |     +--ro priority-code-point?   uint8
          |  |              |     +--ro vlan-id?               uint16
          |  |              +--:(ipv4-tuple)
          |  |              |  +--ro ipv4-tuple
          |  |              |     +--ro source-ip-address?
          |  |              |     |       inet:ipv4-address
          |  |              |     +--ro destination-ip-address?
          |  |              |     |       inet:ipv4-address
          |  |              |     +--ro dscp?                     uint8
          |  |              |     +--ro protocol?                 uint16
          |  |              |     +--ro source-port?              uint16
          |  |              |     +--ro destination-port?         uint16
          |  |              +--:(ipv6-tuple)
          |  |              |  +--ro ipv6-tuple
          |  |              |     +--ro source-ip-address?
          |  |              |     |       inet:ipv6-address
          |  |              |     +--ro destination-ip-address?
          |  |              |     |       inet:ipv6-address
          |  |              |     +--ro dscp?                     uint8
          |  |              |     +--ro protocol?                 uint16
          |  |              |     +--ro source-port?              uint16
          |  |              |     +--ro destination-port?         uint16
          |  |              +--:(time-aware-offset)
          |  |                 +--ro time-aware-offset?       uint32
          |  +--ro status-info
          |  |  +--ro talker-status?     enumeration
          |  |  +--ro listener-status?   enumeration
          |  |  +--ro failure-code?      uint8
          |  +--ro failed-interfaces* [mac-address interface-name]
          |     +--ro mac-address       string
          |     +--ro interface-name    string
          +---x remove_streams
             +---w input
             |  +---w stream-list* [stream-id]
             |     +---w stream-id    tsn:stream-id-type
             +--ro output
                +--ro result?   string

rpcs:
  +---x compute_streams
  |  +---w input
  |  |  +---w domain* [domain-id]
  |  |     +---w domain-id    -> /tsn-uni/domain/domain-id
  |  |     +---w cuc* [cuc-id]
  |  |        +---w cuc-id         -> /tsn-uni/domain/cuc/cuc-id
  |  |        +---w stream-list* [stream-id]
  |  |           +---w stream-id    -> /tsn-uni/domain/cuc/stream/stream-id
  |  +--ro output
  |     +--ro result?   string
  +---x compute_planned_and_modified_streams
  |  +---w input
  |  |  +---w domain* [domain-id]
```

```
   |  |     +---w domain-id    string
   |  |     +---w cuc* [cuc-id]
   |  |        +---w cuc-id    string
   |  +--ro output
   |     +--ro result?   string
   +---x compute_all_streams
   |  +---w input
   |  |  +---w domain* [domain-id]
   |  |     +---w domain-id    string
   |  |     +---w cuc* [cuc-id]
   |  |        +---w cuc-id    string
   |  +--ro output
   |     +--ro result?   string
   +---x request_domain_id
   |  +---w input
   |  |  +---w cuc-id?   string
   |  +--ro output
   |     +--ro result?   string
   +---x request_free_stream_id
      +---w input
      |  +---w domain-id?   string
      |  +---w cuc-id?      string
      +--ro output
         +--ro result?   string

notifications:
  +---n compute_streams_completed
  |  +--ro domain* [domain-id]
  |     +--ro domain-id    string
  |     +--ro cuc* [cuc-id]
  |        +--ro cuc-id    string
  |        +--ro stream* [stream-id]
  |           +--ro stream-id       tsn:stream-id-type
  |           +--ro failure-code?   uint8
  +---n configure_streams_completed
  |  +--ro domain* [domain-id]
  |     +--ro domain-id    string
  |     +--ro cuc* [cuc-id]
  |        +--ro cuc-id    string
  |        +--ro stream* [stream-id]
  |           +--ro stream-id       tsn:stream-id-type
  |           +--ro failure-code?   uint8
  +---n remove_streams_completed
     +--ro domain* [domain-id]
        +--ro domain-id    string
        +--ro cuc* [cuc-id]
           +--ro cuc-id    string
           +--ro stream* [stream-id]
              +--ro stream-id       tsn:stream-id-type
              +--ro failure-code?   uint8
```

Insert the following subclause (48.6.23) after 48.6.22

48.6.23 Definitions for the ieee802-dot1q-tsn-config-uni YANG module

```
module ieee802-dot1q-tsn-config-uni {
```

```
yang-version "1.1";
namespace urn:ieee:std:802.1Q:yang:ieee802-dot1q-tsn-config-uni;
prefix dot1q-tsn-config-uni;
import ieee802-dot1q-tsn-types {
  prefix tsn;
  reference
    "48.6.3 of IEEE Std 802.1Q-2022";
}
organization
  "Institute of Electrical and Electronics Engineers";
contact
  "WG-URL: http://ieee802.org/1/
  WG-EMail: stds-802-1-l@ieee.org

  Contact: IEEE 802.1 Working Group Chair
  Postal: C/O IEEE 802.1 Working Group
        IEEE Standards Association
        445 Hoes Lane
        Piscataway, NJ 08854
        USA

  E-mail: stds-802-1-chairs@ieee.org";
description
  "Time-Sensitive Networking (TSN) User/Network Interface (UNI) for the
  exchange of information between CUC and CNC that are required to
  configure TSN Streams in a TSN network.";
revision 2023-05-16 {
  description
    "Editor's note: this revision statement will be removed prior to
    publication. It is only present while the project is running in
    order to make it easier for the reader to see what changes have
    been introduced while the project is running. Update for D1.1 that
    includes fixes, and updates and adds RPCs, actions, and
    notifications.";
  reference
    "Clause 46.2 of IEEE Std 802.1Q-2022";
}
revision 2022-11-09 {
  description
    "Editor's note: this revision statement will be removed prior to
    publication. It is only present while the project is running in
    order to make it easier for the reader to see what changes have
    been introduced while the project is running. Update for D1.0 that
    rebases the YANG module of IEEE Std 802.1Q-2022 and updates RPCs,
    actions, and notifications.";
  reference
    "Clause 46.2 of IEEE Std 802.1Q-2022";
}
revision 2022-06-03 {
  description
    "Editor's note: this revision statement will be removed prior to
    publication. It is only present while the project is running in
    order to make it easier for the reader to see what changes have
    been introduced while the project is running. Update for D0.3 that
    restructures the data model, adds RPCS, adds actions, and adds
    notifications.";
  reference
    "Clause 46.2 of IEEE Std 802.1Qcc-2018";
}
```

```
revision 2021-12-16 {
  description
    "Initial revision. Note that this module might change in backward
    incompatible ways until approved as a standard.";
  reference
    "Clause 46.2 of IEEE Std 802.1Q-2022";
}
container tsn-uni {
  description
    "Top-level container for the TSN UNI module.";
  list domain {
    key "domain-id";
    description
      "List of Configuration Domains.

      This list exists so CUCs can be associated with the Configuration
      Domain they are located in and can be used to restrict access to
      CUCs, e.g., by using standard mechanism as described in RFC 8341.";
    leaf domain-id {
      type string;
      description
        "The Domain ID is a unique identifier of a Configuration
        Domain. It is used to identify the Configuration Domain a CUC
        belongs to.";
      reference
        "46.2.2.1 of IEEE Std 802.1Qdj-2023";
    }
    leaf cnc-enabled {
      type boolean;
      default "false";
      description
        "cnc-enabled is used to enable or disable the CNC functionality
        of a station capable of acting as a CNC. If this object is set
        to TRUE the CNC functionality is enabled. If it is set to FALSE
        the CNC functionality is disabled.";
      reference
        "46.2.2.3 of IEEE Std 802.1Qdj-2023";
    }
    list cuc {
      key "cuc-id";
      description
        "List of CUCs.

        This list exists so Streams can be associated with the CUC that
        initially requested them and can be used to restrict access to
        Streams, e.g., by using standard mechanisms as described in RFC
        8341.";
      leaf cuc-id {
        type string;
        description
          "The CUC ID is a unique identifier of a CUC. It is used to
          identify the CUC that a Stream belongs to, i.e., that
          requested the creation of a Stream.";
        reference
          "46.2.2.2 of IEEE Std 802.1Qdj-2023";
      }
      list stream {
        key "stream-id";
        description
```

```
        "List of Streams.

        Each Stream consists of a Stream ID, a request container, and
        a configuration container.

        In the fully centralized model of TSN configuration, the
        Stream ID and request originate from the CUC and is delivered
        to the CNC, while the configuration originates from the CNC
        and is delivered to the CUC.";
      leaf stream-id {
        type tsn:stream-id-type;
        description
          "The Stream ID is a unique identifier of a Stream request
          and corresponding configuration. It is used to associate a
          CUC's Stream request with a CNC's corresponding response.";
      }
      leaf stream-status {
        type enumeration {
          enum planned {
            value 0;
            description
              "The Stream has been requested but has not yet been
              configured by the CNC.";
          }
          enum configured {
            value 1;
            description
              "The Stream has been computed and configured by the
              CNC.";
          }
          enum modified {
            value 2;
            description
              "The Stream has been configured but Stream parameters
              have been modified after configuration.";
          }
        }
        config false;
        description
          "The stream-status indicates what status the Stream has in
          the CNC.";
        reference
          "46.2.3.8 of IEEE Std 802.1Qdj-2023";
      }
      container talker {
        description
          "The Talker container contains: - Talker's behavior for
          Stream (how/when transmitted) - Talker's requirements from
          the network - TSN capabilities of the Talker's
          interface(s).";
        uses tsn:group-talker;
        uses tsn:group-status-talker-listener {
          refine "accumulated-latency" {
            config false;
          }
          refine "interface-configuration" {
            config false;
          }
        }
```

```
    }
    list listener {
      key "index";
      description
        "Each Listener list entry contains: - Listener's
        requirements from the network - TSN capabilities of the
        Listener's interface(s).";
      leaf index {
        type uint32;
        description
          "This index is provided in order to provide a unique key
          per list entry.";
      }
      uses tsn:group-listener;
      uses tsn:group-status-talker-listener {
        refine "accumulated-latency" {
          config false;
        }
        refine "interface-configuration" {
          config false;
        }
      }
    }
    uses tsn:group-status-stream {
      refine "status-info" {
        config false;
      }
      refine "failed-interfaces" {
        config false;
      }
    }
  }
  action remove_streams {
    description
      "Removes the Streams with the ids provided in the stream-id
      list.";
    reference
      "46.2.8.1 of IEEE Std 802.1Qdj-2023";
    input {
      list stream-list {
        key "stream-id";
        description
          "List of stream-ids that are used to identify the Streams
          that are requested to be removed.";
        leaf stream-id {
          type tsn:stream-id-type;
          description
            "Unique identifier that is used to request a Stream
            that is to be removed from the configuration.";
        }
      }
    }
    output {
      leaf result {
        type string;
        description
          "Returns status information indicating if Stream removal
          has been successfully started.";
      }
```

```
        }
      }
    }
  }
}

// RPCs
rpc compute_streams {
  description
    "Starts computation of path and resource allocation for one or more
    Stream. The Streams that are included in the computation are the
    ones that have their domain-id, cuc-id, and stream-id provided.
    This RPC can be applied to compute new Streams as well as recompute
    Streams that have been modified.";
  input {
    list domain {
      key "domain-id";
      description
        "List of Configuration Domains.

        This list exists so CUCs can be associated with the
        Configuration Domain they are located in.";
      reference
        "46.2.7.1 of IEEE Std 802.1Qdj-2023";
      leaf domain-id {
        type leafref {
          path '/tsn-uni/domain/domain-id';
        }
        description
          "A unique identifier of a Configuration Domain. It is used to
          identify the Configuration Domain a CUC belongs to.";
      }
      list cuc {
        key "cuc-id";
        description
          "List of CUCs.

          This list exists so Streams can be associated with the CUC
          that initially requested them.";
        leaf cuc-id {
          type leafref {
            path '/tsn-uni/domain/cuc/cuc-id';
          }
          description
            "A unique identifier of a CNC. It is used to identify the
            CUC that a Streams belong to, i.e., that requested the
            creation of a Stream.";
        }
        list stream-list {
          key "stream-id";
          description
            "List of stream-ids that are used to identify the Streams
            that are requested to be computed and configured.";
          leaf stream-id {
            type leafref {
              path '/tsn-uni/domain/cuc/stream/stream-id';
            }
            description
              "Unique identifier that is used to request a Stream that
```

```
                is to be computed and configured.";
            }
          }
        }
      }
    }
    output {
      leaf result {
        type string;
        description
          "Only returns status information indicating if the computation
          has been started. It does not return status information on the
          success or failure of the actual Stream computation. A
          notifcation can be used to inform the caller of this RPC on the
          results of Stream computation after the computation has
          finished.";
      }
    }
  }
  rpc compute_planned_and_modified_streams {
    description
      "Starts computation of path and resource allocation for all Streams
      that are in the domain provided by domain-id and are associated
      with the CUC provided by cuc-id, and that have not been computed
      (i.e., that have a Stream status of planned or modified.";
    reference
      "46.2.7.2 of IEEE Std 802.1Qdj-2023";
    input {
      list domain {
        key "domain-id";
        description
          "List of Configuration Domains.

          This list exists so CUCs can be associated with the
          Configuration Domain they are located in.";
        leaf domain-id {
          type string;
          description
            "A unique identifier of a Configuration Domain. It is used to
            identify the Configuration Domain a CUC belongs to.";
        }
        list cuc {
          key "cuc-id";
          description
            "List of CUCs.

            This list exists so Streams can be associated with the CUC
            that initially requested them.";
          leaf cuc-id {
            type string;
            description
              "A unique identifier of a CNC. It is used to identify the
              CUC that a Streams belong to, i.e., that requested the
              creation of a Stream.";
          }
        }
      }
    }
    output {
```

```
      leaf result {
        type string;
        description
          "Only returns status information indicating if the computation
          has been started. It does not return status information on the
          success or failure of the actual Stream computation. A
          notifcation can be used to inform the caller of this RPC on the
          results of Stream computation after the computation has
          finished.";
      }
    }
  }
  rpc compute_all_streams {
    description
      "Starts computation of path and resource allocation for all Streams
      that are in the domain provided by domain-id and are associated
      with the CUC provided by cuc-id.";
    reference
      "46.2.7.3 of IEEE Std 802.1Qdj-2023";
    input {
      list domain {
        key "domain-id";
        description
          "List of Configuration Domains.

          This list exists so CUCs can be associated with the
          Configuration Domain they are located in.";
        leaf domain-id {
          type string;
          description
            "A unique identifier of a Configuration Domain. It is used to
            identify the Configuration Domain a CUC belongs to.";
        }
        list cuc {
          key "cuc-id";
          description
            "List of CUCs.

            This list exists so Streams can be associated with the CUC
            that initially requested them.";
          leaf cuc-id {
            type string;
            description
              "A unique identifier of a CNC. It is used to identify the
              CUC that a Streams belong to, i.e., that requested the
              creation of a Stream.";
          }
        }
      }
    }
    output {
      leaf result {
        type string;
        description
          "Only returns status information indicating if the computation
          has been started. It does not return status information on the
          success or failure of the actual Stream computation. A
          notifcation can be used to inform the caller of this RPC on the
          results of Stream computation after the computation has
```

```
          finished.";
      }
    }
  }
  rpc request_domain_id {
    description
      "Returns the DomainId of the Configuration Domain that the
      requesting CUC belongs to.";
    reference
      "46.2.7.4 of IEEE Std 802.1Qdj-2023";
    input {
      leaf cuc-id {
        type string;
        description
          "A unique identifier of a CNC. It is used to identify the CUC,
          allowing the CNC to return the DomainId this CUC belongs to.";
      }
    }
    output {
      leaf result {
        type string;
        description
          "Returns the DomainId of the Configuration Domain that the
          requesting CUC belongs to.";
      }
    }
  }
  rpc request_free_stream_id {
    description
      "Returns a free StreamId available for the Configuration Domain
      identified by the DomainId.";
    reference
      "46.2.7.5 of IEEE Std 802.1Qdj-2023";
    input {
      leaf domain-id {
        type string;
        description
          "A unique identifier of a Configuration Domain. It is used to
          identify the Configuration Domain a CUC belongs to.";
      }
      leaf cuc-id {
        type string;
        description
          "A unique identifier of a CNC. It is used to identify the CUC,
          allowing the CNC to return the DomainId this CUC belongs to.";
      }
    }
    output {
      leaf result {
        type string;
        description
          "Returns a free StreamId available for the Configuration Domain
          identified by the DomainId.";
      }
    }
  }

  // Notifications
  notification compute_streams_completed {
```

```
    description
      "Notifies the caller of an RPC or action that initiated the
      computation of one or multiple Streams, that the computation is
      complete. It also returns information on the success or failure for
      each of the Streams in the computation.";
    reference
      "46.2.9.1 of IEEE Std 802.1Qdj-2023";
    list domain {
      key "domain-id";
      description
        "List of Configuration Domains.

        This list exists so CUCs can be associated with the Configuration
        Domain they are located in.";
      leaf domain-id {
        type string;
        description
          "A unique identifier of a Configuration Domain. It is used to
          identify the Configuration Domain a CUC belongs to.";
      }
      list cuc {
        key "cuc-id";
        description
          "List of CUCs.

          This list exists so Streams can be associated with the CUC that
          initially requested them.";
        leaf cuc-id {
          type string;
          description
            "A unique identifier of a CNC. It is used to identify the CUC
            that a Stream belongs to, i.e., that requested the creation
            of a Stream.";
        }
        list stream {
          key "stream-id";
          description
            "List of Streams.

            Each Stream consists of a Stream ID, a request container, and
            a configuration container.

            In the fully centralized model of TSN configuration, the
            Stream ID and request originate from the CUC and is delivered
            to the CNC, while the configuration originates from the CNC
            and is delivered to the CUC.";
          reference
            "46.2.3 of IEEE Std 802.1Qdj-2023";
          leaf stream-id {
            type tsn:stream-id-type;
            description
              "The Stream ID is a unique identifier of a Stream request
              and corresponding configuration. It is used to associate a
              CUC's Stream request with a CNC's corresponding response.";
          }
          leaf failure-code {
            type uint8;
            description
              "A code that indicates if the computation for the Stream
```

```
            was successful (0) or not. In the case of a failure a code
            is returned to indicate what kind of failure occurred.";
        }
      }
    }
  }
}
notification configure_streams_completed {
  description
    "Notifies the caller of an RPC or action that initiated the
    computation of one or multiple Streams, that the computation and
    configuration is complete. It also returns information on the
    success or failure for each of the Streams in the computation and
    configuration.";
  reference
    "46.2.9.2 of IEEE Std 802.1Qdj-2023";
  list domain {
    key "domain-id";
    description
      "List of Configuration Domains.

      This list exists so CUCs can be associated with the Configuration
      Domain they are located in.";
    leaf domain-id {
      type string;
      description
        "A unique identifier of a Configuration Domain. It is used to
        identify the Configuration Domain a CUC belongs to.";
    }
    list cuc {
      key "cuc-id";
      description
        "List of CUCs.

        This list exists so Streams can be associated with the CUC that
        initially requested them.";
      leaf cuc-id {
        type string;
        description
          "A unique identifier of a CNC. It is used to identify the CUC
          that a Streams belong to, i.e., that requested the creation
          of a Stream.";
      }
      list stream {
        key "stream-id";
        description
          "List of Streams.

          Each Stream consists of a Stream ID, a request container, and
          a configuration container.

          In the fully centralized model of TSN configuration, the
          Stream ID and request originate from the CUC and is delivered
          to the CNC, while the configuration originates from the CNC
          and is delivered to the CUC.";
        reference
          "46.2.3 of IEEE Std 802.1Qdj-2023";
        leaf stream-id {
          type tsn:stream-id-type;
```

```
            description
              "The Stream ID is a unique identifier of a Stream request
              and corresponding configuration. It is used to associate a
              CUC's Stream request with a CNC's corresponding response.";
          }
          leaf failure-code {
            type uint8;
            description
              "A code that indicates if the computation and configuration
              for the Stream was successful (0) or not. In the case of a
              failure a code is returned to indicate what kind of failure
              occurred.";
          }
        }
      }
    }
  }
  notification remove_streams_completed {
    description
      "Notifies the caller of an RPC or action that initiated the removal
      of one or multiple Streams, that the removal is complete. It also
      returns information on the success or failure for each of the
      Streams in the removal request.";
    reference
      "46.2.9.3 of IEEE Std 802.1Qdj-2023";
    list domain {
      key "domain-id";
      description
        "List of Configuration Domains.

        This list exists so CUCs can be associated with the Configuration
        Domain they are located in.";
      leaf domain-id {
        type string;
        description
          "A unique identifier of a Configuration Domain. It is used to
          identify the Configuration Domain a CUC belongs to.";
      }
      list cuc {
        key "cuc-id";
        description
          "List of CUCs.

          This list exists so Streams can be associated with the CUC that
          initially requested them.";
        leaf cuc-id {
          type string;
          description
            "A unique identifier of a CNC. It is used to identify the CUC
            that a Streams belong to, i.e., that requested the creation
            of a Stream.";
        }
        list stream {
          key "stream-id";
          description
            "List of Streams.

            Each Stream consists of a Stream ID, a request container, and
            a configuration container.
```

```
1
2             In the fully centralized model of TSN configuration, the
3             Stream ID and request originate from the CUC and is delivered
4             to the CNC, while the configuration originates from the CNC
5             and is delivered to the CUC.";
6           reference
7             "46.2.3 of IEEE Std 802.1Qdj-2023";
8           leaf stream-id {
9             type tsn:stream-id-type;
10            description
11              "The Stream ID is a unique identifier of a Stream request
12              and corresponding configuration. It is used to associate a
13              CUC's Stream request with a CNC's corresponding response.";
14          }
15          leaf failure-code {
16            type uint8;
17            description
18              "A code that indicates if the removal of the Stream was
19              successful (0) or unsuccessful (1).";
20          }
21        }
22      }
23    }
24  }
25 }
26
27
```

Annex X

(informative)

Insert this informative annex at the end of IEEE 802.1Q, but prior to the Bibliography annex, re-lettering as necessary.

TSN features

TSN features are a set of protocols and mechanisms specified by IEEE 802 standards from which one can select the mechanisms that are best suited to meet the needs of the applications supported by a given network. These TSN mechanisms are add-ons to generic networking mechanisms in order to establish a common network that supports TSN Streams as well as other kinds of traffic. The goals of using TSN features typically include providing guaranteed data transport with low and bounded latency, low and bounded delay variation, and extremely low packet loss for TSN Streams. TSN features evolve and new capabilities are added as part of IEEE 802 standardization efforts. Therefore, the following list is incomplete and just provides a snapshot of TSN features:

a) Timing and Synchronization for Time-Sensitive Applications (IEEE Std 802.1AS-2020 [B1])
b) Credit-Based Shaper: (IEEE Std 802.1Q-2022, 5.4.1.5)
c) Frame Preemption (IEEE Std 802.3-2018 [B6] and IEEE Std 802.1Q-2022, 5.26)
d) Scheduled Traffic (IEEE Std 802.1Q-2022, 8.6.8.4)
e) Cyclic Queuing and Forwarding (IEEE Std 802.1Q-2022, 5.4.1.9)
f) Asynchronous Traffic Shaping (IEEE Std 802.1Q-2022, 5.4.1.10)
g) Per-Stream Filtering and Policing (IEEE Std 802.1Q-2022, 5.4.1.8)
h) Frame Replication and Elimination for Reliability (IEEE Std 802.1CB-2017 [B3])
i) Stream Reservation Protocol (IEEE Std 802.1Q-2022, clause 35.)
j) Link-local Registration Protocol (IEEE Std 802.1CS-2020 [B5])
k) Path Control and Reservation (IEEE Std 802.1Q-2022, 5.4.6)
l) TSN Configuration (IEEE Std 802.1Q-2022, 5.29)
m) Configuration Enhancements for Time-Sensitive Networking (IEEE Std 802.1Qdj-2023)

NOTE–There is no need to apply all the TSN features in a network and none of the TSN features are a requirement. The application area or actual deployment determine which TSN features are used in a given network, e.g., whether or not time synchronization is used. TSN profile standards, e.g., IEEE Std 802.1BA [B2] and IEEE Std 802.1CM [B4] select TSN features and give guidelines on their use in a particular application area.

Annex Y

(informative)

Insert the following bibliography references in alphanumeric order

Bibliography

[B1] IEEE Std 802.1AS™-2020, IEEE Standard for Local and Metropolitan Area Networks—Timing and Synchronization for Time Sensitive Applications.

[B2] IEEE Std 802.1BA™-2021, IEEE Standard for Local and Metropolitan Area Networks—Audio Video Bridging (AVB) Systems

[B3] IEEE Std 802.1CB™-2017, IEEE Standard for Local and Metropolitan Area Networks—Frame Replication and Elimination for Reliability.

[B4] IEEE Std 802.1CM™-2018, IEEE Standard for Local and Metropolitan Area Networks—Time-Sensitive Networking for Fronthaul.

[B5] IEEE Std 802.1CS™-2020, IEEE Standard for Local and Metropolitan Area Networks—Link-local Registration Protocol.

[B6] IEEE Std 802.3™-2020, IEEE Standard for Ethernet.