P802.1Qdj/D2.2
February 26, 2024

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

Draft Standard for Local and metropolitan area networks—

Bridges and Bridged Networks

Amendment 38: Configuration Enhancements for Time-Sensitive Networking

Prepared by the
Time-Sensitive Networking (TSN) Task Group of IEEE 802.1

Sponsor
LAN/MAN Standards Committee
of the
IEEE Computer Society

This and the following cover pages are not part of the draft. They provide revision and other information for IEEE 802.1 Working Group members and participants in the IEEE Standards Association ballot process, and will be updated as convenient. The text proper of this draft begins with the Title page.

Current draft

This draft, P802.1Qdj/D2.2, has been prepared for a Standards Association (SA) Recirculation Ballot following the sucessful Initial SA Ballot of P802.1Qdj/D2.0. The comments received on that initial ballot, and their disposition by the CRG (Comment Resolution Group), are available in myProject.

Changes are identified by change bars with the following exceptions:

- — The YANG schema and module incorporated in Clause 48 in the amendment are imported as complete UTF-8 .yang files without the possibility of inadvertent change, which rules out selective change bar marking. The updated ieee802-dot1q-types and ieee802-dot1q-bridge modules are attached as plain text files.
- — A small number of editing instruction changes, which will not affect use of the amendment text or its the rollup into a future revision of the standard have not been change barred. The exact phrasing of editing instructions fall within the scope of pre-publication editing. The present changes are intended to support subsequent and contemporaneous amendments with the minimum of further change (see comment on ‘Base standard’ below).

The complete text of the updated Bibliography has been included in this amendment. This inclusion follows guidance provided during pre-publication editing of prior amendments, and is intended to make it easier for contributors and reviewers to avoid omitting or duplicating references in or between the Bibliography and the Normative References clause.

Base standard

This draft is being submitted for SA Recirculation Ballot prior to completion of the SA Ballot process for another amendment (P802.1Qdx) to the base standard, IEEE Std 802.1Q. There is currently no technical overlap between these amendments, and the editorial interaction is limited to the numbering of minor clauses and figures which are added to the end of their enclosing clause. The current numbering in this draft anticipates the prior approval and pre-publication editing of this proposed standard, P802.1Qdj. The editing instructions for P802.1Qdx accomodate the possibility of further numbering changes as a result of any further numbering changes arising as a result of comment resolution and development of this draft.

MIB and YANG modules

The YANG modules specified by this standard are attached to the draft pdf as plain text (UTF-8) .yang files.

P802.1Qdj/D2.2
February 26, 2024

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

Draft Standard for Local and Metropolitan Area Networks—

Bridges and Bridged Networks

Amendment 38: Configuration Enhancements for Time-Sensitive Networking

Prepared by the
Time-Sensitive Networking (TSN) Task Group of IEEE 802.1

Sponsor
LAN/MAN Standards Committee
of the
IEEE Computer Society

IEEE Standards Department
445 Hoes Lane
Piscataway, NJ 08854, USA

Abstract: This amendment to IEEE Std 802.1Q-2022 as amended by IEEE Std 802.1Qcz-2023, IEEE Std 802.1Qcw-2023, and IEEE Std 802.1Qcj-2023 describes the relationships and division of responsibilities between Centralized User Configuration (CUC) components, that can be used to configure end stations' use of Time-Sensitive Networking (TSN) capabilities, and a Centralized Network Configuration (CNC) component that can be used to configure network resources within an administrative Configuration Domain. A YANG model and modules that can be used by a network configuration protocol, such as NETCONF, to provide communication between the CNC and a CUC is specified.

Keywords: Bridged Network, Centralized Network Configuration, CNC, Centralized User Configuration, CUC, IEEE 802.1Q™, IEEE 802.1Qdj™, LAN, local area network, Time-Sensitive Networking, TSN, Virtual Bridged Network, virtual LAN, VLAN Bridge, YANG.

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

PDF: ISBN 978-X-XXX-XXX-X STDXXXXX
Print: ISBN 978-X-XXX-XXX-X STDPDXXXXX

Important Notices and Disclaimers Concerning IEEE Standards Documents

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

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

IEEE Standards documents are developed within IEEE Societies and subcommittees of IEEE Standards Association (IEEE SA) Board of Governors. IEEE develops its standards through an accredited consensus development process, which brings together volunteers representing varied viewpoints and interests to achieve the final product. IEEE Standards are documents developed by volunteers with scientific, academic, and industry-based expertise in technical working groups. Volunteers are not necessarily members of IEEE or IEEE SA and participate without compensation from IEEE. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

IEEE makes no warranties or representations concerning its standards, and expressly disclaims all warranties, express or implied, concerning this standard, including but not limited to the warranties of merchantability, fitness for a particular purpose and non-infringement. In addition, IEEE does not warrant or represent that the use of the material contained in its standards is free from patent infringement. IEEE standards documents are supplied "AS IS" and "WITH ALL FAULTS."

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

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

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

Translations

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

Official statements

A statement, written or oral, that is not processed in accordance with the IEEE SA Standards Board Operations Manual shall not be considered or inferred to be the official position of IEEE or any of its committees and shall not be considered to be, nor be relied upon as, a formal position of IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that the presenter's views should be considered the personal views of that individual rather than the formal position of IEEE, IEEE SA, the Standards Committee, or the Working Group. Statements made by volunteers may not represent the formal position of their employer(s) or affiliation(s).

Comments on standards

Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE or IEEE SA. However, **IEEE does not provide interpretations, consulting information, or advice pertaining to IEEE Standards documents**.

Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Since IEEE standards represent a consensus of concerned interests, it is important that any responses to comments and questions also receive the concurrence of a balance of interests. For this reason, IEEE and the members of its Societies and subcommittees of the IEEE SA Board of Governors are not able to provide an instant response to comments, or questions except in those cases where the matter has previously been addressed. For the same reason, IEEE does not respond to interpretation requests. Any person who would like to participate in evaluating comments or in revisions to an IEEE standard is welcome to join the relevant IEEE working group. You can indicate interest in a working group using the Interests tab in the Manage Profile & Interests area of the IEEE SA myProject system.[1] An IEEE Account is needed to access the application.

Comments on standards should be submitted using the Contact Us form.[2]

Laws and regulations

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

Data privacy

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

Copyrights

IEEE draft and approved standards are copyrighted by IEEE under US and international copyright laws. They are made available by IEEE and are adopted for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making these documents available for use and adoption by public authorities and private users, neither IEEE nor its licensors waive any rights in copyright to the documents.

[1] Available at: https://development.standards.ieee.org/myproject-web/public/view.html#landing.
[2] Available at: https://standards.ieee.org/content/ieee-standards/en/about/contact/index.html.

Photocopies

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

Updating of IEEE Standards documents

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

Every IEEE standard is subjected to review at least every 10 years. When a document is more than 10 years old and has not undergone a revision process, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE standard.

In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit IEEE Xplore or contact IEEE.[3] For more information about the IEEE SA or IEEE's standards development process, visit the IEEE SA Website.

Errata

Errata, if any, for all IEEE standards can be accessed on the IEEE SA Website.[4] Search for standard number and year of approval to access the web page of the published standard. Errata links are located under the Additional Resources Details section. Errata are also available in IEEE Xplore. Users are encouraged to periodically check for errata.

Patents

IEEE Standards are developed in compliance with the IEEE SA Patent Policy.[5]

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

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

[3] Available at: https://ieeexplore.ieee.org/browse/standards/collection/ieee.
[4] Available at: https://standards.ieee.org/standard/index.html.
[5] Available at: https://standards.ieee.org/about/sasb/patcom/materials.html.

IMPORTANT NOTICE

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

Participants

<<The following lists will be updated in the usual way prior to publication>>

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 V. Rouyer, *Vice Chair*
János Farkas, *Chair, Time-Sensitive Networking Task Group*
Craig Gunther, *Vice 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

Introduction

This introduction is not part of IEEE Std 802.1Qdj™-2024, IEEE Standard for Local and Metropolitan Area Networks—Bridges and Bridged Networks—
Amendment 38: Configuration Enhancements for Time-Sensitive Networks.

IEEE Std 802.1Qdj™-2024: Configuration Enhancements for Time-Sensitive Networks describes the relationships and division of responsibilities between Centralized User Configuration (CUC) components, that can be used to configure end stations' use of Time-Sensitive Networking (TSN) capabilities, and a Centralized Network Configuration (CNC) component that can be used to configure network resources within an administrative Configuration Domain. This standard also includes a YANG model and modules that can be used by a network configuration protocol, such as NETCONF, to provide communication between the CNC and a CUC.

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

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

Contents

Figures

Tables

IEEE Standard for Local and Metropolitan Area Networks—

Bridges and Bridged Networks

Amendment 38: 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 802.1Qcw™-2023, and IEEE Std 802.1Qcj™-2023.)

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

The editing instructions are shown in ***bold italics***. Four editing instructions are used: change, delete, insert, and replace. ***Change*** is used to make corrections in existing text or tables. The editing instruction specifies the location of the change and describes what is being changed by using ~~strikethrough~~ (to remove old material) and <u>underscore</u> (to add new material). ***Delete*** removes existing material. ***Insert*** adds new material without disturbing the existing material. Deletions and insertions may require renumbering. If so, renumbering instructions are given in the editing instruction. ***Replace*** is used to make changes in figures or equations by removing the existing figure or equation and replacing it with a new one. Editing instructions, change markings, and this note will not be carried over into future editions because the changes will be incorporated into the base standard. [6]

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

1. Overview

1.3 Introduction

Change the paragraph beginning "This standard specifies enhancements to protocols, procedures, and managed objects for the configuration of network resources" as follows:

This standard specifies enhancements to protocols, procedures, and managed objects for the configuration of network resources for time-sensitive ~~(i.e., bounded latency)~~ applications that require timely, high probability, delivery of frames without end station retransmission. ~~The enhancements address Time-Sensitive Networking (TSN) application requirements beyond audio/video (AV) traffic.~~ To this end, it:

cm) ~~Specifies a software interface between the user (i.e., time-sensitive application) and network components, such that the user provides Stream requirements (e.g., for bounded latency), and the network configures resources from Talker to Listeners to meet those requirements. This user/network interface (UNI) is specified as an information model that can be applied to any protocol.~~

cn) Describes three approaches to network configuration: ~~Specifies three models for the UNI:~~ fully distributed, centralized network/distributed user, and fully centralized.

co) ~~Specifies enhancements to the Stream Reservation Protocol (SRP), using a new application version, MSRPv1. MSRPv1 integrates the UNI TLVs for the benefits of enhanced configuration. For compatibility, MSRPv1 translates to the previous version (MSRPv0).~~

cp) Specifies ~~enhancements to the~~ managed objects for forwarding and queuing enhancements for time-sensitive streams (FQTSS).

cq) ~~Specifies enhancements to the managed objects for SRP.~~

cr) Describes Centralized User Configuration (CUC) and Centralized Network Configuration (CNC) entities.

cs) Specifies managed objects for configuration of Bridges by a ~~Centralized Network Configuration (~~CNC~~) component~~.

ct) ~~Defines~~ Specifies YANG configuration and operational state models (Clause 48) in support of Scheduled Traffic, Frame Preemption, ~~and~~ Per-Stream Filtering and Policing, and CUC configuration

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 and management scheme, and a single administration.

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

5. Conformance

5.29 TSN CNC station requirements

Change 5.29, as follows:

This subclause (5.29) defines the conformance requirements for a station that supports the Time-Sensitive Networking (TSN) Centralized Network Configuration (CNC) requirements (46.1.3). The TSN CNC station component is implemented within an end station or Bridge.

A TSN CNC station implementation that conforms to the provisions of this standard shall:

a) Support the use of a remote management protocol. The TSN CNC claiming to support remote management shall state the following:
   1) Which remote management protocol standard(s) or specification(s) are supported for the client implementation.
   2) Which standard(s) or specification(s) for managed object definitions and encodings are supported for use by the remote management protocol.

b) Support the managed object definitions and encodings for Stream reservation remote management (12.32).

c) Support the use of at least one protocol for User/network configuration information that complies with the requirements for protocol integration defined in 46.2. The TSN CNC shall state which User/network configuration information protocol standard(s) or specification(s) are supported.

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.~~ If SRP (Clause 35) is supported by the TSN CNC for the User/network configuration information, the TSN CNC shall support MRP External Control (12.32.4).

<u>A TSN CNC station implementation that conforms to the provisions of this standard may:</u>

e) <u>Use the YANG modules identified in 46.3.</u>

46. Time-Sensitive Networking (TSN) configuration

Change the introductory text of Clause 46 as follows:

Time-Sensitive Networking (TSN) is a collection of features in IEEE 802.1 standards that provide the following:

— Time synchronization among Bridges and end stations
— Significant reduction in frame loss due to faults in network equipment
— Significant reduction in, or the elimination of, frame loss due to egress Port congestion
— Bounded latency

This clause provides specifications for the configuration of TSN features in a network. The configuration process begins when Talkers and Listeners pass their requirements to the network and proceeds with the configuration of TSN features in Bridges along a tree from each Talker to its Listener(s).

46.1 Overview of TSN configuration

Change 46.1 as follows:

46.1.1 ~~User/Network Interface (UNI)~~Streams, Talkers, and Listeners

TSN configuration uses the concept of a Stream that is transmitted by a Talker to one or more Listeners. The Talkers and Listeners are located within end stations.

This clause specifies configuration information ~~that is exchanged over a User/Network Interface (UNI). The user side of the interface represents~~ for Streams requested by Talkers and Listeners, or by network management entities acting on their behalf. ~~The network side of the interface represents the Bridges that transfer frames of the Stream from each Talker to its Listeners. Each user specifies~~ Stream requirements ~~for its data, but~~ are initially specified by, or for, Talker and Listener without detailed knowledge of the network. ~~The network~~ Network entities or administrators obtain~~s~~ these requirements ~~from users~~, analyze~~s~~ the topology and ~~TSN~~ capabilities of the Bridges, and configure~~s the~~ Bridges to meet ~~user~~ the requirements. The network returns the success or failure of each Stream's configuration to ~~the user~~ its Talker(s) and Listener(s).

46.1.2 ~~Modeling of user/network~~TSN configuration ~~information~~ protocols

A variety of protocols can be used for the exchange of configuration information ~~over the TSN UNI~~ (e.g., signaling protocols, remote network management protocols). ~~These protocols can exchange the configuration information as text or as binary fields. To enable flexible integration of TSN configuration into a variety of protocols, 46.2 specifies the TSN user/network configuration information in a manner that is independent of schema, encoding, or protocol.~~

Specific TSN-capable products list the ~~user/network~~ protocol that is supported as part of their conformance [e.g., 5.18.3, item c) in 5.29]. Each user/network protocol will specify a specific schema and/or encoding for the configuration information in 46.2. Examples of these protocols are described for each of the TSN configuration models in 46.1.3.

46.1.3 TSN configuration models

~~This subclause describes three~~Three models for TSN user/network configuration are described. These models provide an architectural context for subsequent specifications. Each model specification shows the logical flow of ~~user/network~~ configuration information between various entities in the network.

46.1.3.1 Fully distributed model

In the fully distributed model, the end stations that contain~~ users of Streams (i.e.,~~ Talkers and Listeners~~)~~ communicate the user requirements directly~~ over the TSN user/network protocol~~ to the neighboring Bridge. The network is configured in a fully distributed manner, without a centralized network configuration entity. The distributed network configuration is performed using a protocol that propagates~~ TSN user/network~~ configuration information along the active topology for the Stream (i.e., Bridges in the tree from Talker to Listeners).

As~~ user~~ Stream requirements propagate through each Bridge, management of the Bridge's resources is effectively performed locally. This local management is limited to the information that the Bridge has knowledge of and does not necessarily include knowledge of the entire network.

Figure 46-1 provides a graphical representation of the fully distributed model.

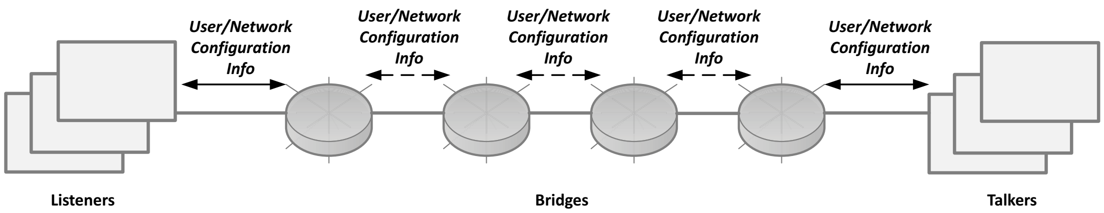

Figure 46-1—Fully distributed model

In the figure, the solid arrows represent the protocol that is used~~ as the UNI for~~ to exchange~~ of configuration~~ information between Talkers/Listeners (users) and Bridges (network). This~~ configuration~~ information is specified in 46.2.

In the figure, the dashed arrows represent the protocol that propagates~~ configuration~~ information within the network. This protocol carries~~ the TSN user/network~~ Stream configuration information (46.2) as well as additional information~~ that is~~ specific to network configuration.

The following TSN features can be configured by Bridges using this model:

a) Credit-based shaper algorithm (8.6.8.2) and its configuration (Clause 34)

The Stream Reservation Protocol (SRP) of Clause 35 can be used~~ as the UNI~~ by Talkers and Listeners, and to propagate configuration information throughout the network of Bridges.~~ SRP exchanges configuration information as binary fields using the Type-Length-Value (TLV) technique. Using this technique, the protocol's top-level message contains a list of one or more TLVs. Each TLV consists of a Type field that specifies what the Value field contains, a Length field that specifies the number of octets in the Value field, and the Value field. In SRP specifications, each TLV Type identifies one of the groups specified in 46.2, and the TLV Value contains a binary representation of the elements in that group.~~

46.1.3.2 Centralized network/distributed user model

Some TSN use cases are computationally complex. For example, for scheduled traffic (8.6.8.4), computation of the gate control list of each Port can take significant time. For such use cases, it is helpful to centralize the computation in a single entity (Bridge or end station), rather than perform the computation in all Bridges.

Some TSN use cases can benefit from a complete knowledge of all Streams in the network. For example, if the bandwidth for multiple Streams is greater than the available bandwidth along the shortest path between Talkers and Listeners, it is helpful to forward a subset of those Streams along a path other than the shortest. For these use cases, a centralized entity can gather information for the entire network in order to find the best configuration.

~~The~~ In 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~~ to the neighboring Bridge, just as in the fully distributed model. However, in the centralized network/distributed user model, a Centralized Network Configuration (CNC, 46.1.6) entity configures Bridges to pass Talker/ Listener Stream requirements directly to the CNC, rather than propagating that information along the path to be taken by Stream data. ~~In contrast, in the centralized network/distributed user model, the configuration information is directed to/from a Centralized Network Configuration (CNC~~, 46.1.6~~) entity.~~ All configuration of Bridges for ~~TSN~~ Streams is performed by this CNC using a remote network management protocol.

The CNC has a complete view of the physical topology of the network as well as the capabilities of each Bridge. This enables the CNC to centralize complex computations. The CNC can exist in either an end station or a Bridge.

~~The CNC knows the address of all Bridges at the edge of the network (those with an end station connected). The CNC configures those edge Bridges to act as a proxy, transferring Talker/Listener information directly between the edge Bridge and the CNC, rather than propagate the information to the interior of the network.~~

Figure 46-2 provides a graphical representation of the centralized network/distributed user model.

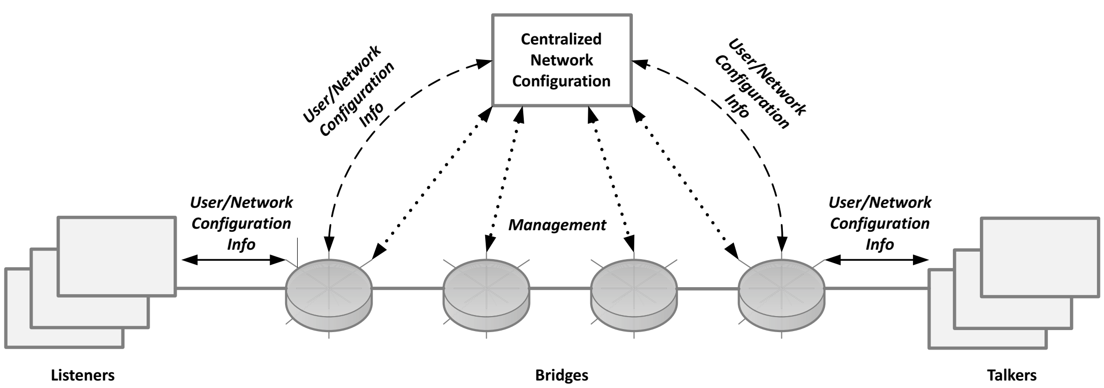

Figure 46-2—Centralized network/distributed user model

In the figure, the solid arrows represent the protocol that is used ~~as the UNI for~~ to exchange ~~of configuration~~ Stream information between Talkers/Listeners (users) and Bridges (network). This ~~configuration~~ information is specified in 46.2.

In the figure, the dashed arrows represent the protocol that transfers Stream configuration information between ~~edge~~ Bridges and the CNC. This ~~configuration~~ information is specified in 46.2.

In the figure, dotted arrows represent the remote network management protocol. The CNC acts as the management client, and each Bridge acts as the management server. The CNC uses remote management to discover physical topology, retrieve Bridge capabilities, and configure TSN features in each Bridge. Talkers and Listeners are not required to participate in this remote network management protocol. The information carried by the remote network management protocol is specified in Clause 12.

NOTE 1—If the Talker/Listener protocol of the fully distributed model is selected to be the same as the Talker/Listener protocol of the centralized network/distributed user model, end stations can support both models without explicit knowledge of how the network is configured.

The following TSN features can be configured by the CNC using this model:

a) Credit-based shaper algorithm (8.6.8.2) and its configuration (Clause 34)
b) Frame preemption (6.7.2)
c) Scheduled traffic (8.6.8.4, 8.6.9)
d) Frame Replication and Elimination for Reliability (IEEE Std 802.1CB)
e) Per-Stream Filtering and Policing (8.6.5.1)
f) Cyclic queuing and forwarding (Annex T)

SRP (Clause 35) can be used as the ~~UNI~~ Talker/Listener protocol (solid arrows of Figure 46-2). SRP's MRP External Control (12.32.4) feature can be used to exchange configuration information with the CNC component (dashed arrows of Figure 46-2). SRP exchanges configuration information using the TLV technique to reference elements in 46.2 (see 46.1.3.1). Examples of a remote network management protocol (dotted arrows of Figure 46-2) include Simple Network Management Protocol (SNMP), NETCONF (IETF RFC 6241 [B40]), and RESTCONF (IETF RFC 8040 [B48]).

NOTE 2—NETCONF and RESTCONF specify a startup datastore: nonvolatile configuration that is applied when the Bridge powers on. The startup datastore feature enables a ~~TSN~~ CNC to configure Bridges and then remove itself from the network. SNMP does not specify a startup datastore feature. If SNMP is used by a ~~TSN~~ CNC, this can be mitigated by a) using a proprietary (Bridge-specific) startup datastore feature or b) ensuring that the ~~TSN~~ CNC is always active in the network in order to reconfigure Bridges that cycle power.

46.1.3.3 Fully centralized model

Many TSN use cases require significant user configuration in the end stations that act as Talkers and Listeners. For example, in many automotive and industrial control applications, the timing of physical inputs and outputs (I/Os) is determined by the physical environment under control, and the timing requirements for ~~TSN~~ Streams are derived from that I/O timing. In some use cases, these I/O timing requirements can be computationally complex and involve detailed knowledge of the application software/hardware within each end station.

In order to accommodate this sort of TSN use case, the fully centralized model enables a Centralized User Configuration (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~~ not specified in this standard.

From a network perspective, the primary difference between the fully centralized model and the centralized network/distributed user model is that all user requirements are exchanged between the CNC and CUC. ~~Therefore, the TSN UNI exists between the CNC and CUC.~~

Figure 46-3 provides a graphical representation of the fully centralized model with multiple CUCs.

Replace Figure 46-3 with the following figure:

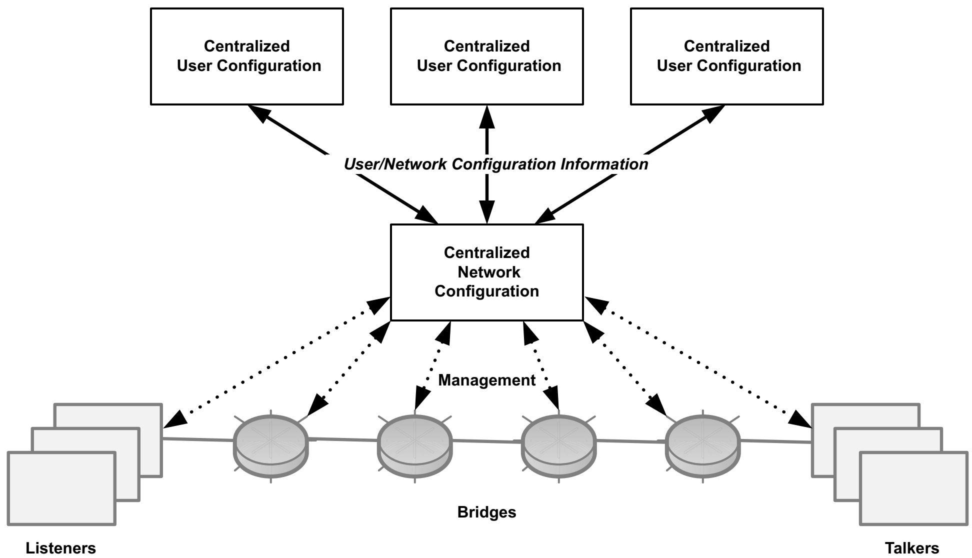

Figure 46-3—Fully centralized model

In the figure, the solid arrows represent the protocol that is used ~~as the UNI for~~ to exchange ~~of configuration~~ Stream information between the CUC and the CNC. This ~~configuration~~ information is specified in 46.2.

In the figure, the dotted arrows represent the remote network management protocol. The CNC acts as the management client, and each Bridge acts as the management server. The CNC uses remote management to discover physical topology, retrieve Bridge capabilities, and configure TSN features in each Bridge. Talkers and Listeners are not required to participate in this remote network management protocol. The information carried by the remote network management protocol is specified in Clause 12.

In this fully centralized model, a protocol is used between the CUC and end stations (Talkers and Listeners) to retrieve end station capabilities and requirements and to configure the end stations. ~~Since that protocol is user-to-user, its configuration information is considered to be outside the scope of this standard, and it is not shown in Figure 46-3.~~

The following TSN features can be configured by the CNC using this model:

a) Credit-based shaper algorithm (8.6.8.2) and its configuration (Clause 34)
b) Frame preemption (6.7.2)
c) Scheduled traffic (8.6.8.4, 8.6.9)
d) Frame Replication and Elimination for Reliability (IEEE Std 802.1CB)
e) Per-Stream Filtering and Policing (8.6.5.1)
f) Cyclic queuing and forwarding (Annex T)

YANG (IETF RFC 7950) is a data modeling language used to model configuration data and state data for remote network management protocols. The remote network management protocol uses a specific encoding such as XML or JSON. For a particular feature, a YANG module specifies the organization and rules for the feature's management data, and a mapping from YANG to the specific encoding enables the data to be understood correctly by both client (e.g., network manager) and server (e.g., Bridge). ~~Technically speaking, the TSN user/network configuration is not network management, in that information is exchanged between~~

~~user and network, and not between a network manager and the network's Bridges (Clause 12). Nevertheless, the concepts are sufficiently similar that YANG is useful for modeling the configuration and state data for the TSN user/network configuration information.~~

In order to support the use of YANG-based protocols for the fully centralized model, 46.3 specifies a YANG module. The YANG module specifies a YANG typedef/grouping for each group of information in 46.2.

~~NOTE—At the time that this clause was developed, specific protocol implementations for the fully centralized model were a work in progress. One protocol explored for the UNI between CUC and CNC is RESTCONF (IETF RFC 8040 [B48]). A complete YANG module for the TSN UNI can be specified in a document other than IEEE Std 802.1Q. In order to conform with this clause, the complete TSN UNI YANG module imports the YANG module of 46.3 for use within its containers and lists. The complete TSN UNI YANG module will presumably specify features outside the scope of this clause, such as operations to control the deployment of Stream configuration to the network. The JSON encoding can be used with RESTCONF. Although the TSN UNI is technically not network management, use of RESTCONF provides a simple and effective application programming interface (API) for TSN configuration.~~

For an informative example workflow using the fully centralized model, refer to U.2.

46.1.4 Stream identification and transformation

TSN configuration uses the concept of a Stream of data that is transmitted by a Talker to one or more Listeners.~~ The Talkers and Listeners are end stations.~~

In order to apply TSN behavior to a Stream~~s~~ (e.g., to reserve~~d~~ bandwidth~~ guarantees~~), the network~~ must be able to~~ distinguishes~~ one~~ Streams from~~ another Stream and distinguish Streams~~ each other and from non-TSN traffic (e.g., best-effort).~~ Therefore, each~~ Each frame of~~ the~~ a Stream~~ must~~ contains fields~~ in its header~~ that uniquely identify the Stream.

The goal of TSN configuration is to allow Talkers and Listeners to use their existing transport layer and application layer protocols for data, rather than requiring a TSN-specific frame format. TSN achieves this goal by identifying each Stream using fields from well-established frame formats such as Transmission Control Protocol (TCP), User Datagram Protocol (UDP), and IEEE 802.1 (i.e., MAC addresses and VLAN identifier).

As the frames of each Stream cross the user/network boundary, the identification of the Stream in its frames can be different between the network and the user. For example, the user can use UDP without an awareness of VLAN IDs, but the network can require a specific VLAN ID in order to apply TSN features. In order to support this sort of difference in frame format, the TSN user/network configuration information (46.2) provides features to enable transformation of the Stream's identification at the user/network boundary. The user identification translates to/from the network identification at the boundary, either within the end station or a nearest Bridge. This transformation has the benefit of allowing the user's identification to match its higher layer application protocol and the network's identification to match the bridging technology.

Stream transformation can be accomplished using the functions specified in IEEE Std 802.1CB. The functions of IEEE Std 802.1CB can be implemented in the end station (Talker/Listener) or within the nearest Bridge. The descriptions in this clause focus on Stream transformation in the end station and use features of the TSN user/network configuration information (46.2).

NOTE 1—In this clause, Stream transformation refers to changes to the fields of a frame that identify the Stream. IEEE Std 802.1CB specifies Stream transformation for identification as well as for frame replication and elimination (redundancy).

NOTE 2—Stream transformation is an optional capability of end stations and Bridges. If stream transformation is not supported, the user's identification of the Stream must be the same as the network's identification, and the user must use an identification that is consistent with bridging as specified for TSN features in this standard (e.g., VLAN tag and group destination MAC address).

Figure 46-4 provides an example of Stream transformation in the Talker end station. Stream transformation in the Listener end station is similar. The example of Figure 46-4 assumes use of the centralized network/distributed user model (46.1.3.2). Use of the fully centralized model (46.1.3.3) is similar.

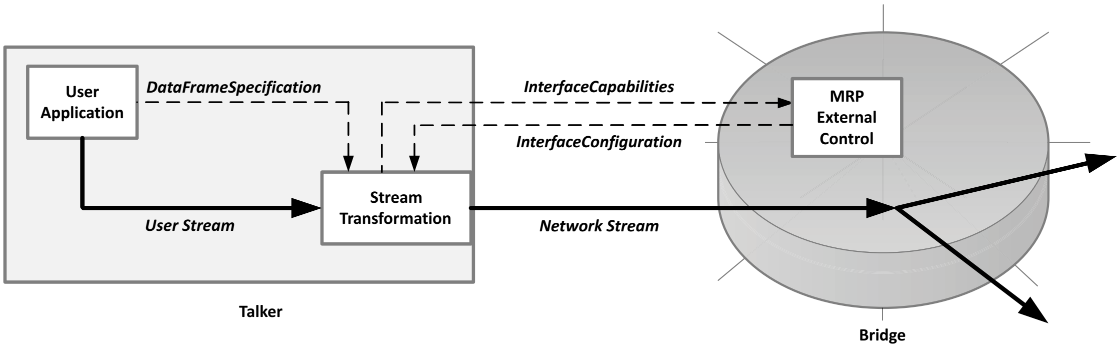

Figure 46-4—Example of Stream transformation in Talker end station

For Stream transformation in the end station, the end station's interface to the network can provide the transformation capability, acting as a network entity within the user's end station.

The identification of the Stream in the user originates from the User Application block, specified by the Talker as the DataFrameSpecification (46.2.3.4). The end station knows this user identification, but not the identification that the network requires. To negotiate the network identification, the Talker uses SRP to transmit InterfaceCapabilities (46.2.3.7) that describe its Stream transformation capabilities to the nearest Bridge. The Bridge uses MRP External Control (12.32.4) to send the InterfaceCapabilities to a CNC. The CNC consults its configuration of network identification and uses MRP External Control to send InterfaceConfiguration (46.2.5.3) along with successful Status (46.2.5) back to the Bridge. When the Bridge receives this information, it propagates back to the Talker using SRP. The InterfaceConfiguration provides the network identification, which the end station uses to perform Stream transformation for data frames.

NOTE 3—The network identification typically entails allocation of a group MAC address for the Stream. If a CNC is used, the CNC can allocate a group MAC address from a pool that it maintains.

Figure 46-5 provides an example of IEEE 802.1CB functions within the Stream Transformation block in the Talker end station. The example assumes that the user identification uses an Internet Protocol (IP) packet for identification and that the frame conveying the IP packet does not use the appropriate MAC address and VLAN tag for TSN features in Bridges (e.g., IP packet unicast destination MAC address, untagged). The IEEE 802.1CB function for IP Stream identification uses fields of the IP packet to identify the packet as a specific ~~TSN s~~Stream. That stream identification is then applied to the IEEE 802.1CB function for Active Destination MAC and VLAN Stream identification to replace the destination MAC address and add a

VLAN tag for TSN Bridge features. The IP fields of the packet are not changed. The IEEE 802.1CB functions can be implemented in software (e.g., operating system driver) or in hardware.

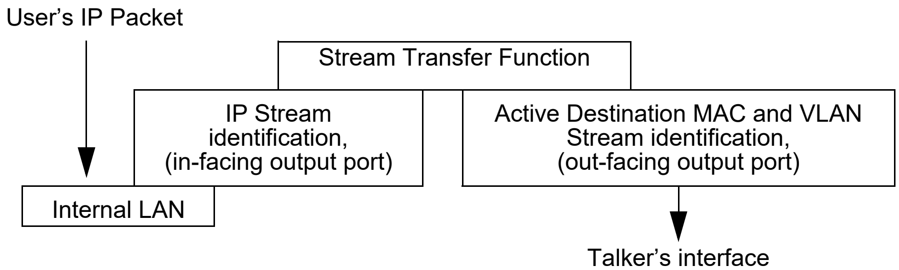

Figure 46-5—Example of IEEE 802.1CB functions in Talker end station

Figure 46-6 provides a corresponding example of IEEE 802.1CB functions within the Stream Transformation block in the Listener end station. In this direction, the IEEE 802.1CB function for Active Destination MAC and VLAN Stream identification provides both functions. The IEEE 802.1CB function uses the group destination MAC address and VLAN tag of the received frame to identify a specific Stream. The IEEE 802.1CB function then transforms the destination MAC address and VLAN tag to restore the Stream's frame to its original format (as transmitted by the Talker).

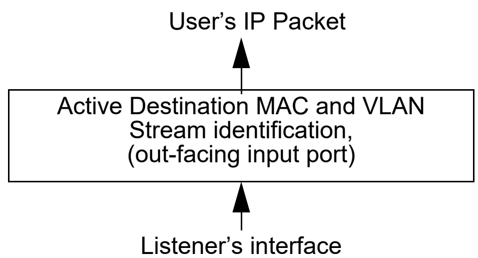

Figure 46-6—Example of IEEE 802.1CB functions in Listener end station

Insert 46.1.5, 46.1.6, and 46.1.7 after 46.1.4 as follows:

46.1.5 Centralized User Configuration

A 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 network system.

The CUC is responsible for:

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

NOTE–The CNC is responsible for the assignment of a unique StreamID to each Stream. For this a remote procedure call (RPC) RequestFreeStreamId (46.2.7.5) is available so the CUC can request a free (i.e., available) 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 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 (46.2). The CUC can request computation of paths and configurations for Streams in the following ways:

e) 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.
f) 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.
g) Request computation of the paths and configurations for all Streams of a CUC, using the protocol operation described in 46.2.7.3. The computation is performed by the CNC on all Streams in a Configuration Domain that belong to the CUC specified in the request.
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. Computation for the changes has to be triggered via RPC.
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. Computation for the changes has to be triggered via RPC.

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 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 for a 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. Filtering Database (8.8) 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 user/network configuration information specified in 46.2. It communicates with the stations using the managed objects defined in this and other IEEE 802.1 standards. There can only be one active CNC per Configuration Domain.

46.1.7 Configuration Domain

A Configuration Domain is a set of stations that are under a common configuration and management scheme, and a single administration. The Configuration Domain provides boundary information for the common management scheme and to support the responsibilities of the CUC and CNC regarding 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.

46.2 User/network configuration information

Change the introductory text of 46.2 as follows:

~~This subclause specifies the user/network configuration information that is used for the three TSN configuration models (46.1.3). The semantics for the TSN user/network configuration information is specified independent of schema, encoding, or protocol.~~

~~A schema or encoding of a protocol for the TSN user/network configuration information will reference 46.2 as part of its normative specifications.~~ <u>User/network configuration information for the three TSN configuration models (46.1.3) is specified as follows.</u> For the two distributed models, SRP specifies TLVs in 35.2.2.10 that reference information in 46.2. For the fully centralized model, the YANG module in 46.3 references information in 46.2.

Within this subclause, the word element refers to a single item of information used for TSN configuration. The word group refers to a collection of related elements. Groups are organized hierarchically, such that a group can be contained within another group. A single low-level group can be contained within multiple higher level groups. The dot-separated notation is used to refer to a specific element in text. For example, "Talker.StreamID.UniqueId" refers to the UniqueId element of the StreamID ~~group~~ that is contained within the Talker group.

This subclause specifies each group in a table. Each row of the table specifies an element of the group. Each element's row specifies its name, data type, and a reference to normative text that specifies its semantics (i.e., meaning). The data type of each element uses one of the values from 46.2.1. Other specifications such as conformance (i.e., required or optional), direction of transfer (i.e., user to network or network to user), default value, and range limitations are specified with normative text instead of with the table.

Each element name uses a camel-case naming convention (e.g., “MacAddress”) to align with naming conventions used in other clauses of IEEE Std 802.1Q. A specific protocol can use a different naming convention (e.g., 46.3) as long as the protocol’s name for the element can be associated with the element’s specification in 46.2.

46.2.1 Data types

Change 46.2.1 as follows:

~~The data type of each element is limited to semantics, independent of a specific encoding or protocol.~~ Data types in the tables include the following:

a) Boolean
b) int8, for a signed 8-bit integer
c) int16, for a signed 16-bit integer
d) int32, for a signed 32-bit integer
e) uint8, for an unsigned 8-bit integer
f) uint16, for an unsigned 16-bit integer
g) uint32, for an unsigned 32-bit integer
h) string
i) enumeration, for a collection of named values
j) rational, for a rational number consisting of a uint32 numerator and uint32 denominator
k) mac-address-type, for an IEEE 802 MAC address
l) ipv4-address-type, for an IPv4 address (IETF RFC 791)
m) ipv6-address-type, for an IPv6 address (IETF RFC 8200)
n) sequence of <X>, for a list of zero or more instances of data type <X> (e.g., sequence of uint32)

If the data type shown in the table is not from the preceding list, then the data type is specified in the normative text for the element.

46.2.2 Protocol integration

Change the paragraph beginning “Each TSN configuration protocol shall use the StreamID“ as follows:

~~Each TSN configuration protocol shall use the~~ The StreamID of ~~this clause (~~46.2.3.1~~)~~ uniquely identifies the Stream~~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.

Insert the following NOTE after the dashed list item beginning "— Response: Bridge":

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

Change the paragraph beginning “The protocol message(s) that invoke the join or leave operation“ 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~~ 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 46.2.2.1, 46.2.2.2, 46.2.2.3 as follows:

46.2.2.1 DomainID

DomainID uniquely identifies 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 and is used in configuration models that include a CNC. It is used along with the DomainID to associate Streams with a CUC.

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 paragraph beginning "The Talker group contains the following groups:" 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)
- — <u>StreamStatus (46.2.3.8)</u>

Insert the following sentence as a new paragraph, prior to 46.2.3.1:

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

Insert 46.2.3.8 and Table 46-12 as follows, 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 46.2.6, 46.2.7, 46.2.8, and 46.2.7 at the end of 46.2, as follows:

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 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 and configuration can take an arbitrary amount of time. The notifications

ComputeStreamsCompleted (46.2.9.1) and ConfigureStreamsCompleted (46.2.9.2) are available to the CNC to return information on success or failure of the Stream computation and configuration, after the actions have 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 and configuration can take an arbitrary amount of time. The notifications ComputeStreamsCompleted (46.2.9.1) and ConfigureStreamsCompleted (46.2.9.2) are available to the CNC to return information on success or failure of the Stream computation and configuration, after the actions have 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 and configuration can take an arbitrary amount of time. The notifications ComputeStreamsCompleted (46.2.9.1) and ConfigureStreamsCompleted (46.2.9.2) are available to the CNC to return information on success or failure of the Stream computation and configuration, after the actions have 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 (46.2.3.1) from a CNC. Requesting a free StreamID allows a CUC to provide an unused, i.e., unique, StreamID 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. Each Stream that is to be removed is specified by providing its associated StreamID (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:

- — ComputeStreamsCompleted (46.2.9.1)
- — ConfigureStreamsCompleted (46.2.9.2)
- — RemoveStreamsCompleted (46.2.9.3)

46.2.9.1 ComputeStreamsCompleted

This notification is used by the CNC to inform a CUC that has requested the computation of one or more 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 one or more 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 one or more 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 removal 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 modules.

~~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 specified in 48.6.3 [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~~ apply 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

48.2 IEEE 802.1Q YANG models

Insert 48.12.12 at the end of 48.2 as follows:

48.2.12 CNC-configuration model

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

The CNC-configuration model 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 and Figure 48-22.

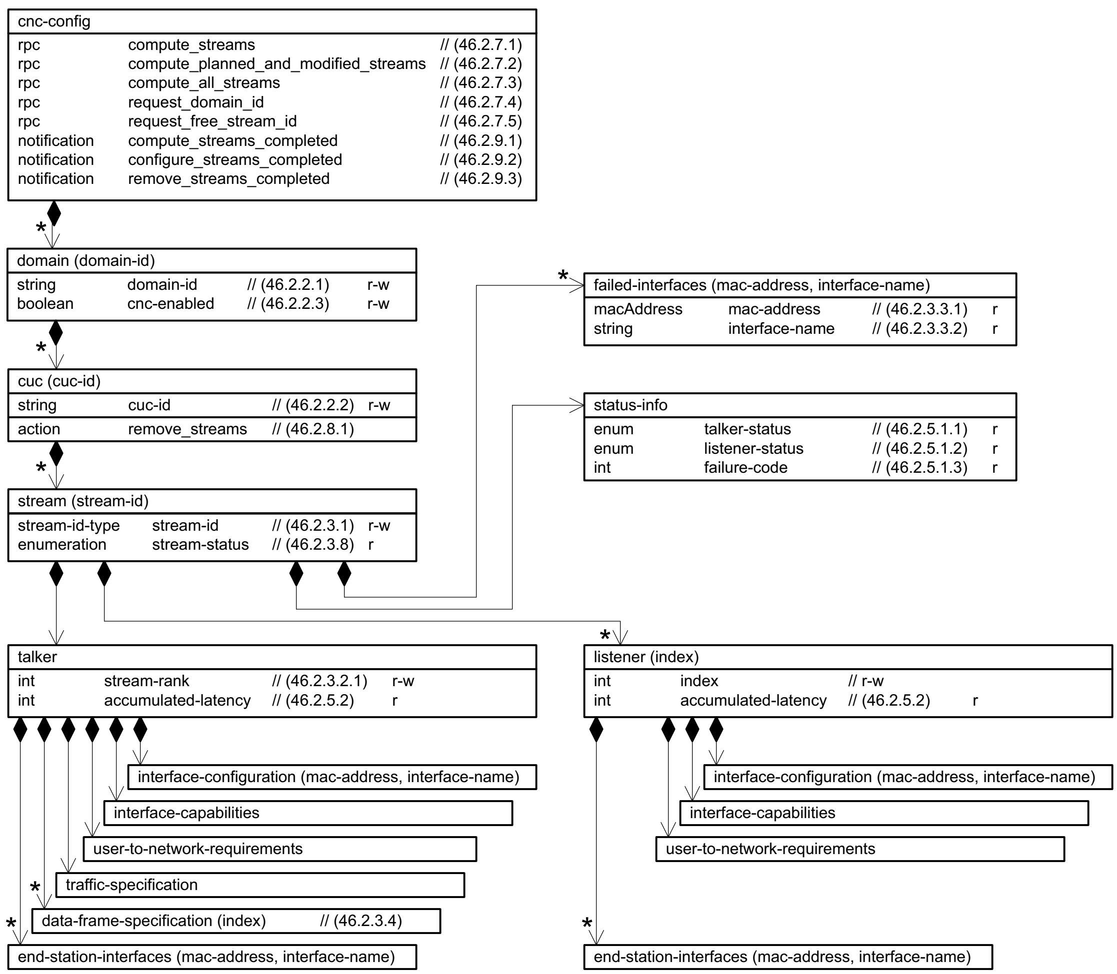

Figure 48-21—CNC-configuration model A

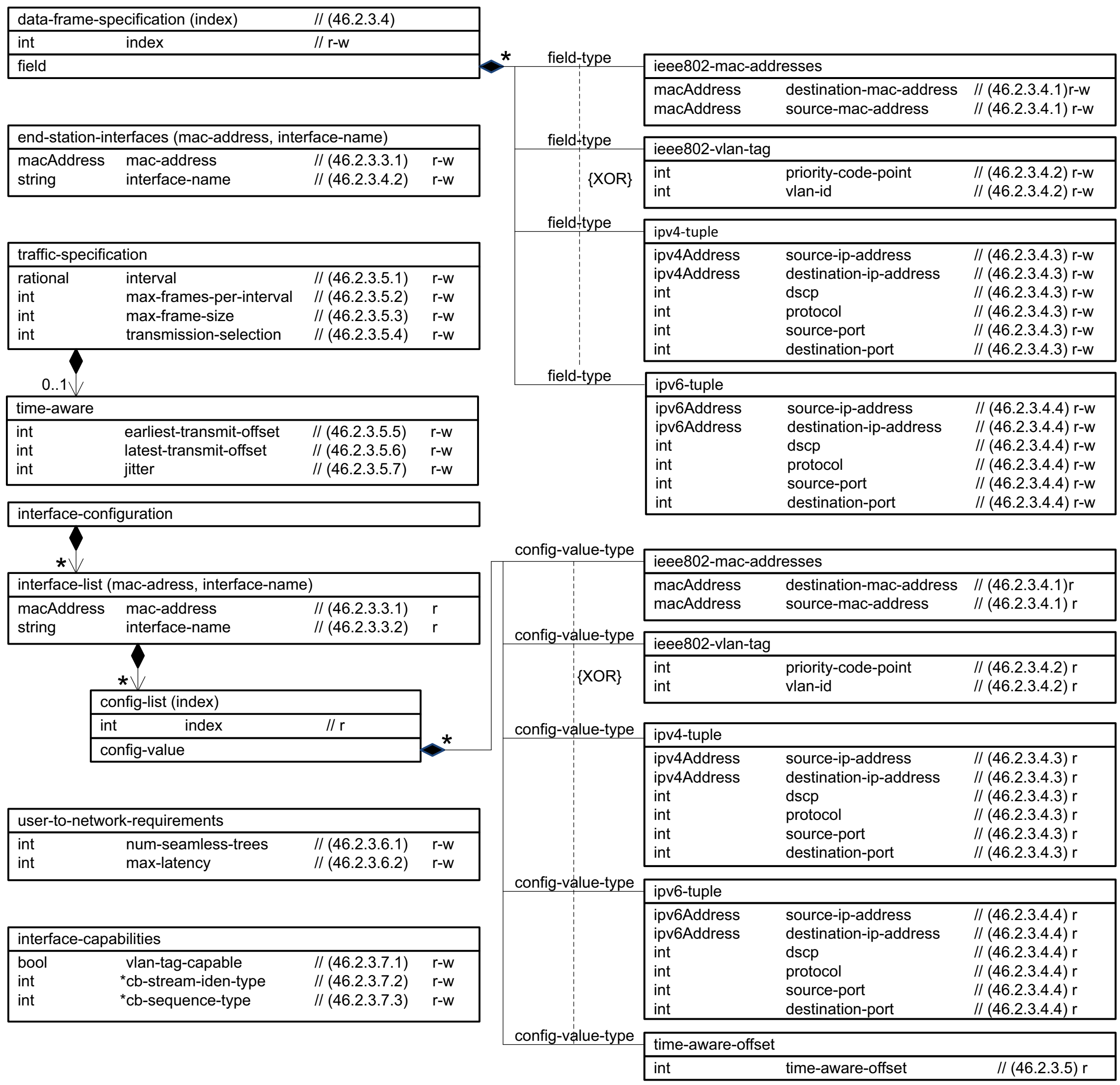

Figure 48-22—CNC-configuration model B

48.3 Structure of the YANG models

Insert the following row(s) at the end of Table 48-1 as follows (unchanged rows not shown):

Table 48-1—Summary of the YANG modules

Module	References	Managed functionality	Initial YANG specification Notes
ieee802-dot1q-cnc-config	48.5.23, 48.6.23	46.1.5, 46.1.6, 46.2	IEEE Std 802.1Qdj CNC configuration

Insert 48.3.12 and Table 48-13 at the end of 48.3 as follows:

48.3.12 User/Network Interface model

A station implementing the CNC-configuration model (48.2.12) implements the YANG modules in Table 48-13.

Table 48-13—CNC-configuration model YANG modules

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

48.4 Security considerations

Insert 48.4.12 at the end of 48.4, as follows:

48.4.12 Security considerations of the CNC-configuration model

The ieee802-dot1q-cnc-config YANG module is structured in a way that allows access to specific parts of the YANG tree to be restricted, e.g., by using standard mechanisms as described in IETF RFC 8341 [B49].

The following objects in the ieee802-dot1q-cnc-config YANG module could be manipulated to interfere with the operation of streams in a configuration domain and, for example, be used to cause network instability:

cnc-config/domain/cuc/stream

cnc-config/domain/cuc/remove_streams

48.5 YANG schema tree definitions

Insert 48.5.23, as follows:

48.5.23 Schema for the ieee802-dot1q-cnc-config YANG module

```
module: ieee802-dot1q-cnc-config
  +--rw cnc-config
     +--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    -> /cnc-config/domain/domain-id
  |  |     +---w cuc* [cuc-id]
  |  |        +---w cuc-id         -> /cnc-config/domain/cuc/cuc-id
  |  |        +---w stream-list* [stream-id]
  |  |           +---w stream-id
  |  |                   -> /cnc-config/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
```

48.6 YANG modules[7] [8] [9]

Insert 48.6.23 and the following YANG module at the end of 48.6:

48.6.23 The ieee802-dot1q-cnc-config YANG module

```
module ieee802-dot1q-cnc-config {
  yang-version "1.1";
  namespace urn:ieee:std:802.1Q:yang:ieee802-dot1q-cnc-config;
  prefix dot1q-cnc-config;
  import ieee802-dot1q-tsn-types {
    prefix tsn;
    reference
      "48.6.3 of IEEE Std 802.1Q";
  }
  organization
    "IEEE 802.1 Working Group";
  contact
    "WG-URL: http://ieee802.org/1/
    WG-EMail: stds-802-1-l@ieee.org

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

    E-mail: stds-802-1-chairs@ieee.org";
  description
    "This module supports management of a Time-Sensitive Networking (TSN)
    Centralized Network Configuration (CNC) component, its Configuration
    Domains - each supporting one or more Centralized User Configuration (CUC)
    components, and the Streams created by those CUCs. RPCs support
    path computation and resource allocation across all or specified
    Configuration Domains, CUCs, and Streams.

    Copyright (C) IEEE (2024).

    This version of this YANG module is part of IEEE Std 802.1Q; see
    the standard itself for full legal notices.";
  revision 2024-01-31 {
    description
      "Published as part of IEEE Std 802.1Qdj-2024.

      The following reference statement identifies each referenced IEEE
      Standard as updated by applicable amendments.";
    reference
      "IEEE Std 802.1Q Bridges and Bridged Networks:
       IEEE Std 802.1Q-2022, IEEE Std 802.1Qcz-2023, IEEE Std 802.1Qcw-2023,
       IEEE Std 802.1Qcj-2023, IEEE Std 802.1Qdj-2024.";
  }
  container cnc-config {
    description
      "Top-level container for the CNC module.";
    list domain {
      key "domain-id";
      description
        "A list of Configuration Domains, each supporting one or more CUCs.
        Access to Streams and resources associated with particular Configuration
        Domains can be restricted.";
      leaf domain-id {
        type string;
```

[7] Copyright release for YANG: Users of this standard may freely reproduce the YANG modules contained in this standard so that they can be used for their intended purpose.

[8] An ASCII version of each YANG module is attached to the PDF of this standard and can also be obtained from the IEEE 802.1 Website at https://1.ieee802.org/yang-modules/.

[9] References in this standard's YANG module definitions are not clickable, as each module has been incorporated unchanged after development and verification using YANG tools.

```
        description
          "The DomainID uniquely identifies a Configuration Domain.";
        reference
          "46.2.2.1 of IEEE Std 802.1Q";
      }
      leaf cnc-enabled {
        type boolean;
        default "false";
        description
          "cnc-enabled is used to enable or disable CNC functionality. If TRUE,
          the CNC functionality is enabled. If FALSE, the CNC functionality is
          disabled.";
        reference
          "46.2.2.3 of IEEE Std 802.1Q";
      }
      list cuc {
        key "cuc-id";
        description
          "A list of CUCs in the Configuration Domain, each with its associated
          Streams. Access to  Streams and resources associated with particular
          CUCs can be restricted.";
        leaf cuc-id {
          type string;
          description
            "The CucID uniquely identifies the CUC.";
          reference
            "46.2.2.2 of IEEE Std 802.1Q";
        }
        list stream {
          key "stream-id";
          description
            "A list of Streams created by the CUC, with their status and
            configuration (talker and listener parameters) in the CNC.";
          leaf stream-id {
            type tsn:stream-id-type;
            description
              "The StreamID uniquely identifies a Stream.";
          }
          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 is the status of the Stream in the CNC.";
            reference
              "46.2.3.8 of IEEE Std 802.1Q";
          }
          container talker {
            description
              "The Talker container for the Stream comprises the following:
              - The Talker’s behavior (how/when frames are transmitted).
              - The Talker’s network requirements.
              - The Talker's interface(s) TSN capabilities.";
            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
              "A list of parameters for each of the Stream's Listeners,
               each list entry comprising:
              - The Listener's interface(s).
              - The Listener’s network requirements.
              - The Listener’s interface(s) TSN capabilities.";
            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 selected Streams.";
          reference
            "46.2.8.1 of IEEE Std 802.1Q";
          input {
            list stream-list {
              key "stream-id";
              description
                "List of stream-ids for the Streams to be removed.";
              leaf stream-id {
                type tsn:stream-id-type;
                description
                  "The stream-id uniquely identifies a Stream.";
              }
            }
          }
          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
    Streams, each identified by its domain-id, cuc-id, and stream-id.
    This RPC can compute paths and resource allocations for both new and
    modified Streams.";
  input {
    list domain {
      key "domain-id";
      description
        "A list of Configuration Domains containing the Streams for which
        paths and resource allocations are to be computed.";
      reference
        "46.2.7.1 of IEEE Std 802.1Q";
      leaf domain-id {
        type leafref {
          path '/cnc-config/domain/domain-id';
        }
        description
          "The referenced DomainID uniquely identifies a Configuration
          Domain.";
      }
      list cuc {
        key "cuc-id";
        description
          "A list of CUCs in the Configuration Domain, each with Streams whose
          paths and resource allocations are to be computed. ";
        leaf cuc-id {
          type leafref {
            path '/cnc-config/domain/cuc/cuc-id';
          }
          description
            "The referenced CucID uniquely identifies a CUC.";
        }
        list stream-list {
          key "stream-id";
          description
            "A list of Streams, created by the CUC, whose paths and resource
            allocations are to be computed.";
          leaf stream-id {
            type leafref {
              path '/cnc-config/domain/cuc/stream/stream-id';
            }
            description
              "The referenced StreamID uniquely identifies 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
        notification 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 specified CUCs in specified domains, 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 802.1Q";
  input {
```

```
      list domain {
        key "domain-id";
        description
          "A list of Configuration Domains containing the CUCs with Streams
          whose paths and resource allocations are to be computed.";
        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
            "A list of CUCs, in the Configuration Domain, that have Streams
            whose paths and resource allocations are to be computed.";
          leaf cuc-id {
            type string;
            description
              "The CucID uniquely identifies the CUC.";
          }
        }
      }
    }
    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
          notification can be used to inform the caller of this RPC of 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 specified CUCs in specified domains.";
    reference
      "46.2.7.3 of IEEE Std 802.1Q";
    input {
      list domain {
        key "domain-id";
        description
          "A list of Configuration Domains containing the CUCs with Streams
          whose paths and resource allocations are to be computed";
        leaf domain-id {
          type string;
          description
            "The DomainID uniquely identifies a Configuration Domain.";
        }
        list cuc {
          key "cuc-id";
          description
            "A list of CUCs in the Configuration Domain, each with Streams whose
            paths and resource allocations are to be computed.";
          leaf cuc-id {
            type string;
            description
              "The CucID uniquely identifies the CUC.";
          }
        }
      }
    }
    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
          notification can be used to inform the caller of this RPC of the
          results of Stream computation after the computation has
          finished.";
      }
    }
  }
  rpc request_domain_id {
    description
      "Returns the DomainID of the Configuration Domain to which a specified
      CUC belongs.";
    reference
      "46.2.7.4 of IEEE Std 802.1Q";
    input {
      leaf cuc-id {
        type string;
        description
          "The CucID uniquely identifies the CUC.";
      }
    }
    output {
      leaf result {
        type string;
        description
          "Returns the DomainID of the Configuration Domain for the
          specified CUC.";
      }
    }
  }
  rpc request_free_stream_id {
    description
      "Returns a free (i.e., available) StreamID for use by a specified
      CUC in a specified Configuration Domain.";
    reference
      "46.2.7.5 of IEEE Std 802.1Q";
    input {
      leaf domain-id {
        type string;
        description
          "The DomainID uniquely identifies the Configuration Domain.";
      }
      leaf cuc-id {
        type string;
        description
          "The CucID uniquely identifies the CUC.";
      }
    }
    output {
      leaf result {
        type string;
        description
          "Returns a free (i.e., available) StreamID for use by the
          specified CUC in the specified Configuration Domain.";
      }
    }
  }

  // Notifications
  notification compute_streams_completed {
    description
      "Notifies the caller of an RPC or action that initiated computation for
      one or more Streams, that the computation is complete. Returns information
      on the success or failure of computation for each of those Streams.";
    reference
      "46.2.9.1 of IEEE Std 802.1Q";
    list domain {
      key "domain-id";
      description
        "The list of Configuration Domains for which computation was requested,
        with each list entry specifying the CUCs, and the Streams for each of
        those CUCs for which computation was requested, with the result of the
```

```
        computation (success or failure) for each Stream.";
      leaf domain-id {
        type string;
        description
          "The DomainID uniquely identifies a Configuration Domain.";
      }
      list cuc {
        key "cuc-id";
        description
          "The list of CUCs for the specified Configuration Domain, with each
          list entry specifying the Streams for which computation was requested
          and the result of the computation for each Stream.";
        leaf cuc-id {
          type string;
          description
            "The CucID uniquely identifies the CUC.";
        }
        list stream {
          key "stream-id";
          description
            "The list of Streams, for which computation was requested, for the
            specified Configuration Domain and CUC, with the result of the
            computation (success or failure) for each Stream.";
          reference
            "46.2.3 of IEEE Std 802.1Q";
          leaf stream-id {
            type tsn:stream-id-type;
            description
              "The StreamID uniquely identifies a Stream.";
          }
          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 computation for
      one or more Streams, that computation and configuration is complete.
      Returns information on the success or failure information of computation
      and configuration for each of those Streams.";
    reference
      "46.2.9.2 of IEEE Std 802.1Q";
    list domain {
      key "domain-id";
      description
        "The list of Configuration Domains for which computation was requested,
        with each list entry specifying the CUCs, and the Streams for each of
        those CUCs for which computation was requested, with the result of the
        computation (success or failure) for each Stream.";
      leaf domain-id {
        type string;
        description
          "The DomainID uniquely identifies a Configuration Domain.";
      }
      list cuc {
        key "cuc-id";
        description
          "The list of CUCs for the specified Configuration Domain, with each
          list entry specifying the Streams for which computation was requested
          and the result of the computation for each Stream.";
        leaf cuc-id {
          type string;
          description
            "The CucID uniquely identifies the CUC.";
        }
```

```
        list stream {
          key "stream-id";
          description
            "The list of Streams, for which computation was requested, for the
            specified Configuration Domain and CUC, with the result of the
            computation (success or failure) for each Stream.";
          reference
            "46.2.3 of IEEE Std 802.1Q";
          leaf stream-id {
            type tsn:stream-id-type;
            description
              "The StreamID uniquely identifies a Stream.";
          }
          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 computation for
      one or more Streams, that the removal is complete. Returns information
      on the success or failure of removal for each of those Streams.";
    reference
      "46.2.9.3 of IEEE Std 802.1Q";
    list domain {
      key "domain-id";
      description
        "A list of Configuration Domains with each list entry specifying the
        CUCs, and the Streams for each of those CUCs for which removal was
        requested, with the result of the removal attempt (success or failure)
        for each Stream.";
      leaf domain-id {
        type string;
        description
          "The DomainID uniquely identifies a Configuration Domain.";
      }
      list cuc {
        key "cuc-id";
        description
          "The list of CUCs for the specified Configuration Domain, with each
          list entry specifying the Streams for which removal was requested
          and the result of the removal attempt for each Stream.";
        leaf cuc-id {
          type string;
          description
            "The CucID uniquely identifies the CUC.";
        }
        list stream {
          key "stream-id";
          description
            "The list of Streams, for which removal was requested, for the
            specified Configuration Domain and CUC, with the result of the
            removal (success or failure) for each Stream.";
          reference
            "46.2.3 of IEEE Std 802.1Q";
          leaf stream-id {
            type tsn:stream-id-type;
            description
              "The StreamID uniquely identifies a Stream.";
          }
          leaf failure-code {
            type uint8;
            description
              "A code that indicates if the removal of the Stream was
```

```
              successful (0) or unsuccessful (1).";
          }
        }
      }
    }
  }
}
```

Annex A

(normative)

PICS proforma—Bridge implementations[1]

[1] *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.

A.49 TSN Centralized Network Configuration (CNC) station

Change A.49 as follows:

Item	Feature	Status	References	Support
	If the functionality of a Centralized Network Configuration station (CNC-S of A.5) is not supported, mark N/A and ignore the remainder of this table.			N/A [ ]
CNC-S-1	What management protocol standard(s) or specification(s) are supported (client side)?	M	5.29 item a)	____________ ____________ ____________
CNC-S-2	Does the implementation support the managed object definitions and encodings for Stream reservation remote management?	M	5.29 item b), 12.32	Yes [ ]
CNC-S-3	Does the implementation support the managed object definitions and encodings for scheduled traffic?	O	12.29	Yes [ ] No [ ]
CNC-S-4	Does the implementation support the managed object definitions and encodings for frame preemption?	O	12.30	Yes [ ] No [ ]
CNC-S-5	Does the implementation support the managed object definitions and encodings for IEEE Std 802.1AS?	O	IEEE Std 802.1AS	Yes [ ] No [ ]
CNC-S-6	Does the implementation support the managed object definitions and encodings for IEEE Std 802.1CB?	O	IEEE Std 802.1CB	Yes [ ] No [ ]
CNC-S-7	Does the implementation support MRP External Control for the MSRP application?	O	12.32.4	Yes [ ] No [ ]
CNC-S-8	Does the implementation support MRP External Control for the MVRP application?	O	12.32.4	Yes [ ] No [ ]
CNC-S-9	Does the implementation support MRP External Control for the MMRP application?	O	12.32.4	Yes [ ] No [ ]
CNC-S-10	What user/network configuration protocol standard(s) or specification(s) are supported?	M	5.29 item c), 46.2.2	____________ ____________ ____________
CNC-S-11	Does the implementation ~~conform to the conditional requirements for use of a YANG-based protocol~~support the YANG modules identified in 46.3?	~~M~~O	5.29 item e), 46.2, 46.3	Yes [ ]
CNC-S-13	Does the implementation conform to the conditional requirements for use of SRP?	M	5.29 item d), 46.2, 12.32.4	Yes [ ]

Annex X

(informative)

Bibliography

Change Annex X as follows:

Bibliographical references are resources that provide additional or helpful material but do not need to be understood or used to implement this standard. Reference to these resources is made for informational use only.

[B1] Alizadeh, M., B. Atikoglu, A. Kabbani, A. Lakshmikantha, R. Pan, B. Prabhakar, and M. Seaman, "Data Center Transport Mechanisms: Congestion Control Theory and IEEE Standardization," *Proceedings of the 46th Annual Allerton Conference on Communication, Control and Computing*, Urbana-Champaign, Sept. 2008.

[B2] Asynchronous Transfer Mode (ATM): A collection of equipment and standards used for telecommunications and data transfer, https://www.itu.int/ITU-T/ and https://www.broadband-forum.org.

[B3] Calculating the Delay Added by Qav Stream Queue, https://www.ieee802.org/1/files/public/docs2009/av-fuller-queue-delay-calculation-0809-v02.pdf.

[B4] Duato, J., "A Necessary and Sufficient Condition for Deadlock-Free Routing in Cut-Through and Store-and-Forward Networks," *IEEE Transactions on Parallel and Distributed Systems*," vol. 7, no. 8, pp. 841–854, Aug. 1996. doi:10.1109/71.532115.

[B5] Hu, S., Y. Zhu, P. Cheng, C. Guo, K. Tan, J. Padhye, and K. Chen "Tagger: Practical PFC Deadlock Prevention in Data Center Networks," *Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies (CoNEXT '17)*, ACM, New York, NY, USA, pp. 451–463. doi:10.1145/3143361.3143382.

[B6] IEC 62439-3:2016, Industrial communications networks—High availability automation networks—Part 3: Parallel Redundancy Protocol (PRP) and High-availability Seamless Redundancy (HSR).[2]

[B7] IEEE Std 802®-2014, IEEE Standard for Local and Metropolitan Area Networks—Overview and Architecture.[3, 4]

[B8] IEEE Std 802.1AB™-2005, IEEE Standard for Local and metropolitan area networks—Station and Media Access Control Connectivity Discovery.

[B9] IEEE Std 802.1AB™-2009, IEEE Standard for Local and metropolitan area networks—Station and Media Access Control Connectivity Discovery.

[B10] IEEE Std 802.1AC™-2016, IEEE Standard for Local and metropolitan area networks—Media Access Control (MAC) Service Definition.

[B11] IEEE Std 802.1BA™, IEEE Standard for Local and Metropolitan Area Networks—Audio Video Bridging (AVB) Systems.

[2] IEC publications are available from the International Electrotechnical Commission (https://www.iec.ch) and the American National Standards Institute (https://www.ansi.org/).
[3] The IEEE standards or products referred to in Annex X are trademarks owned by The Institute of Electrical and Electronics Engineers, Incorporated.
[4] IEEE publications are available from The Institute of Electrical and Electronics Engineers (https://standards.ieee.org/).

[B12] IEEE Std 802.1CM™-2018, IEEE Standard for Local and Metropolitan Area Networks—Time-Sensitive Networking for Fronthaul.

[B13] IEEE Std 802.1D™, 1993 Edition [ISO/IEC 10038:1993], IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local area networks—Media Access Control (MAC) bridges.

[B14] IEEE Std 802.1D™, 1998 Edition [ISO/IEC 15802-3:1998], IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Common specifications—Part 3: Media Access Control (MAC) Bridges.

[B15] IEEE Std 802.1D™-2004, IEEE Standard for Local and metropolitan area networks—Media Access Control (MAC) Bridges.

[B16] IEEE Std 802.3™-2018, IEEE Standard for Ethernet.

[B17] IEEE Std 802.11™-2016, IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.

[B18] IEEE Std 1588-2019, IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurements and Control Systems.

[B19] IEEE Std 1722™, IEEE Standard for a Transport Protocol for Time-Sensitive Applications in Bridged Local Area Networks.

NOTE—See also IEC 61883-6:2005, Consumer audio/video equipment—Digital Interface—Part 6: Audio and music data transmission protocol and IEC 61883-4:2004, Consumer audio/video equipment—Digital Interface—Part 4: MPEG2-TS data transmission.

[B20] IETF RFC 793 (STD0007), Transmission Control Protocol, September 1981.[5]

[B21] IETF RFC 1321, The MD5 Message-Digest Algorithm, April 1992.

[B22] IETF RFC 1633, Integrated Services in the Internet Architecture: an Overview, June 1994.

[B23] IETF RFC 2210, The Use of RSVP with IETF Integrated Services, September 1997.

[B24] IETF RFC 2211, Specification of the Controlled-Load Network Element Service, September 1997.

[B25] IETF RFC 2212, Specification of Guaranteed Quality of Service, September 1997.

[B26] IETF RFC 2215, General Characterization Parameters for Integrated Service Network Elements, September 1997.

[B27] IETF RFC 2475, An Architecture for Differentiated Services, December 1998.

[B28] IETF RFC 2597, Assured Forwarding PHB Group, June 1999.

[B29] IETF RFC 2814, SBM (Subnet Bandwidth Manager): A Protocol for RSVP-based Admission Control over IEEE 802-style Networks, May 2000.

[B30] IETF RFC 2815, Integrated Service Mappings on IEEE 802 Networks, May 2000.

[B31] IETF RFC 2816, A Framework for Integrated Services Over Shared and Switched IEEE 802 LAN Technologies, May 2000.

[5] IETF RFCs are available from the Internet Engineering Task Force (https://www.ietf.org/).

[B32] IETF RFC 3031, Multiprotocol Label Switching Architecture, January 2001.

[B33] IETF RFC 3246, An Expedited Forwarding PHB (Per-Hop Behavior), March 2002.

[B34] IETF RFC 3270, Multi-Protocol Label Switching (MPLS) Support of Differentiated Services, May 2002.

[B35] IETF RFC 4655, A Path Computation Element (PCE)-Based Architecture, August 2006.

[B36] IETF RFC 4663, Transferring MIB Work from IETF Bridge MIB WG to IEEE 802.1 WG, September 2006.

[B37] IETF RFC 4960, Stream Control Transmission Protocol, September 2007.

[B38] IETF RFC 5306, Restart Signaling for IS-IS, October 2008.

[B39] IETF RFC 6087, Guidelines for Authors and Reviewers of YANG Data Model Documents, January 2011.

[B40] IETF RFC 6241, Network Configuration Protocol (NETCONF), June 2011.

[B41] IETF RFC 6242, Using the NETCONF Protocol over Secure Shell (SSH), June 2011.

[B42] IETF RFC 6536, Network Configuration Protocol (NETCONF) Access Control Model, March 2012.

[B43] IETF RFC 6762, Multicast DNS, February 2013.

[B44] IETF RFC 7223, A YANG Data Model for Interface Management, May 2014.

[B45] IETF RFC 7319, IANA Considerations for Connectivity Fault Management (CFM) Code Points, July 2014.

[B46] IETF RFC 7567, IETF Recommendations Regarding Active Queue Management, July 2015.

[B47] IETF RFC 8014, An Architecture for Data-Center Network Virtualization over Layer 3 (NVO3), December 2016.

[B48] IETF RFC 8040, RESTCONF Protocol, January 2017.

[B49] IETF RFC 8341, Network Configuration Access Control Model, March 2018, https://datatracker.ietf.org/doc/rfc8341/

[B48] ITU-T Recommendation G.806, Characteristics of transport equipment—Description methodology and generic functionality.[6]

[B49] ITU-T Recommendation G.8031/Y.1342, Ethernet linear protection switching.

[B50] ITU-T Recommendation I.610 (02/1999), B-ISDN operation and maintenance principles and functions.

[B51] ITU-T Recommendation X.25 (10/1996), Interface between Data Terminal Equipment (DTE) and Data Circuit-terminating Equipment (DCE) for terminals operating in the packet mode and connected to public data networks by dedicated circuit.

[B52] MEF Technical Specification 4 (MEF 4), Metro Ethernet Network Architecture Framework—Part 1: Generic Framework, May 2004.[7]

[6] ITU-T publications are available from the International Telecommunications Union (https://www.itu.int/).
[7] MEF publications are available from the MEF Forum (https://www.mef.net/).

[B53] MEF Technical Specification 16 (MEF 16), Ethernet Local Management Interface (E-LMI), January 2006.

[B54] MEF Technical Specification 26 (MEF 26), External Network Network Interface (ENNI)—Phase 1, January 2010.

[B55] MEF 35.1, Service OAM Performance Monitoring Implementation Agreement.

[B56] MoCA MAC/PHY Specification Extensions v1.1, MoCA-M/P-SPEC-V1.1-06162009, June 2009.[8]

[B57] MoCA MAC/PHY Specification v2.0, MoCA_Specification_v2-131121, November 2013.

[B58] Multiprotocol Label Switching (MPLS): A standard for label-based forwarding in an IP network. The standard is specified in several RFCs, (see https://datatracker.ietf.org/doc/charter-ietf-mpls/) and ITU-T recommendations (see https://www.itu.int/ITU-T/).

[B59] OMG Unified Modeling Language (OMG UML), Version 2.5.1, December 2017.

[B60] Rocher-Gonzalez, J., J. Escudero-Sahuquillo, P. J. Garcia, F. J. Quiles, "On the Impact of Routing Algorithms in the Effectiveness of Queuing Schemes in High-Performance Interconnection Networks," *2017 IEEE 25th Annual Symposium on High-Performance Interconnects (HOTI)*, Santa Clara, CA, 2017, pp. 65–72. doi:10.1109/HOTI.2017.16.

[B61] Seaman, M., "Preemption and MACsec replay protection*,"* available at https://www.ieee802.org/1/files/public/docs2014/ae-seaman-preemption-1114-v04.pdf, November 2014.

[B62] SMPTE 259M-2008, SMPTE Standard for Television—SDTV Digital Signal/Data—Serial Digital Interface, 2008. See section 8.[9]

[B63] SMPTE 292M-2008, SMPTE Standard 1.5 Gb/s Signal/Data Serial Interface, 2008.

[B64] SMPTE 424M-2012, SMPTE Standard for Television—3 Gb/s Signal/Data Serial Interface.

[B65] Specht, J., and S. Samii, "Urgency-Based Scheduler for Time-Sensitive Switched Ethernet Networks," *28th Euromicro Conference on Real-Time Systems (ECRTS)*, pp. 75–85, 2016.

[B66] Teener, M. J., "Peristaltic Shaper: updates, multiple speeds," available at https://www.ieee802.org/1/files/public/docs2014/new-tsn-mjt-peristaltic-shaper-0114.pdf, January 2014.

[B67] Xu, L., K. Harfoush, and I. Rhee, "Binary increase congestion control (BIC) for fast long-distance networks," *IEEE INFOCOM 2004*, vol. 4, pp. 2514-2524, Mar. 2004.

[B68] Zhu, Y., H.Eran, D. Firestone, C. Guo, M. Lipshteyn, Y. Liron, J. Padhye, S. Raindel, M. H. Yahia, and M. Zhang, 2015. "Congestion Control for Large-Scale RDMA Deployments," *ACM SIGCOMM Computer Communication Review*, 45(4), pp. 523–536. doi:10.1145/2785956.2787484.

[8] MoCa publications are available from the Multimedia over Coax Alliance (https://mocalliance.org).
[9] SMPTE publications are available from the Society of Motion Picture and Television Engineers (https://www.smpte.org).