

IEEE P802.1Qdj/D1.1

June 2, 2023

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

IEEE P802.1Qdj/D1.1

**Draft Standard for
Local and metropolitan area networks—**

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

Sponsor

LAN/MAN Standards Committee of the IEEE Computer Society

Prepared by the Time Sensitive Networking Task Group of IEEE 802.1

DRAFT STATUS:

17 Draft for second Working Group ballot.

18 Copyright © 2023 by the Institute of Electrical and Electronics Engineers, Inc.

19 Three Park Avenue

20 New York, New York 10016-5997, USA

21 All rights reserved.

22

23 This document is an unapproved draft of a proposed IEEE Standard. As such, this document is subject to change. USE AT
24 YOUR OWN RISK! IEEE copyright statements SHALL NOT BE REMOVED from draft or approved IEEE standards, or
25 modified in any way. Because this is an unapproved draft, this document must not be utilized for any conformance/
26 compliance purposes. Permission is hereby granted for officers from each IEEE Standards Working Group or Committee to
27 reproduce the draft document developed by that Working Group for purposes of international standardization
28 consideration. IEEE Standards Department must be informed of the submission for consideration prior to any reproduction
29 for international standardization consideration (stds.ipr@ieee.org). Prior to adoption of this document, in whole or in part,
30 by another standards development organization, permission must first be obtained from the IEEE Standards Department
31 (stds.ipr@ieee.org). When requesting permission, IEEE Standards Department will require a copy of the standard
32 development organization's document highlighting the use of IEEE content. Other entities seeking permission to reproduce
33 this document, in whole or in part, must also obtain permission from the IEEE Standards Department.

34

35 IEEE Standards Activities Department

36 445 Hoes Lane

37 Piscataway, NJ 08854, USA

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

5

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

1 Important Notices and Disclaimers Concerning IEEE Standards Documents

2 IEEE documents are made available for use subject to important notices and legal disclaimers. These notices
3 and disclaimers, or a reference to this page, appear in all standards and may be found under the heading
4 “Important Notice” or “Important Notices and Disclaimers Concerning IEEE Standards Documents.”

5 Notice and Disclaimer of Liability Concerning the Use of IEEE Standards 6 Documents

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

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

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

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

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

41 Translations

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

1 Official statements

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

8 Comments on standards

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

19 Comments on standards should be submitted to the following address:

20 Secretary, IEEE-SA Standards Board
21 445 Hoes Lane
22 Piscataway, NJ 08854 USA

23 Laws and regulations

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

29 Copyrights

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

36 Photocopies

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

1 Updating of IEEE Standards documents

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

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

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

15 Errata

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

19 Patents

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

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

1 Participants

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

4 **Glenn Parsons, *Chair***
5 **Jessy Rouyer, *Vice Chair***
6 **János Farkas, *Chair Time-Sensitive Networking Task Group***
7 **Stephan Kehrler, *Editor***

<<TBA>>

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

<<TBA>>

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

3 <<TBA>>

<<TBA>>

4
5 *Member Emeritus
6

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

8
9 <<TBA>>

1 Introduction

2

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

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

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

9

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

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

14 0312: <https://www.802-1.org/items/423>.

15 >>

16 <<Editor’s Introduction to historical draft changes

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

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

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

23 — **Comment #9** and **comment #39**

24 — **Comment #20** and **comment #64**

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

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

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

4 >>

5 <<Editor’s Introduction to historical draft changes

6 **P802.1Qdj/D1.0:** This draft was prepared by Stephan Kehrer for first Working Group ballot as a result of
7 comment resolution on Draft D0.3, finalized during the meeting of the TSN Task Group on September 12,
8 2022.

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

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

12 >>

13 <<Editor’s Introduction to historical draft changes

14 **P802.1Qdj/D0.3:** This draft was prepared by Stephan Kehrer for third Task Group ballot as a result of
15 comment resolution on Draft D0.2, finalized during the electronic meeting of the TSN Task Group on May
16 9, 2022.

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

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

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

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

25 >>

26 <<Editor’s Introduction to historical draft changes

27 **P802.1Qdj/D0.2:** This draft was prepared by Stephan Kehrer for second Task Group ballot as a result of
28 comment resolution on Draft D0.1, finalized during the electronic meeting of the TSN Task Group on March
29 15, 2021.

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

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

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

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

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

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

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

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

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

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

39 >>

40 <<Editor’s Introduction to historical draft changes

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

44 >>

1 <<Editor's Introduction to historical draft changes

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

5 >>

1	1.	Overview.....	16
2	1.3	Introduction.....	16
3	3.	Definitions	17
4	5.	Conformance.....	18
5	5.29	TSN CNC station requirements	18
6	46.	Time-Sensitive Networking (TSN) configuration	19
7	46.1	Overview of TSN configuration	19
8	46.1.3	TSN configuration models	19
9	46.1.5	Centralized User Configuration	20
10	46.1.6	Centralized Network Configuration	21
11	46.1.7	Configuration Domain	21
12	46.2.2	Protocol integration	21
13	46.2.3	Talker	22
14	46.2.6	Protocol operations	23
15	46.2.7	Remote Procedure Calls	23
16	46.2.8	Actions	24
17	46.2.9	Notifications	25
18	46.3	YANG for TSN user/network configuration	25
19	48.	YANG Data Models.....	27
20	48.2.12	User/Network Interface model	27
21	48.3	Structure of the YANG models	29
22	48.3.12	User/Network Interface model	29
23	48.4	Security Considerations	29
24	48.4.12	Security considerations of the User/Network Interface model	29
25	48.5	YANG schema tree definitions	29
26	48.5.23	Schema for the ieee802-dot1q-tsn-config-uni YANG module	29
27	48.6.23	Definitions for the ieee802-dot1q-tsn-config-uni YANG module	33
28	Annex X		
29	(informative) TSN features		46
30	Annex Y		
31	(informative) Bibliography		47

1	Figure 46-3	Fully centralized model.....	19
---	-------------	------------------------------	----

1	Table 46-12 StreamStatus enumeration	23
2	Table 1 Summary of the YANG modules.....	29
3	Table 48-13 User/Network Interface model YANG modules	29

1 IEEE P802.1Qdj/D1.1

2 Draft Standard for Local and metropolitan area 3 networks—

4 Bridges and Bridged Networks— 5 Amendment XX: Configuration 6 Enhancements for Time-Sensitive 7 Networking

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

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

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

21

22 1. Overview

23 1.3 Introduction

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

- 26 a) Defines the Centralized User Configuration (CUC) (46.1.5), and the CNC (46.1.6).
- 27 b) Specifies a Configuration Domain (46.1.7).
- 28 c) Defines YANG configuration and operational state models (48.6.23) in support of the UNI (Clause
29 46).

30

^bNotes in text, tables and figures are given for information only, and do not contain requirements needed to implement the standard.

1 3. Definitions

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

4 **3.1 Configuration Domain:** A set of stations that are under a common configuration, management scheme,
5 and responsibility.

6 **3.2 TSN features:** The protocols and mechanisms that define the set of tools available for building a time-
7 sensitive network.

8 NOTE—See Annex X for more information.

1 5. Conformance

2 5.29 TSN CNC station requirements

3 *Change item d) in 5.29, as follows*

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

1 46. Time-Sensitive Networking (TSN) configuration

2 46.1 Overview of TSN configuration

3 46.1.3 TSN configuration models

4 46.1.3.2 Centralized network/distributed user model

5 *Change the third paragraph in 46.1.3.2, as follows.*

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

11 46.1.3.3 Fully centralized model

12 *Change the second paragraph in 46.1.3.3, as follows.*

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

17 *Replace Figure 46-3 with the following figure.*

18

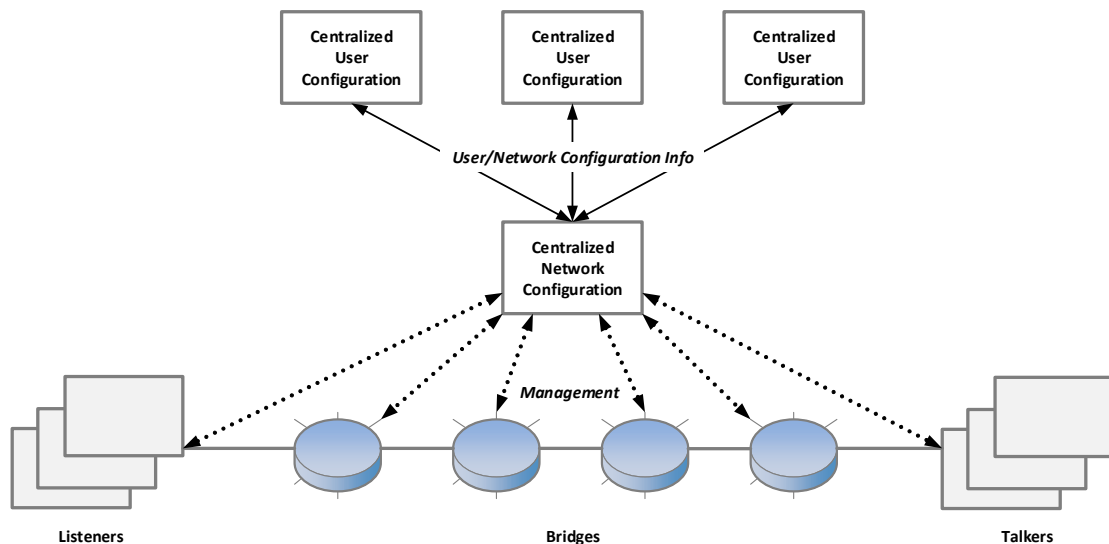


Figure 46-3—Fully centralized model

19

20 *Insert the following subclauses after 46.1.4*

1 46.1.5 Centralized User Configuration

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

5 The CUC is responsible for:

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

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

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

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

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

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

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

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

1 46.1.6 Centralized Network Configuration

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

4 The CNC is responsible for:

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

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

22 46.1.7 Configuration Domain

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

28 *Change clause 46.2.2, as follows:*

29 46.2.2 Protocol integration

30 *Change the third paragraph in clause 46.2.2, as follows:*

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

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

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

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

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

1 *Change the last paragraph in clause 46.2.2, as follows:*

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

9 *Insert the following subclauses after 46.2.2.*

10 46.2.2.1 DomainID

11 DomainID is a unique identifier that specifies the Configuration Domain of a CUC, and the Streams
12 associated with that CUC. DomainID is only used if the centralized network/distributed user model
13 (46.1.3.2) or the fully centralized model (46.1.3.3) is used.

14 46.2.2.2 CucID

15 CucID uniquely identifies a CUC within a Configuration Domain. It is used along with the DomainID to
16 associate Streams with a CUC. CucID is only used if the centralized network/distributed user model
17 (46.1.3.2) or the fully centralized model (46.1.3.3) is used.

18 46.2.2.3 CncEnabled

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

22 46.2.3 Talker

23 *Change the third paragraph in clause 46.2.3, as follows:*

24 The Talker group contains the following groups:

- 25 — StreamID (46.2.3.1)
- 26 — StreamRank (46.2.3.2)
- 27 — EndStationInterfaces (46.2.3.3)
- 28 — DataFrameSpecification (46.2.3.4)
- 29 — TrafficSpecification (46.2.3.5)
- 30 — UserToNetworkRequirements (46.2.3.6)
- 31 — InterfaceCapabilities (46.2.3.7)
- 32 — [StreamStatus \(46.2.3.8\)](#)

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

34 [For the join and leave operation, StreamStatus shall be included.](#)

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

1 46.2.3.8 StreamStatus

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

5

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..

6 *Insert the following subclauses after 46.2.5*

7 46.2.6 Protocol operations

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

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

16 46.2.7 Remote Procedure Calls

17 The TSN user/network configuration provides the following RPCs:

- 18 — ComputeStreams (46.2.7.1)
- 19 — ComputePlannedAndModifiedStreams (46.2.7.2)
- 20 — ComputeAllStreams (46.2.7.3)
- 21 — RequestDomainId (46.2.7.4)
- 22 — RequestFreeStreamId (46.2.7.5)

23 46.2.7.1 ComputeStreams

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

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

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

3 **46.2.7.2 ComputePlannedAndModifiedStreams**

4 This RPC starts the computation of path and resource allocation for Streams that have not been configured
5 or that have been configured and have been modified since configuration. The Streams that are to be
6 included in the computation are specified by providing their associated `DomainID` (46.2.2.1) and `CucID`
7 (46.2.2.2). The object `StreamStatus` (46.2.3.8) is used to determine if a Stream is included in the computation
8 initiated by this RPC.

9 The RPC returns information that indicates only if the Stream computation has been started successfully or
10 not. It does not return information on whether the Stream configuration itself has been successful or not,
11 because computation can take an arbitrary amount of time. The notification `ConfigureStreamsCompleted`
12 (46.2.9.1) is available to the CNC to return information on success or failure of the Stream computation,
13 after the computation has finished.

14 **46.2.7.3 ComputeAllStreams**

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

18 The RPC returns information that indicates only if the Stream computation has been started successfully or
19 not. It does not return information on whether the Stream configuration itself has been successful or not,
20 because computation can take an arbitrary amount of time. The notification `ConfigureStreamsCompleted`
21 (46.2.9.1) is available to the CNC to return information on success or failure of the Stream computation,
22 after the computation has finished.

23 **46.2.7.4 RequestDomainId**

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

27 **46.2.7.5 RequestFreeStreamId**

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

31 **46.2.8 Actions**

32 The TSN user/network configuration provides the following actions:

33 — `RemoveStreams` (46.2.8.1)

34 **46.2.8.1 RemoveStreams**

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

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

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

5 **46.2.9 Notifications**

6 The TSN user/network configuration provides the following notifications:

- 7 — ComputeStreamsCompleted (46.2.9.1)
- 8 — ConfigureStreamsCompleted (46.2.9.2)
- 9 — RemoveStreamsCompleted (46.2.9.3)

10 **46.2.9.1 ComputeStreamsCompleted**

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

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

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

21 **46.2.9.2 ConfigureStreamsCompleted**

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

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

30 **46.2.9.3 RemoveStreamsCompleted**

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

36 **46.3 YANG for TSN user/network configuration**

37 *Change 46.3, as follows:*

1 In order to support the use of YANG-based protocols for the fully centralized model (46.1.3.3), 48.6.3 [and](#)
2 [48.6.23](#) ~~specifies~~ [specify](#) YANG modules.

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

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

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

15 In the YANG module definitions of 48.6.3 [and 48.6.23](#), if any discrepancy between the “description” text
16 and the corresponding specifications in 46.2 occurs, the specifications in 46.2 take precedence.

1 48. YANG Data Models

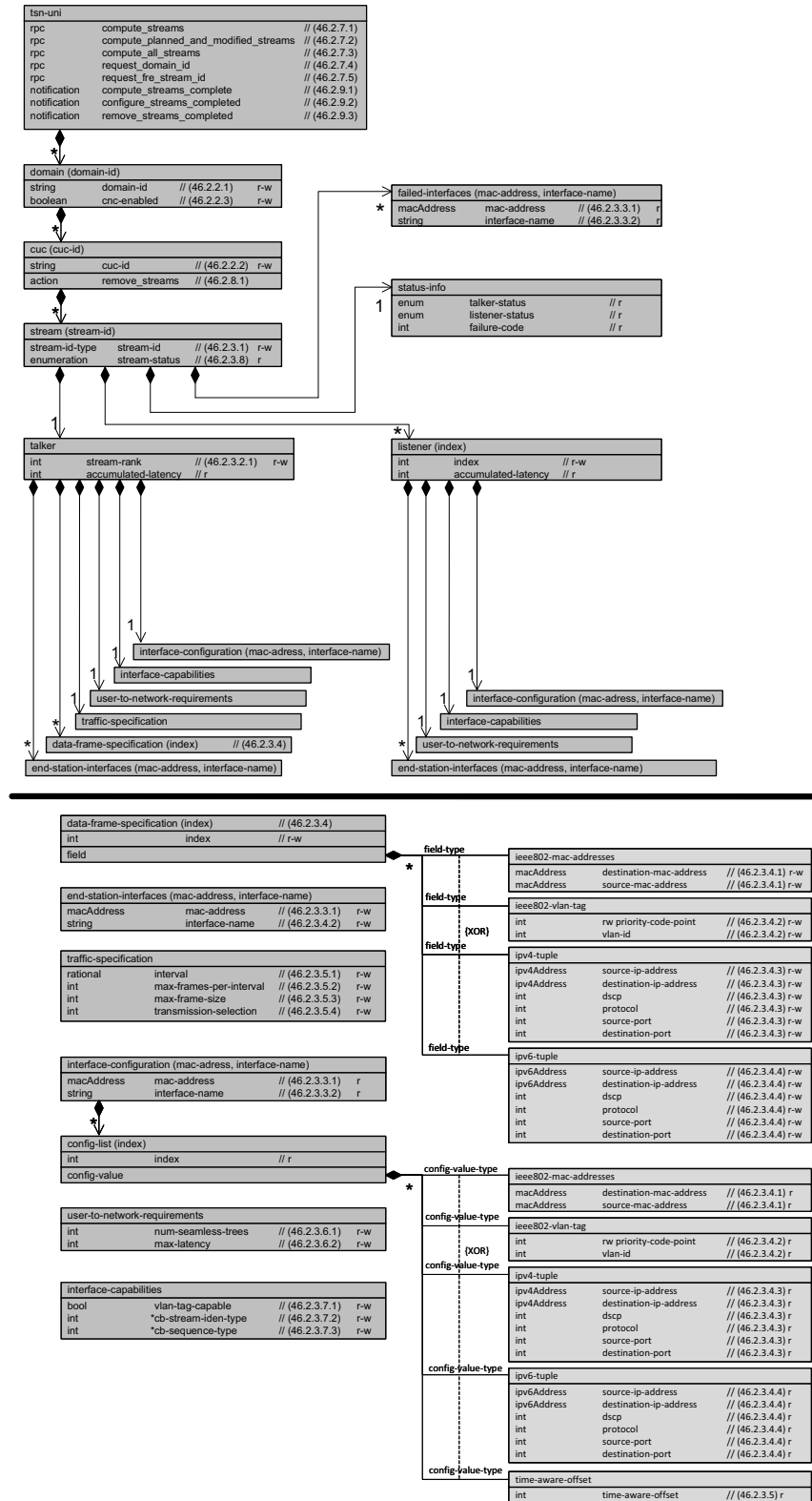
2 Insert the following subclause (48.2.12) after 48.2.11

3 48.2.12 User/Network Interface model

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

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

1



2

Figure 48-21—User/Network Interface model

1 48.3 Structure of the YANG models

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

Table 48-1—Summary of the YANG modules

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

3 *Insert the following subclause (48.3.12) after 48.3.11*

4 48.3.12 User/Network Interface model

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

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

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

7 48.4 Security Considerations

8 *Insert the following subclause (48.4.12) after 48.4.11*

9 48.4.12 Security considerations of the User/Network Interface model

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

16 tsn-config-uni configuration of the UNI

17 tsn-uni/domain/cuc/stream

18 tsn-uni/domain/cuc/remove_stream

19 48.5 YANG schema tree definitions

20 *Insert the following subclause (48.5.23) after 48.5.22*

21 48.5.23 Schema for the ieee802-dot1q-tsn-config-uni YANG module

22

```

1 module: ieee802-dot1q-tsn-config-uni
2   +--rw tsn-uni
3     +--rw domain* [domain-id]
4       +--rw domain-id      string
5       +--rw cnc-enabled?    boolean
6       +--rw cuc* [cuc-id]
7         +--rw cuc-id        string
8         +--rw stream* [stream-id]
9           | +--rw stream-id      tsn:stream-id-type
10          | +--ro stream-status?  enumeration
11          | +--rw talker
12            | +--rw stream-rank
13              | +--rw rank?      uint8
14              | +--rw end-station-interfaces* [mac-address interface-name]
15                | +--rw mac-address      string
16                | +--rw interface-name    string
17                | +--rw data-frame-specification* [index]
18                  | +--rw index          uint8
19                  | +--rw (field)?
20                    | +--:(ieee802-mac-addresses)
21                      | +--rw ieee802-mac-addresses
22                        | +--rw destination-mac-address?  string
23                        | +--rw source-mac-address?        string
24                        +--:(ieee802-vlan-tag)
25                          | +--rw ieee802-vlan-tag
26                          | +--rw priority-code-point?    uint8
27                          | +--rw vlan-id?                 uint16
28                          +--:(ipv4-tuple)
29                            | +--rw ipv4-tuple
30                              | +--rw source-ip-address?    inet:ipv4-address
31                              | +--rw destination-ip-address? inet:ipv4-address
32                              | +--rw dscp?                 uint8
33                              | +--rw protocol?             uint16
34                              | +--rw source-port?          uint16
35                              | +--rw destination-port?     uint16
36                              +--:(ipv6-tuple)
37                                | +--rw ipv6-tuple
38                                | +--rw source-ip-address?    inet:ipv6-address
39                                | +--rw destination-ip-address? inet:ipv6-address
40                                | +--rw dscp?                 uint8
41                                | +--rw protocol?             uint16
42                                | +--rw source-port?          uint16
43                                | +--rw destination-port?     uint16
44                                +--rw traffic-specification
45                                  | +--rw interval
46                                    | +--rw numerator?      uint32
47                                    | +--rw denominator?    uint32
48                                    +--rw max-frames-per-interval? uint16
49                                    +--rw max-frame-size?    uint16
50                                    +--rw transmission-selection? uint8
51                                    +--rw time-aware!
52                                      +--rw earliest-transmit-offset? uint32
53                                      +--rw latest-transmit-offset?  uint32
54                                      +--rw jitter?              uint32
55                                  +--rw user-to-network-requirements
56                                    | +--rw num-seamless-trees?  uint8
57                                    | +--rw max-latency?         uint32
58                                  +--rw interface-capabilities
59                                    | +--rw vlan-tag-capable?     boolean

```

```

1      | | | +--rw cb-stream-iden-type-list*  uint32
2      | | | +--rw cb-sequence-type-list*   uint32
3      | | +--ro accumulated-latency?       uint32
4      | | +--ro interface-configuration
5      | |   +--ro interface-list* [mac-address interface-name]
6      | |     +--ro mac-address            string
7      | |     +--ro interface-name         string
8      | |     +--ro config-list* [index]
9      | |       +--ro index                  uint8
10     | |       +--ro (config-value)?
11     | |         +--:(ieee802-mac-addresses)
12     | |           | +--ro ieee802-mac-addresses
13     | |             | +--ro destination-mac-address?  string
14     | |             | +--ro source-mac-address?       string
15     | |         +--:(ieee802-vlan-tag)
16     | |           | +--ro ieee802-vlan-tag
17     | |             | +--ro priority-code-point?     uint8
18     | |             | +--ro vlan-id?                  uint16
19     | |         +--:(ipv4-tuple)
20     | |           | +--ro ipv4-tuple
21     | |             | +--ro source-ip-address?
22     | |               | inet:ipv4-address
23     | |             | +--ro destination-ip-address?
24     | |               | inet:ipv4-address
25     | |             | +--ro dscp?                      uint8
26     | |             | +--ro protocol?                   uint16
27     | |             | +--ro source-port?                uint16
28     | |             | +--ro destination-port?           uint16
29     | |         +--:(ipv6-tuple)
30     | |           | +--ro ipv6-tuple
31     | |             | +--ro source-ip-address?
32     | |               | inet:ipv6-address
33     | |             | +--ro destination-ip-address?
34     | |               | inet:ipv6-address
35     | |             | +--ro dscp?                      uint8
36     | |             | +--ro protocol?                   uint16
37     | |             | +--ro source-port?                uint16
38     | |             | +--ro destination-port?           uint16
39     | |         +--:(time-aware-offset)
40     | |           +--ro time-aware-offset?              uint32
41     | +--rw listener* [index]
42     | | +--rw index                  uint32
43     | | +--rw end-station-interfaces* [mac-address interface-name]
44     | | | +--rw mac-address          string
45     | | | +--rw interface-name       string
46     | | +--rw user-to-network-requirements
47     | | | +--rw num-seamless-trees?  uint8
48     | | | +--rw max-latency?         uint32
49     | | +--rw interface-capabilities
50     | | | +--rw vlan-tag-capable?     boolean
51     | | | +--rw cb-stream-iden-type-list*  uint32
52     | | | +--rw cb-sequence-type-list*     uint32
53     | | +--ro accumulated-latency?       uint32
54     | | +--ro interface-configuration
55     | |   +--ro interface-list* [mac-address interface-name]
56     | |     +--ro mac-address            string
57     | |     +--ro interface-name         string
58     | |     +--ro config-list* [index]
59     | |       +--ro index                  uint8

```

```

1      | |      +---ro (config-value)?
2      | |      +---:(ieee802-mac-addresses)
3      | |      | +---ro ieee802-mac-addresses
4      | |      | +---ro destination-mac-address?    string
5      | |      | +---ro source-mac-address?        string
6      | |      +---:(ieee802-vlan-tag)
7      | |      | +---ro ieee802-vlan-tag
8      | |      | +---ro priority-code-point?      uint8
9      | |      | +---ro vlan-id?                  uint16
10     | |      +---:(ipv4-tuple)
11     | |      | +---ro ipv4-tuple
12     | |      | +---ro source-ip-address?
13     | |      | |      inet:ipv4-address
14     | |      | +---ro destination-ip-address?
15     | |      | |      inet:ipv4-address
16     | |      | +---ro dscp?                      uint8
17     | |      | +---ro protocol?                  uint16
18     | |      | +---ro source-port?               uint16
19     | |      | +---ro destination-port?          uint16
20     | |      +---:(ipv6-tuple)
21     | |      | +---ro ipv6-tuple
22     | |      | +---ro source-ip-address?
23     | |      | |      inet:ipv6-address
24     | |      | +---ro destination-ip-address?
25     | |      | |      inet:ipv6-address
26     | |      | +---ro dscp?                      uint8
27     | |      | +---ro protocol?                  uint16
28     | |      | +---ro source-port?               uint16
29     | |      | +---ro destination-port?          uint16
30     | |      +---:(time-aware-offset)
31     | |      +---ro time-aware-offset?          uint32
32     | +---ro status-info
33     | | +---ro talker-status?    enumeration
34     | | +---ro listener-status?  enumeration
35     | | +---ro failure-code?     uint8
36     | +---ro failed-interfaces* [mac-address interface-name]
37     | +---ro mac-address        string
38     | +---ro interface-name     string
39 +---x remove_streams
40     +---w input
41     | +---w stream-list* [stream-id]
42     | +---w stream-id     tsn:stream-id-type
43 +---ro output
44     +---ro result?    string
45
46 rpcs:
47 +---x compute_streams
48     | +---w input
49     | | +---w domain* [domain-id]
50     | | +---w domain-id    -> /tsn-uni/domain/domain-id
51     | | +---w cuc* [cuc-id]
52     | | +---w cuc-id      -> /tsn-uni/domain/cuc/cuc-id
53     | | +---w stream-list* [stream-id]
54     | | +---w stream-id    -> /tsn-uni/domain/cuc/stream/stream-id
55     | +---ro output
56     | +---ro result?    string
57 +---x compute_planned_and_modified_streams
58     | +---w input
59     | | +---w domain* [domain-id]

```



```

1  | |      +---w domain-id    string
2  | |      +---w cuc* [cuc-id]
3  | |      +---w cuc-id     string
4  | +--ro output
5  |      +--ro result?    string
6  +---x compute_all_streams
7  | +---w input
8  | | +---w domain* [domain-id]
9  | |      +---w domain-id    string
10 | |      +---w cuc* [cuc-id]
11 | |      +---w cuc-id     string
12 | +--ro output
13 |      +--ro result?    string
14 +---x request_domain_id
15 | +---w input
16 | | +---w cuc-id?    string
17 | +--ro output
18 |      +--ro result?    string
19 +---x request_free_stream_id
20 | +---w input
21 | | +---w domain-id?    string
22 | | +---w cuc-id?      string
23 | +--ro output
24 |      +--ro result?    string
25
26 notifications:
27 +---n compute_streams_completed
28 | +--ro domain* [domain-id]
29 |      +--ro domain-id    string
30 |      +--ro cuc* [cuc-id]
31 |      +--ro cuc-id     string
32 |      +--ro stream* [stream-id]
33 |      +--ro stream-id    tsn:stream-id-type
34 |      +--ro failure-code? uint8
35 +---n configure_streams_completed
36 | +--ro domain* [domain-id]
37 |      +--ro domain-id    string
38 |      +--ro cuc* [cuc-id]
39 |      +--ro cuc-id     string
40 |      +--ro stream* [stream-id]
41 |      +--ro stream-id    tsn:stream-id-type
42 |      +--ro failure-code? uint8
43 +---n remove_streams_completed
44 | +--ro domain* [domain-id]
45 |      +--ro domain-id    string
46 |      +--ro cuc* [cuc-id]
47 |      +--ro cuc-id     string
48 |      +--ro stream* [stream-id]
49 |      +--ro stream-id    tsn:stream-id-type
50 |      +--ro failure-code? uint8

```

53 *Insert the following subclause (48.6.23) after 48.6.22*

54 **48.6.23 Definitions for the ieee802-dot1q-tsn-config-uni YANG module**

```

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

```

```
1  yang-version "1.1";
2  namespace urn:ieee:std:802.1Q:yang:ieee802-dot1q-tsn-config-uni;
3  prefix dot1q-tsn-config-uni;
4  import ieee802-dot1q-tsn-types {
5    prefix tsn;
6    reference
7      "48.6.3 of IEEE Std 802.1Q-2022";
8  }
9  organization
10   "Institute of Electrical and Electronics Engineers";
11  contact
12   "WG-URL: http://ieee802.org/1/
13   WG-EMail: stds-802-1-1@ieee.org
14
15   Contact: IEEE 802.1 Working Group Chair
16   Postal: C/O IEEE 802.1 Working Group
17           IEEE Standards Association
18           445 Hoes Lane
19           Piscataway, NJ 08854
20           USA
21
22   E-mail: stds-802-1-chairs@ieee.org";
23  description
24   "Time-Sensitive Networking (TSN) User/Network Interface (UNI) for the
25   exchange of information between CUC and CNC that are required to
26   configure TSN Streams in a TSN network.";
27  revision 2023-05-16 {
28    description
29     "Editor's note: this revision statement will be removed prior to
30     publication. It is only present while the project is running in
31     order to make it easier for the reader to see what changes have
32     been introduced while the project is running. Update for D1.1 that
33     includes fixes, and updates and adds RPCs, actions, and
34     notifications.";
35    reference
36     "Clause 46.2 of IEEE Std 802.1Q-2022";
37  }
38  revision 2022-11-09 {
39    description
40     "Editor's note: this revision statement will be removed prior to
41     publication. It is only present while the project is running in
42     order to make it easier for the reader to see what changes have
43     been introduced while the project is running. Update for D1.0 that
44     rebases the YANG module of IEEE Std 802.1Q-2022 and updates RPCs,
45     actions, and notifications.";
46    reference
47     "Clause 46.2 of IEEE Std 802.1Q-2022";
48  }
49  revision 2022-06-03 {
50    description
51     "Editor's note: this revision statement will be removed prior to
52     publication. It is only present while the project is running in
53     order to make it easier for the reader to see what changes have
54     been introduced while the project is running. Update for D0.3 that
55     restructures the data model, adds RPCs, adds actions, and adds
56     notifications.";
57    reference
58     "Clause 46.2 of IEEE Std 802.1Qcc-2018";
59  }
```

```
1  revision 2021-12-16 {
2    description
3      "Initial revision. Note that this module might change in backward
4      incompatible ways until approved as a standard.";
5    reference
6      "Clause 46.2 of IEEE Std 802.1Q-2022";
7  }
8  container tsn-uni {
9    description
10     "Top-level container for the TSN UNI module.";
11    list domain {
12      key "domain-id";
13      description
14        "List of Configuration Domains.
15
16        This list exists so CUCs can be associated with the Configuration
17        Domain they are located in and can be used to restrict access to
18        CUCs, e.g., by using standard mechanism as described in RFC 8341.";
19      leaf domain-id {
20        type string;
21        description
22          "The Domain ID is a unique identifier of a Configuration
23          Domain. It is used to identify the Configuration Domain a CUC
24          belongs to.";
25        reference
26          "46.2.2.1 of IEEE Std 802.1Qdj-2023";
27      }
28      leaf cnc-enabled {
29        type boolean;
30        default "false";
31        description
32          "cnc-enabled is used to enable or disable the CNC functionality
33          of a station capable of acting as a CNC. If this object is set
34          to TRUE the CNC functionality is enabled. If it is set to FALSE
35          the CNC functionality is disabled.";
36        reference
37          "46.2.2.3 of IEEE Std 802.1Qdj-2023";
38      }
39      list cuc {
40        key "cuc-id";
41        description
42          "List of CUCs.
43
44          This list exists so Streams can be associated with the CUC that
45          initially requested them and can be used to restrict access to
46          Streams, e.g., by using standard mechanisms as described in RFC
47          8341.";
48        leaf cuc-id {
49          type string;
50          description
51            "The CUC ID is a unique identifier of a CUC. It is used to
52            identify the CUC that a Stream belongs to, i.e., that
53            requested the creation of a Stream.";
54          reference
55            "46.2.2.2 of IEEE Std 802.1Qdj-2023";
56        }
57        list stream {
58          key "stream-id";
59          description
```

```
1      "List of Streams.
2
3      Each Stream consists of a Stream ID, a request container, and
4      a configuration container.
5
6      In the fully centralized model of TSN configuration, the
7      Stream ID and request originate from the CUC and is delivered
8      to the CNC, while the configuration originates from the CNC
9      and is delivered to the CUC.";
10     leaf stream-id {
11         type tsn:stream-id-type;
12         description
13             "The Stream ID is a unique identifier of a Stream request
14             and corresponding configuration. It is used to associate a
15             CUC's Stream request with a CNC's corresponding response.";
16     }
17     leaf stream-status {
18         type enumeration {
19             enum planned {
20                 value 0;
21                 description
22                     "The Stream has been requested but has not yet been
23                     configured by the CNC.";
24             }
25             enum configured {
26                 value 1;
27                 description
28                     "The Stream has been computed and configured by the
29                     CNC.";
30             }
31             enum modified {
32                 value 2;
33                 description
34                     "The Stream has been configured but Stream parameters
35                     have been modified after configuration.";
36             }
37         }
38         config false;
39         description
40             "The stream-status indicates what status the Stream has in
41             the CNC.";
42         reference
43             "46.2.3.8 of IEEE Std 802.1Qdj-2023";
44     }
45     container talker {
46         description
47             "The Talker container contains: - Talker's behavior for
48             Stream (how/when transmitted) - Talker's requirements from
49             the network - TSN capabilities of the Talker's
50             interface(s).";
51         uses tsn:group-talker;
52         uses tsn:group-status-talker-listener {
53             refine "accumulated-latency" {
54                 config false;
55             }
56             refine "interface-configuration" {
57                 config false;
58             }
59         }
```

```
1      }
2      list listener {
3          key "index";
4          description
5              "Each Listener list entry contains: - Listener's
6              requirements from the network - TSN capabilities of the
7              Listener's interface(s).";
8          leaf index {
9              type uint32;
10             description
11                 "This index is provided in order to provide a unique key
12                 per list entry.";
13         }
14         uses tsn:group-listener;
15         uses tsn:group-status-talker-listener {
16             refine "accumulated-latency" {
17                 config false;
18             }
19             refine "interface-configuration" {
20                 config false;
21             }
22         }
23     }
24     uses tsn:group-status-stream {
25         refine "status-info" {
26             config false;
27         }
28         refine "failed-interfaces" {
29             config false;
30         }
31     }
32 }
33 action remove_streams {
34     description
35         "Removes the Streams with the ids provided in the stream-id
36         list.";
37     reference
38         "46.2.8.1 of IEEE Std 802.1Qdj-2023";
39     input {
40         list stream-list {
41             key "stream-id";
42             description
43                 "List of stream-ids that are used to identify the Streams
44                 that are requested to be removed.";
45             leaf stream-id {
46                 type tsn:stream-id-type;
47                 description
48                     "Unique identifier that is used to request a Stream
49                     that is to be removed from the configuration.";
50             }
51         }
52     }
53     output {
54         leaf result {
55             type string;
56             description
57                 "Returns status information indicating if Stream removal
58                 has been successfully started.";
59         }
60     }
```

```

1      }
2    }
3  }
4 }
5 }
6
7 // RPCs
8 rpc compute_streams {
9   description
10    "Starts computation of path and resource allocation for one or more
11    Stream. The Streams that are included in the computation are the
12    ones that have their domain-id, cuc-id, and stream-id provided.
13    This RPC can be applied to compute new Streams as well as recompute
14    Streams that have been modified.";
15   input {
16     list domain {
17       key "domain-id";
18       description
19         "List of Configuration Domains.
20
21         This list exists so CUCs can be associated with the
22         Configuration Domain they are located in.";
23       reference
24         "46.2.7.1 of IEEE Std 802.1Qdj-2023";
25       leaf domain-id {
26         type leafref {
27           path '/tsn-uni/domain/domain-id';
28         }
29         description
30           "A unique identifier of a Configuration Domain. It is used to
31           identify the Configuration Domain a CUC belongs to.";
32       }
33       list cuc {
34         key "cuc-id";
35         description
36           "List of CUCs.
37
38           This list exists so Streams can be associated with the CUC
39           that initially requested them.";
40         leaf cuc-id {
41           type leafref {
42             path '/tsn-uni/domain/cuc/cuc-id';
43           }
44           description
45             "A unique identifier of a CNC. It is used to identify the
46             CUC that a Streams belong to, i.e., that requested the
47             creation of a Stream.";
48         }
49         list stream-list {
50           key "stream-id";
51           description
52             "List of stream-ids that are used to identify the Streams
53             that are requested to be computed and configured.";
54           leaf stream-id {
55             type leafref {
56               path '/tsn-uni/domain/cuc/stream/stream-id';
57             }
58             description
59               "Unique identifier that is used to request a Stream that

```

```
1         is to be computed and configured.";
2     }
3 }
4 }
5 }
6 }
7 output {
8     leaf result {
9         type string;
10        description
11            "Only returns status information indicating if the computation
12            has been started. It does not return status information on the
13            success or failure of the actual Stream computation. A
14            notification can be used to inform the caller of this RPC on the
15            results of Stream computation after the computation has
16            finished.";
17    }
18 }
19 }
20 rpc compute_planned_and_modified_streams {
21     description
22         "Starts computation of path and resource allocation for all Streams
23         that are in the domain provided by domain-id and are associated
24         with the CUC provided by cuc-id, and that have not been computed
25         (i.e., that have a Stream status of planned or modified.";
26     reference
27         "46.2.7.2 of IEEE Std 802.1Qdj-2023";
28     input {
29         list domain {
30             key "domain-id";
31             description
32                 "List of Configuration Domains.
33
34                 This list exists so CUCs can be associated with the
35                 Configuration Domain they are located in.";
36             leaf domain-id {
37                 type string;
38                 description
39                     "A unique identifier of a Configuration Domain. It is used to
40                     identify the Configuration Domain a CUC belongs to.";
41             }
42             list cuc {
43                 key "cuc-id";
44                 description
45                     "List of CUCs.
46
47                     This list exists so Streams can be associated with the CUC
48                     that initially requested them.";
49                 leaf cuc-id {
50                     type string;
51                     description
52                         "A unique identifier of a CNC. It is used to identify the
53                         CUC that a Streams belong to, i.e., that requested the
54                         creation of a Stream.";
55                 }
56             }
57         }
58     }
59     output {
```

```
1     leaf result {
2         type string;
3         description
4             "Only returns status information indicating if the computation
5             has been started. It does not return status information on the
6             success or failure of the actual Stream computation. A
7             notification can be used to inform the caller of this RPC on the
8             results of Stream computation after the computation has
9             finished.";
10    }
11 }
12 }
13 rpc compute_all_streams {
14     description
15         "Starts computation of path and resource allocation for all Streams
16         that are in the domain provided by domain-id and are associated
17         with the CUC provided by cuc-id.";
18     reference
19         "46.2.7.3 of IEEE Std 802.1Qdj-2023";
20     input {
21         list domain {
22             key "domain-id";
23             description
24                 "List of Configuration Domains.
25
26                 This list exists so CUCs can be associated with the
27                 Configuration Domain they are located in.";
28             leaf domain-id {
29                 type string;
30                 description
31                     "A unique identifier of a Configuration Domain. It is used to
32                     identify the Configuration Domain a CUC belongs to.";
33             }
34             list cuc {
35                 key "cuc-id";
36                 description
37                     "List of CUCs.
38
39                     This list exists so Streams can be associated with the CUC
40                     that initially requested them.";
41             leaf cuc-id {
42                 type string;
43                 description
44                     "A unique identifier of a CNC. It is used to identify the
45                     CUC that a Streams belong to, i.e., that requested the
46                     creation of a Stream.";
47             }
48         }
49     }
50 }
51 output {
52     leaf result {
53         type string;
54         description
55             "Only returns status information indicating if the computation
56             has been started. It does not return status information on the
57             success or failure of the actual Stream computation. A
58             notification can be used to inform the caller of this RPC on the
59             results of Stream computation after the computation has
```



```
1         finished.";
2     }
3 }
4 }
5 rpc request_domain_id {
6     description
7         "Returns the DomainId of the Configuration Domain that the
8         requesting CUC belongs to.";
9     reference
10        "46.2.7.4 of IEEE Std 802.1Qdj-2023";
11     input {
12         leaf cuc-id {
13             type string;
14             description
15                 "A unique identifier of a CNC. It is used to identify the CUC,
16                 allowing the CNC to return the DomainId this CUC belongs to.";
17         }
18     }
19     output {
20         leaf result {
21             type string;
22             description
23                 "Returns the DomainId of the Configuration Domain that the
24                 requesting CUC belongs to.";
25         }
26     }
27 }
28 rpc request_free_stream_id {
29     description
30         "Returns a free StreamId available for the Configuration Domain
31         identified by the DomainId.";
32     reference
33        "46.2.7.5 of IEEE Std 802.1Qdj-2023";
34     input {
35         leaf domain-id {
36             type string;
37             description
38                 "A unique identifier of a Configuration Domain. It is used to
39                 identify the Configuration Domain a CUC belongs to.";
40         }
41         leaf cuc-id {
42             type string;
43             description
44                 "A unique identifier of a CNC. It is used to identify the CUC,
45                 allowing the CNC to return the DomainId this CUC belongs to.";
46         }
47     }
48     output {
49         leaf result {
50             type string;
51             description
52                 "Returns a free StreamId available for the Configuration Domain
53                 identified by the DomainId.";
54         }
55     }
56 }
57
58 // Notifications
59 notification compute_streams_completed {
```

```
1  description
2    "Notifies the caller of an RPC or action that initiated the
3    computation of one or multiple Streams, that the computation is
4    complete. It also returns information on the success or failure for
5    each of the Streams in the computation.";
6  reference
7    "46.2.9.1 of IEEE Std 802.1Qdj-2023";
8  list domain {
9    key "domain-id";
10   description
11     "List of Configuration Domains.
12
13     This list exists so CUCs can be associated with the Configuration
14     Domain they are located in.";
15   leaf domain-id {
16     type string;
17     description
18       "A unique identifier of a Configuration Domain. It is used to
19       identify the Configuration Domain a CUC belongs to.";
20   }
21   list cuc {
22     key "cuc-id";
23     description
24       "List of CUCs.
25
26       This list exists so Streams can be associated with the CUC that
27       initially requested them.";
28     leaf cuc-id {
29       type string;
30       description
31         "A unique identifier of a CNC. It is used to identify the CUC
32         that a Stream belongs to, i.e., that requested the creation
33         of a Stream.";
34     }
35     list stream {
36       key "stream-id";
37       description
38         "List of Streams.
39
40         Each Stream consists of a Stream ID, a request container, and
41         a configuration container.
42
43         In the fully centralized model of TSN configuration, the
44         Stream ID and request originate from the CUC and is delivered
45         to the CNC, while the configuration originates from the CNC
46         and is delivered to the CUC.";
47       reference
48         "46.2.3 of IEEE Std 802.1Qdj-2023";
49       leaf stream-id {
50         type tsn:stream-id-type;
51         description
52           "The Stream ID is a unique identifier of a Stream request
53           and corresponding configuration. It is used to associate a
54           CUC's Stream request with a CNC's corresponding response.";
55       }
56       leaf failure-code {
57         type uint8;
58         description
59           "A code that indicates if the computation for the Stream
```

```
1      was successful (0) or not. In the case of a failure a code
2      is returned to indicate what kind of failure occurred.";
3  }
4  }
5  }
6  }
7  }
8  notification configure_streams_completed {
9      description
10     "Notifies the caller of an RPC or action that initiated the
11     computation of one or multiple Streams, that the computation and
12     configuration is complete. It also returns information on the
13     success or failure for each of the Streams in the computation and
14     configuration.";
15     reference
16     "46.2.9.2 of IEEE Std 802.1Qdj-2023";
17     list domain {
18         key "domain-id";
19         description
20         "List of Configuration Domains.
21
22         This list exists so CUCs can be associated with the Configuration
23         Domain they are located in.";
24         leaf domain-id {
25             type string;
26             description
27             "A unique identifier of a Configuration Domain. It is used to
28             identify the Configuration Domain a CUC belongs to.";
29         }
30         list cuc {
31             key "cuc-id";
32             description
33             "List of CUCs.
34
35             This list exists so Streams can be associated with the CUC that
36             initially requested them.";
37             leaf cuc-id {
38                 type string;
39                 description
40                 "A unique identifier of a CNC. It is used to identify the CUC
41                 that a Streams belong to, i.e., that requested the creation
42                 of a Stream.";
43             }
44             list stream {
45                 key "stream-id";
46                 description
47                 "List of Streams.
48
49                 Each Stream consists of a Stream ID, a request container, and
50                 a configuration container.
51
52                 In the fully centralized model of TSN configuration, the
53                 Stream ID and request originate from the CUC and is delivered
54                 to the CNC, while the configuration originates from the CNC
55                 and is delivered to the CUC.";
56                 reference
57                 "46.2.3 of IEEE Std 802.1Qdj-2023";
58                 leaf stream-id {
59                     type tsn:stream-id-type;
```

```
1      description
2          "The Stream ID is a unique identifier of a Stream request
3          and corresponding configuration. It is used to associate a
4          CUC's Stream request with a CNC's corresponding response.";
5      }
6      leaf failure-code {
7          type uint8;
8          description
9              "A code that indicates if the computation and configuration
10             for the Stream was successful (0) or not. In the case of a
11             failure a code is returned to indicate what kind of failure
12             occurred.";
13      }
14  }
15  }
16  }
17  }
18  notification remove_streams_completed {
19      description
20          "Notifies the caller of an RPC or action that initiated the removal
21          of one or multiple Streams, that the removal is complete. It also
22          returns information on the success or failure for each of the
23          Streams in the removal request.";
24      reference
25          "46.2.9.3 of IEEE Std 802.1Qdj-2023";
26      list domain {
27          key "domain-id";
28          description
29              "List of Configuration Domains.
30
31              This list exists so CUCs can be associated with the Configuration
32              Domain they are located in.";
33          leaf domain-id {
34              type string;
35              description
36                  "A unique identifier of a Configuration Domain. It is used to
37                  identify the Configuration Domain a CUC belongs to.";
38          }
39          list cuc {
40              key "cuc-id";
41              description
42                  "List of CUCs.
43
44                  This list exists so Streams can be associated with the CUC that
45                  initially requested them.";
46              leaf cuc-id {
47                  type string;
48                  description
49                      "A unique identifier of a CNC. It is used to identify the CUC
50                      that a Streams belong to, i.e., that requested the creation
51                      of a Stream.";
52              }
53              list stream {
54                  key "stream-id";
55                  description
56                      "List of Streams.
57
58                      Each Stream consists of a Stream ID, a request container, and
59                      a configuration container.
```

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

1 Annex X

2 (informative)

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

5 TSN features

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

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

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

31

Annex Y

(informative)

Insert the following bibliography references in alphanumeric order

Bibliography

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

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

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

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

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

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

16