Second IEEE 802.1 Working Group ballot on Draft 2.1 of the

IEC/IEEE 60802 Time-Sensitive Networking Profile for Industrial Automation

Working Group ballot start date: 2023-10-02

Working Group ballot closing date: 2023-11-02

This is an unapproved draft prepared by the IEC/IEEE 60802 Joint Project.

NOTE – This page is not subject to ballot comments.

CONTENTS

Time-sensitive networking profile for industrial automation

FOREWORD

1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC document(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation.

   IEEE Standards documents are developed within IEEE Societies and Standards Coordinating Committees of the IEEE Standards Association (IEEE SA) Standards Board. IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of IEEE and serve without compensation. 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 contained in its standards. Use of IEEE Standards documents is wholly voluntary. *IEEE documents are made available for use subject to important notices and legal disclaimers (see* https://standards.ieee.org/ipr/disclaimers.html *for more information).*

   IEC collaborates closely with IEEE in accordance with conditions determined by agreement between the two organizations. This Dual Logo International Standard was jointly developed by the IEC and IEEE under the terms of that agreement.

2) The formal decisions of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees. The formal decisions of IEEE on technical matters, once consensus within IEEE Societies and Standards Coordinating Committees has been reached, is determined by a balanced ballot of materially interested parties who indicate interest in reviewing the proposed standard. Final approval of the IEEE standards document is given by the IEEE Standards Association (IEEE SA) Standards Board.

3) IEC/IEEE Publications have the form of recommendations for international use and are accepted by IEC National Committees/IEEE Societies in that sense. While all reasonable efforts are made to ensure that the technical content of IEC/IEEE Publications is accurate, IEC or IEEE cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.

4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications (including IEC/IEEE Publications) transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC/IEEE Publication and the corresponding national or regional publication shall be clearly indicated in the latter.

5) IEC and IEEE do not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC and IEEE are not responsible for any services carried out by independent certification bodies.

6) All users should ensure that they have the latest edition of this publication.

7) No liability shall attach to IEC or IEEE or their directors, employees, servants or agents including individual experts and members of technical committees and IEC National Committees, or volunteers of IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE SA) Standards Board, for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC/IEEE Publication or any other IEC or IEEE Publications.

8) Attention is drawn to the normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.

9) Attention is drawn to the possibility that implementation of this IEC/IEEE Publication may require use of material covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. IEC or IEEE shall not be held responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patent 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.

IEC/IEEE 60802 was prepared by subcommittee 65C: Industrial networks, of IEC technical committee 65: Industrial-process measurement, control and automation, in cooperation with IEEE 802.1: Higher Layer LAN Protocols Working Group of IEEE 802: LAN/MAN Standards Committee of the IEEE computer society, under the IEC/IEEE Dual Logo Agreement between IEC and IEEE. It is an International Standard.

This document is published as an IEC/IEEE Dual Logo standard.

The text of this International Standard is based on the following IEC documents:

Draft	Report on voting
XX/XX/FDIS	XX/XX/RVD

Full information on the voting for its approval can be found in the report on voting indicated in the above table.

The language used for the development of this International Standard is English.

This document was drafted in accordance with the rules given in the ISO/IEC Directives, Part 2, available at www.iec.ch/members_experts/refdocs. The main document types developed by IEC are described in greater detail at www.iec.ch/publications/.

The IEC Technical Committee and IEEE Working Group have decided that the contents of this document will remain unchanged until the stability date indicated on the IEC website under webstore.iec.ch in the data related to the specific document. At this date, the document will be

- reconfirmed,
- withdrawn, or
- revised.

IMPORTANT – The 'colour inside' logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

____

INTRODUCTION

Editor's note: This profile has been drafted per the IEC style guide. The use of introductory boilerplate text, editorial conventions and clause numbering may differ from those outlined in the IEEE style guide.

For the convenience of the reviewer, references to the IEC/IEEE cooperation process and to the IEC organization and style guidelines are listed below:

IEC-IEEE 60802 Joint Project Cooperation Process:

http://www.ieee802.org/1/files/public/docs2018/admin-IEC-IEEE-JWG-cooperation-process-0118.pdf

IEC organization:

http://www.iec.ch/

Home of the IEC organization of the Joint Project IEC/IEEE 60802 is: IEC 65C/WG18:

https://www.iec.ch/dyn/www/f?p=103:14:506767825234046::::FSP_ORG_ID,FSP_LANG_ID:26299,25Assistance to experts drafting IEC documents:

http://www.iec.ch/standardsdev/resources/draftingpublications/

Reference material:

http://www.iec.ch/members_experts/refdocs/

ISO/IEC Directives, Part 2:2021

edition 9.0 (2021-05)

Principles and rules for the structure and drafting of ISO and IEC documents:

https://www.iec.ch/members_experts/refdocs/iec/isoiecdir2%7Bed9.0%7Den.pdf

ISO/IEC Directives, Part 1:2023 + IEC Supplement:2023

https://www.iec.ch/system/files/2023-05/consolidated_iso-iec_part-1_iecsupplement_2023.pdf

Procedures for the technical work - Procedures specific to IEC:

https://www.iec.ch/members_experts/refdocs/iec/isoiecdir1-consolidatedIECsup%7Bed17.0%7Den.pdf

This document defines a Time-Sensitive Networking profile for industrial automation. The profile selects features, options, configurations, defaults, protocols, and procedures of bridges, end stations, and LANs to build industrial automation networks.

The profile meets the industrial automation market objective of converging Operations Technology (OT) and Information Technology (IT) networks by defining a common, standardized network infrastructure. This objective is accomplished by taking advantage of the improvements that Time-Sensitive Networking provides to IEEE 802.1 and IEEE 802.3 standard Ethernet networks by providing guaranteed data transport with bounded low latency, low latency variation, zero congestion loss for critical traffic, and high availability.

The profile helps the convergence of industrial communication networks by referring only to international standards to build the lower layers of the communication stack and their management.

Ethernet extended with Time-Sensitive Networking technology provides the features required in the area of industrial communication networks, such as:

- Meeting low latency and latency variation requirements concerning data transmission.
- Efficient exchange of data records on a frequent time period.
- Reliable communications with calculable downtime.
- High availability meeting application requirements.
- Efficient mechanisms for bandwidth utilization of exchanges of data records, with zero congestion loss.
- Improved clock synchronization mechanisms, including support of multiple gPTP domains.

Time-sensitive networking profile for industrial automation

1 Scope

This document defines time-sensitive networking profiles for industrial automation. The profiles select features, options, configurations, defaults, protocols, and procedures of bridges, end stations, and LANs to build industrial automation networks. This document also specifies YANG modules defining read-only information available online and offline as a digital data sheet. This document also specifies YANG modules for remote procedure calls and actions to address requirements arising from industrial automation networks.

2 Normative References

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 9594-1:2020 (ITU-T Recommendation X.500), *Information technology: Open systems interconnection – Part 1: The Directory: Overview of concepts, models and services*

ISO/IEC 9594-2:2020 (ITU-T Recommendation X.501), *Information technology: Open systems interconnection Part 2: The Directory: Models*

IEEE Draft Std P1588e[1] (Draft 0.2, March 2022), *Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems Amendment: MIB and YANG Data Model*s

IEEE Std 802.1AB-2016[2], *IEEE Standard for Local and Metropolitan Area Networks: Station and Media Access Control Connectivity Discovery*

IEEE Std 802.1ABcu-2021, *IEEE Standard for Local and Metropolitan Area Networks: Station and Media Access Control Connectivity Discovery Amendment 1: YANG Data Model*

IEEE Std 802.1AR-2018, *IEEE Standard for Local and Metropolitan Area Networks: Secure Device Identity*

IEEE Std 802.1AS-2020, *IEEE Standard for Local and Metropolitan Area Networks: Timing and Synchronization for Time-Sensitive Applications*

IEEE Draft Std P802.1ASdm (Draft 0.5, January 2022), *IEEE Standard for Local and Metropolitan Area Networks: Timing and Synchronization for Time-Sensitive Applications Amendment: Hot Standby*

IEEE Std 802.1CB-2017, *IEEE Standard for Local and Metropolitan Area Networks: Frame Replication and Elimination for Reliability*

IEEE Std 802.1CBcv-2021, IEEE *Standard for Local and Metropolitan Area Networks: Frame Replication and Elimination for Reliability — Amendment 1: Information Model, YANG Data Model and Management Information Base Module*

IEEE Std 802.1Q-2022, *IEEE Standard for Local and Metropolitan Area Network: Bridges and Bridged Networks*

[1] Numbers preceded by P are IEEE authorized standards projects that were not approved by the IEEE SA Standards Board at the time this publication went to Sponsor ballot/press. For information about obtaining drafts, contact the IEEE.

[2] The IEEE standards or products referred to in Clause 2 are trademarks of The Institute of Electrical and Electronics Engineers, Incorporated

IEEE Draft Std P802.1Qcw (Draft 1.3, February 2021), *Draft Standard for Local and Metropolitan Area Networks: Bridges and Bridged Networks, Amendment: YANG Data Models for Scheduled Traffic, Frame Preemption, and Per-Stream Filtering and Policing*

IEEE Draft Std P802.1Qdj (Draft 0.3, June 2022), *Draft Standard for Local and Metropolitan Area Networks: Bridges and Bridged Networks, Amendment: Configuration Enhancements for Time-Sensitive Networking*

IEEE Draft Std P802.1Qdx, *Draft Standard for Local and Metropolitan Area Networks: Bridges and Bridged Networks, Amendment: YANG Data Models for the Credit-Based Shaper*

IEEE Std 802.3-2022, *IEEE Standard for Ethernet*

IEEE Std 802.3.2-2019, *IEEE Standard for Ethernet YANG Data Model Definitions*

IEEE Draft Std P802.3de (Draft 3.0, March 2022), *Draft Standard for Ethernet Amendment 6: Enhancements to MAC Merge and Time Synchronization Service Interface for Point-to-Point 10 Mb/s Single-Pair Ethernet*

IETF RFC 2131, Droms, R., *Dynamic Host Configuration Protocol,* March 1997, available at https://www.rfc-editor.org/info/rfc2131

IETF RFC 2986, Nystrom, M. and Kaliski, B., *PKCS #10: Certification Request Syntax Specification Version 1.7,* November 2000, available at https://www.rfc-editor.org/info/rfc2986

IETF RFC 3986, Berners-Lee, T., Fielding. R., and Masinter, L., *Uniform Resource Identifier (URI): Generic Syntax,* January 2005, available at https://www.rfc-editor.org/info/rfc3986

IETF RFC 5246, Dierks, T. and Rescorla, E., *The Transport Layer Security (TLS) Protocol,* August 2008, available at https://www.rfc-editor.org/info/rfc5246

IETF RFC 5277, Chisholm, S. and Trevino, H., *NETCONF Event Notification,* July 2008, available at https://www.rfc-editor.org/info/rfc5277

IETF RFC 5280, Turner, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and Polk, W., *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*, May 2008, available at https://www.rfc-editor.org/info/rfc5280

IETF RFC 5480, Cooper, S., Brown, D., Yiu., K., Housley, R., and Polk, T., *Elliptic Curve Cryptography Subject Public Key Information*, March 2009, available at https://www.rfc-editor.org/info/rfc5480

IETF RFC 6022, Scott, M. and Bjorklund, M., *YANG Module for NETCONF Monitoring*, October 2010, available at https://www.rfc-editor.org/info/rfc6022

IETF RFC 6024, Reddy, R. and Wallace, C., *Trust Anchor Management Requirements*, October 2010, available at https://www.rfc-editor.org/info/rfc6024

IETF RFC 6066, Eastlake, D, *Transport Layer Security (TLS) Extensions: Extension Definitions*, January 2011, available at https://www.rfc-editor.org/info/rfc6066

IETF RFC 6125, Saint-Andre, P. and Hodges, J., *Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS),* March 2011, available at https://www.rfc-editor.org/info/rfc6125

IETF RFC 6241, Enns, R., Bjorklund, M., Schoenwaelder, J. and Bierman, A., *Network Configuration Protocol (NETCONF)*, June 2011, available at https://www.rfc-editor.org/info/rfc6241

IETF RFC 6242, Wasserman, M., *Using the NETCONF Protocol over Secure Shell (SSH)*, June 2011, available at https://www.rfc-editor.org/info/rfc6242

IETF RFC 6961, Pettersen, Y., *The Transport Layer Security (TLS) Multiple Certificate Status Request Extension*, June 2013, available at https://www.rfc-editor.org/info/rfc6961

IETF RFC 7317, Bierman, A. and Bjorklund, M., *A YANG Data Model for System Management*, August 2014, available at https://www.rfc-editor.org/info/rfc7317

IETF RFC 7589, Badra, M., Luchuk, A. and Schoenwaelder, J., *Using the NETCONF Protocol over Transport Layer Security (TLS) with Mutual X.509 Authentication*, June 2015, available at https://www.rfc-editor.org/info/rfc7589

IETF RFC 7748, Langley, A., Hamburg, M., and Turner, S., *Elliptic Curves for Security*, January 2016, available at https://www.rfc-editor.org/info/rfc7748

IETF RFC 7950, Bjorklund, M., *The YANG 1.1 Data Modeling Language*, August 2016, available at https://www.rfc-editor.org/info/rfc7950

IETF RFC 8032, Josefsson, S., and Liusvaara, I., *Edwards-Curve Digital Signature Algorithm (EdDSA)*, January 2017, available at https://www.rfc-editor.org/info/rfc8032

IETF RFC 8069, Thomas, A., *URN Namespace for IEEE*, February 2017, available at https://www.rfc-editor.org/info/rfc8069

IETF RFC 8141, Saint-Andre, P., and Klensin. J., *Uniform Resource Names (URNs)*, April 2017, available at https://www.rfc-editor.org/info/rfc8141

IETF RFC 8341, Bierman, A. and Bjorklund, M., *Network Configuration Access Control Model*, March 2018, available at https://www.rfc-editor.org/info/rfc8341

IETF RFC 8342, Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K. and Wilton, R., *Network Management Datastore Architecture (NMDA)*, March 2018, available at https://www.rfc-editor.org/info/rfc8342

IETF RFC 8343, Bjorklund, M., *YANG Data Model for Interface Management*, March 2018, available at https://www.rfc-editor.org/info/rfc8343

IETF RFC 8348, Bierman, A., Bjorklund, M., Dong, J., and Romascanu, D., *A YANG Data Model for Hardware Management*, March 2018, available at https://www.rfc-editor.org/info/rfc8348

IETF RFC 8410, Josefsson, S., and Schaad, J., *Algorithm Identifiers for Ed25519, Ed448, X25519, and X448 for Use in the Internet X.509 Public Key Infrastructure*, August 2018, available at https://www.rfc-editor.org/info/rfc8410

IETF RFC 8446, Rescorla, E., *The Transport Layer Security (TLS) Protocol Version 1.3*, August 2018, available at https://www.rfc-editor.org/info/rfc8446

IETF RFC 8525, Bierman, A., Bjorklund, M., Schoenwaelder, J., Watsen, K. and Wilton, R., *YANG Library*, March 2019, available at https://www.rfc-editor.org/info/rfc8525

IETF RFC 8526, Bierman, A., Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K. and Wilton, R., *NETCONF Extensions to Support the Network Management Datastore Architecture*, March 2019, available at https://www.rfc-editor.org/info/rfc8526

IETF RFC 8639, Voit, E., Clemm, A., Gonzalez Prieto, A., Nilsen-Nygaard, E., and Tripathy, A., *Subscription to YANG Notifications*, September 2019, available at https://www.rfc-editor.org/info/rfc8639

IETF RFC 8640, Voit, E., Clemm, A., Gonzalez Prieto, A., Nilsen-Nygaard, E. and Tripathy, A., Dynamic *Subscription to YANG Events and Datastores over NETCONF*, September 2019, available at https://www.rfc-editor.org/info/rfc8640

IETF RFC 8641, Clemm, A. and Voit, E., *Subscription to YANG Notifications for Datastore Updates*, September 2019, available at https://www.rfc-editor.org/info/rfc8641

IETF RFC 9195, Lengyel, B. and Claise, B., *A File Format for YANG Instance Data*, February 2022, available at https://www.rfc-editor.org/info/rfc9195

IETF RFC 9196, Lengyel, B., Clemm, A. and Claise, B., *YANG Modules Describing Capabilities for Systems and Datastore Update Notifications*, February 2022, available at https://www.rfc-editor.org/info/rfc9196

Editor's note: The "„Internet-Draft (I-D)"" will be substituted before IEEE SA ballot and IEC CDV with the IETF RFC numbers, which are not yet known. The reference to the draft will also disappear.

IETF RFC „Internet-Draft (I-D)", *Updates to Using the NETCONF Protocol over Transport Layer Security (TLS) with Mutual X.509 Authentication* (draft-ietf-netconf-over-tls13-02), Internet Draft, Work in Progress by NETCONF WG, available at https://datatracker.ietf.org/doc/draft-ietf-netconf-over-tls13/

IETF RFC „Internet-Draft (I-D)", *A YANG Data Model for a Truststore* (draft-ietf-netconf-trust-anchors-19), Internet Draft, Work in Progress by NETCONF WG, available at https://datatracker.ietf.org/doc/draft-ietf-netconf-trust-anchors/19/

IETF RFC „Internet-Draft (I-D)", *A YANG Data Model for a Keystore* (draft-ietf-netconf-keystore-26), Internet Draft, Work in Progress by NETCONF WG, available at https://datatracker.ietf.org/doc/draft-ietf-netconf-keystore/26/

IETF RFC „Internet-Draft (I-D)", *YANG Data Types and Groupings for Cryptography* (draft-ietf-netconf-crypto-types-25), Internet Draft, Work in Progress by NETCONF WG, available at https://datatracker.ietf.org/doc/draft-ietf-netconf-crypto-types/25/

NIST FIPS 180-4, *Secure Hash Standard (SHS),* August 2015, available at https://csrc.nist.gov/publications/detail/fips/180/4/final

NIST FIPS 186-5, *Digital Signature Standard (DSS),* February 2023, available at https://csrc.nist.gov/publications/detail/fips/186/5/final

NIST SP 800-186, *Recommendations for Discrete Logarithm-based Cryptography: Elliptic Curve Domain Parameters,* February 2023, available at https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-186.pdf

Editor's note: Any draft standards will be removed prior to CDV and SA Ballot.

3 Terms, definitions, symbols, abbreviated terms and conventions

3.1 General

For the purposes of this document, the terms and definitions given in ITU-T G.8260, IEEE Std 802-2014, IEEE Std 802.3-2022, IEEE Std 802.1Q-2022, IEEE Std 802.1AS-2020, and the following apply:

- IEC Electropedia: available at https://www.electropedia.org/
- ISO Online browsing platform: available at https://www.iso.org/obp
- IEEE Standards Dictionary Online: available at https://dictionary.ieee.org
- ITU-T Terms and Definitions database: available at https://www.itu.int/br_tsb_terms/#/

NOTE Definitions in IEC 60050 can be found in the Electropedia link above.

3.2 List of terms, abbreviated terms and definitions given in various standards

For the purposes of this document, the terms and definitions given in Table 1 apply.

Editor’s note: Any standard referenced in the section title but not referenced in the table will be removed prior to CDV and sponsor ballot.

For ease of understanding, the most important terms used within this document are listed in Table 1 but the definitions are not repeated.

Table 1 – List of terms

Term	Source
BMCA	IEEE Std 802.1AS-2020
Bridge	IEEE Std 802.1Q-2022
Bridge Port	IEEE Std 802.1Q-2022
CFM	IEEE Std 802.1Q-2022
Clock	IEEE Std 802.1AS-2020
ClockTimeTransmitter	IEEE Std 802.1AS-2020
ClockTimeReceiver	IEEE Std 802.1AS-2020
ClockSource	IEEE Std 802.1AS-2020
ClockTarget	IEEE Std 802.1AS-2020
CNC	IEEE Std 802.1Q-2022
constant time error (cTE)	ITU-T G.8260
Customer Virtual Local Area Network (C-VLAN) component	IEEE Std 802.1Q-2022
CUC	IEEE Std 802.1Q-2022
device	IEEE Std 802.1AR-2018
DLL	IEEE Std 802-2014
DTE	IEEE Std 802.3-2022
dynamic time error (dTE)	ITU-T G.8260
end entity (EE)	NIST Special Publication 800-57 Part 2, Revision 1
end station	IEEE Std 802-2014
Ethernet	IEEE Std 802.3-2022
FDB	IEEE Std 802.1Q-2022
FID	IEEE Std 802.1Q-2022
fingerprint	IETF RFC 7589
FQTSS	IEEE Std 802.1Q-2022
fractional frequency offset	IEEE Std 802.1AS-2020
frame	IEEE Std 802.1Q-2022
frame preemption	IEEE Std 802.1Q-2022
FRER	IEEE Std 802.1CB-2017
gating cycle	IEEE Std 802.1Q-2022
gPTP communication path	IEEE Std 802.1AS-2020
gPTP domain	IEEE Std 802.1AS-2020
Grandmaster Clock	IEEE Std 802.1AS-2020
Grandmaster PTP Instance	IEEE Std 802.1AS-2020
Independent Virtual Local Area Network [VLAN] Learning (IVL)	IEEE Std 802.1Q-2022
IST	IEEE Std 802.1Q-2022
LAN	IEEE Std 802-2014
latency	IEEE Std 802.1Q-2022

Term	Source
Listener	IEEE Std 802.1Q-2022
LLDP	IEEE Std 802.1AB-2016
LLDPDU	IEEE Std 802.1AB-2016
local clock	IEEE Std 802.1AS-2020
LocalClock	IEEE Std 802.1AS-2020
logical link	IEEE Std 802-2014
LPI	IEEE Std 802.3-2022
MAC	IEEE Std 802.1Q-2022
MMRP	IEEE Std 802.1Q-2022
MST	IEEE Std 802.1Q-2022
MVRP	IEEE Std 802.1Q-2022
NETCONF	IETF RFC 6241
PCP	IEEE Std 802.1Q-2022
PDU	IEEE Std 802.1Q-2022
PHY	IEEE Std 802.3-2022
PLS	IEEE Std 802.3-2022
Port	IEEE Std 802.1Q-2022
preciseOriginTimestamp	IEEE Std 802.1AS-2020
primary domain	IEEE Draft Std P802.1ASdm
PSFP	IEEE Std 802.1Q-2022
PTP End Instance	IEEE Std 802.1AS-2020
PTP Instance	IEEE Std 802.1AS-2020
PTP Link	IEEE Std 802.1AS-2020
PTP Port	IEEE Std 802.1AS-2020
PTP Relay Instance	IEEE Std 802.1AS-2020
PVID	IEEE Std 802.1Q-2022
redundancy	IEC 60050-192
residence time	IEEE Std 802.1AS-2020
secondary domain	IEEE Draft Std P802.1ASdm
station	IEEE Std 802-2014
stream	IEEE Std 802.1Q-2022
synchronized time	IEEE Std 802.1AS-2020
Talker	IEEE Std 802.1Q-2022
time error	ITU-T G.8260
time-sensitive stream	IEEE Std 802.1Q-2022
traffic class	IEEE Std 802.1Q-2022
TLV	IEEE Std 802.3-2022
Configuration Domain	IEEE P802.1Qdj
UNI	IEEE Std 802.1Q-2022
VID	IEEE Std 802.1Q-2022
VLAN	IEEE Std 802.1Q-2022
YANG	IETF RFC 6020

578

3.3 Terms defined in this document

3.3.1

application clock

clock used by the application to time events

Note 1 to entry: Events can be periodic or aperiodic.

3.3.2

Bridge component

Customer Virtual Local Area Network (C-VLAN) component as defined in IEEE Std 802.1Q-2022

3.3.3

control latency

time delay between the input to a sensor application and the output from an actuator application

Note 1 to entry: For the purposes of this document, control latency does not include latencies in the sensor, actuator, or the physical system in a process.

3.3.4

deadline

application defined fixed time reference point that represents a time when data is required by the application

3.3.5

digital data sheet

information about the capabilities of an IA-station, for example, states, configurations, and supported features

3.3.6

end station component

end station entity as defined in IEEE Std 802-2014

3.3.7

Global Time

synchronized time, derived from a gPTP domain, that is traceable to the PTP timescale

3.3.8

IA-controller

industrial automation function, consisting of a comparing element and a controlling element, that performs a specified control function

Note 1 to entry: An IA-controller exchanges data with several IA-devices or other IA-controllers for the purpose of control of a system.

Note 2 to entry: The primary categories of IA-controllers are distributed control system (DCS), programmable logic controller (PLC), and programmable automation controller (PAC).

3.3.9

IA-device

industrial automation function, consisting of sensor and/or actuator elements to read and/or write process data

Note 1 to entry: An IA-device exchanges data with an IA-controller or other IA-devices for the purpose of control of a system.

3.3.10

IA-station

material element or assembly of one or more end station components, and zero, one or more bridge components

Note 1 to entry: IA-controllers and IA-devices are industrial automation functions of IA-stations.

Note 2 to entry: An IA-station is often colloquially called an “IA-controller” or “IA-device” based on its primary function, for example, “IA-controller” for an IA-station that includes an IA-controller function and an IA-device function.

3.3.11
imprinting
<security> equipping IA-stations with an LDevID-NETCONF credential as defined in IEEE Std 802.1AR-2018, corresponding trust anchor as defined in IETF RFC 6024, and certificate-to-name mapping instructions as defined in IETF RFC 7589, Clause 7

3.3.12
management entity
IA-station function responsible for configuration of Bridge components, end station components and ports

Note 1 to entry: The management entity interacts with remote management.

3.3.13
network diameter
longest of all the calculated shortest paths between each pair of nodes in the network

Note 1 to entry: The shortest path between 2 nodes is the path between the two nodes that contains the fewest number of logical links.

3.3.14
network provisioning
process of defining a consistent network configuration, which is applied to all stations

3.3.15
nominal frequency
ideal frequency with zero uncertainty

Note 1 to entry: The nominal frequency of the PTP timescale is further explained in IEEE Std 1588-2019, 7.2.1, 7.2.2, and Annex B.

3.3.16
ppm
μHz/Hz

Note 1 to entry: The term "ppm" refers to a pure multiplicator of 0,000 001 and is used in the context of this document as an SI unit term to allow readable terms conformant to various rules related to expressions.

3.3.17
Working Clock
synchronized time, derived from a gPTP domain, that is traceable to the PTP timescale, or to an ARB timescale that is continuous

Note 1 to entry: In general, the Working Clock is traceable to an ARB timescale; however, the Working Clock time can be correlated to a recognized timing standard.

3.4 Abbreviated terms and acronyms

AEAD	Authenticated Encryption with Associated Data
AES	Advanced Encryption Standard
ARB	Arbitrary
ASCII	American Standard Code for Information Interchange
ASN	Abstract Syntax Notation
BMCA	Best Master Clock Algorithm
CA	Certification Authority
CBC	Cipher Block Chaining
ccA	Conformance Class A
ccB	Conformance Class B
CFM	Connectivity Fault Management
CMLDS	Common Mean Link Delay Service

CMS	Cryptographic Message Syntax
CN	Common Name
CNC	Centralized Network Configuration
CRL	Certificate Revocation List
CRUDX	Create Read Update Delete eXecute
CSR	Certificate Signing Request
CUC	Centralized User Configuration
C-VLAN	Customer VLAN
DAC	Discretionary Access Control
DER	Distinguished Encoding Rules
DH	Diffie-Hellman
DHE	Diffie-Hellman Ephemeral
DLL	Data Link Layer
DMAC	Destination MAC Address
DNS	Domain Name Service
DSA	Digital Signature Algorithm
DTE	Data Terminal Equipment
EC	Elliptic Curve
ECC	Elliptic Curve Cryptography
ECDSA	Elliptic Curve Digital Signature Algorithm
EdDSA	Edwards-Curve Digital Signature Algorithm
EE	End Entity
FDB	Filtering Database
FID	Filtering Identifier
FQDN	Fully Qualified Domain Name
FQTSS	Forwarding and Queuing Enhancements for time-sensitive streams
FRER	Frame Replication and Elimination for Reliability
GCM	Galois Counter Mode
gPTP	generalized Precision Time Protocol
HMAC	Keyed-Hashing for Message Authentication Code
HW	HardWare
IA	Industrial Automation
IDevID	Initial Device IDentifier
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
I-LAN	Internal Local Area Network
ISO	International Organization for Standardization
ISS	Internal Sublayer Service
IST	Internal Spanning Tree
IT	Information Technology
ITU	International Telecommunication Union
IVL	Independent Virtual Local Area Network Learning
LDevID	Locally significant Device IDentifier

LLDP	Link Layer Discovery Protocol
LLDPDU	Link Layer Discovery Protocol Data Unit
LPI	Low Power Idle
LRP	Link-local Registration Protocol
MAC	Media Access Control
MD	Media-Dependent
MDI	Media Dependent Interface
MMRP	Multiple MAC Registration Protocol
MST	Multiple Spanning Tree
MVRP	Multiple VLAN Registration Protocol
N/A	Not applicable
NACM	Network configuration Access Control Model
NETCONF	Network Configuration Protocol
NMDA	Network Management Datastore Architecture
NPE	Network Provisioning Entity
NRR	Neighbor Rate Ratio
OCSP	Online Certificate Status Protocol
OID	Object Identifier
OT	Operations Technology
OUI	Organizational Unique Identifier
PCP	Priority Code Point
PCS	Profile Conformance Statement
PDU	Protocol Data Unit
PE	Path Entity
PEM	Privacy Enhanced Mail
PFS	Perfect Forward Secrecy
PHY	Physical Layer devices
PII	Personally Identifiable Information
PKCS	Public Key Cryptography Standards
PKI	Public Key Infrastructure
PKIX	Public Key Infrastructure X.509
PLS	Physical Signaling Sublayer
PPS	Pulse Per Second
PSFP	Per-Stream Filtering and Policing
PTP	Precision Time Protocol
PVID	Port VLAN Identifier
RBAC	Role-Based Access Control
RFC	Request for Comments
RPC	Remote Procedure Call
RSA	Rivest-Shamir-Adleman
RAE	Resource Allocation Entity
SAN	Subject Alternative Name
SHA	Secure Hash Algorithm

STE	Sync Tree Entity
TDE	Topology Discovery Entity
TLS	Transport Layer Security
TLV	Type, Length, Value
TOFU	Trust On First Use
TSN	Time-Sensitive Networking
TSN-IA	Time-Sensitive Networking for Industrial Automation
TTP	Trusted Third Party
UNI	User/Network Interface
URL	Uniform Resource Locator
URN	Uniform Resource Name
VID	VLAN Identifier
VLAN	Virtual Local Area Network
YANG	Yet Another Next Generation data modeling language

3.5 Conventions

3.5.1 Convention for capitalizations

Capitalized terms are either based on the rules given in the ISO/IEC Directives Part 2 or emphasize that these terms have a specific meaning throughout this document.

Throughout this document "bridge" can be used instead of "Bridge", except when

- it occurs at the beginning of a sentence or
- it is being used as (or part of) a specific term such as "VLAN Bridge" rather than being used to identify bridges (potentially of any type) in general. If "VLAN Bridge" is meant where only "Bridge" is written, a change to "VLAN Bridge" would be appropriate.

3.5.2 Unit conventions

This document uses:

- Gb/s for gigabits per second and
- Mb/s for megabits per second.

3.5.3 Conventions for YANG contents

YANG modules and XML instance data for YANG shown in this document use the following style:

Text style `higher-layer-if` text style

Contents of a YANG module use the following style:

```
<ieee802-dot1q-bridge xmlns="urn:ietf:params:xml:ns:yang:ieee802-dot1q-bridge">
    <bridges>
        <bridge> <!-- list -->
            <name>functional-unit-x</name>
            ...
```

YANG modules in which only parent nodes are listed always include all their child leaves.

3.5.4 Conventions for YANG selection / Digital Datasheet

YANG nodes in 6.4 marked with [m], are mandatory nodes in the digital datasheet, nodes marked with [c] are conditional mandatory if the IA-station supports the corresponding optional functionality. Nodes marked with [o], are optional nodes in the digital datasheet.

4 Overview of TSN in industrial automation

4.1 Industrial application operation

Industrial network applications are based on three main types of building blocks, which can be combined in one IA-controller or provided as a combination of an IA-controller and IA-devices interconnected through a suitable communication network.

These basic building blocks are:

- IA-device Sensor subsystems, which provide input signals indicating the value of the parameter or state being monitored, such as temperature, pressure, or discrete input information.
- IA-controller subsystems, which operate on combinations of measurements and external demand settings to develop output requests, such as position corrections in a motion application.
- IA-device Actuator subsystems, which implement output requests that result in physical changes to the process or machine under control, such as a level in a storage tank, the speed of a printing press, or movement of a robot.

NOTE 1 In general, all subsystems have an internal state, based upon initial settings, and derived from execution; therefore, the application inputs are combined with the internal state to develop an updated internal state and associated outputs.

A control loop is formed when the process or machine responds to the actuator output and produces a new measured value at the sensor. The complete loop is shown in Figure 1 where an IA-controller and IA-devices are connected as end stations in the network.

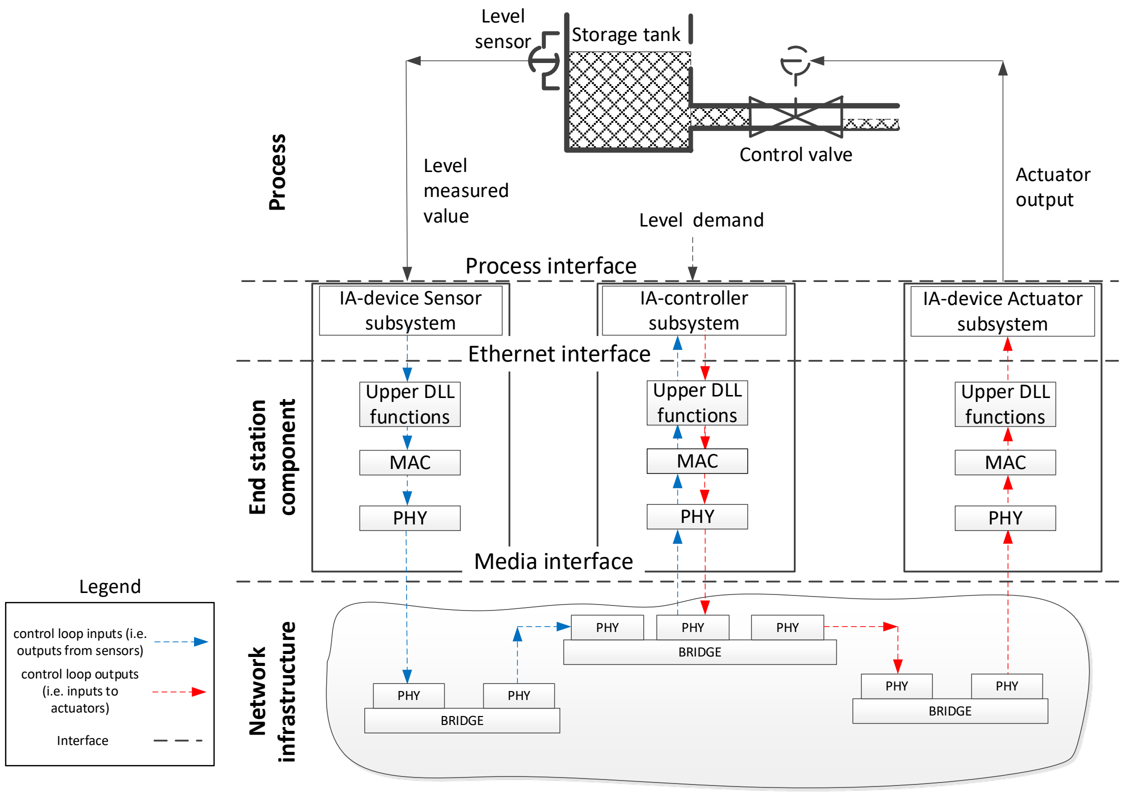


Figure 1 – Data flow in a control loop

In operation, the IA-device Sensor subsystem samples the measured value and the sampled values are transferred through the network as data packets for the IA-controller subsystem to compare with the demand value. After the required computational time, the required output is transferred from the IA-controller subsystem to the IA-device Actuator subsystem for implementation as a change in the external process.

This sequence repeats continuously as a regular operation using a Working Clock. The Working Clock is traceable to an ARB timescale or to the PTP timescale. Traceability to the PTP timescale is not required by all applications. For stability, the time constant of the process response needs to be on the order of five to ten times (or more) the sequence repetition time (i.e., sampling time).

NOTE 2 In common Industrial Network deployments, it has been observed that a ratio of 5 to 10 (or more) provides effective control of the automated process. The actual ratio of the process response time constant to sampling time required for stability depends on the implementation.

Control latency is a critical factor in all types of control and needs to be bounded. Components contributing to the control latency time are shown in Figure 1.

- Application time for sampling, computation, and processing within each IA-controller and IA-device. These are specific to the IA-device and IA-controller and known to the IA-device or IA-controller makers.
- The time for data transfer through the upper DLL functions, MAC and PHY layers within each IA-controller and IA-device. This time depends on the implementation of these components, their situation-dependent load and performance, and configuration elements related to QoS supported by these components.
- End Station and Bridge scheduling and transfer time through the network. These are influenced by the configuration process, which allocates available bandwidth and priorities to various types of application messages.

Offline engineering of the network is possible, including the calculation of the control latency time. During system operation, management services are provided for diagnostics and checking the performance indicators of an installed network.

4.2 Industrial applications

4.2.1 General

Industrial applications can contain multiple tasks. These tasks are executed based upon time or other events. Thus, an industrial application can have multiple tasks executing on different cycles as shown in Figure 2 and Figure 3.

Examples of these tasks include:

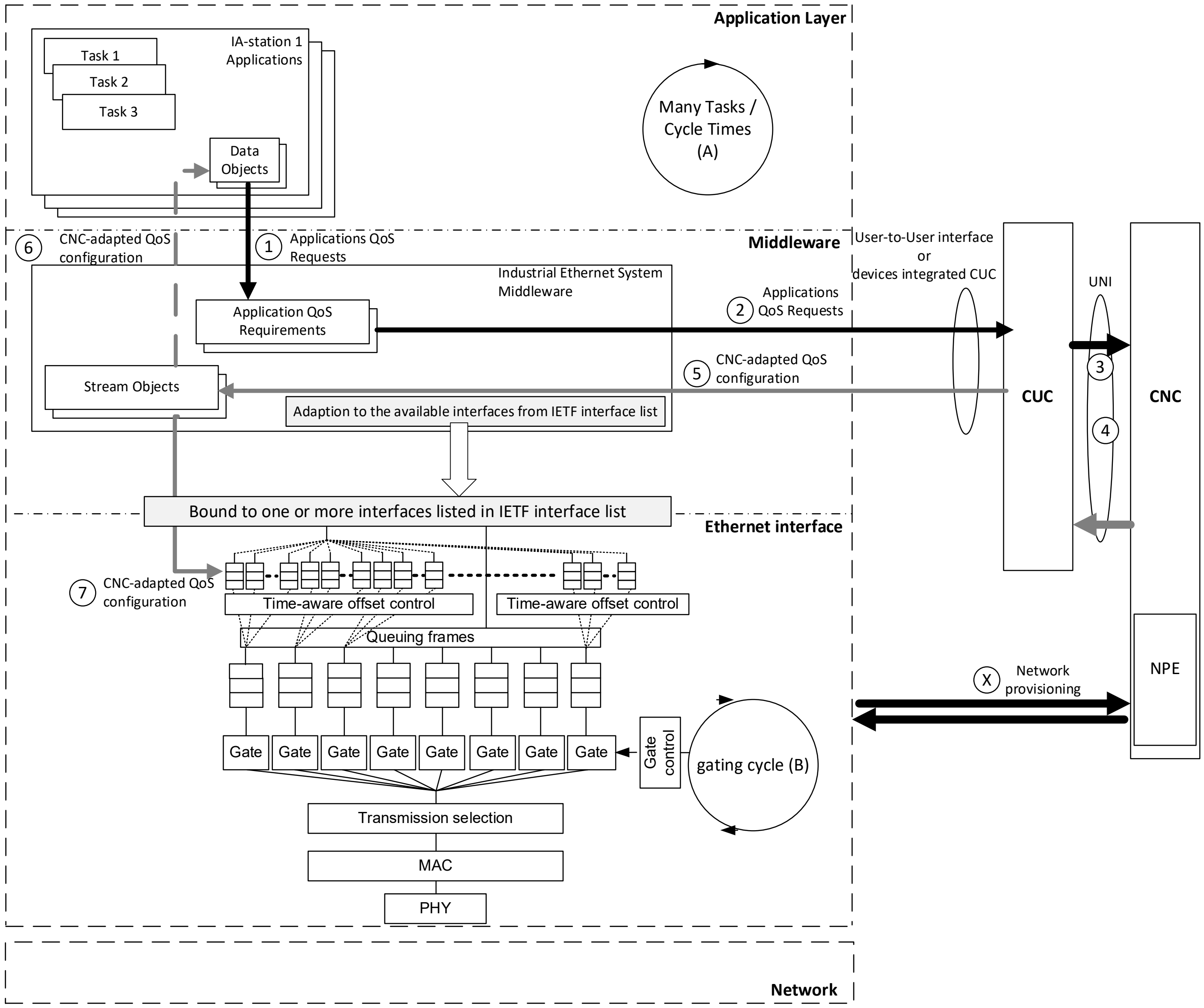
- Background tasks, which are executed when no other task is running. There can be zero, one, or more such tasks in an industrial application.
- Main task which executes periodically. The start and execution of this task is often based upon the ARB timescale. There can be zero or one such task, in an industrial application.
- Global Time tasks. The start and execution of these tasks is often based upon Global Time (for example, at noon every day, at noon every Friday, etc). There can be zero, one or more such tasks in an industrial application.
- Process driven tasks which are started by an event (for example, a sensor value reaches a defined point, a process fault occurs, etc.). There can be zero, one or more such tasks in an industrial application.
- Control loop tasks which are bound to Working Clock and started periodically. There can be zero, one or more such tasks in an industrial application.

A user defines the required automation tasks along with the data objects required as output and input for these tasks and the end station which hosts these tasks. Thus, these tasks are bound to data objects, which need to be exchanged between end stations per the user's definition. Many of these tasks have timing requirements, which are added as attributes to the assigned data objects. Examples of these attributes include:

- [DataObject_Update_Interval] an update interval (time between two consecutive updates at the transmitting end station);
- [DataObject_Deadline] a deadline (latest receive time at the end station, relative to the start of the DataObject Update Interval);

- [DataObject_Data_Size] the size of the DataObject;
- Other attributes as needed to form a stream-list request according to IEEE Draft P802.1Qdj, 46.1.5.

NOTE These attributes are provided for illustration purposes. The list is not representative of all industrial applications. These are not network attributes.


Application Layer
IA-station 1 Applications
Task 1
Task 2
Task 3
Data Objects
Many Tasks / Cycle Times (A)
6
CNC-adapted QoS configuration
1
Applications QoS Requests
Middleware
Industrial Ethernet System Middleware
Application QoS Requirements
Stream Objects
Adaption to the available interfaces from IETF interface list
User-to-User interface or devices integrated CUC
2
Applications QoS Requests
5
CNC-adapted QoS configuration
CUC
UNI
3
4
CNC
Bound to one or more interfaces listed in IETF interface list
Ethernet interface
7
CNC-adapted QoS configuration
Time-aware offset control
Time-aware offset control
Queuing frames
Gate
Gate
Gate
Gate
Gate
Gate
Gate
Gate
Gate control
gating cycle (B)
Transmission selection
MAC
PHY
X
Network provisioning
NPE
Network

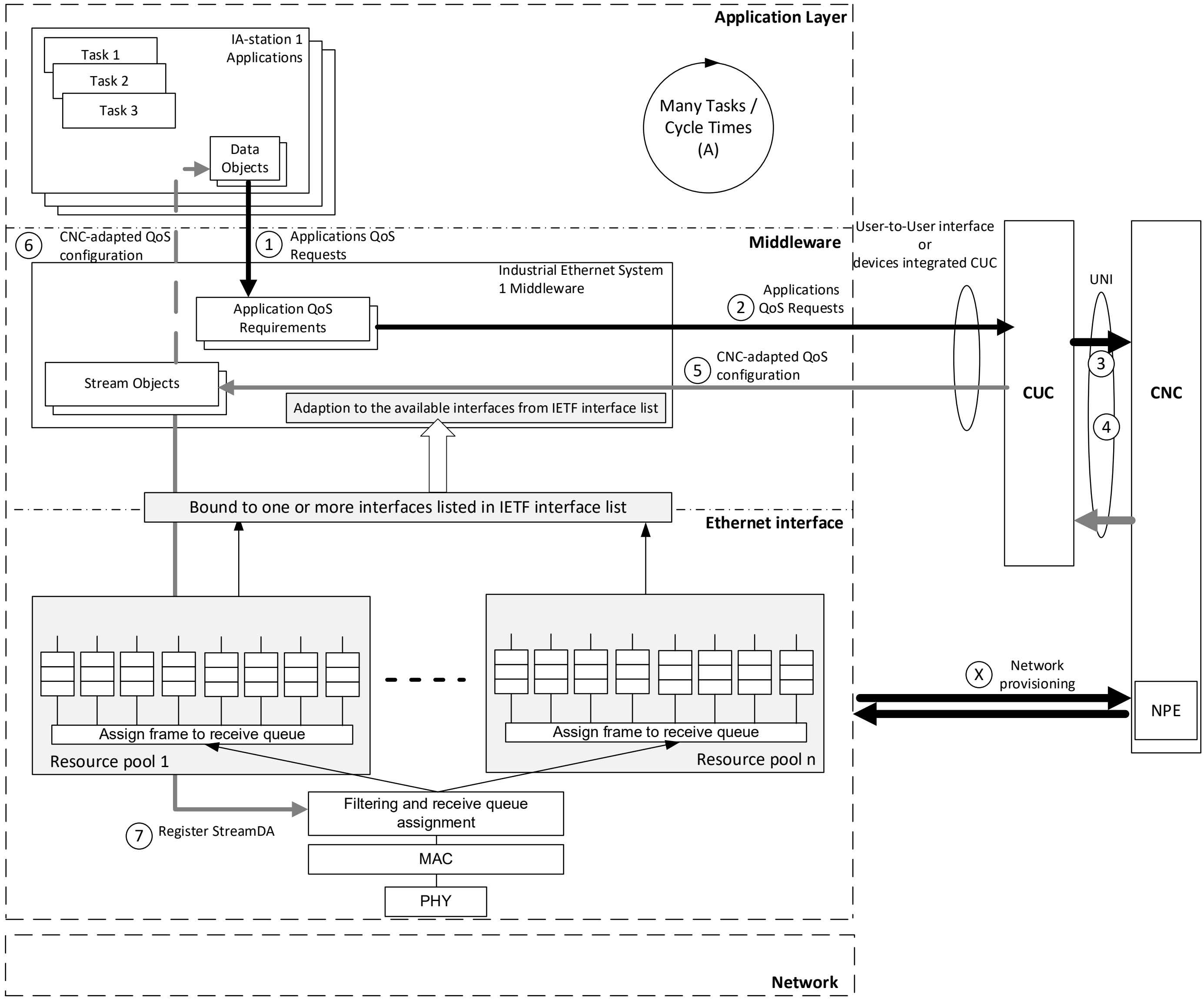
Legend:

Ⓧ IA-stations are configured by the CNC before any CUC to CNC action happens. For example, this configuration can include the gating cycle and transmission selection. Later accesses can happen concurrently with a CUC to CNC sequence.

① … ⑦ Sequence of actions taken

Time-aware offset control Queues are shared between multiple middleware components.

Figure 2 – IA-station interaction with CNC – Transmit path


Application Layer
Task 1
Task 2
Task 3
IA-station 1 Applications
Data Objects
Many Tasks / Cycle Times (A)
Middleware
6
CNC-adapted QoS configuration
1
Applications QoS Requests
Industrial Ethernet System 1 Middleware
Application QoS Requirements
Stream Objects
Adaption to the available interfaces from IETF interface list
User-to-User interface or devices integrated CUC
2
Applications QoS Requests
5
CNC-adapted QoS configuration
UNI
3
4
CUC
CNC
Bound to one or more interfaces listed in IETF interface list
Ethernet interface
Assign frame to receive queue
Resource pool 1
Assign frame to receive queue
Resource pool n
X
Network provisioning
NPE
Filtering and receive queue assignment
7
Register StreamDA
MAC
PHY
Network

Legend:

Ⓧ IA-stations are configured by the CNC before any CUC to CNC action happens. Configuring includes for example the gating cycle and transmission selection. Later accesses may happen concurrently to a CUC to CNC sequence.

① … ⑦ Sequence of actions taken

Resource pool x Different resource pools to support decoupling of middleware components.

Figure 3 – IA-station interaction with CNC – Receive path

4.2.2 Control loop tasks

Control loops rely on the behavior of synchronized tasks by each of the IA-devices and IA-controllers involved in that control loop. For example, this behavior can be implemented by using a common Working Clock, a common starting point relative to the Working Clock and a common duration for this control loop task at the involved IA-devices and IA-controllers. The data objects associated with the control loop share common values for some attributes (for example, the same values for DataObject_Update_Interval and DataObject_Deadline). Multiple control loop tasks can be implemented and running in parallel at the involved automation devices.

4.2.3 Start of control loop tasks

The calculation of the starting point for a control loop task is independent from the time when the device is powered up or connected to the Configuration Domain. The start of a control loop task, which is based on the Working Clock, can be calculated in the following manner:

> Divide the Working Clock value, expressed as an integer, by the duration of the control loop task, expressed as an integer, whenever the Working Clock value increases by one. A remainder of zero provides the basis for the start of the control loop task.

NOTE The units of the Working Clock value and the duration of the control loop task are the same.

Stations in the network associated with the control loop synchronize to a Working Clock using IEEE Std 802.1AS-2020.

4.3 IA-stations

An IA-station can be a simple end station acting as source or destination for control data traffic. In addition, an IA-station can be a combined functional unit that includes an end station component together with a Bridge component in one chassis. IA-stations, incorporating multiple functional units with several end station components and Bridge components within one chassis, can also be found in industrial automation. Within this kind of combined IA-station various components can be connected by internal ports and internal LANs. All components utilize a common management entity as shown in Figure 4.

Figure 4 shows an example IA-station incorporating four functional units in one chassis. Functional unit 1 and functional unit 2 each consist of a Bridge component and an end station component. The end station components are connected by internal ports via internal LANs to the Bridge components. The Bridge components include two external ports each. Functional unit 3 includes only a single end station component with one external port. Functional unit 4 includes a single end station component with two external ports.

IA-controllers and IA-devices as well as the management entity are IA-station functions acting as source of and/or destination for link layer data traffic. Thus, each IA-station incorporates at least one end station component where these functions can be located. Figure 4 shows that IA-station functions can either reside in a single end station component (IA-device 1, IA-controller 1, IA-device 2, IA-device 3, IA-controller 3) or in multiple end station components (IA-controller 2, management entity).

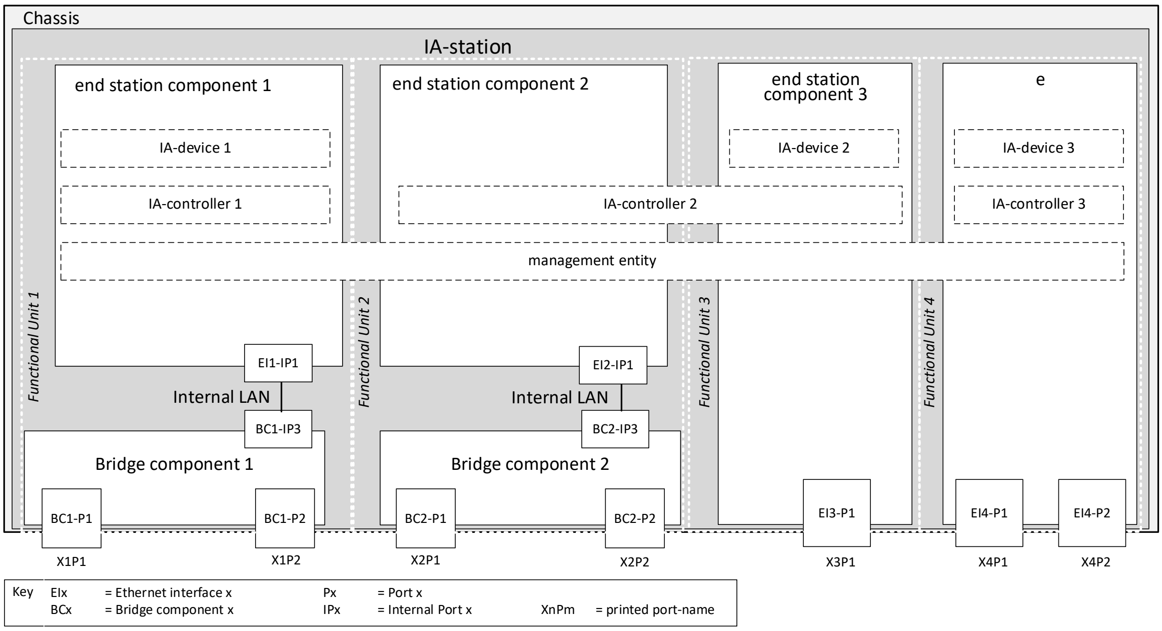


Figure 4 – IA-station example

4.4 Ethernet interface

One or more middleware components act as a layer between applications and the Ethernet interface. Figure 2 and Figure 3 show the relation between applications, middleware, Ethernet interface and the network. Various applications can run in parallel on an automation device. Data objects represent the information exchanged between applications running in different end stations. The application requirements contained in these data objects are translated by the middleware into stream requirements for use by the CUC. This translation can be accomplished in one or both of the following ways:

a) The user defines the data objects and translates them into stream requirements and end-station communication-configurations. A user-specific mechanism is used to configure the network components, establish paths, and the time-aware offset control.

b) The user defines the data objects and associates them with QoS requirements for each stream (application QoS requirements). These can be forwarded as stream requirement requests by a CUC to a CNC. The CNC responds by providing a stream configuration response. The request and response are specified in IEEE P802.1Qdj. This information is used to configure the time-aware offset control, which utilizes per-stream queues. The CUC can be integrated into the end station or can be accessed via a user-to-user protocol. The middleware uses this information for configuring Talkers and Listeners. This information is also used to add additional timing information to the data objects for application usage.

Time-aware offset control utilizes per-stream queues (see IEEE Std 802.1Q-2022, Figure 34-1) and the traffic specification of the streams, including transmission offsets, provided by the CNC to ensure the order of stream transmission.

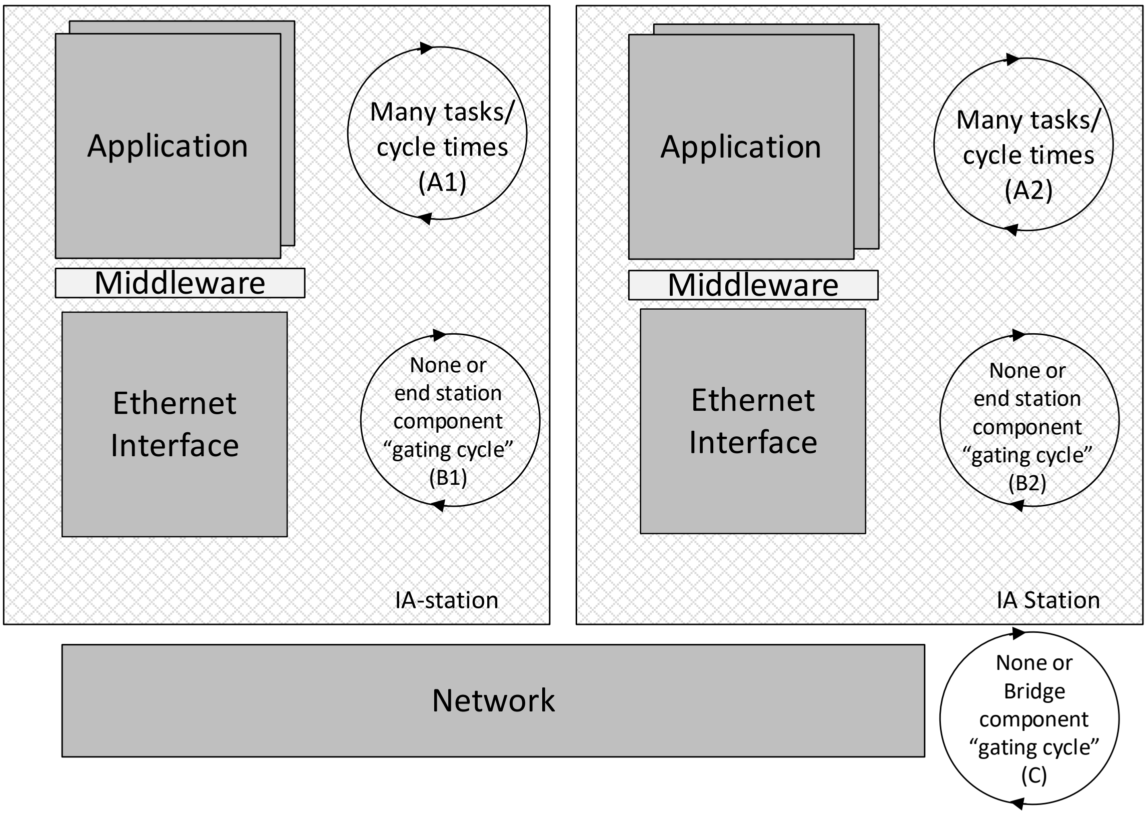

Application
Many tasks/ cycle times (A1)
Middleware
Ethernet Interface
None or end station component “gating cycle” (B1)
IA-station
Application
Many tasks/ cycle times (A2)
Middleware
Ethernet Interface
None or end station component “gating cycle” (B2)
IA Station
Network
None or Bridge component “gating cycle” (C)

Figure 5 – Model for cycles

These automation systems, which are built from various end stations and connected via bridges, can share a common gating cycle or each station can have its own gating cycle. Alternatively, a bridge or end station can have no gating cycle (expressed as "none" in Figure 5).

4.5 Mechanisms that can be used to meet control loop latency requirements

Meeting latency requirements on a network can be accomplished using one or more combinations of the mechanisms enumerated below. The choice of a mechanism or a subset of the mechanisms listed below depends on the nature of the application(s) and the corresponding latency requirements:

a) Defining, testing, and simulating all possible application combinations and associated traffic patterns,
b) Overprovisioning the network,
c) Providing scheduled time slots for each application to transmit on the network,
d) Preempting lower priority traffic,
e) Providing scheduled time slots for certain traffic classes,
f) Time-aware offset control,
g) Enforcing deterministic queuing delays in bridges.

NOTE This list is not comprehensive and not all mechanisms mentioned here are part of this specification. For specific mechanisms covered by this document please refer to Clause 5.

Frame preemption is specified in IEEE Std 802.1Q-2022 and IEEE Std 802.3-2022.

Reserving time on the network for certain traffic types can be done through enhancements for scheduled traffic according to IEEE Std 802.1Q-2022, 8.6.8.4. An aligned gating cycle needs to be defined for this method to work. Once a gating cycle is defined, portions of a cycle time can either be allocated to streams or classes of streams.

Multiple Talker/Listener(s) pairs can be used for streams between end stations. Engineered time-triggered transmit can be used to coordinate transmission of all the traffic that shares a network to meet application requirements.

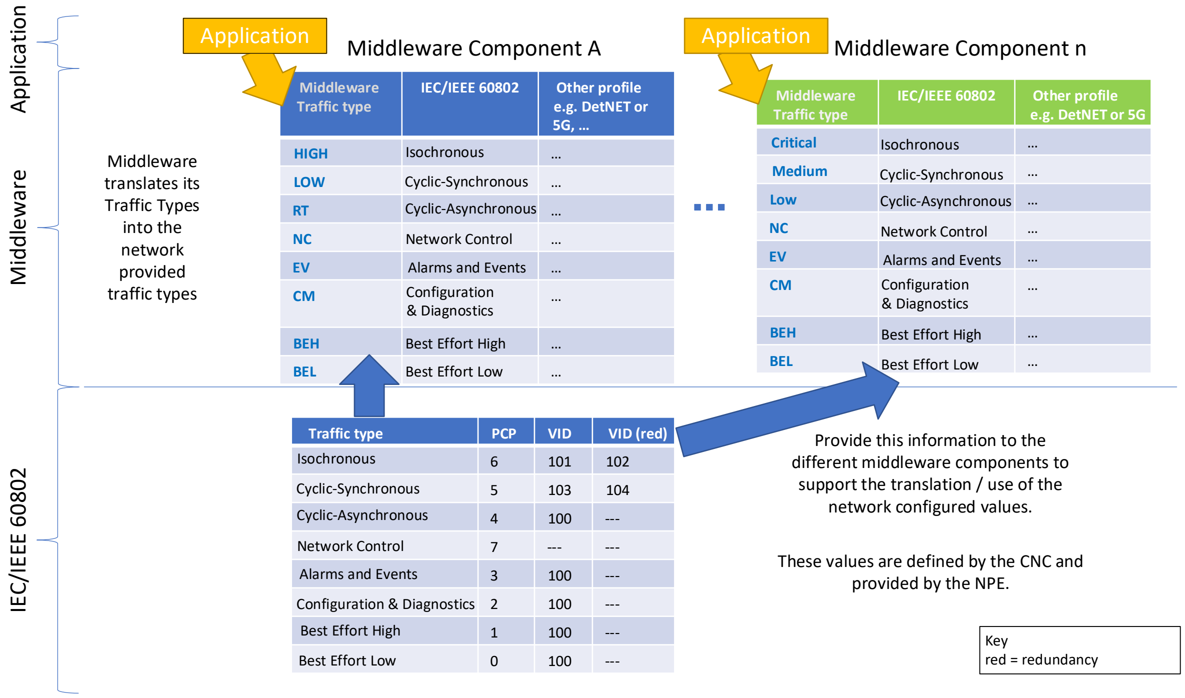
Creating a traffic load model in advance allows analysis of resulting traffic. It can be used to select and implement appropriate mechanisms to achieve latency requirements.

4.6 Translation between middleware and network provisioning

4.6.1 Interfaces of type l2vlan

Application engineering can be done without knowledge of the network provisioning. Since the application is not aware of the network provisioning, it cannot directly map to the network configuration, for example, the use of PCP or VID as configured in the network. This problem is solved by providing a translation table, in the form of a YANG module definition, to the middleware. The IA-station's local YANG datastore contains this information.

Figure 6 and Figure 7 show examples of the translation models.



Middleware Traffic type	IEC/IEEE 60802	Other profile e.g. DetNET or 5G, ...
HIGH	Isochronous	...
LOW	Cyclic-Synchronous	...
RT	Cyclic-Asynchronous	...
NC	Network Control	...
EV	Alarms and Events	...
CM	Configuration & Diagnostics	...
BEH	Best Effort High	...
BEL	Best Effort Low	...

Middleware Traffic type	IEC/IEEE 60802	Other profile e.g. DetNET or 5G
Critical	Isochronous	...
Medium	Cyclic-Synchronous	...
Low	Cyclic-Asynchronous	...
NC	Network Control	...
EV	Alarms and Events	...
CM	Configuration & Diagnostics	...
BEH	Best Effort High	...
BEL	Best Effort Low	...

Traffic type	PCP	VID	VID (red)
Isochronous	6	101	102
Cyclic-Synchronous	5	103	104
Cyclic-Asynchronous	4	100	---
Network Control	7	---	---
Alarms and Events	3	100	---
Configuration & Diagnostics	2	100	---
Best Effort High	1	100	---
Best Effort Low	0	100	---

Provide this information to the different middleware components to support the translation / use of the network configured values.

These values are defined by the CNC and provided by the NPE.

Figure 6 – Traffic type translation example

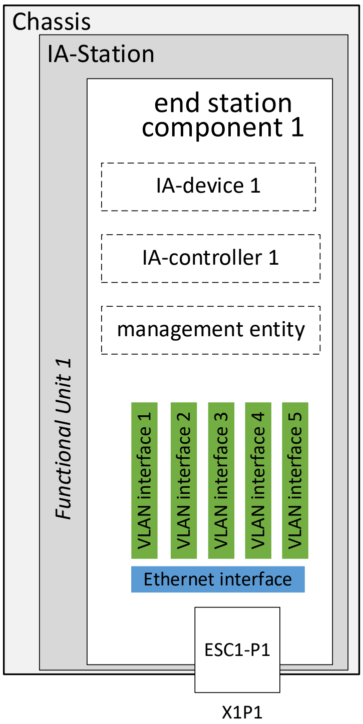


Figure 7 – IETF Interfaces used for Traffic Type Translation

Interfaces of type l2vlan (IETF RFC 7224) can be used to provide the required mapping information to all installed middleware and applications.

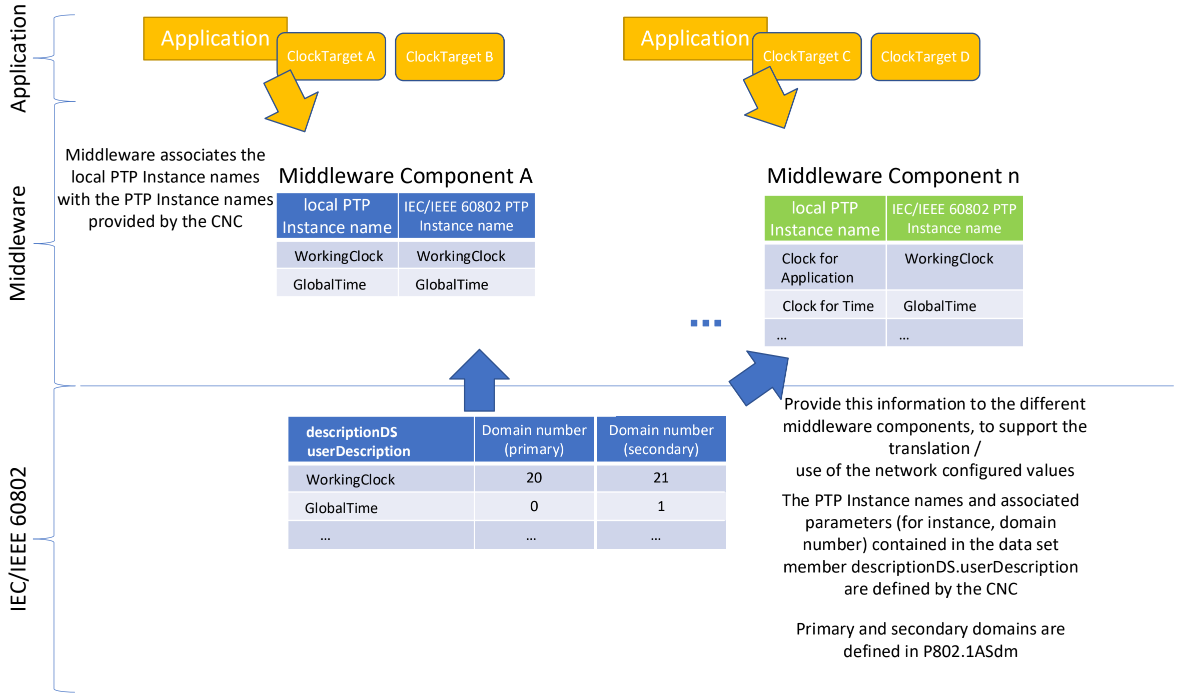
The name string of the l2vlan interfaces can provide the vlan-id, the assigned traffic types with their PCP values and redundancy information (see 6.4.2.5).

4.6.2 PTP Instances

PTP domain numbers are also configured during network provisioning. The middleware needs to know which PTP domain is assigned to which target clock. This is done by providing descriptionDS.userDescription names according to IEEE Std 1588-2019, 8.2.5.5 to create a translation table.

descriptionDS.userDescription names allow the support of multiple middleware components at one IA-station using the same PTP Instances (see 6.2.13). An IA-station's local database stores this information.

Figure 8 and Figure 9 show examples of the translation models.



Middleware associates the local PTP Instance names with the PTP Instance names provided by the CNC

local PTP Instance name	IEC/IEEE 60802 PTP Instance name
WorkingClock	WorkingClock
GlobalTime	GlobalTime

local PTP Instance name	IEC/IEEE 60802 PTP Instance name
Clock for Application	WorkingClock
Clock for Time	GlobalTime
...	...

descriptionDS userDescription	Domain number (primary)	Domain number (secondary)
WorkingClock	20	21
GlobalTime	0	1
...	...	...

Provide this information to the different middleware components, to support the translation / use of the network configured values

The PTP Instance names and associated parameters (for instance, domain number) contained in the data set member descriptionDS.userDescription are defined by the CNC

Primary and secondary domains are defined in P802.1ASdm

Figure 8 – PTP Instance Translation Example

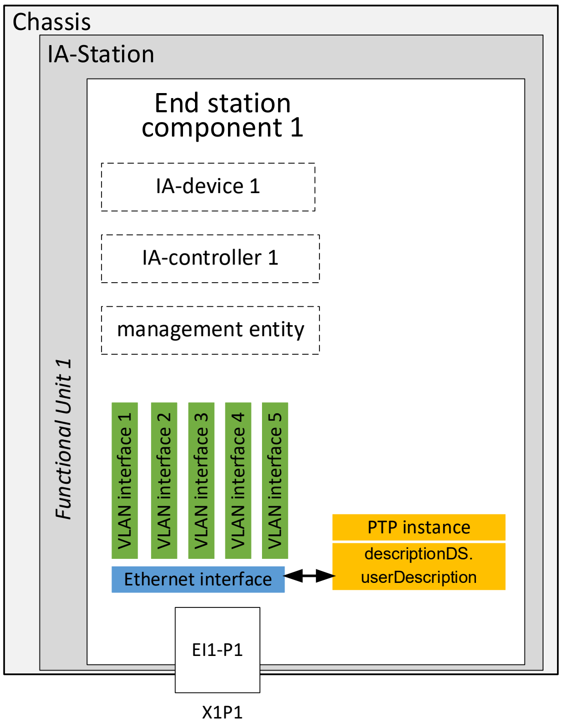


Figure 9 – descriptionDS.userDescription used for PTP Instance Translation

The userDescription contains the clock type (i.e., WorkingClock, GlobalTime, or both). This information is used by the middleware to align to the intended ClockTarget or ClockSource (see 6.2.13).

4.7 Industrial traffic types

4.7.1 General

Industrial automation applications make use of different traffic schemes/types for different functionalities (for example, parameterization, control, alarming). The various traffic patterns have different characteristics, and thus impose different requirements on a network. To specify these traffic types, a two-step approach is used:

a) First define characteristics of generic traffic types (traffic-type-categories) and

b) Second define instances of the generic traffic types, i.e., the traffic types.

4.7.2 Traffic type characteristics

The traffic type characteristics in Table 2 enable the identification of several distinct traffic types that are shared among sets of industrial applications.

Table 2 – Traffic type characteristics

Characteristic	Description
Cyclic	Traffic types consist of frames that can either be transmitted on a reoccurring time period (cyclic) or at no set period (acyclic). Available selections are: • Required: traffic frames are transmitted cyclically • Optional: Implementation of cyclic traffic is at the discretion of the user.

Characteristic	Description
Data delivery requirements	Denotes the delivery constraints for the traffic. Four options are specified: • Frame Latency: data delivery of a frame for a given Talker-Listener pair occurs within a bounded timespan. • Flow Latency: data delivery up to a certain number of frames or data size (including bursts of frames) occurring over a defined period. • Deadline: data delivery of a frame to a given Listener occurs at or before a specific point in time. • No: Denotes the case of traffic types with no special data delivery requirements
Time-triggered transmission	Talker data transmission occurs at a specific point in time based upon the Working Clock. Available selections are: • Required • Optional: Implementation of time-triggered transmission is at the discretion of the user. Enhancements of scheduled traffic is only one means of achieving time-triggered transmission. Other, application-based, methods are possible

4.7.3 Traffic type categories

4.7.3.1 General

The two-step approach described in 4.7.1 allows a clear differentiation between characteristics as seen from the “network” point of view and “application” point of view. Traffic-type-categories allow different IEEE 802 feature selections to achieve the goals of a specific network deployment. Four traffic-type-categories are identified in industrial automation systems:

a) IA time-aware stream,

b) IA stream,

c) IA traffic engineered non-stream,

d) IA non-stream.

4.7.3.2 IA time-aware stream

The characteristics of this traffic type category are shown in Table 3.

Table 3 – IA time-aware stream characteristics

Characteristics	
Cyclic	Required
Data delivery requirement	Deadline or Frame Latency
Time-triggered transmission	Required

4.7.3.3 IA stream

The characteristics of this traffic type category are shown in Table 4.

Table 4 – IA stream characteristics

Characteristics	
Cyclic	Required
Data delivery requirement	Frame Latency
Time-triggered transmission	Optional

4.7.3.4 IA traffic engineered non-stream

The characteristics of this traffic type category are shown in Table 5.

Table 5 – IA traffic engineered non-stream characteristics

Characteristics	
Cyclic	Optional
Data delivery requirement	Flow Latency
Time-triggered transmission	Optional

4.7.3.5 IA non-stream

The characteristics of this traffic type category are shown in Table 6.

Table 6 – IA non-stream characteristics

Characteristics	
Cyclic	Optional
Data delivery requirement	No
Time-triggered transmission	Optional

4.7.4 Traffic types

4.7.4.1 General

Table 7 summarizes relevant industrial automation traffic types and their associated characteristics. In an industrial automation system, other applications, such as audio or video, utilizes one of these traffic types. Traffic Type codes are needed for the VLAN naming scheme defined in this document. See 6.4.2.4 for more information.

Table 7 – Industrial automation traffic types summary

Traffic type name	Traffic type code	Cyclic	Data delivery requirements	Time-triggered transmission	Traffic-type-category
Isochronous	H	Required	Deadline	Required	IA time-aware-stream
Cyclic-synchronous	G	Required	Frame Latency	Required	IA time-aware-stream
Cyclic-asynchronous	F	Required	Frame Latency	Optional	IA stream
Alarms & Events	E	Optional	Flow Latency	Optional	IA traffic engineered non-stream
Configuration & Diagnostics	D	Optional	Flow Latency	Optional	IA traffic engineered non-stream
Network Control	C	Optional	Flow Latency	Optional	IA traffic engineered non-stream
Best Effort High	B	Optional	No	Optional	IA non-stream
Best Effort Low	A	Optional	No	Optional	IA non-stream

4.7.4.2 Isochronous

A type of IA time-aware stream traffic. This type of traffic is transmitted cyclically using time-triggered transmission. Listeners have individual deadline requirements. Cycle times are typically in the range of microseconds to tens of milliseconds. Frame size is typically below 500 octets. Talker-Listener pairs are synchronized to the Working Clock. The network is configured by the CNC to provide zero congestion loss for this traffic type. This type of traffic is normally used in control loop tasks.

4.7.4.3 Cyclic-synchronous

A type of IA time-aware stream traffic. This type of traffic is transmitted cyclically using time-triggered transmission. Talker-Listener pairs have individual latency requirements. Cycle times are typically in the range of hundreds of microseconds to hundreds of milliseconds. Frame size is unconstrained except as indicated in 5.5.1. Talker-Listener pairs are synchronized to the Working Clock. The network is configured by the CNC to provide zero congestion loss for this traffic type. This type of traffic is normally used in control loop tasks.

4.7.4.4 Cyclic-asynchronous

A type of IA stream traffic. This type of traffic is transmitted cyclically with latency requirements bounded by the interval as defined in IEEE Std 802.1Q-2022, 46.2.3.5.1. Talker-Listener pairs have individual latency requirements. Cycle times are typically in the range of milliseconds to seconds. Frame size is unconstrained except as indicated in 5.5.1. Data exchanges between Talker-Listener pairs are typically not dependent on the Working Clock. This traffic type typically tolerates limited congestion loss. The network is configured by the CNC to handle this traffic type without loss, up to a certain number of frames or data size.

4.7.4.5 Alarms and events

A type of IA traffic engineered non-stream. This type of traffic is transmitted cyclically or acyclically. This traffic expects bounded latency including time for retransmission in the range of milliseconds to hundreds of milliseconds. The source of the alarm or event typically limits the bandwidth allocated to this traffic. Frame size is unconstrained except as indicated in 5.5.1. Congestion loss can occur. Retransmission to mitigate frame loss is expected. The network is configured by the CNC to handle these frames, including bursts of frames, up to a certain number of frames or data size over a defined period.

4.7.4.6 Configuration and diagnostics

A type of IA traffic engineered non-stream. This type of traffic is transmitted cyclically or acyclically. This traffic expects bounded latency, up to seconds, including time for retransmission. The source of configuration or diagnostics frames typically limits the bandwidth allocated to this traffic. Frame size is unconstrained except as indicated in 5.5.1. Congestion loss can occur. Retransmission to mitigate frame loss is expected. The network is configured by the CNC to handle these frames, including bursts of frames, up to a certain number of frames or data size over a defined period.

4.7.4.7 Network control

A type of IA traffic engineered non-stream. This type of traffic can be transmitted cyclically or acyclically. This traffic expects bounded latency including time for retransmission. Frame size is unconstrained except as indicated in 5.5.1. The network is configured by the CNC to handle these frames, including bursts of frames, up to a certain number of frames or data size over a defined period. If these limits are exceeded congestion loss can occur. Network control is comprised of services required to maintain network operation. Examples include time synchronization, loop prevention, and topology detection.

4.7.4.8 Best effort

A type of IA non-stream. The network is configured by the CNC so that these frames do not interfere with other traffic types. These frames are forwarded when resources are available. Congestion loss resulting in frame drop can occur. It is sometimes desirable to have more than one traffic class for best effort traffic (see Table 8).

4.7.4.9 Traffic class to traffic type mapping

Table 8 provides an example for the usage of traffic classes based on the traffic type:

Table 8 – Example traffic class to traffic type mapping

Traffic class	PCP (8 Queues)	PCP (4 Queues)	Traffic Type
7	6	2	Isochronous

6	5	1	Cyclic-Synchronous
5	4	1	Cyclic-Asynchronous
4	7	3	Network Control
3	3	0	Alarms and Events
2	2	0	Configuration & Diagnostics
1	1	0	Best Effort High
0	0	0	Best Effort Low
NOTE An example mapping of PCP and traffic type to an application is provided in Figure 6.			

The traffic-type-categories definition allows different IEEE 802 feature selections to achieve specified goals. Moreover it helps in identification of the traffic protection mechanisms. Adherence to this example of a common mapping helps minimize potential conflicts between traffic types.

4.8 Security for TSN-IA

4.8.1 General

Subclause 4.8 describes selected aspects of TSN-IA security. Protecting the management of industrial communication is the main objective of TSN-IA security. The protection of communications that use industrial traffic types is not addressed by this document.

4.8.2 Security configuration model

Security configuration is a part of system engineering and configuration. The security configuration in this document does not encompass the supply of configuration objects for middleware and application security. Security configuration settles the prerequisites for protecting the establishment and management of communications that use industrial traffic types (see 4.7). It ensures that the security features of IA-stations (including CNCs) can be used for protecting message exchanges and authorizing the resource accesses during stream establishment and management. This security configuration supplies deployment-specific configuration objects to IA-stations. They encompass:

- Instructions about cryptographic algorithms,
- Credentials and trust anchors,
- Instructions to interpret the outcome of peer entity authentication while enforcing resource access controls, and
- Access control rules and permissions

This security configuration uses NETCONF/YANG request/response exchanges:

- The to-be-configured IA-stations act in NETCONF server role with respect to their security configuration.
- A NETCONF client is responsible for setting-up IA-stations for security. This NETCONF client possesses information about the security relationship to be established during security configuration or about the expectations on the IA-stations in a configuration domain. It can be implemented as part of an interactive or automated process (for example an engineering tool, or CNC operation). As an implication, the security configuration includes options for interactive and automated setup, i.e., security configuration is done by human and/or non-human actors.

  NOTE NETCONF notifications can also be used to recognize events such as a near-term end-of-life of certificate objects, especially EE certificate objects (see IETF RFC 4210, 3.1.1).
- The security configuration exchanges supply deployment-specific objects (trust anchors, credentials etc.) to IA-stations and manages them. IA-stations that are in factory default state can only possess manufacturer-specific security objects (trust anchors, credentials

etc.) when booting initially. The protected NETCONF/YANG exchanges with IA-stations that are in factory default state are outlined in 4.8.3 to 4.8.6.

4.8.3 NETCONF/YANG processing

Securing NETCONF/YANG resources on NETCONF servers is specified by IETF RFC 6241 (NETCONF). Therefore, message exchange protection between NETCONF clients and servers as well as resource access authorization by NETCONF servers is needed:

- IETF RFC 7589 and IETF draft-ietf-netconf-over-tls13 (NETCONF-over-TLS) specify a solution to protect NETCONF message exchanges by TLS.
- IETF RFC 8341 (NACM) specifies three access control points, covering the request/response and notification model in NETCONF according to IETF RFC 8341, 2.1.

NETCONF servers enforce security as shown in Figure 10. The processing steps are executed upon the current configuration of the NETCONF server's YANG modules.

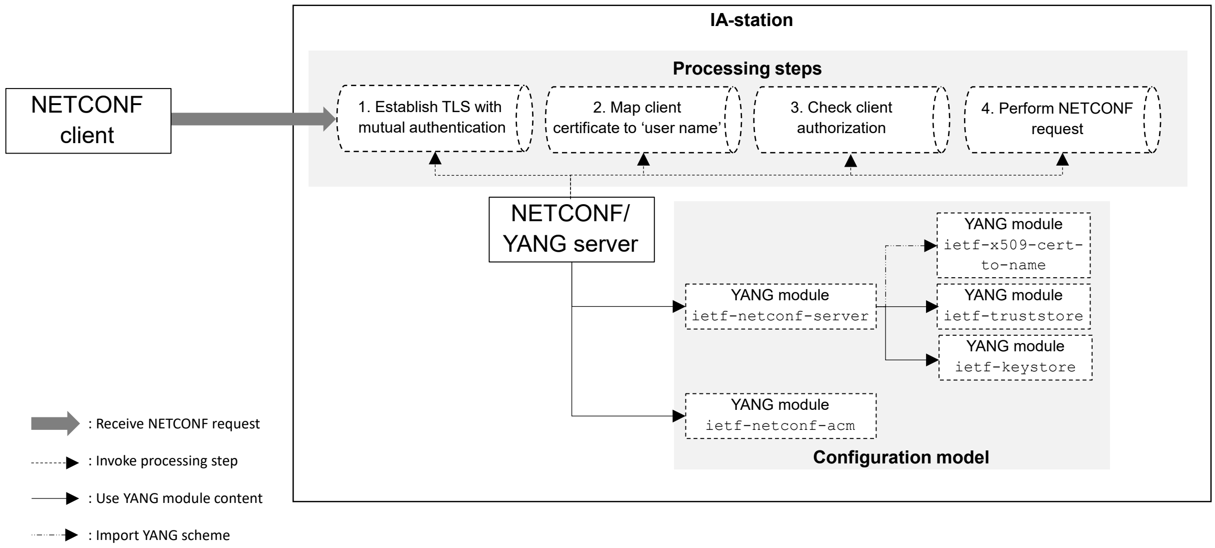

IA-station
Processing steps
NETCONF client
1. Establish TLS with mutual authentication
2. Map client certificate to 'user name'
3. Check client authorization
4. Perform NETCONF request
NETCONF/ YANG server
YANG module ietf-x509-cert-to-name
YANG module ietf-netconf-server
YANG module ietf-truststore
YANG module ietf-keystore
YANG module ietf-netconf-acm
Configuration model
: Receive NETCONF request
: Invoke processing step
: Use YANG module content
: Import YANG scheme

Figure 10 – NETCONF/YANG security processing steps

The processing steps on the side of NETCONF servers are:

1) Establish a TLS connection with mutual authentication: The NETCONF server acts as TLS server and awaits connection requests of NETCONF clients (TLS clients). At the beginning of the TLS handshake, the TLS client and server negotiate the TLS protocol version to be used. During the TLS handshake the NETCONF server authenticates itself towards the NETCONF client by a credential from its ietf-keystore YANG module. In addition, the NETCONF server challenges the NETCONF client for authentication and verifies its authentication by trust anchors in its ietf-truststore YANG module according to 6.3.4. A successful mutual authentication is a prerequiste for proceeding to the next step.

2) Map the client certificate to a username: The NETCONF server maps the authenticated TLS client certificate to a "NETCONF username"[3] by applying an ordered list of mapping instructions. These instructions are provided in its ietf-x509-cert-to-name YANG module. The applicable list item is identified by matching its configured fingerprint (according to IETF RFC 7589, Clause 7) against the certification path that was used for TLS client authentication (an end entity certificate or a CA certificate). According to the map type

[3] In this document, NETCONF username' values do not represent references to human users – in almost all cases.

of the identified list item, the NETCONF server determines the “NETCONF username”. This can be done by extracting information from the end entity certificate of the NETCONF client. A successful certificate-to-”NETCONF username” mapping is a prerequiste for proceeding to the next step.

3) Check client authorization: The NETCONF server checks if the NETCONF client has the permission to access the requested NETCONF/YANG resource based on its “NETCONF username” and the access control rules available in its ietf-netconf-acm YANG module. See 4.8.4 for more information about NETCONF/YANG access control. A successful authorization is a prerequiste for proceeding to the next step.

4) Perform NETCONF request: If all preceding steps succeeded, the NETCONF server performs the NETCONF request.

4.8.4 NETCONF/YANG access control

NACM defines a YANG information model for describing permitted/denied access operations. NETCONF servers are responsible for enforcing access control to their resources according to the information in their ietf-netconf-acm YANG modules. NACM allows the description of access-controlled resources in terms of NETCONF protocol operations, nodes in YANG datastores and/or types of notification events. NACM uses character strings to represent the subject actors i.e., NETCONF clients. These character strings are known as “NETCONF username”. The NACM access control information of a NETCONF server is created, updated, and deleted per IA-station. The management of this information happens along the IA-station lifecycle for example, manufacturing, bootstrapping, operation, maintaining, re-owning, destructing. Moreover, the management of the NACM access control information itself is subject to NACM access control.

This document employs multiple YANG data models for fulfilling its purposes. This extends beyond the above identified YANG modules (see 4.8.3). The NETCONF server on an IA-station enforces access control for NETCONF/YANG resources. To meet this objective, the NETCONF server on an IA-station is supplied with access control information for the used NETCONF/YANG resources. NACM is employed for this purpose and profiles default access control information for the NETCONF/YANG resources (see 6.3.2.2). This relieves other organizations or individuals for example, manufacturers, integrators, operators, owners from being responsible to create NACM access control information for the respective NETCONF/YANG resources.

NACM relies on character strings (known as “NETCONF username”) to refer to clients. NACM access control information as specified in this document, populates the “NETCONF username” character strings in NACM with role names specified in 6.3.2.1.4, c). This allows to create default NACM information without knowing actual names of individual entities. A role name can refer to 0, 1 or more individual entities. It is the responsibility of users to assign role names to individual entities. This happens by binding the assigned role names to the credentials of individual entities. The current form to express this binding is a role extension in the identity certificates of end entities defined in this document. These are NETCONF clients, i.e., these role extensions appear in the end entity certificates of LDevID credentials for NETCONF clients.

As initial step NACM maps the NETCONF username to a set of groups. The set of groups determines the set of rules to be applied for access-controlled resources.

4.8.5 Identity checking

IETF RFC 7589 (NETCONF-over-TLS) specifies that NETCONF clients check the identity of NETCONF servers and that NETCONF servers check the identity of NETCONF clients.

The NETCONF server identity check happens inside NETCONF clients. It matches an actual against an expectation:

- The actual server identity is established by the end entity certificate of the NETCONF server (authenticated by means of TLS).
- The expectations on server identity are established by the information that is used to connect to the NETCONF server.

IETF RFC 7589 refers to IETF RFC 6125, Clause 6, for the details of retrieving the actual and comparing it against the expected.

The NETCONF client identity check happens inside NETCONF servers. It also matches an actual against an expectation:

- The actual client identity is established by the end entity certificate of the NETCONF client (authenticated by means of TLS).
- The expectations on client identity are established by the contents of the YANG modules ietf-netconf-acm and ietf-x509-cert-to-name.

The details of this check are subject to the requested NETCONF operation. IETF RFC 7589, Clause 7, specifies the mapping of an authenticated client certificate to a “NETCONF username” whose permissions are then enforced by IETF RFC 8341 (NACM). More information is provided in 4.8.3, steps 2 and 3.

4.8.6 Secure device identity

4.8.6.1 Device Identity

The term ‘device’ originates from IEEE Std 802.1AR-2018. It matches the term IA-station in this document.

The device identity refers to a set of information items about a device that:

- describes a device as a physical or virtual entity in a distributed system (identifier and/or attribute information);
- is used by a device to describe itself as such entity (identifier and/or attribute information);
- allows to interact with this device (addressing information i.e., a specific identifier class).

The targeted use case, for example application data exchanges, configuration exchanges, inventory, or ordering, determines the required amount of identity information about a device.

The device identity of any single IA-station encompasses:

- MAC addresses, IP addresses, TCP ports, DNS names.
- ietf-hardware YANG module contents (IETF RFC 8348).

4.8.6.2 Verifiable Device Identity

Certain aspects of device identity are verified before relying on them during online interactions. These are examples.

- DNS names or IP addresses are used to call the management entity of an IA-station i.e., its NETCONF/YANG server. Their value represents the caller's expectation on the identity of their responder in network communications. Verification of the responder's identity helps defeat DNS spoofing, component impersonation and man-in-the-middle attacks. This is specified by IETF RFC 7589 and described in IETF RFC 6125, Clause 6. Passing this check is a prerequisite before NETCONF application exchanges can happen.
- mfg-name values in instances of the ietf-hardware YANG module. These values make claims about the IA-station manufacturer. Their verification is a means to protect against counterfeiting.

The verification of IA-station identity happens according to a model that is fully specified by this document. That verification can be done in a manufacturer-agnostic manner. This verification is important before supplying locally significant credentials especially LDevID-NETCONF to IA-stations that are in factory-default state.

4.8.6.3 Verification Support Mechanisms

4.8.6.3.1 General

Subclause 4.8.6.3 considers mechanisms that support device identity verification during online interactions with IA-stations.

4.8.6.3.2 Secure Transports

Sending information in plain form over a protected channel, e.g., ietf-hardware YANG module contents via NETCONF-over-TLS protects the transferred information during its transit through the network but does not vouch for the correctness of the received information e.g., the mfg-name value.

4.8.6.3.3 Secure Information

Protecting information objects by means of a cryptographic authentication code or digital signature enables verification of the authenticity and integrity of that information. These cryptographic authentication codes can use symmetric or asymmetric schemes. In case of asymmetric schemes, raw and self-signed public keys need to be distinguished from CA-signed public keys.

Asymmetric schemes with CA-signed public keys are preferable for the verifiable device identity use case: claimants and verifiers share a public key; the claimant possesses the corresponding private key. The establishment and storage of the shared public keys uses public key certificates. For this approach self-signed CA certificates are to be established in an authentic manner. The number of self-signed CA certificates is independent from the number of verifiers (CNCs) as well as claimants (IA-stations).

4.8.6.3.4 IDevID and LDevID Credentials

IDevID and LDevID credentials are specified by IEEE Std 802.1AR-2018. These objects are comprised of a certification path and a private key. The certification path encompasses an end entity certificate which contains verifiable device identity in a CA-signed form. The device identity verification happens after validating the certification path (IETF RFC 5280, Clause 6) and checking the proof-of-possession for the private key. The certification path validation demands trust anchors as input arguments (IETF RFC 5280, 6.1.1 input argument (d)).

Two types of credentials are distinguished by IEEE Std 802.1AR-2018:

- IDevIDs are issued by device manufacturers. They represent an initial identity as it is known at device production-time. The initial device identity is not locally significant: it cannot contain deployment-specific information such as DNS names or IP addresses.
- LDevIDs are issued by other actors e.g., a device user. They represent a locally significant device identity: they can contain deployment-specific information e.g., DNS names or IP addresses.

IEEE Std 802.1AR-2018, Clause 6, uses signature suites to describe the subject public key and the signature fields in IDevID and LDevID certification paths. This notion is different from TLS cipher suites.

NOTE IDevID and LDevID credentials also serve purposes beyond secure device identity, for instance the realization of secure transports. This facilitates the use case of NETCONF/YANG security setup from factory default state.

4.8.6.3.5 IDevID Items beyond IEEE Std 802.1AR-2018

IEEE Std02.1AR allows verification of the following identity items:

- certificate issuer (not necessarily: manufacturer) by issuer field (data type: ASN.1 Name) and
- if present: device instance by serialNumber value (data type: ASN.1 PrintableString).

NOTE 1 IEEE Std 802.1AR-2018 represents the initial device identity as an optional serialNumber attribute (OID 2.5.4.5) in the subject field of the EE certificate. This value is unique within the domain of significance of the EE certificate issuer.

NOTE 2 This verification can happen after certification path validation and the proof-of-possession checking for the private key.

The following bullet points describe options beyond IEEE Std 802.1AR-2018 for verifying the device identity of IA-stations in factory default state. It also identifies informational items needed for the corresponding checks:

- IA-station manufacturer check: using names that identify IA-station manufacturers e.g., mfg-name in ietf-hardware YANG module,
- IA-station type check: using attributes that identify IA-station types e.g., model-name, hw-revision, description in ietf-hardware YANG module, and
- IA-station instance check: using values that identify IA-station instances e.g., serial-num in ietf-hardware YANG module.

The following model described in the bullet points applies to the verification of the initial device identity of IA-stations:

- the set of to-be-conducted checks is determined by IA-station and CNC users,
- an IA-station uses IDevID credentials to prove its device identity. The checking happens by means of online interactions in the operational network. It happens automatically and is done by CNCs. This does not depend on configuration-domain external repositories, and
- other stakeholders e.g., middleware/application consortia or individual manufactures are allowed to additionally express information items in IDevID credentials to reflect their device identity model. CNCs do not assess such additional information.

4.8.6.3.6 Device Identity Representation in IDevID and LDevID Credentials

The best practices for representing verifiable device identity information in IDevID and LDevID credentials (see 6.3.3.2.2 for more information) are:

- Corresponding information (actual values or references to them) appears in EE certificates:
  - IDevID EE certificates bind initial device identity items that are known by the device manufacturer at production time e.g., mfg-name.
  - LDevID EE certificates bind locally significant device identity items that are known by other actors such as device users e.g., DNS names or IP addresses. They can also bind initial device identity information.
- Items that encode device naming information appear in the subjectAltName extension.

  NOTE This is specified in IETF RFC 5280, 4.2.1.6. It is further explained in IETF RFC 6125, 2.3.
- A binding can take one of following forms. Multiple forms can appear in one EE certificate:
  - By-value: the verifiable device identity information is represented by its value inside the IDevID resp. LDevID EE certificate. Examples are:
    - the product serialNumber in IDevID credentials (IEEE Std 802.1AR-2018) and,
    - the hostname of the NETCONF/YANG server in LDevID-NETCONF credentials (IETF RFC 6125, Clause 6).
  - By-ref: the verifiable device identity information is represented by a reference inside the IDevID resp. LDevID EE certificate, not by its value:
    - The actual value can be provided by the device itself or by a device-external source, and
  - If it is provided in form of an unprotected information object, then the reference object that is embedded to EE certificates includes a digest value.

5 Conformance

5.1 General

A claim of conformance to this document is a claim that the behavior of an implementation of an IA-station (see 5.5, 5.6) with its Bridge components (see 5.7, 5.8) and end station components (see 5.9, 5.10) meets the mandatory requirements of this document and may support options identified in this document. Furthermore this document includes conformance requirements for CNC and CUC implementations (see 5.11, 5.13).

5.2 Requirements terminology

The verbal forms for required expressions of provisions follow the conventions:

a) Requirements terminology is provided in the ISO/IEC Directives Part 2:2021, Clause 7. This document can be found at www.iec.ch/members_experts/refdocs.

b) The Profile Conformance Statement (PCS) proformas (see Annex A) reflect the occurrences of the words “shall,” “may,” and “should” within this document.

c) This document avoids needless repetition and apparent duplication of its formal requirements by using is, is not, are, and are not for definitions and the logical consequences of conformant behavior. Behavior that is permitted but is neither always required nor directly controlled by an implementer or administrator, or whose conformance requirement is detailed elsewhere, is described by can. Behavior that never occurs in a conformant implementation or system of conformant implementations is described by cannot. The word allow is used as a replacement for the phrase “Support the ability for,” and the word capability means “can be configured to.”

5.3 Profile conformance statement (PCS)[4]

The supplier of an implementation that is claimed to conform to this document shall provide the information necessary to identify both the supplier and the implementation and shall complete a copy of the PCS proforma provided in Annex A.

5.4 Conformance classes

This document includes conformance requirements and options that are related to an entire station, as well as conformance requirements and options that are related to single Bridge or end station components within an IA-station. Figure 11 illustrates this conformance model.

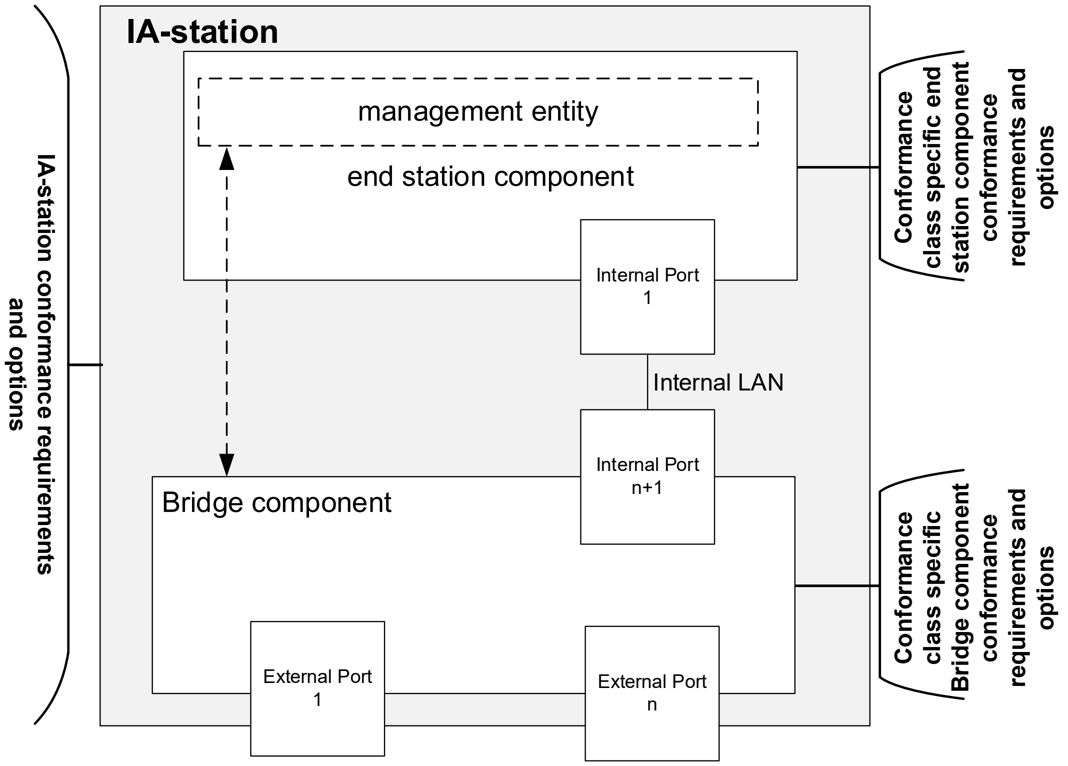

IA-station
management entity
end station component
Internal Port 1
Internal LAN
Internal Port n+1
Bridge component
External Port 1
External Port n
IA-station conformance requirements and options
Conformance class specific end station component conformance requirements and options
Conformance class specific Bridge component conformance requirements and options

Figure 11 – IA-station conformance model

This document supports a variety of industrial use cases. In some of these use cases, support of certain TSN features might be mandatory, while in others, supporting these features could lead to non-optimal implementations. Therefore, this document defines two conformance

[4] Copyright release for the PCS: Users of this document may freely reproduce the PCS contained in this document so that they can be used for their intended purpose.

classes that are applicable both to Bridge components and end station components. Conformance Class A (ccA) is feature rich, i.e., tailored to use cases requiring support of many TSN-IA features. Conformance Class B (ccB) targets implementations that are more resource constrained. The details for the conformance classes are specified in 5.7 and 5.8 for Bridge components, and in 5.9 and 5.10 for end station components.

NOTE 1 It is the responsibility of the IA-station manufacturer to carefully consider the implications of mixing ccA and ccB Bridge components and end station components in a single IA-station.

NOTE 2 It is the responsibility of the user to carefully consider the implications of mixing ccA and ccB Bridge components and end station components in a single Configuration Domain.

NOTE 3 Any Bridge compliant to this document is an IA-station. Any IA-station contains a management entity (i.e., an end station component).

5.5 IA-station requirements

5.5.1 IA-station PHY and MAC requirements for external ports

IA-stations for which a claim of conformance to this document is made shall support the following list of requirements for external ports.

a) Media Access Control (MAC) service specification according to IEEE Std 802.3-2022, Clause 2.
    a) Media Access Control (MAC) frame and packet specifications according to IEEE Std 802.3-2022, Clause 3, especially the MAC Client Data field size according to IEEE Std 802.3-2022, 3.2.7, item c).
    b) Layer Management according to IEEE Std 802.3-2022, 5.2.4.
    c) Implement at least one IEEE Std 802.3-2022 MAC that shall operate in full-duplex mode, and associated IEEE Std 802.3-2022 PHY with a data rate of at least one of speed: 10 Mb/s, 100 Mb/s, 1 000 Mb/s, 2,5 Gb/s, 5 Gb/s, or 10 Gb/s together with the corresponding managed objects:
        1) 10BASE-T1L MAU type according to IEEE Std 802.3-2022, Clauses 22 and 146,
        2) 100BASE-TX and 100BASE-FX MAU types according to IEEE Std 802.3-2022, Clauses 21, 22, 24, 25, 26, 30, 31 and IEEE Std 802.3-2022, Annexes 23A, 28A, 28B, 28C, 28D, 31A, 31B, 31C, and 31D,
        3) 1000BASE-T and 1000BASE-SX MAU types according to IEEE Std 802.3-2022, Clauses 28, 34, 35, 36, 37, 38, and 40,
        4) 2.5GBASE-T and 5GBASE-T MAU types according to IEEE Std 802.3-2022, Clauses 28, 125, and 126,
        5) 2.5GBASE-T1 and 5GBASE-T1 MAU types according to IEEE Std 802.3-2022, Clause 149,
        6) 10GBASE-T and 10GBASE-SR MAU types according to IEEE Std 802.3-2022, Clauses 44, 46, 47, 49, 51, 52, 55, and IEEE Std 802.3-2022, Annexes 48A and 55A,
        7) 10GBASE-T1 MAU type according to IEEE Std 802.3-2022, Clause 149,
        8) 100BASE-T1 MAU type according to IEEE Std 802.3-2022, Clause 96 and,
        9) 1000BASE-T1 MAU type according to IEEE Std 802.3-2022, Clause 97.

d) Support the YANG features and leaves of the ieee802-ethernet-interface module according to 6.4.9.2.1.

e) Ethernet support for time synchronization protocols according to IEEE Std 802.3-2022, Clause 90.

NOTE Clauses and subclauses not mentioned can be implemented but are not part of a conformity assessment.

5.5.2 IA-station topology discovery requirements

IA-stations for which a claim of conformance to this document is made shall support the following list of requirements.

a) The required capabilities according to IEEE Std 802.1AB-2016, 5.3 and IEEE Std 802.1ABcu-2021, 5.3.

b) Topology discovery and verification according to 6.5.

c) The YANG features and leaves of the ieee802-dot1ab-lldp module according to 6.4.9.2.2.

5.5.3 IA-station requirements for time synchronization

These requirements are related to the entire IA-station with all its PTP Instances and PTP Ports. IA-stations for which a claim of conformance to this document is made shall support the following list of requirements.

a) PTP Instance requirements according to IEEE Std 802.1AS-2020, 5.4.1 items a) through i).

NOTE A gPTP domain in a PTP End Instance can be used for Global Time, Working Clock, or both.

b) Timing and synchronization management according to IEEE Std 802.1AS-2020, 5.4.2 items j) and k).

c) PTP Instance requirements according to 6.2.2.

d) PTP Protocol requirements according to 6.2.3.

e) Error generation limits according to 6.2.4.

f) PtpInstanceSyncStatus state machine according to 6.2.5.

g) The transmission of the Drift_Tracking TLV according to IEEE P802.1ASdm, 5.4.2 item n).

h) The PtpInstanceSyncStatus according to 6.2.5.

i) External port configuration capability according to IEEE Std 802.1AS-2020, 5.4.2 item g).

j) MAC-specific timing and synchronization methods for IEEE Std 802.3 full-duplex links according to IEEE Std 802.1AS-2020, 5.5 items a) through d) and item h).

k) The YANG features and leaves of the:

   i) ieee1588-ptp module according to 6.4.9.2.3.1,

   ii) ieee802-dot1as-ptp module according to 6.4.9.2.3.2, and

   iii) ieee802-dot1as-hs module according to 6.4.9.2.3.3.

l) The message timestamp point according to IEEE802.1AS-2020, 11.3.9.

m) The Common Mean Link Delay Service (CMLDS) according to IEEE802.1AS-2020, 11.2.17.

n) The descriptionDS according to IEEE Std 1588-2019, 8.2.5.

5.5.4 IA-station requirements for management

5.5.4.1 General

These requirements are related to the secured management of an entire IA-station independent of the internal component structure.

5.5.4.2 Secure management exchanges

IA-stations for which a claim of conformance to this document is made shall support the following list of requirements.

a) NETCONF server functionality according to IETF RFC 6241 including:

   1) Candidate configuration capability as described in IETF RFC 6241, 8.3,

   2) Rollback-on-Error capability as described in IETF RFC 6241, 8.5, and

   3) Validate capability as described in IETF RFC 6241, 8.6.

NOTE The SSH transport protocol, which is mandatory in IETF RFC 6241, 2.3, is not used by IA-stations conformant to this document.

b) NETCONF-over-TLS server supporting TLS version 1.2, according to IETF RFC 7589, with the cipher suite TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256, based on the signature algorithm ECDSA with SHA-256 and Curve P-256 (NIST FIPS 186-5 and NIST SP 800-186, 3.2.1.3), according to 6.3.2.1 and 6.3.4.

c) Secure Device Identity according to 6.3.3 and IEEE Std 802.1AR-2018, 5.3 a) using the signature suite in IEEE Std 802.1AR-2018 9.2, 5.3 d), and 5.3 i).

d) PKIX (IETF RFC 5280) according to 6.3.2.1.4 and IETF RFC 5280, 4.1, 4.2.1.1-3, 4.2.1.6, 6.1, 6.2.

e) NACM (IETF RFC 8341) supporting six different roles according to 6.3.2.1.4 c).

f) The YANG features and leaves of the:

    1) [draft-]ietf-keystore module according to 6.4.9.2.4.1,

    2) ietf-netconf-acm module according to 6.4.9.2.4.2 and,

    3) [draft-]ietf-truststore according to 6.4.9.2.4.3.

g) NETCONF Event Notifications according to IETF RFC 5277 including operations according to IETF RFC 5277, Clause 2.

h) Dynamic Subscription to YANG Events and Datastores over NETCONF as described in IETF RFC 8640.

i) NETCONF Extensions to Support the Network Management Datastore Architecture (NMDA) as described in IETF RFC 8526.

j) DHCP client according to IETF RFC 2131, 4.1, 4.2, and 4.4.

5.5.4.3 IA-station management YANG modules

IA-stations for which a claim of conformance to this document is made shall support the YANG features and leaves for IA-station management of the:

a) ietf-system-capabilities module according to 6.4.9.2.5.1,

b) ietf-yang-library module as according to 6.4.9.2.5.2,

c) ietf-yang-push module according to and 6.4.9.2.5.3,

d) ietf-notification-capabilities module according to 6.4.9.2.5.4,

e) ietf-subscribed-notifications module according to 6.4.9.2.5.5,

f) Diagnostics using YANG-Push subscriptions according to 6.4.7,

g) ietf-netconf-monitoring module according to 6.4.9.2.5.6,

h) ietf-system module according to 6.4.9.2.5.7,

i) ietf-hardware module according to 6.4.9.2.5.8,

j) ietf-interfaces module according to 6.4.9.2.5.9,

k) ieee802-dot1q-bridge module according to 6.4.9.2.5.10,

l) iecieee60802-ethernet-interface module according to 6.4.9.2.5.11 and,

m) ietf-netconf-server according to 6.4.9.2.5.12.

5.5.4.4 Digital data sheet

IA-stations for which a claim of conformance to this document is made shall provide a 60802 instance data file according to 6.4.8. The instance data file shall contain at least the YANG nodes of 6.4.9 that are marked with [m] or [c].

NOTE It is the users responsibility to ensure that the filename is unique by using a standardized mechanism (for example, GUID, URL, or ReverseDomainName).

5.6 IA-station options

5.6.1 IA-station PHY and MAC options for external ports

IA-stations for which a claim of conformance to this document is made may support the following list of requirements.

a) Power over Ethernet (PoE) over 2 Pairs according to IEEE Std 802.3-2022, Clause 33.
b) Power Interfaces according to IEEE Std 802.3-2022, Clause 104.
c) Power over Ethernet according to IEEE Std 802.3-2022 Clause 145.

5.6.2 IA-station options for time synchronization

IA-stations for which a claim of conformance to this document is made may support the following list of requirements.

a) The media-independent master capability according to IEEE Std 802.1AS-2020, 5.4.2 item b).
b) Grandmaster PTP Instance capability according to IEEE Std 802.1AS-2020, 5.4.2 item c).
c) More than one PTP port as a PTP Relay Instance according to IEEE Std 802.1AS-2020, 5.4.2 item d).
d) Transmit of the Signaling message according to IEEE Std 802.1AS-2020, 5.4.2 item e).
e) support more than 1 PTP Instance according to IEEE Std 802.1AS-2020, 5.4.2 item f).
f) The SyncIntervalSetting state machine according to IEEE Std 802.1AS-2020, 5.4.2 item h),
g) One or more application interfaces according to IEEE Std 802.1AS-2020, 5.4.2 item i).
h) Hot standby redundancy requirements according to P802.1ASdm, 5.4.2, item m).

5.6.3 IA-station options for management

IA-stations for which a claim of conformance to this document is made may support the following list of requirements.

a) Writable-Running capability according to IETF RFC 6241, 8.2.
b) Confirmed Commit capability according to IETF RFC 6241, 8.4.
c) Distinct Startup capability according to IETF RFC 6241, 8.7.
d) URL capability according to IETF RFC 6241, 8.8.
e) XPath capability according to IETF RFC 6241, 8.9.
f) NETCONF-over-TLS server supporting TLS version 1.2, according to IETF RFC 7589, with one or more of the following cipher suites.
    - TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 according to IETF RFC 5289, 3.2, Clause 5, and
    - TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 according to IETF RFC 7905, 2 and Clause 3.

    and based on one or more of the following signature algorithms:
    - ECDSA with SHA-512 and Curve P-521 according to NIST FIPS 186-5 and NIST SP 800-186, 3.2.1.5,
    - Ed25519 according to IETF RFC IETF RFC 8032, 5.1, and
    - Ed448 according to IETF RFC 8032, 5.2.
g) NETCONF-over-TLS server supporting TLS version 1.3, according to IETF RFC 7589 and IETF draft-ietf-netconf-over-tls13, with one or more of the following cipher suites according to IETF RFC 8446, 9.1.
    - TLS_AES_128_GCM_SHA256,
    - TLS_AES_256_GCM_SHA384, and
    - TLS_CHACHA20_POLY1305_SHA256.

    and one or more of the following signature schemes:
    - ecdsa_secp256r1_sha256 according to NIST FIPS 186-5 and NIST SP 800-186, 3.2.1.3,

- ecdsa_secp521r1_sha512 according to NIST FIPS 186-5 and NIST SP 800-186, 3.2.1.5,
- ed25519 according to IETF RFC 8032, 5.1, and
- ed448 according to IETF RFC 8032, 5.2.

h) Support the YANG features and leaves of the ietf-keystore (IETF RFC “Internet-Draft (I-D) “ A YANG Data Model for a Keystore - draft-ietf-netconf-keystore) with component-internal or component-external generation of asymmetric key pairs according to 6.3.4.3.

i) PKIX according to IETF RFC 5280, 4.2.1.13, Clause 5, and 6.3.

IA-stations for which a claim of conformance to this document is made should support Internal key generation according to 6.3.4.3.2.

5.7 Bridge component requirements

5.7.1 Common Bridge component requirements

A Bridge component implementation of any conformance class for which a claim of conformance to this document is made shall support the following list of requirements.

a) C-VLAN component requirements according to IEEE Std 802.1Q-2022, 5.5 and 5.4 except item o) in IEEE Std 802.1Q-2022, 5.4.

b) The use of Customer VLAN Identifiers (C-VID).

c) FDB to contain Static and Dynamic VLAN Registration Entries for a minimum of 10 VIDs, up to a maximum of 4 094 VIDs, according to IEEE Std 802.1Q-2022, 8.8.

NOTE 1 An example use case for 8 VIDs: 2 VIDs for IA time-aware stream or IA stream traffic, 2 VIDs for IA time-aware stream or IA stream redundancy, 4 VIDs for IA traffic engineered non-stream or IA non-stream traffic, 1 isolation VID, and 1 default VID (see 6.4.5.2).

d) Translation of VIDs through support of the VID Translation Table or through support of both the VID Translation Table and Egress VID translation table on one or more Bridge Ports according to IEEE Std 802.1Q-2022, 6.9.

e) The strict priority algorithm for transmission selection on each port for each traffic class according to IEEE Std 802.1Q-2022, 8.6.8.1.

f) The capability to disable Priority-based flow control if it is implemented according to IEEE Std 802.1Q-2022, Clause 36.

g) The Priority Regeneration requirements according to IEEE Std 802.1Q-2022, 5.4.1, item o).

h) MST according to IEEE Std 802.1Q-2018, 5.4.1.1 a) to i) and k) to o) and 6.4.2.4.

i) TE-MSTID according to IEEE Std 802.1Q-2022, 8.6. and 8.8 and IEEE Std 802.1Q-2022, 5.5.2.

j) Spanning tree, VLAN, and TE-MSTID configuration according to 6.4.2.4.

k) The l2vlan interface types per 6.4.2.5.

l) Flow meters including support of at least 3 flow meters per port, according to IEEE Std 802.1Q-2022 8.6.5.3 items a), b), and f) and 8.6.5.5 items a) through c). A flow meter should set following IEEE Std 802.1Q-2022, 8.6.5.5 parameters to values:

- Item d) Excess Information Rate (EIR) = 0,
- Item e) Excess burst size (EBS) = 0, and
- Item g) Color mode (CM) = color_blind.

NOTE 2 When CM = color_blind, DropOnYellow (IEEE Std 802.1Q-2022, 8.6.5.1.3, item h), MarkAllFramesRed (IEEE Std 802.1Q-2022, 8.6.5.1.3, item j), and MarkAllFramesRedEnable (IEEE Std 802.1Q-2022, 8.6.5.1.3, item i) are not used.

NOTE 3 For example, an implementation could contain one flow meter for broadcast traffic, one flow meter for multicast traffic and one flow meter for unicast traffic.

5.7.2 ccA Bridge component requirements

A Bridge component implementation for which a claim of conformance to ccA of this document is made shall support the following list of requirements.

a) Common Bridge component requirements according to 5.7.1.

b) At least 2 PTP Instances according to 5.5.3.

c) Eight queues according to IEEE Std 802.1Q-2022, 8.6.6.

d) Enhancements for scheduled traffic for data rates of 100 Mb/s and 1 Gb/s according to IEEE Std 802.1Q-2022, 5.4.1 items ab) and ac) including:

  1) a tick granularity of less than or equal to 10 ns according to IEEE Std 802.1Q-2022, 8.6.8.4,

  2) The allowable error budget between the transmission selection timing point and the on-the-wire timing point, less any error budget for the PHY (IEEE Std 802.1Q-2022, Figure 12.6), of less than or equal to 10 ns, and

  NOTE Transmission selection timing points have a granularity of 1 ns; however, operation is determined by the precision of the "tick" event.

  3) Support the YANG features and leaves of the ieee802-dot1q sched module according to 6.4.9.3.2.

e) Frame preemption according to IEEE Std 802.1Q-2022, 5.4.1 item ad), for data rates of 100 Mb/s and 1 Gb/s, including:

  1) Support of Interspersing Express Traffic with preemptable traffic according to IEEE Std 802.3-2022, Clause 99, including support of the Additional Ethernet Capabilities for TLV in an LLDPDU to indicate supported functions of frame preemption according to IEEE Std 802.3-2022, 79.3.7, and

  2) Support of the YANG features and leaves of the ieee802-dot1q-preemption module according to 6.4.9.3.4.

5.7.3 ccB Bridge component requirements

A Bridge component implementation for which a claim of conformance to ccB of this document is made shall support the following list of requirements.

a) Common Bridge component requirements according to 5.7.1.

b) At least 1 PTP Instance according to 5.5.3.

c) At least four queues according to IEEE Std 802.1Q-2022, 8.6.6.

5.8 Bridge component options

5.8.1 Common Bridge component options

A Bridge component implementation of any conformance class for which a claim of conformance to this document is made may support the operation of the credit-based shaper algorithm according to 802.1Q, 8.6.8.2 on all Ports as the transmission selection algorithm for at least 4 traffic classes including support of the YANG features and leaves of the <ieee-cbs> module according to 6.4.9.3.5.

5.8.2 ccA Bridge component options

A Bridge component implementation for which a claim of conformance to ccA of this document is made may support the following list of requirements.

a) Any or none of the common Bridge component options according to 5.8.1.

b) More than 2 PTP Instances according to 5.5.3.

c) Enhancements for scheduled traffic for data rates of 10 Mb/s, 2,5 Gb/s, 5 Gb/s, and 10 Gb/s according to IEEE Std 802.1Q-2022, 5.4.1 items ab) and ac) including:

1) a tick granularity of less than or equal to 10 ns according to IEEE Std 802.1Q-2022, 8.6.8.4,

2) The allowable error budget between the transmission selection timing point and the on-the-wire timing point, less any error budget for the PHY (IEEE Std 802.1Q-2022, Figure 12.6), of less than or equal to 10 ns, and

3) Support the YANG features and leaves of the ieee802-dot1q sched module according to 6.4.9.3.2.

d) Frame preemption according to IEEE Std 802.1Q-2022, 5.4.1 item ad), for data rates for data rates of 10 Mb/s, 2,5 Gb/s, 5 Gb/s, and 10 Gb/s, including:

NOTE IEEE Std 802.3de-2022, 99.1, comprises 10 Mb/s.

1) Support of Interspersing Express Traffic with preemptable traffic according to IEEE Std 802.3-2022, Clause 99, including support of the Additional Ethernet Capabilities for TLV in an LLDPDU to indicate supported functions of frame preemption according to IEEE Std 802.3-2022, 79.3.7, and

2) Support of the YANG features and leaves of the ieee802-dot1q-preemption module according to 6.4.9.3.4.

5.8.3 ccB Bridge component options

A Bridge component implementation for which a claim of conformance to ccB of this document is made may support the following list of requirements.

a) Any or none of the common Bridge component options according to 5.8.1.

b) Up to eight queues according to IEEE Std 802.1Q-2022, 8.6.6.

c) More than 1 PTP Instance according to 5.5.3.

d) Enhancements for scheduled traffic according to IEEE Std 802.1Q-2022, 5.4.1 items ab) and ac) including:

1) a tick granularity of less than or equal to 10 ns according to IEEE Std 802.1Q-2022, 8.6.8.4,

2) The allowable error budget between the transmission selection timing point and the on-the-wire timing point, less any error budget for the PHY (IEEE Std 802.1Q-2022, Figure 12.6), of less than or equal to 10 ns, and

3) Support the YANG features and leaves of the ieee802-dot1q sched module according to 6.4.9.3.2.

e) Frame preemption according to IEEE Std 802.1Q-2022, 5.4.1 item ad), including:

1) Support of Interspersing Express Traffic with preemptable traffic according to IEEE Std 802.3-2022, Clause 99 including support of the Additional Ethernet Capabilities for TLV in an LLDPDU to indicate supported functions of frame preemption according to IEEE Std 802.3-2022, 79.3.7, and

2) Support of the YANG features and leaves of the ieee802-dot1q-preemption module according to 6.4.9.3.4.

5.9 End station component requirements

5.9.1 Common end station Component requirements

An end station component implementation of any conformance class for which a claim of conformance to this document is made shall support the following list of requirements.

a) The use of at least one customer VID for IA traffic engineered non-stream or IA non-stream traffic.

b) The use of an additional customer VID for IA time-aware stream traffic if that traffic type category is supported.

c) The use of an additional customer VID for IA stream traffic if that traffic type category is supported.

d) The use of an additional customer VID for IA time-aware stream traffic if redundancy for that traffic type category is supported.

e) The use of an additional customer VID for IA stream traffic if redundancy for that traffic type category is supported.

f) Participate in only a single configuration domain.

5.9.2 ccA end station component requirements

An end station component implementation for which a claim of conformance to ccA of this document is made shall support the following list of requirements.

a) Common end station component requirements according to 5.9.1.

b) At least 2 PTP Instances according to 5.5.3.

c) End station requirements for enhancements for scheduled traffic according to IEEE Std 802.1Q-2022, 5.25, for data rates of 100 Mb/s and 1 Gb/s including:

   1) a tick granularity of less than or equal to 10 ns according to IEEE Std 802.1Q-2022, 8.6.8.4,

   2) The allowable error budget between the transmission selection timing point and the on-the-wire timing point, less any error budget for the PHY (IEEE Std 802.1Q-2022, Figure 12.6), of less than or equal to 10 ns, and

   3) Support the YANG features and leaves of the ieee-dot1q-sched module according to 6.4.9.3.2.

d) End station requirements for frame preemption according to IEEE Std 802.1Q-2022, 5.26, for data rates of 100 Mb/s, and 1 Gb/s, if the IA time-aware stream traffic or the IA stream traffic type categories are supported, including:

   1) Support of Interspersing Express Traffic according to IEEE Std 802.3-2022, Clause 99, including support of the Additional Ethernet Capabilities TLV in an LLDPDU to indicate supported functions of frame preemption according to IEEE Std 802.3-2022, 79.3.7 and Table 79-8, and

   2) Support of the YANG features and leaves of the ieee802-dot1q-preemption module according to 6.4.9.3.4.

5.9.3 ccB end station component requirements

An end station component implementation for which a claim of conformance to ccB of this document is made shall support the following list of requirements.

a) Common end station component requirements according to 5.9.1.

b) At least 1 PTP Instance according to 5.5.3.

5.10 End station component options

5.10.1 Common end station component options

An end station component implementation of any conformance class for which a claim of conformance to this document is made may support the following list of requirements.

a) The operation of the credit-based shaper algorithm according to IEEE Std 802.1Q-2022, 8.6.8.2 including support of the YANG features and leaves of the <ieee-cbs> module according to 6.4.9.3.5.

b) Talker end system behaviors according to IEEE Std 802.1CB-2017, 5.6, 5.7 b) and 5.8 a) to b), as amended by IEEE Std 802.1CBdb-2021 and IEEE Std 802.1CBcv-2021 including support of the ieee802-dot1cb-stream-identification and ieee802-dot1cb-frer YANG modules according to 6.4.9.3.6.

c) Listener end system behaviors according to IEEE Std 802.1CB-2017, 5.9, 5.11 a) to b) as amended by IEEE Std 802.1CBdb-2021" and IEEE Std 802.1CBcv-2021 including support of the ieee802-dot1cb-stream-identification and ieee802-dot1cb-frer YANG modules according to 6.4.9.3.6.

5.10.2 ccA end station component options

An end station component implementation for which a claim of conformance to ccA of this document is made may support the following list of requirements.

a) Common end station options according to 5.10.1

b) More than 2 PTP Instances according to 5.5.3.

c) End station requirements for enhancements for scheduled traffic according to IEEE Std 802.1Q-2022, 5.25, for data rates of 10 Mb/s, 2,5 Gb/s, 5 Gb/s, and 10 Gb/s including:

   1) a tick granularity of less than or equal to 10 ns according to IEEE Std 802.1Q-2022, 8.6.8.4,

   2) The allowable error budget between the transmission selection timing point and the on-the-wire timing point, less any error budget for the PHY (IEEE Std 802.1Q-2022, Figure 12.6), of less than or equal to 10 ns, and

   3) Support the YANG features and leaves of the ieee802-dot1q sched module according to 6.4.9.3.2.

d) End station requirements for frame preemption according to IEEE Std 802.1Q-2022, 5.26, for data rates of 10 Mb/s, 2,5 Gb/s, 5 Gb/s, and 10 Gb/s.

   NOTE IEEE Std 802.3de-2022, 99.1, comprises 10 Mb/s.

   1) Support of Interspersing Express Traffic according to IEEE Std 802.3-2022, Clause 99, and IEEE P802.3de, 99.1, including support of the Additional Ethernet Capabilities TLV in an LLDPDU to indicate supported functions of frame preemption according to IEEE Std 802.3-2022, 79.3.7 and Table 79-8, and

   2) Support of the YANG features and leaves of the ieee802-dot1q-preemption module according to 6.4.9.3.4.

5.10.3 ccB end station component options

An end station component implementation for which a claim of conformance to ccB of this document is made may support the following list of requirements.

a) Common end station component options according to 5.10.1.

b) More than 1 PTP Instance according to 5.5.3.

c) End station requirements for enhancements for scheduled traffic according to IEEE Std 802.1Q-2022, 5.25 including:

   1) a tick granularity of less than or equal to 10 ns according to IEEE Std 802.1Q-2022, 8.6.8.4,

   2) The allowable error budget between the transmission selection timing point and the on-the-wire timing point, less any error budget for the PHY (IEEE Std 802.1Q-2022, Figure 12.6), of less than or equal to 10 ns, and

   3) Support the YANG features and leaves of the ieee802-dot1q sched module according to 6.4.9.3.2.

d) End station requirements for frame preemption according to IEEE Std 802.1Q-2022, 5.26 including:

   1) Support of Interspersing Express Traffic according to IEEE Std 802.3-2022, Clause 99, and IEEE P802.3de, 99.1, including support of the Additional Ethernet Capabilities TLV in an LLDPDU to indicate supported functions of frame preemption according to IEEE Std 802.3-2022, 79.3.7 and Table 79-8, and

   2) Support of the YANG features and leaves of the ieee802-dot1q-preemption module according to 6.4.9.3.4.

5.11 CNC requirements

CNCs for which a claim of conformance to this document is made shall support the following list of requirements.

a) TSN CNC station requirements according to IEEE Std 802.1Q-2022, 5.29.

b) NETCONF-over-TLS server and related client functionality 5.5.4.2.

c) The common YANG modules, features, and leaves according to 6.4.9.2.

d) The optional YANG modules, features, and leaves according to 6.4.9.3.

e) Be integrated in an IA-station that supports the use of at least one customer VLAN Identifier for an isolation VLAN.

5.12 CNC options

There are no optional CNC features.

5.13 CUC requirements

CUCs for which a claim of conformance to this document is made shall support the following list of requirements.

a) Be integrated in an IA-Station that supports NETCONF-over-TLS client functionality with client related security requirements according to 5.5.4.2.

b) The TSN UNI YANG module, features, and leaves according to 6.4.9.4.1.

c) The ietf-netconf-client module according to 6.4.9.4.1.

5.14 CUC options

There are no optional CUC features.

6 Required functions for an industrial network

6.1 General

Clause 6 provides requirements specific to this document and the industrial use case.

6.2 Synchronization

6.2.1 General

An IA-station can contain more than one Grandmaster PTP Instance and PTP End Instance to support:

a) hot-standby use cases, or

b) Working Clock or Global Time.

6.2.2 PTP Instance requirements

A Grandmaster PTP Instance, a PTP Relay Instance and a PTP End Instance, and the Working Clock or Global Time clocks connected to them, shall meet the following requirements under their allowed working conditions and for their lifetime:

a) The fractional frequency offset of the LocalClock relative to the nominal frequency shall be according to Table 9.

b) The range of the rate of change of fractional frequency offset of the LocalClock shall be according to Table 9.

c) During operation, the Working Clock and Global Time at Grandmaster PTP Instances and PTP End Instances shall increase monotonically, where monotonic means that for a time *y* that occurs after time *x*, the ClockTarget's timestamp of *y* is greater than or equal to the ClockTarget's timestamp of *x*.

d) The Working Clock and Global Time at a PTP End Instance can be controlled by applying a frequency change over a period of time. This also results in a phase change of the Working Clock or Global Time, as the phase change of a clock due to an applied frequency change is the product of the applied frequency change and the duration of time of the frequency change. The frequency applied can have a fine resolution to speed up or slow down the clock smoothly, and it has a total range of frequency adjustment.

e) For the Global Time at a PTP End Instance, the maximum value of frequency adjustment shall be according to Table 9.

f) For the Working Clock at a PTP End Instance, the maximum value of frequency adjustment shall be according to Table 9.

For Working Clock or Global Time, decoupled from a ClockTarget, a higher maximum rate of frequency adjustments and maximum rate of change of fractional frequency offset are allowed. As soon as it is coupled (or coupled again) a) to f) apply.

Table 9 – Required values

Topic	Value
Local Clock at non-Grandmaster PTP Instance, range of fractional frequency offset relative to the nominal frequency	± 50 ppm
Local Clock at non-Grandmaster PTP Instance, range of rate of change of fractional frequency offset with respect to the nominal frequency	± 1 ppm/s
Working Clock (acting as ClockSource) and Local Clock at Grandmaster PTP Instance, range of fractional frequency offset with respect to the nominal frequency	± 25 ppm
Working Clock (acting as ClockSource) and Local Clock at Grandmaster PTP Instance, range of rate of change of fractional frequency offset with respect to the nominal frequency (steady state, see Annex X)	± 1 ppm/s
Working Clock (acting as ClockSource) at Grandmaster PTP Instance, range of rate of change of fractional frequency offset (transient, see Annex X)	± 3 ppm/s
Working Clock at PTP End Instance, maximum value of frequency adjustment	± 250 ppm over any observation interval of 1 ms
NOTE 1 If the Grandmaster PTP Instance implementation is such that its Working Clock and Local Clock are the same or otherwise locked to the same frequency, the normative requirements on the Working Clock take priority over those on the Local Clock. NOTE 2 The Maximum value of frequency adjustment represents an upper bound that limits how much a PTP End Instance can change the frequency of its Working Clock or Global Time during a given period. However, these adjustments are incremental rather than instantaneous over the defined interval.	

6.2.3 PTP protocol requirements

Table 10 shows the required protocol times.

Table 10 – Protocol settings

Topic	Value
Nominal time between successive Announce messages (announce interval)	1 s
Nominal time between successive Pdelay_Req messages (Pdelay_Req message transmission interval)	125 ms

Topic	Value
Range of allowed time between successive Pdelay_Req messages	119 ms to 131 ms
Nominal time between successive Sync messages at the Grandmaster (Sync message transmission interval)	125 ms
Range of allowed time between successive Sync messages at the Grandmaster	119 ms to 131 ms
Time between reception of a Sync message and transmission of the subsequent Sync message (i.e. residence time) at a PTP Relay instance	Maximum: 15 ms Measured Mean: ≤ 5 ms
Maximum time between transmission of a Sync message and transmission of the related Follow_Up message	2,5 ms
Time between reception of a Pdelay_Req message and transmission of the subsequent Pdelay_Resp message (i.e. Pdelay turnaround time).	Maximum: 15 ms
ClockTimeReceiver (servo controller)	Maximum Bandwidth (Hz): 1,0 Hz Maximum Gain Peaking (dB): 2,2 dB Minimum absolute value of Roll-off: 20 dB/decade
NOTE 1 A consequence of having a single allowed value of mean sync interval is that syncLocked mode is achieved. If the master port sync interval is the same as that of the slave port, syncLocked mode is achieved. NOTE 2 The values contained in this table apply to both the Working Clock and Global Time.	

6.2.4 Error Generation Limits

Table 11 shows the required limits on error generation at a Grandmaster PTP instance. A limit on error generation for a Grandmaster PTP Instance is a limit on the amount of error it generates in the output Sync message compared to its Working Clock (acting as ClockSource) and Local Clock.

Table 11 – Error generation limits for Grandmaster PTP Instance

Topic	Value
(preciseOriginTimestamp + correctionField) in Sync message minus Working Clock at Grandmaster when Sync message is transmitted	Allowable range of the measured mean: - 10 ns to + 10 ns Range around the measured mean within which 90% of measurements fall: ± 7 ns Range around the measured mean within which 100% of measurements fall: ± 10 ns
Rate Ratio between Working Clock at Grandmaster and Local Clock when Sync message is transmitted minus rateRatio field in Sync message	Mean 0 ppm ± 0,1 ppm Standard deviation ≤ 0,1 ppm
syncEgressTimestamp in Drift_Tracking TLV minus Local Clock when Sync message is transmitted	Range around the measured mean within which 90% of measurements fall: ± 7 ns Range around the measured mean within which 100% of measurements fall: ± 10 ns

Table 12 shows the required limits on error generation at a PTP Relay instance. A limit on error generation for a PTP Relay Instance is a limit on the amount of error it adds to the output Sync message compared the to input Sync message. These requirements are written for the case when errors due to change of fractional frequency offset of its Local Clock with respect to the nominal frequency and errors in the input Sync message are negligible with respect to the specified error generation limits. See D.3.5.

Table 12 – Error generation limits for PTP Relay Instance

Topic	Value
(preciseOriginTimestamp + correctionField) in the Sync message transmitted by PTP Relay Instance minus Working Clock at Grandmaster when the Sync message is transmitted, while… • Working Clock (acting as ClockSource) at Grandmaster is stable. • Local Clock at upstream PTP Instance is stable • meanLinkDelay between upstream PTP Instance and PTP Relay Instance is negligible	Allowable range of the measured mean: - 2 ns to + 2 ns Range around the measured mean within which 90% of measurements fall: ± 10 ns Range around the measured mean within which 100% of measurements fall: ± 20 ns
rateRatio field in the Sync message transmitted by PTP Relay Instance minus the Rate Ratio from the PTP Relay Instance's Local Clock to the WorkingClock at the Grandmaster, while… • Working Clock (acting as ClockSource) at Grandmaster is stable. • Local Clock at upstream PTP Instance is stable.	Allowable range of the measured mean: - 0,1 ppm to + 0,1 ppm Allowable measured standard deviation around the measured mean: 0,02 ppm
rateRatio field in the Sync message transmitted by PTP Relay Instance minus the Rate Ratio from the PTP Relay Instance's Local Clock to the WorkingClock at the Grandmaster, while… • WorkingClock (acting as ClockSource) at Grandmaster PTP Instance, fractional frequency offset with respect to the nominal frequency is increasing at 1 ppm/s • Local Clock at upstream PTP Instance is stable.	Allowable range of the measured mean: - 0,1 ppm to + 0,1 ppm Allowable measured standard deviation around the measured mean: 0,08 ppm
rateRatio field in the Sync message transmitted by PTP Relay Instance minus the Rate Ratio from the PTP Relay Instance's Local Clock to the WorkingClock at the Grandmaster, while… • WorkingClock (acting as ClockSource) at Grandmaster PTP Instance, fractional frequency offset with respect to the nominal frequency is increasing at 1 ppm/s • Local Clock at upstream PTP Instance, fractional frequency offset with respect to the nominal frequency is increasing at 1 ppm/s	Allowable range of the measured mean: - 0,1 ppm to + 0,1 ppm Allowable measured standard deviation around the measured mean: 0,08 ppm
rateRatioDrift field in the Sync message transmitted by PTP Relay Instance minus the Rate Ratio Drift from the PTP Relay Instance's Local Clock to the WorkingClock at the Grandmaster, while… • WorkingClock (acting as ClockSource) at Grandmaster is stable. • Local Clock at upstream PTP Instance is stable.	Allowable range of the measured mean: - 0,1 ppm/s to + 0,1 ppm/s Allowable measured standard deviation around the measured mean: 0,02 ppm/s
rateRatioDrift field in the Sync message transmitted by PTP Relay Instance minus the Rate Ratio Drift from the PTP Relay Instance's Local Clock to the WorkingClock at the Grandmaster, while… • WorkingClock (acting as ClockSource) at Grandmaster PTP Instance, fractional frequency offset with respect to the nominal frequency is increasing at 1 ppm/s • Local Clock at upstream PTP Instance is stable.	Allowable range of the measured mean: - 0,1 ppm/s to + 0,1 ppm/s Allowable measured standard deviation around the measured mean: 0,08 ppm/s

Topic	Value
rateRatioDrift field in the Sync message transmitted by PTP Relay Instance minus the Rate Ratio Drift from the PTP Relay Instance's Local Clock to the WorkingClock at the Grandmaster, while… • WorkingClock (acting as ClockSource) at Grandmaster PTP Instance, fractional frequency offset with respect to the nominal frequency is increasing at 1 ppm/s • Local Clock at upstream PTP Instance, fractional frequency offset with respect to the nominal frequency is increasing at 1 ppm/s	Allowable range of the measured mean: - 0,1 ppm/s to + 0,1 ppm/s Allowable measured standard deviation around the measured mean: 0,08 ppm/s
syncEgressTimestamp in Drift_Tracking TLV minus Local Clock when Sync message is transmitted	Range around the measured mean within which 90% of measurements fall: ± 7 ns Maximum difference of any measurement from the measured mean: ± 10 ns

Table 13 shows the required limits on error generation at a PTP End Instance. A limit on error generation for a PTP End Instance is a limit on the amount of error it adds to its Working Clock (acting as ClockTarget) compared to the input Sync message. These requirements are written for the case when errors due to change of fractional frequency offset of its Local Clock with respect to the nominal frequency and errors in the input Sync message are negligible with respect to the specified error generation limits. See D.3.6.

Table 13 – Error generation limits for PTP End Instance

Topic	Value
Working Clock (acting as ClockTarget) at PTP End Instance minus Working Clock (acting as Clock Source) at Grandmaster, while… • WorkingClock (acting as ClockSource) at Grandmaster is stable. • Local Clock at upstream PTP Instance is stable. • meanLinkDelay between upstream PTP Instance and PTP Relay Instance is negligible	Allowable range of cTE: - 10 ns to + 10 ns Allowable range of dTE: - 15 ns to + 15 ns
Working Clock (acting as ClockTarget) at PTP End Instance minus Working Clock (acting as Clock Source) at Grandmaster, while… • WorkingClock (acting as ClockSource) at Grandmaster PTP Instance, fractional frequency offset with respect to the nominal frequency is increasing at 1 ppm/s • Local Clock at upstream PTP Instance is stable. • meanLinkDelay between upstream PTP Instance and PTP Relay Instance is negligible	Allowable range of cTE: - 10 ns to + 10 ns Allowable range of dTE: - 17 ns to + 17 ns
Working Clock (acting as ClockTarget) at PTP End Instance minus Working Clock (acting as Clock Source) at Grandmaster, while… • WorkingClock (acting as ClockSource) at Grandmaster PTP Instance, fractional frequency offset with respect to the nominal frequency is increasing at 1 ppm/s • Local Clock at upstream PTP Instance, fractional frequency offset with respect to the nominal frequency is increasing at 1 ppm/s • meanLinkDelay between upstream PTP Instance and PTP Relay Instance is negligible	Allowable range of cTE: - 10 ns to + 10 ns Allowable range of dTE: - 17 ns to + 17 ns

6.2.5 Clock states

Industrial automation systems monitor the synchronization status of each PTP Instance to determine the viability of operations. This status is obtained from the isSynced global variable defined in IEEE P802.1ASdm, 18.4.1.

PtpInstanceSyncStatus state machine in IEEE P802.1ASdm shall be supported independent whether hot standby is supported. The interface primitives of 9.3.3, 9.4.3, 9.5.3, 9.6.2 of IEEE P802.1ASdm shall be supported.

6.2.6 Grandmaster PTP Instance requirements

The behavior of a ClockSource coupled to a ClockMaster of a Grandmaster PTP Instance allows a controlled/disciplined ClockTarget to stay in the ranges stated in 6.2.2 and 6.2.3. This includes the cases in which the ClockSource is controlled (effect of rate and offset compensation) by another ClockSource, for example, a GPS time source.

NOTE A Grandmaster can lose and regain its source of time, leading to large discontinuities in the value of grandmaster time. In such situations, the application can decouple from the grandmaster (see Figure 12). After the grandmaster has regained a source of time, the decision to re-couple to the grandmaster is an application decision.

Figure 12 shows an example of additional factors influencing the maximum rate of change of fractional frequency offset.

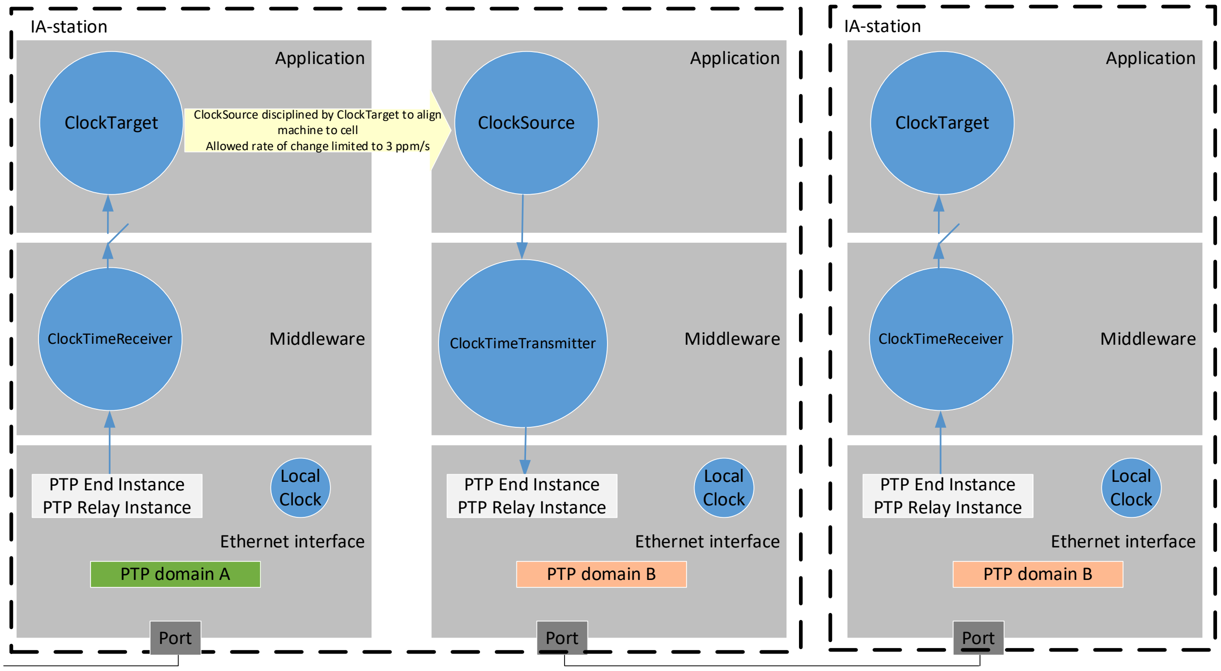


Figure 12 – Externally controlled ClockSource of a Grandmaster

Coupled machines, for example newspaper printing machines, use multiple PTP domains to allow different combinations over time without influencing the main production path. This is done by application coupling between PTP domain A and B as shown in the left-hand IA-station in Figure 12. In this IA-station, the alignment of the ClockSource of PTP domain B to the ClockTarget of PTP domain A is accomplished by some means not addressed by this document.

6.2.7 Application framework

Any step change in the time of a ClockSource or ClockTarget whose absolute value exceeds a user-defined threshold (for example 1 µs) leads to action being taken by the application or by a higher-layer entity.

If the change is in Global Time, it is desirable that all consumers of that time be made aware of this change (i.e., a jump in Global Time from the value A to the value B), so that the actual time interval between the time corresponding to A and the time corresponding to B can be evaluated.

In the case of Working Clock, a time change that exceeds the user-defined threshold (for example 1 μs) is avoided to protect assets and prevent damage. Thus, the ClockSource or ClockTarget can be decoupled (see Figure 14) from the PTP-maintained clock when such a time change occurs.

In Figure 14, two ClockTargets are traceable to a reliable source of time, which should be synchronized to Global Time or Working Clock.

The status of a ClockSource, ClockTarget, ClockTimeTransmitter or ClockTimeReceiver is given by the state of the clock (see 6.2.5) as shown in Figure 13. When timestamps are provided to the application, the current ClockSource or ClockTarget state can also be provided to the application.

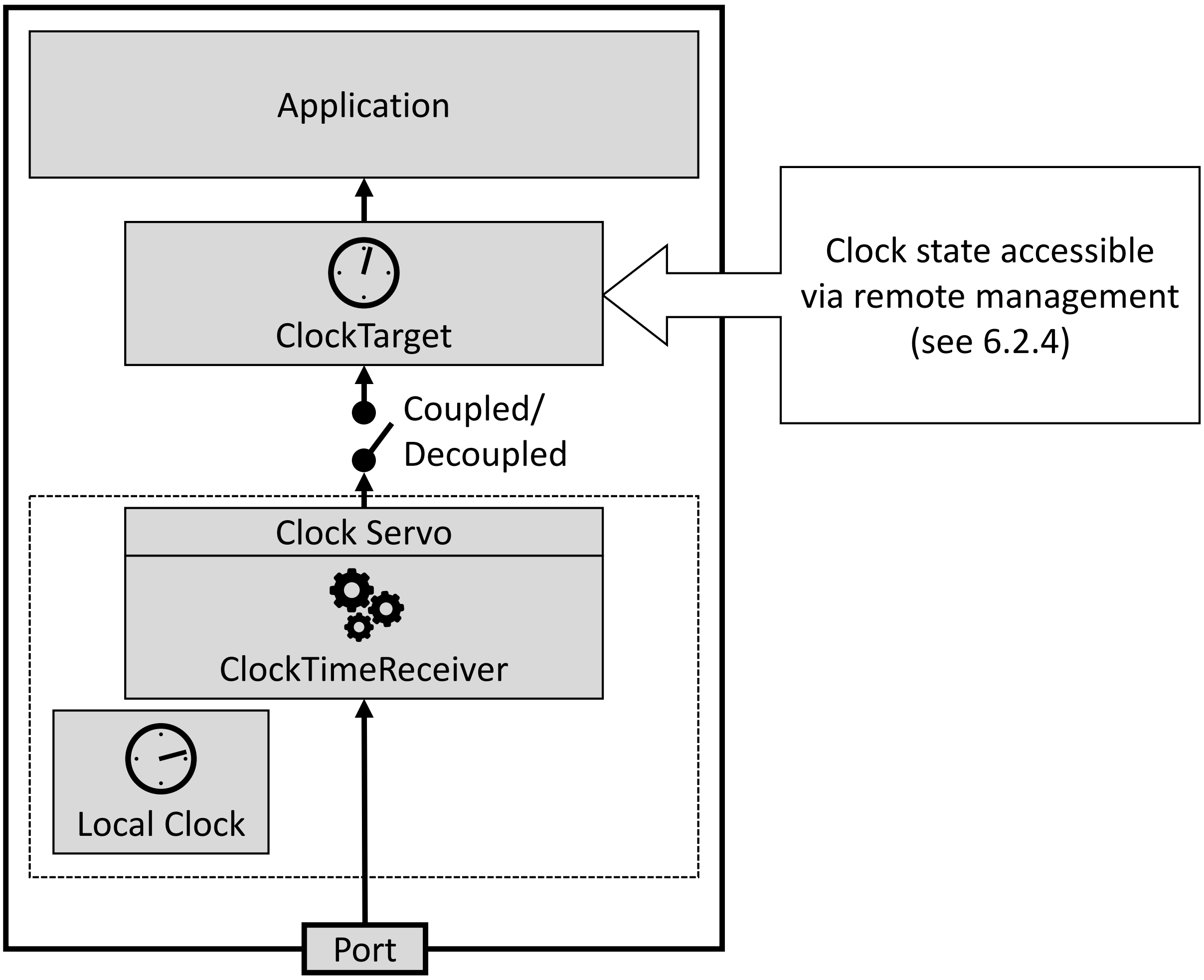


Figure 13 – Clock model

6.2.8 Working Clock domain framework

The gPTP domainNumber of a Working Clock domain is assigned by the CNC. In industrial applications, when stepsRemoved, as specified in IEEE Std 802.1AS-2020, between the Grandmaster PTP Instance and any PTP End Instance, as determined by the Best Master Clock Algorithm, is less than or equal to 64, max|TE_R| of the synchronized time of any ClockTarget, relative to the Grandmaster ClockSource, is less than or equal to 1 μs (see error budget A in Figure 16). Thus it is incumbent upon any PTP Instance to ensure that the requirements specified in 5.5.3, 6.2.2, and 6.2.3 are met.

6.2.9 Global Time domain framework

The gPTP domainNumber of a Global Time domain is assigned by the CNC. In industrial applications, when stepsRemoved, as specified in IEEE Std 802.1AS-2020, between the Grandmaster PTP Instance and any PTP End Instance, as determined by the Best Master Clock Algorithm, is less than or equal to 100, max$|TE_R|$ of the synchronized time of any ClockTarget, relative to the Grandmaster ClockSource, is less than or equal to 100 µs (see error budget A in Figure 16). Thus it is incumbent upon any PTP Instance to ensure that the requirements specified in 5.5.3, 6.2.2, and 6.2.3 are met.

6.2.10 IA-station model for clocks

Industrial automation applications, as described in 4.1, require synchronized time that is traceable to a known source (i.e., Global Time) and a source of time synchronized to the Working Clock. Figure 14 and Figure 15 show examples of the IA-station internal model for clocks with the two PTP Instances. It is possible for the ClockSource or ClockTarget to start decoupled or become decoupled from the ClockTimeReceiver or ClockTimeTransmitter of a PTP Instance; the ClockSource or ClockTarget runs independently of the availability of the network or a Grandmaster. For example, if the PTP Instance enters a clock state other than SYNCED, the application might choose to decouple its clock from the PTP Instance and continue to run on its internal clock. If the PTP Instance reenters SYNCED, the application can choose to again synchronize to the PTP Instance.

Figure 14 shows the IA-station internal model for clocks, with the two PTP instances used as ClockTimeReceiver/ClockTarget.

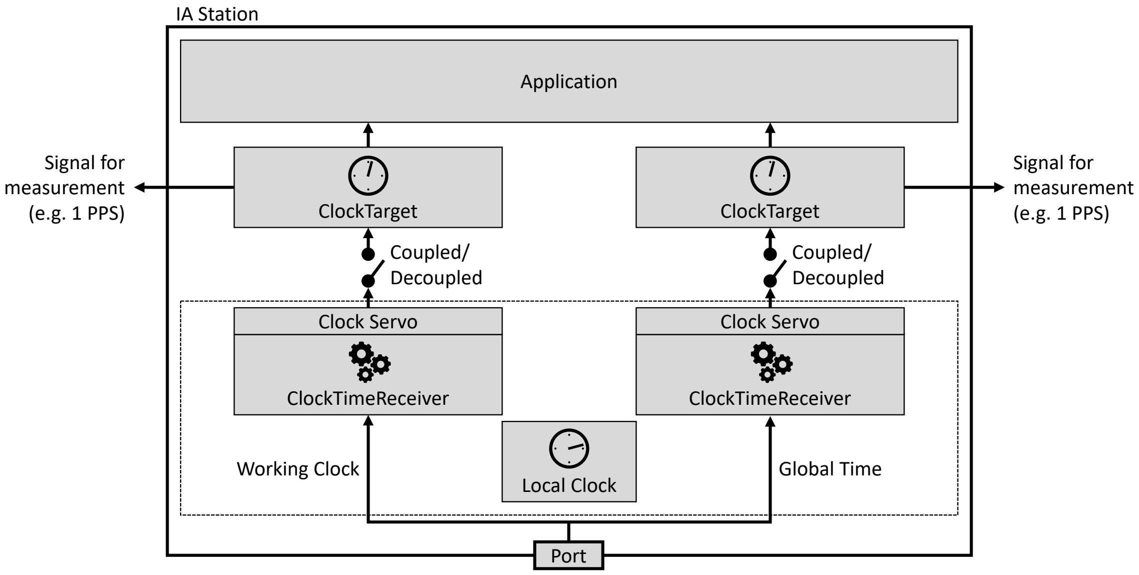


Figure 14 – Example clock usage principles for PTP End Instances

Figure 15 shows the IA-station internal model for clocks, with the two PTP instances used as Grandmaster.

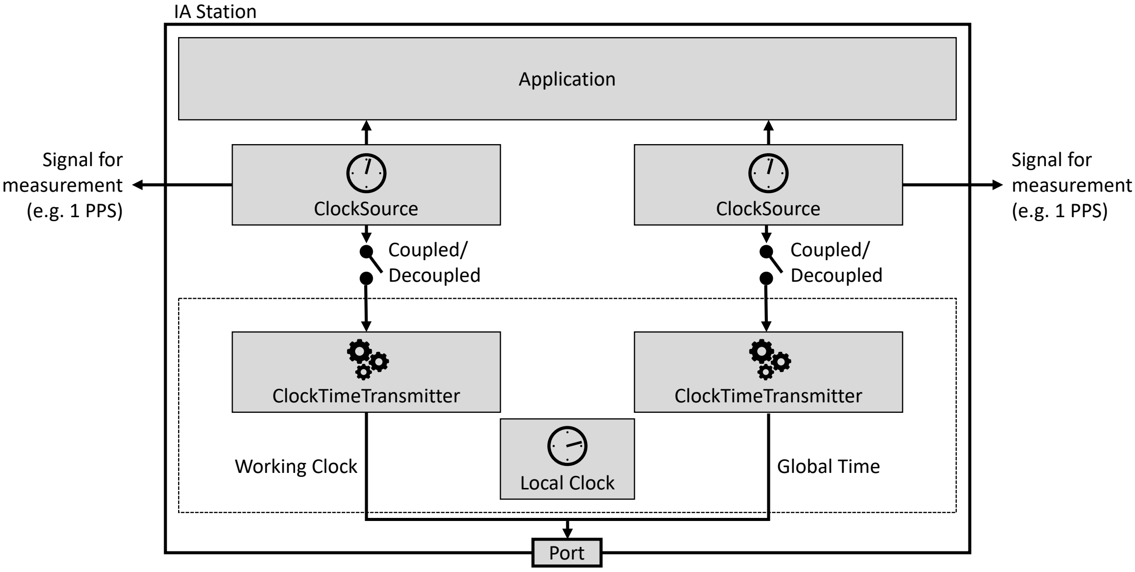


Figure 15 – Example clock usage principles for Grandmaster PTP Instances

6.2.11 Clock usage for the Ethernet interface

6.2.11.1 Time-aware offset control

Time-aware offset control (see 4.4) needs an assigned source of time and a definition when to start or to stop, which are dependent on the clock state.

The clock used is the ClockTarget or, in the case of a Grandmaster PTP Instance, the ClockSource.

IA time-aware streams are only transmitted while the chosen ClockSource or ClockTarget is in clock state SYNCED (see 6.2.5).

Thus, changes of the clock state directly influence the transmission of frames.

6.2.11.2 Gating cycle

To control the gating cycle, the gate control list needs an assigned source of time. Enabling and disabling the gate control list is dependent on the clock state.

The clock used is the ClockTarget or, in the case of a Grandmaster PTP Instance, the ClockSource.

The gating cycle is run using the chosen ClockSource or ClockTarget in all clock states (see 6.2.5).

6.2.12 Error model

Synchronization is transported over the entire path, from the Grandmaster PTP Instance to the PTP End Instance, through the intermediate PTP Relay Instances. All time errors, cTE and dTE, are accumulated during this process.

Time error can arise in the following processes:

a) the transporting of time in PTP Instances and via PTP Links that connect PTP Instances,

b) the providing of time to the Grandmaster PTP Instance, from the ClockSource entity via the ClockTimeTransmitter entity, and

c) the providing of time to a ClockTarget entity (end application) via the ClockTimeReceiver entity.

NOTE Item a) includes time error introduced in a PTP End Instance between the slave port and the ClockTimeReceiver entity, and between the ClockTimeTransmitter entity and a master port.

An output synchronization signal (for example, 1 pulse per second (PPS)) synchronized to the Working Clock as shown in Figure 14 and Figure 15, at any PTP Instance, is used to measure the time error between the ClockSource of the Grandmaster and the ClockTarget of a PTP Instance that is not the Grandmaster. The additional error introduced by implementation of the output synchronization signal is in the range of -10 ns to +10 ns. Figure 16 shows the error budget principle used. These budgets do not include any deviation from the PTP timescale. Representative budgets are provided in Annex D.

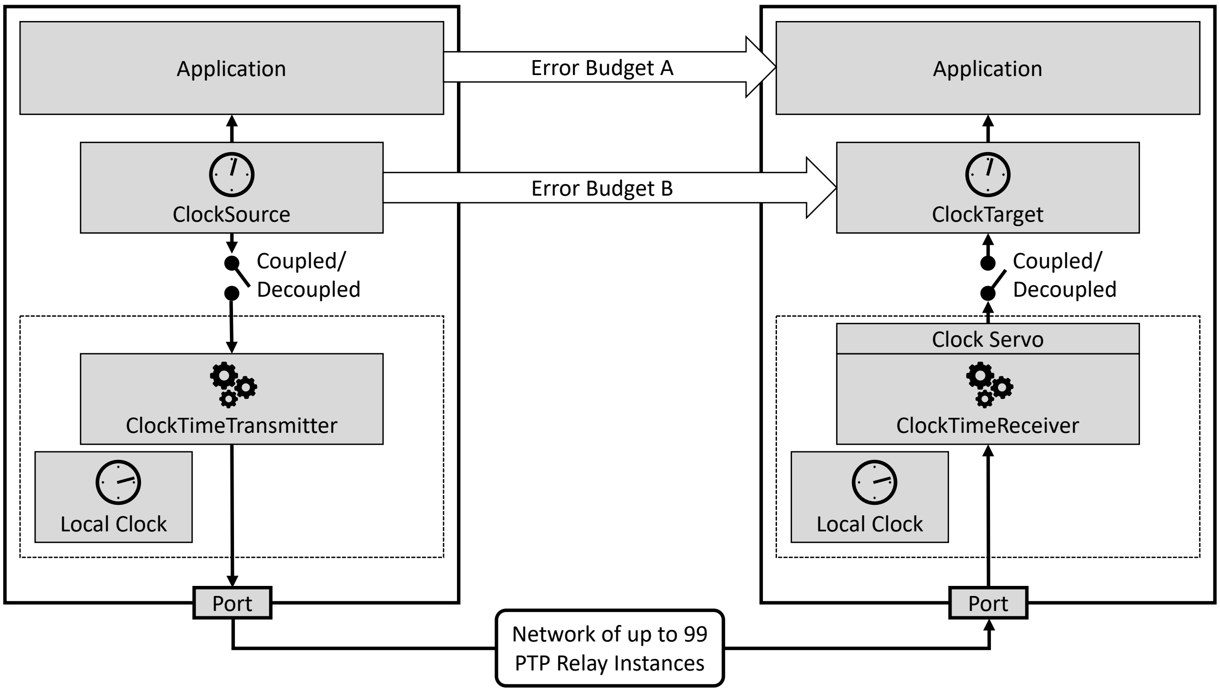


Figure 16 – Error budget scheme

Table 14 shows example values for the splitting of the available error budgets (see Figure 16).

Table 14 – Error budget

Domain	Error budget A	Error budget B
Working Clock	1 µs	900 ns
Global Time	100 µs	99,9 µs

Global time is often used for tracking events in industrial applications (i.e., sequence of events). Any usage of Global time for time stamping of application events is allowed an error budget of 1 ms.

6.2.13 gPTP domains and PTP Instances

Any valid gPTP domain number as specified in IEEE 802.1AS-2020 can be used. The IEEE Std 1588-2019 attribute descriptionDS.userDescription shall be used according to Table 1 to support the translation of PTP Instances and middleware as described in 4.6.2. One gPTP domain can be used for both Working Clock and Global Time. If only one gPTP domain is used, then the requirements for the Working Clock apply (see 6.2.8).

Table 15 – descriptionDS.userDescription of gPTP Domains

gPTP Domain	descriptionDS.userDescription
Working Clock (no hot standby configured)	“60802-WorkingClock”
Primary Working Clock (with configured hot standby)	“60802-Primary-WorkingClock”
Secondary Working Clock (with configured hot standby)	“60802-Secondary-WorkingClock”
Global Time (no hot standby configured)	“60802-GlobalTime”
Primary Global Time (with configured hot standby)	“60802-Primary-GlobalTime”
Secondary Global Time (with configured hot standby)	“60802-Secondary-GlobalTime”
GlobalTime and WorkingClock (no hot standby configured)	“60802-GlobalTime-WorkingClock”
Primary GlobalTime and WorkingClock (with configured hot standby)	“60802-Primary-GlobalTime-WorkingClock”
Secondary GlobalTime and WorkingClock (with hot standby configured)	“60802-Secondary-GlobalTime-WorkingClock”

The descriptionDS.userDescription attribute is represented in the ieee1588-ptp YANG module by the `user-description` leaf in the `description-ds` container of a PTP Instance.

The linking between a gPTP domain and the IETF interfaces is provided by the `underlying-interface` leaves in the `port` list of the PTP Instance that implements the gPTP domain.

6.2.14 Split and combine cases for a PTP domain

Modular machines or production cells allow the splitting and combining of machines if this is required by the production process. To minimize the production disruption, the second machine is connected to the first machine during operation.

Combining the machines does not disturb the first machine, which keeps producing goods. Thus, the Grandmaster of the first machine is the Grandmaster of the combined PTP domain.

Splitting the machines does not disturb the first machine, which keeps producing goods. The Grandmaster of the second machine starts after splitting to allow standalone production for the second machine.

Figure 17 and Figure 18 shows the split and combine use case. The following steps are intended to avoid jumps in synchronization that could potentially disrupt operation.

- Splitting:
  - Grandmaster of machine 2 controls machine 2 and Grandmaster of machine 1 controls machine 1.
  - Machine 1 and machine 2 are separated. Machine 1 continues production. The Grandmaster located in Machine 1 provides synchronization.
  - Machine 2 can be moved to a different location or just used stand alone to produce some goods. The Grandmaster in machine 2 provides synchronization for machine 2.
- Combining:
  - Grandmaster of machine 2 follows the Grandmaster from machine 1.
  - Machine 2 is done with its production process and is combined with machine 1 again. Machine 1 can still be producing while machine 2 is combined with machine 1 again.
  - Machine 1 is undisturbed and machine 2 is starting to use the Grandmaster from machine 1.

  NOTE When machine 2 is starting to use the Grandmaster from machine 1, the time difference between the PTP instances in machine 2 and the Grandmaster from machine

1 can be > 1µs. Therefore, splitting and combining while using hot standby can result in errors > 1µs.

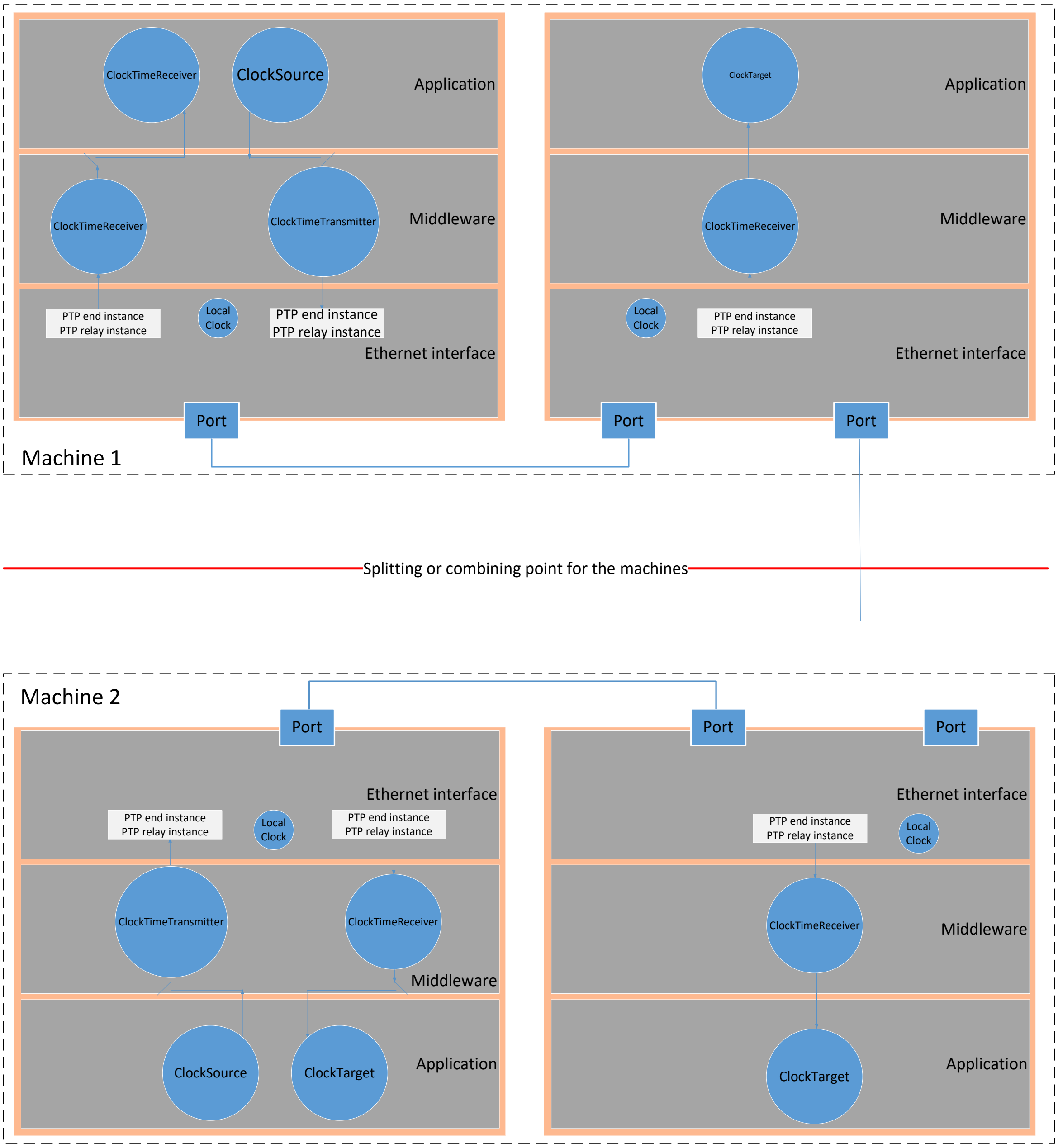


Figure 17 – Split and combine using BMCA

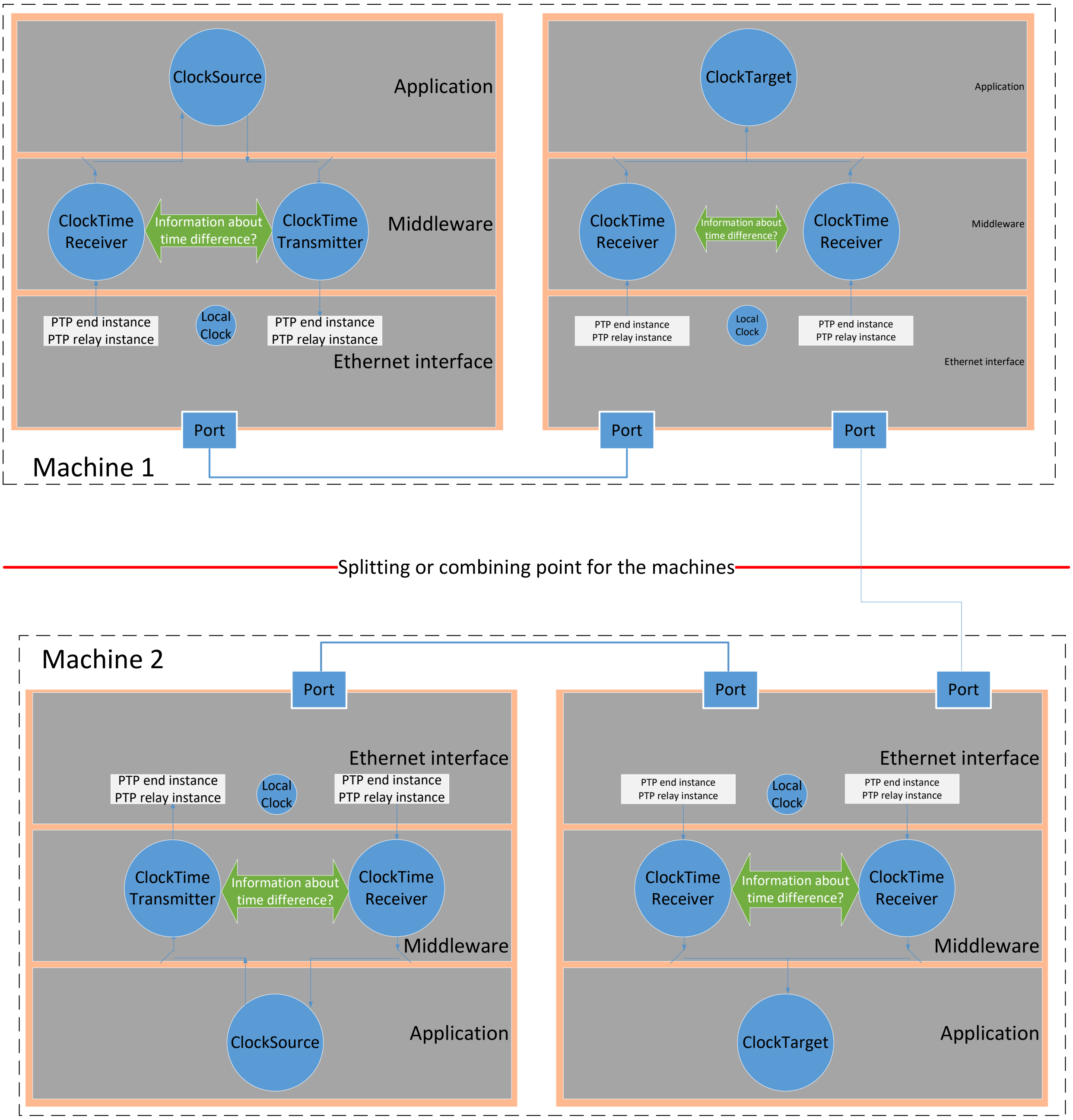


Figure 18 – Split and combine using hot standby

6.3 Security model

6.3.1 General

Subclause 6.3 specifies the security model starting with NETCONF/YANG. It describes the security functionality, the security objects in factory default state, the imprinting of Configuration Domain-specific security objects and the secure configuration based on Configuration Domain-specific security objects.

NOTE Securing the transport of time synchronization is not covered in this document. Techniques for securing time synchronization exist; however, such techniques can have performance ramifications.

6.3.2 Security functionality

6.3.2.1 Message exchange protection

6.3.2.1.1 General

Network configuration with NETCONF/YANG shall be protected by NETCONF-over-TLS according to IETF RFC 7589 and IETF draft-ietf-netconf-over-tls13. NETCONF-over-SSH according to IETF RFC 6242 shall not be used. The to-be-configured IA-stations shall act in the NETCONF server role.

NOTE This document selects TLS as a secure transport for NETCONF since TLS is the better match for the case of configuration clients that rely upon unattended or automated operation. This case is dominant in industrial automation. To avoid complexity, this document deselects SSH as a secure transport for NETCONF.

6.3.2.1.2 TLS profile

TLS protocol version 1.2 according to IETF RFC 5246, 6.2.3.3, 7.4.7.2 and 8.1.2 shall be used with mutual authentication according to the following list of requirements.

a) Mutual authentication in conjunction with the IDevID and LDevID-NETCONF credentials according to 6.3.4 and 6.3.5.

b) The cipher suite TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 shall be supported. The cipher suites TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 and TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 may be supported.

c) IETF RFC 7589 implicitly mandates the cipher suite TLS_RSA_WITH_AES_128_CBC_SHA by referring to IETF RFC 5246. This cipher suite shall not be supported because it requires excessive asymmetric key lengths, it is not an Authenticated Encryption with Associated Data (AEAD) scheme, and it does not provide perfect forward secrecy.

d) IETF draft-ietf-netconf-over-tls13 mandates the cipher suite TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256. This cipher suite shall not be supported because it requires excessive asymmetric key lengths.

e) Signature algorithm ECDSA with SHA-256 and Curve P-256 according to NIST FIPS 186-5 Digital Signature Standard (DSS) shall be supported.

f) Signature algorithms ECDSA with SHA-512 and Curve P-521 according to NIST FIPS 186-5, Ed25519 according to IETF RFC 8032, 5.1, and Ed448 according to IETF RFC 8032, 5.2, may be supported.

TLS protocol version 1.3 according to IETF RFC 8446, may be used with mutual authentication for NETCONF/YANG as follows:

g) The cipher suites TLS_AES_128_GCM_SHA256, TLS_AES_256_GCM_SHA384 and TLS_CHACHA20_POLY1305_SHA256 may be supported.

h) The signature schemes ecdsa_secp256r1_sha256, ecdsa_secp521r1_sha512, ed25519 and ed448 may be supported.

Independent from the TLS version, The TLS Certificate message from the TLS client and server shall contain the self-signed root certificate. This approach allows to simplify/flatten the PKI hierarchy on base of the current TLS client certificate to NETCONF username mapping algorithm in IETF RFC 7589. Implementations shall support TLS Certificate message with at least 2 certificate objects.

6.3.2.1.3 Certificate-to-name mapping

The certificate-to-name mapping procedure in IETF RFC 7589 shall be as follows.

NOTE IETF RFC 7589, Clause 7, specifies that NETCONF servers map client certificates to “NETCONF usernames” and specifies a concrete mapping procedure for this purpose. This mapping is represented by the YANG module ietf-x509-cert-to-name.

The list of mapping entries has a single element containing:

- fingerprint: the fingerprint of the trust anchor for the Configuration Domain, and
- map_type: ext-60802-roles.

The mapping entry provides the assigned role names for the NETCONF client that are extracted from the id-60802-pe-roles certificate extension of the client's TLS-authenticated END ENTITY certificate.

6.3.2.1.4 Role extension

The id-60802-pe-roles extension in LDevID-NETCONF END ENTITY certificates shall be constructed as follows:

a) Extension field extnID

The extnID shall provide the following OBJECT IDENTIFIER to identify the id-60802-pe-roles extension:

```
id-60802 OBJECT IDENTIFIER ::= { <60802-specific OID> }

id-60802-pe OBJECT IDENTIFIER ::= { id-60802  1 }

id-60802-pe-roles OBJECT IDENTIFIER ::= { id-60802-pe  1 }
```

Editor's note: A 60802-specific OID cannot be provided until SA Ballot.

b) Extension field critical

The id-60802-pe-roles extension shall not be marked as critical (critical:= FALSE).

c) Extension field extnValue

```
60802RoleNamesSyntax ::= SEQUENCE OF 60802RoleName

60802RoleName   ::= ENUMERATED {
                              TruststoreAdminRole (0),
                              KeystoreAdminRole (1),
                              UserMappingAdminRole (2),
                              ConfiguratorRole (3),
                              StreamConfiguratorRole (4),
                              SubscriberRole (5)}
```

NOTE The extnValue provides an OCTET STRING that contains the DER-encoded 60802RoleNamesSyntax value. The output of the certificate-to-name mapping is the list of assigned role names representing the input for checking access permissions with NACM.

6.3.2.2 Resource access authorization

Access control to NETCONF/YANG resources shall be protected by NACM according to IETF RFC 8341.

NACM specifies a YANG data model (ietf-netconf-acm) for expressing rules to control access to NETCONF/YANG resources. This document profiles NACM to deliver role-based access control.

NOTE 1 NACM does not natively deliver role-based access control but can be geared by profiling.

This role-based model for security resources should be applied according to the following list of requirements.

- The global switch enable-nacm is set to true.
- The set of NETCONF/YANG resources of an IA-station is partitioned according to the YANG modules specified in 6.4.9 with a permission-to-role assignment as listed below. An access

operation is allowed through the keyword “permitted” and not allowed through the keyword “denied”.

NOTE 2 NACM recognizes following “access-operations": create, read, update, delete, exec and uses the term write access for the access operations "create", "delete", and "update". This document uses the terms read, write and exec access.

- All authenticated entities (default rules): All YANG modules: read access permitted, write access denied, exec-access denied.

NOTE 3 The default rules apply for YANG modules that are listed in 6.4.9 but are not listed in the rules of the individual roles.

- Rules for StreamConfiguratorRole: YANG module ieee802-dot1q-tsn-config: write and execute operations permitted.
- Rules for SubscriberRole:
  - YANG module ietf-subscribed-notifications: write and execute operations permitted, and
  - YANG module ietf-yang-push: write and execute operations permitted.
- Rules for ConfiguratorRole: All YANG modules except those listed below, write and execute operations permitted:
  - YANG modules for security configuration, i.e., ietf-truststore, ietf-keystore, path to cert-to-name nodes of ietf-netconf-server,
  - YANG modules for stream configuration, i.e., ieee802-dot1q-tsn-config, and
  - YANG modules for subscription configuration, i.e., ietf-subscribed-notifications, ietf-yang-push.
- Rules for TruststoreAdminRole:
  - YANG module ietf-truststore, path to certificate node of IDevID trust anchor: write and execute operations denied, and
  - YANG module ietf-truststore (besides path to certificate node of IDevID trust anchor): write and execute operations permitted.
- Rules for KeystoreAdminRole:
  - YANG module ietf-keystore, path to asymmetric-key node of IDevID credential: write and execute operations denied, and
  - YANG module ietf-keystore (besides path to asymmetric-key node of IDevID credential): write and execute operations permitted.
- Rules for UserMappingAdminRole:
  - YANG module ietf-netconf-server (besides path to cert-to-name nodes): write and execute operations denied, and
  - YANG module ietf-netconf-server, path to cert-to-name nodes: write and execute operations permitted.

In addition, the following access control should be applied for NETCONF protocol operations:

- <lock>, <unlock>: permitted for any role defined in this document,
- <partial-lock>, <partial-unlock>: denied (not used in this document),
- <get> and <get-config>: mapped to a "read" access operation to the target datastore,
- <edit-config>: permitted for any role defined in this document,
- <copy-config>: permitted for ConfiguratorRole,
- <delete-config>: denied (not used in this document),
- <commit>: permitted for any role defined in this document,
- <discard-changes>: permitted for any role defined in this document,
- <close-session>: permitted for any role defined in this document, and

- <kill-session>: denied (not used in in this document).

This document does not specify the assignment of role names to actual system entities. This is a duty of system owners or operators.

6.3.3 IDevID Profile

6.3.3.1 General

IA-stations shall possess IDevID credentials according to 6.3.3. CNCs shall contain trust anchors for validating IDevID credentials.

6.3.3.2 Object Contents

6.3.3.2.1 General

The IDevID credential contents shall comply to 6.3.3.2.2 and IEEE Std 802.1AR-2018, Clause 6.

6.3.3.2.2 IA-station Identity

Any IDevID EE certificate of an IA-station shall take one of the following forms:

- raw form: the IDevID EE certificate complies to IEEE Std 802.1AR-2018, Clause 8, and
- extended form: the IDevID EE certificate complies to requirements provided in 6.3.3.2.2 and IEEE Std 802.1AR-2018, Clause 8

The extended form of an IDevID EE certificate shall be constructed as follows:

- the verifiable device identity shall appear as a URN in a GeneralName of type uniformResourceIdentifier in the subjectAltName extension,
- the URN value shall be constructed according to IETF RFC 8141 and as follows:
  - namespace identifier: ieee (see IETF RFC 8069), and
  - namespace-specific string: iec-ieee-60802#verifiable-device-identity,
  - q-component (see IETF RFC 8141, 2.3.2) to parameterize the named resource: an ampersand-separated list of keyword=value tuples with following keywords and values. These tuples can appear in any order inside the q-component,
    - The keywords: description, hardware-rev, serial-num, mfg-name, model-name, and
    - Their corresponding values from the single ‘component’ list entry in the ietf-hardware YANG module that represents the management entity of the IA-station respectively from its pre-material form in percent-encoding (see IETF RFC 3986).

NOTE 1 These are the items with the YANG property config-false from the ‘component’ list entry that represents the management entity of the IA-station. The config-false items firmware-rev and software-rev are excluded to avoid IDevID credential updates in case of FW or SW updates.

NOTE 2 An object looks like urn:ieee:iec-ieee-60802#verifiable-device-identity?=mfg-name=<mfg-name>&model-name=<model-name>&hardware-rev=<hardware-rev>&serial-num=<serial-num>&description=<description>.

NOTE 3 One IDevID EE certificate can have one subjectAltName extension which can have one or more GeneralName entries. In particular: there can be one or more GeneralName entries of type uniformResourceIdentifier. This allows other organizations e.g., middleware and application consortia or individual manufacturers to also represent their perception of verifiable device identity in addition to the perception of this document.

6.3.3.2.3 Signature Suites

An IDevID shall utilize the signature suite: ECDSA P-256/SHA-256 according to IEEE Std 802.1AR-2018, 9.2.

An IDevID may utilize the following signature suites:

- ECDSA P-521/SHA-512 according to NIST FIPS 186-5/180-4 and using the algorithm identifiers according to IETF RFC 5480,

- EdDSA instance Ed25519 according to IETF RFC 8032 using Curve25519 according to IETF RFC 7748 and using the algorithm identifiers according to IETF RFC 8410, and
- EdDSA instance Ed448 according to IETF RFC 8032 using Curve448 according to IETF RFC 7748 and using the algorithm identifiers according to IETF RFC 8410.

6.3.3.3 Information Model

6.3.3.3.1 General

The information model for IDevID credentials and trust anchors shall comply to YANG and NMDA, in particular the YANG modules ietf-keystore and ietf-truststore, as well as subsequent subclauses of 6.3.3.3.

6.3.3.3.2 Entries

IDevID credentials shall be provided in form of built-in keys of an IA-station by its manufacturer. In YANG, they are modeled as config-false nodes and are represented in the ‘keystore’ container that is instantiated by the YANG module ietf-keystore. The private key shall use the private-key-type choice hidden-private-key i.e., the IDevID private key is not presented in NETCONF/YANG. The details of storing and protecting IDevID private keys as well as using them for signing purposes are implementation specific.

Trust anchors for IDevID credentials are CNC user-configured data objects: these objects shall be available as applied configuration (IETF RFC 8342) upon CNCs. In YANG, they are modeled as config-true nodes and are represented in the ‘truststore’ container that is instantiated by the YANG module ietf-truststore.

NOTE IA-station built-in trust anchors for use cases such as FW/SW update are not addressed in this document.

6.3.3.3.3 Entry Manifoldness

An IA-station shall possess one IDevID credential with a certification path plus trust anchor information issued under the required signature suite according to 6.3.3.2.3 as part of its factory default state.

If an IA-station supports an optional signature suite according to 6.3.3.2.3, it shall possess in addition one IDevID credential with a certification path plus trust anchor information issued under the optional signature suite as part of its factory default state.

An IA-station can have additional IDevID credential(s) with a certification path plus trust anchor information issued under a combination of any required or any supported optional DevID signature suites.

If an IA-station possesses multiple IDevID credentials, then they shall be issued by the same organization (the IA-station manufacturer). Their EE certificates shall contain the same device identity information.

A CNC shall support at least one trust anchor for IDevID credentials per supported IA-station manufacturer.

6.3.3.3.4 Entry Naming

IDevID credentials shall be present in an ‘asymmetric-key’ entry that is identified as: /ietf-keystore:keystore/asymmetric-keys/asymmetric-key/name=
IDevID-<SignatureSuiteName>-<CertificateSerialNumberOfEECertificate>.

IDevID trust anchors shall be present in ‘certificate’ entries that are identified as: /ietf-truststore:truststore/certificate-bags/certificate-bag/certificate/name=
IDevID-<SignatureSuiteName>-<CertificateSerialNumberOfCACertificate>.

Such entries shall be present underneath a ‘certificate-bag’ entry that is identified as: /ietf-truststore:truststore/certificate-bags/certificate-bag/name=IDevID.

6.3.3.4 Processing Model

6.3.3.4.1 General

The processing model for IDevID credentials and trust anchors shall comply to IEEE Std 802.1AR and 6.3.3.4.

6.3.3.4.2 Credentials

6.3.3.4.2.1 General

IDevID credentials are used in following use cases:

- NETCONF/YANG security setup from factory default; the number of such events scales with the number of factory resets i.e., this use case is performed sporadically. It is conducted by CNCs and encompasses a device identity verification, and
- Device identity verification happens as a subtask during NETCONF/YANG security setup from factory default. It can also at the discretion of the CNC user. The details of device identity verification are also subject to given policy.

In these use cases, IA-stations act in claimant role and CNCs act in verifier role:

- IA-stations shall present the certification path of and prove private key possession for an IDevID credential.
- CNCs shall validate the certification path, check the proof-of-possession for the private key, and verify the obtained device identity information.

6.3.3.4.2.2 Creation

IA-station manufacturers select the form factor for representing verifiable device identity in IDevID credentials: raw or extended form. The details of the IDevID credential issuance process are manufacturer-specific and not addressed in this document.

IA-station manufacturers are not required to offer an update feature for IDevID credentials.

6.3.3.4.2.3 Distribution

IA-stations shall supply IDevID credentials in form of built-in keys, see 6.3.3.3.

6.3.3.4.2.4 Use

Verifiers (CNCs) shall perform the following checks when they challenge claimants (IA-stations) to authenticate themselves by means of an IDevID credential.

- IDevID certification path validation according to IETF RFC 5280, Clause 6. Whether this validation happens with or without revocation checks is at the discretion of the CNC user.
  - It is the responsibility of the CNC user to supply a trust anchor configuration (set of trusted certificates or trusted public keys), a revocation check instruction (Boolean) and optionally CRL objects to CNCs. The certification path validation is passed if and only if the IDevID EE certificate is the leaf of a valid certification path that ends with a CA certificate which is signed by a configured trust anchor and which is not revoked (if revocation check is enabled).
- Proof-of-possession checking for the private key. The proof-of-possession check is passed if and only if the IA-station possesses the private key which matches the public key in the IDevID EE certificate.
- Device identity verification:
  - It is the responsibility of the CNC user to establish and supply to CNCs: a device identity verification policy which determines the verifiable device identity subset that shall be checked by the CNC for the IA-stations in a configuration domain. This is a subset of {description, hardware-rev, serial-num, mfg-name, model-name}. The empty subset (“no-identity-check”) as well as the whole set are allowed.
  - The device identity verification for an IA-station instance shall behave as follows:

- If this subset is empty, then the device identity check is passed. If the user chooses not to verify identity, information about the devices is considered unreliable. Tracking the unverified status of such devices is the responsibility of user. It is the responsibility of the user to establish policies for the use of such devices.
- If this subset is non-empty, then the CNC performs the following expected vs. actual check for each verifiable device identity item in this subset:
  - The check for any item in this subset is passed if the expected value (from ietf-hardware YANG module) matches the actual value (from the verifiable device identity URN value for this document in the subjectAltName extension of the IDevID EE certificate). This check fails if the IDevID has raw form.
  - The device identity check is passed if it is passed for all items in the subset.

IDevIDs in raw form (without verifiable device identity URN) can be used if the device identity verification setting option "no-identity-check" is employed. This allows to perform the NETCONF/YANG security setup from factory default for IA-stations with IDevID credentials in raw form. From CNC perspective these IA-stations remain anonymous.

NOTE This document does not specify a mechanism for device identity verification for IDevIDs in raw form. Whether and how device identity checks for such IA-stations are done in an offline mode is at the discretion of CNC users.

6.3.3.4.2.5 Storage

IDevID credentials shall be stored persistently upon an IA-station. The details for implementing this persistent storage are IA-station manufacturer-specific and not addressed in this document.

6.3.3.4.2.6 Revocation

It is the responsibility of IA-station manufacturers to report revocation for the IDevID credentials issued by them in form of X.509 CRL objects. These objects are made available in a form that allows relying parties i.e., CNC users to retrieve them at their own discretion.

CNC users decide whether they support IDevID certification path validation with or without revocation:

- if revocation checks are disabled, then certificate path validation shall be performed according to IETF RFC 5280, 6.1 Basic Path Validation, and
- if revocation checks are enabled, then certificate path validation shall be performed according to IETF RFC 5280, 6.1 Basic Path Validation and 6.3 CRL Validation

NOTE It is the responsibility of CNC users to obtain up-to-date X.509 CRL objects from manufactures and make them locally available for verifiers.

6.3.3.4.3 Trust Anchors

6.3.3.4.3.1 General

Trust anchors are input arguments for certification path validation according to IETF RFC 5280, 6.1.1 input argument (d). Relying parties decide about these input arguments in a discretionary fashion i.e., these objects are not created and distributed as literal trust anchor objects but in a pre-material form of self-signed certificate objects.

NOTE The digital signature in self-signed certificates do not vouch for authenticity of this object: Actor X can issue self-signed certificates featuring the name of actor A that cannot be distinguished from self-signed certificates issued by A. The mechanisms to verify the authenticity of self-signed certificates are not addressed in this document.

The trust anchors for use cases where IA-stations act in claimant role are determined by CNC users.

6.3.3.4.3.2 Creation

The details of the issuance and update processes for self-signed root certificates for validation of IDevID credentials are not addressed by this document.

6.3.3.4.3.3 Distribution

With respect to use cases where IA-stations act in claimant role e.g., NETCONF/YANG security setup and device identity verification the following model applies:

- issuers (IA-station manufacturers) create and distribute self-signed root certificates. Issuers also provide out-of-band means that allow relying parties to check the authenticity of these objects, and
- relying parties (CNC users) check the authenticity of self-signed root certificates and decide about their acceptance as trust anchors for certification path validation in a discretional manner and configure their verifiers (CNCs) accordingly.

The details of distribution and validation of self-signed root certificates are not addressed by this document.

6.3.3.4.3.4 Use

Trust anchors for IDevID credentials are used for certification path validation according to IETF RFC 5280, 6.1.1 d). This concerns CNCs with respect to the use cases NETCONF/YANG security setup from factory default, device identity verification.

6.3.3.4.3.5 Storage

Trust anchors for IDevID credentials shall be stored persistently upon CNCs. The details for implementing this persistent storage are not addressed in this document.

6.3.3.4.3.6 Revocation

IA-station manufacturers are not required to support an authority revocation feature for IDevID credential certification authorities.

6.3.4 Security setup based on IDevID

6.3.4.1 General

IA-stations in factory default state shall conduct a security setup sequence for the Configuration Domain. This sequence consists of the following steps, each step is described in 6.3.4.

- imprintTrustAnchor: imprint of a Configuration Domain specific trust anchor to an IA-station that allows to validate LDevID-NETCONF certificates presented by communication partners.
- imprintCredential: imprint of a Configuration Domain specific credential to an IA-station, i.e., a private key and the corresponding X.509 v3 end entity certificate (plus intermediate CA certificates, if applicable) plus self-signed root CA certificate that serves as own LDevID-NETCONF credential.
- imprintCertToNameMapping: imprint a Configuration Domain specific certificate-to-name mapping to an IA-station.

6.3.4.2 imprintTrustAnchor

IA-stations in factory default state shall support the imprinting of a single Configuration Domain specific trust anchor via NETCONF-over-TLS according to a procedure called “provisional accept of client certificate”, which uses an IDevID credential on NETCONF and TLS server side (IA-station) and a LDevID-NETCONF credential on NETCONF and TLS client side (for example, a CNC) and operates as follows at the NETCONF and TLS server.

a) Challenge the client for TLS client authentication according to IETF RFC 7589 by sending a CertificateRequest message with an empty certificate_authorities entry.

b) Perform certification path validation according to IETF RFC 5280, Clause 6, for the contents of the client’s Certificate message. This certification path validation fails due to a missing trust anchor for the LDevID-NETCONF credential.

c) Provisionally accept the failing certification path validation when the reason is “no matching trust anchor” (and only this reason) and proceed with the TLS exchange.

d) Expect the client to send a trust anchor for LDevID-NETCONF over the provisionally accepted TLS session (no other object type).

e) If the trust anchor in the NETCONF application payload was accepted, then redo the priorly failing certification path validation using this trust anchor, see step b).

f) If this certification path revalidation is successful, then keep the TLS session alive and send an <rpc-reply> with success. The client then is expected to perform the NETCONF exchanges for imprintCredential (described in 6.3.4.3) and for imprintCertToNameMapping (described in 6.3.4.4) via the already established TLS session.

g) If this certification path revalidation is not successful, then terminate the TLS session. The usual NETCONF/YANG hygiene applies. This is expected to remove the entry in the ietf-truststore that was created in step d).

NOTE This "provisional accept of client certificate" is a mirrored version of the "provisional accept of server cert" in IETF RFC 8995.

The "provisional accept of client cert" in factory default state shall skip the certificate-to-name mapping and shall use the NACM recovery session, i.e., skip permission checking. In this model all authenticated clients are accepted as authorized for doing the first imprinting of the LDevID-NETCONF credential and the corresponding trust anchor. Only contextual checks such as "once only when being in factory default state" are feasible. This model is also known as "trust on first use" (TOFU) and, e.g., also allows to read contents of the ietf-hardware module by the client for an extended identity check.

The imprinting NETCONF client should check the actual server identity that is stated by the IA-station on TLS level by matching against the following.

- End entity certificate contents: a list of accepted (or blocked) manufacturers.
- A list of accepted (or blocked) product instances by their product serial number per accepted manufacturer.
- End entity certificate object as a whole: a list of pinned certificates.

Details of how this matching happens depend on the implementation of the client that performs this imprinting.

The LDevID-NETCONF trust anchor certificate shall be imprinted using the truststore container of the ietf-truststore module with:

- /ts:truststore/ts:certificate-bags/ts:certificate-bag/ts:name = IEC60802,
- /ts:truststore/ts:certificate-bags/ts:certificate-bag/[ts:name=IEC60802]/,
- ts:certificate/ts:name = IEC60802-LDevID,
- ts:certificate/ts:cert-data containing the IEC60802-LDevID trust anchor certificate data object of type trust-anchor-cert-cms according to ietf-crypto-types, i.e., enveloped in Base64-encoded CMS SignedData in degenerated form "certs-only" (no signature value), and
- The imprintTrustAnchor step shall use the NETCONF operation <edit-config> according to IETF RFC 6241 for the truststore container. The NETCONF operation <commit> shall not yet be applied, but rather after successful completion of all security setup sequence steps.

6.3.4.3 imprintCredential

6.3.4.3.1 General

The LDevID-NETCONF end entity certificate shall be provided as X.509 v3 public key certificate according to IETF RFC 5280, Clause 4, with the following criteria.

- Contains the FQDN of the NETCONF server in its subjectAltName extension according to IETF RFC 7589, Clause 6, and IETF RFC 6125, 2.2 and B.7.
- Contains an ECDSA public key and shall be signed with ECDSA according to the selected cryptographic algorithm.
- Contains a digitalSignature in its keyUsage extension.
- Has a finite validity period.

NOTE The actual length of the validity period is at the discretion of the user of the Configuration Domain.

Dependent on the key generation capabilities, different steps are applied to this keystore container.

6.3.4.3.2 Internal key generation

For IA-station with internal key generation capabilities, two NETCONF exchanges are performed. Processing steps for the first NETCONF exchange shall be applied as follows at the NETCONF server.

a) Receive and process the NETCONF request message with action <generate-csr> and input values
    - /ks:keystore/ks:asymmetric-keys/ks:asymmetric-key/[ks:name=LDevID_NETCONF]/ks: generate-csr/ks:input/ks:csr-format containing identity p10-csr according to ietf-crypto-types, and
    - /ks:keystore/ks:asymmetric-keys/ks:asymmetric-key/[ks:name=LDevID_NETCONF]/ks: generate-csr/ks:input/ks:csr-info containing a Base64-encoded PKCS#10 CertificationRequestInfo according to IETF RFC 2986, Clause 4.
b) Base64-decode the <csr-info> value and parse it as a PKCS#10 CertificationRequestInfo object.
c) Extract the algorithm information from the child element SubjectPublicKeyInfo of CertificationRequestInfo and randomly generate a key pair for the specified algorithm.
d) Internally store the private key together with its metadata for example, algorithm information, <name> value in a secure manner.
e) Put the public key into the (parsed) PKCS#10 CertificationRequestInfo.
f) Serialize the PKCS#10 CertificationRequestInfo (including the public key).
g) Use the private key to create signature value for the (serialized) PKCS#10 CertificationRequestInfo (including the public key).
h) Create a NETCONF reply message with /ks:keystore/ks:asymmetric-keys/ks:asymmetric-key/[ks:name=LDevID-NETCONF]/ks:generate-csr/ks:output/ks:p10-csr containing the data object of the previous step.

In the second NETCONF exchange, the LDevID-NETCONF end entity certificate (plus intermediate CA certificates) shall be imprinted using the keystore container of the ietf-keystore module with:

- /ks:keystore/ks:asymmetric-keys/ks:asymmetric-key/ks:name = LDevID-NETCONF,
- /ks:keystore/ks:asymmetric-keys/ks:asymmetric-key/[ks:name=LDevID-NETCONF]/,
- ks:certificates/ks:certificate/ks:name = LDevID-NETCONF, and
- ks:certificates/ks:certificate/ks:cert-data containing the certificate chain LDevID-NETCONF end entity certificate (plus intermediate CA certificates, if applicable) plus self-signed root CA certificate as data object of type end-entity-cert-cms according to ietf-crypto-types

The imprintCredential step shall use the NETCONF operation <edit-config> according to IETF RFC 6241 for the keystore container. The NETCONF operation <commit> shall not yet be applied, but rather after successful completion of all security setup sequence steps.

6.3.4.3.3 External key generation

External key generation can be used for IA-stations without internal key generation capability. For external key generation, one NETCONF exchange is performed.

The LDevID-NETCONF private key and end entity certificate (plus intermediate CA certificates) shall be imprinted using the keystore container of the ietf-keystore module with:

- /ks:keystore/ks:asymmetric-keys/ks:asymmetric-key/ks:name = LDevID-NETCONF,

- /ks:keystore/ks:asymmetric-keys/ks:asymmetric-key/[ks:name=LDevID-NETCONF]/,
- ks:certificates/ks:certificate/ks:name = LDevID-NETCONF,
- ks:certificates/ks:certificate/ks:public-key-format describing the encoding of the public key of the selected cryptographic algorithm according to ietf-crypto-types,
- ks:certificates/ks:certificate/ks:public-key containing the public key value in the selected public-key-format,
- ks:certificates/ks:certificate/ks:private-key-format describing the encoding of the private key of the selected cryptographic algorithm according to ietf-crypto-types,
- ks:certificates/ks:certificate/ks:cleartext-private-key containing the private key value in the selected private-key-format,

NOTE The option <cleartext-private-key> was picked to make the first description as simple as possible. This is not meant as the recommended or preferred form.

- ks:certificates/ks:certificate/ks:name = LDevID-NETCONF, and
- ks:certificates/ks:certificate/ks:cert-data containing the certificate chain LDevID-NETCONF end entity certificate (plus intermediate CA certificates, if applicable) plus self-signed root CA certificate as data object of type end-entity-cert-cms according to ietf-crypto-types.

The imprintCredential step shall use the NETCONF operation <edit-config> according to IETF RFC 6241 for the keystore container. The NETCONF operation <commit> shall not yet be applied, but rather after successful completion of all security setup sequence steps.

External key generation can introduce security vulnerabilities during the generation and loading process. Ensuring those processes are secure is the responsibility of the user and not addressed in this document.

6.3.4.4 imprintCertToNameMapping

The Configuration Domain specific certificate-to-name mapping shall be imprinted using the x509c2n container in the ietf-x509-cert-to-name module with:

- x509c2n:cert-to-name/,
- id = 1,
- x509c2n:tls-fingerprint containing the Configuration Domain specific fingerprint of the LDevID-NETCONF trust anchor, and
- x509c2n:map-type <xmlns=" urn:ieee:std:60802:security”> = ext-60802-roles

The application of this map-type is described in 6.3.4.2, steps e) and f).

The imprintCertToNameMapping step shall use the NETCONF operation <edit-config> according to IETF RFC 6241 for the x509c2n container. Afterwards the NETCONF operation <commit> shall be applied to finalize the security setup sequence steps and to leave the factory default state.

6.3.5 Secure configuration based on LDevID-NETCONF

Configuration by NETCONF/YANG is protected by NETCONF-over-TLS as described in 6.3.2.1 and NACM as described in 6.3.2.2. The NETCONF/YANG servers and clients shall use LDevID-NETCONF credentials for authentication.

The procedure called “provisional accept of client certificate” as described in 6.3.4.2 shall not be applied anymore if the IA-station has left the factory default state. Instead, after successful establishment of a TLS session according to IETF RFC 7589 and IETF draft-ietf-netconf-over-tls13, the NETCONF server shall perform a certificate-to-name mapping and authorization check as follows.

a) Compare the fingerprint of the trust anchor of the NETCONF client’s certification path with the fingerprint contained in cert-to-name list entries of the x509c2n container for equal values.

b) If no cert-name list entry match is found, then terminate the TLS session.

c) If a cert-to-name list entry match is found, then verify if the map-type is equal to ext-60802-roles.

d) If the map-type does not match, then terminate the TLS session.

e) If the map-type value matches, then extract the role values from the id-60802-pe-roles certificate extension of the NETCONF client’s TLS-authenticated end entity certificate. The output is a list of string values from the enumeration of defined role names according to this document.

f) The list of role name string values is provided as input to NACM for permission checking. The access to the requested resource is checked according to the rules configured in the nacm container of the ietf-netconf-acm YANG module.

The NETCONF client checks if the expected identity to address the NETCONF server (IP address or DNS name) matches to the actual server identity that is stated by the IA-station on TLS level. This shall be done by comparing the expected identity with the subjectAltName extension of the TLS authenticated LDevID-NETCONF end entity server certificate.

6.4 Management

6.4.1 General

Subclause 6.4 describes a model for configuration, deployment, and management of an industrial automation network.

6.4.2 IA-station management model

6.4.2.1 General

The management model of IA-stations covers simple end station IA-stations as well as combined IA-stations as described in 4.3. The IA-station management model is applied for topology discovery, network provisioning and stream establishment.

6.4.2.2 IEEE 802.1Q management model

In industrial automation both Bridge and end station components make use of IEEE 802.1Q defined functionality (for example, traffic classes, gate control). Thus, the IEEE 802.1Q management model is the basic management model to be applied to all IA-stations. Figure 19 shows the implementation of the IEEE Std 802.1Q Bridge model in YANG as specified in IEEE Std 802.1Q-2022, Clause 48. The IETF Interface Management YANG data model is specified in IETF RFC 8343.

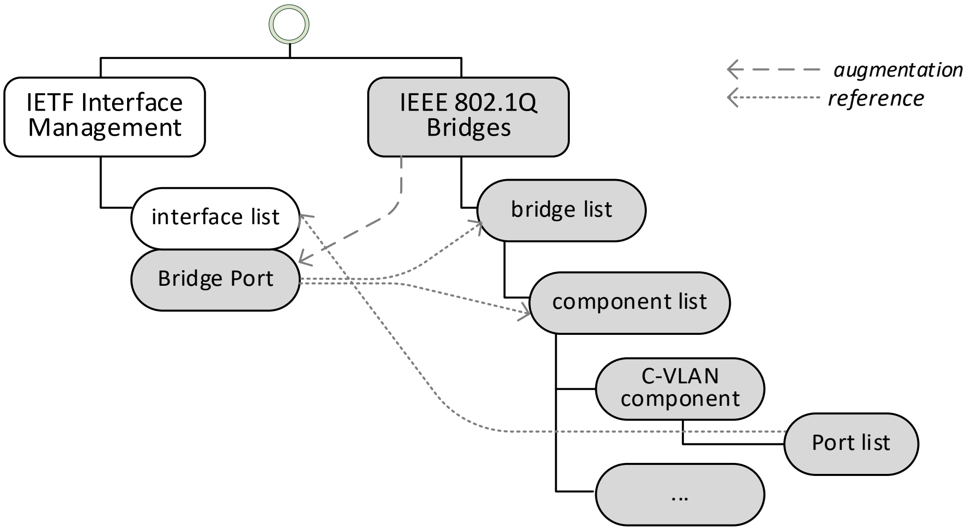


Figure 19 – Generic IEEE 802.1Q YANG Bridge management model

The IEEE 802.1Q Bridge model is organized as a bridge list where each bridge includes an underlying component list (for example, C-VLAN components). Each component has a Port list attached with references to the representation of the ports in the IETF interface list. The managed data of the ports is defined as Bridge Port augmentation to the IETF interface model. Each Bridge Port includes a reference to its bridge and component instances in the IEEE 802.1Q Bridge model.

The YANG data model for an IEEE 802.1Q Bridge is applied to IA-stations:

- Each functional unit of an IA-station is modeled as bridge entry in the bridge list.
- Each Bridge and end station component of an IA-station is modeled as C-VLAN component.
- The IA-station components belonging to a common functional unit are added to the component list of this functional unit’s bridge entry.
- Each IA-station external or internal port is modeled as Bridge Port.

The IA-station ports belonging to a common component are added to the Port list of the related component list entry.

Further YANG data models which are relevant for IA-stations are described in 6.4.9.

6.4.2.3 Internal LAN connection model

The modeling of internal connections between C-VLAN components within an IA-station is aligned to IEEE Std 802.1Q-2022, 17.3.2.2, which includes an I-LAN interface. As shown in Figure 20, the I-LAN interface is modeled as an ilan IETF interface object (see IETF RFC 7224) together with appropriate `higher-layer-if` and `lower-layer-if` reference objects to describe the internal connection.

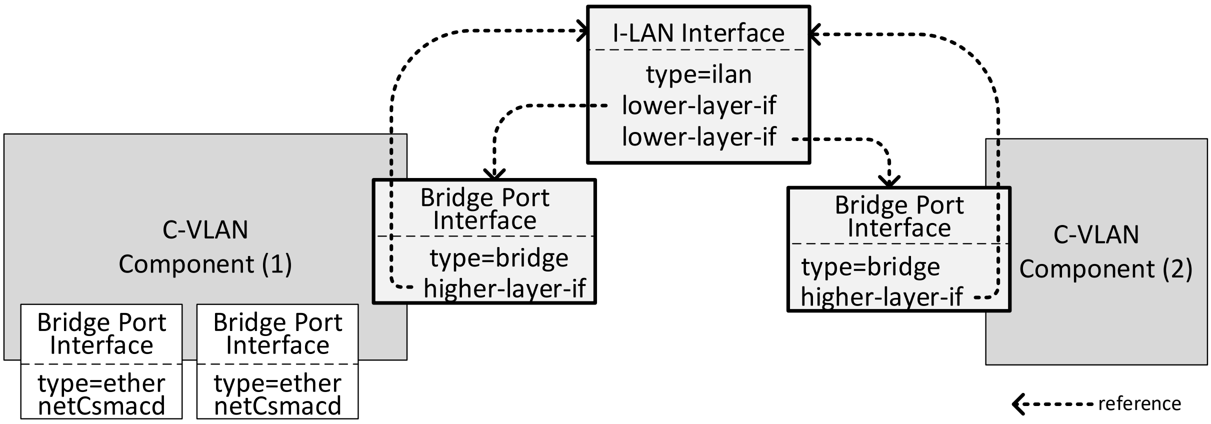


Figure 20 – Internal LAN connection management model

This internal LAN connection model comprises three configuration steps:

- The internal Ports of the C-VLAN components are modeled as IETF interfaces of type bridge with Bridge Port augmentation.
- An additional I-LAN interface of type ilan according to IETF RFC 7224 is created.
- The I-LAN interface references the internal Bridge Port interfaces of the connected C-VLAN components as lower-layer-if, and
- the internal Bridge Port interfaces of the connected C-VLAN components reference the I-LAN interface as higher-layer-if.

Figure 21 shows the application of this model to the example IA-station of Figure 20.

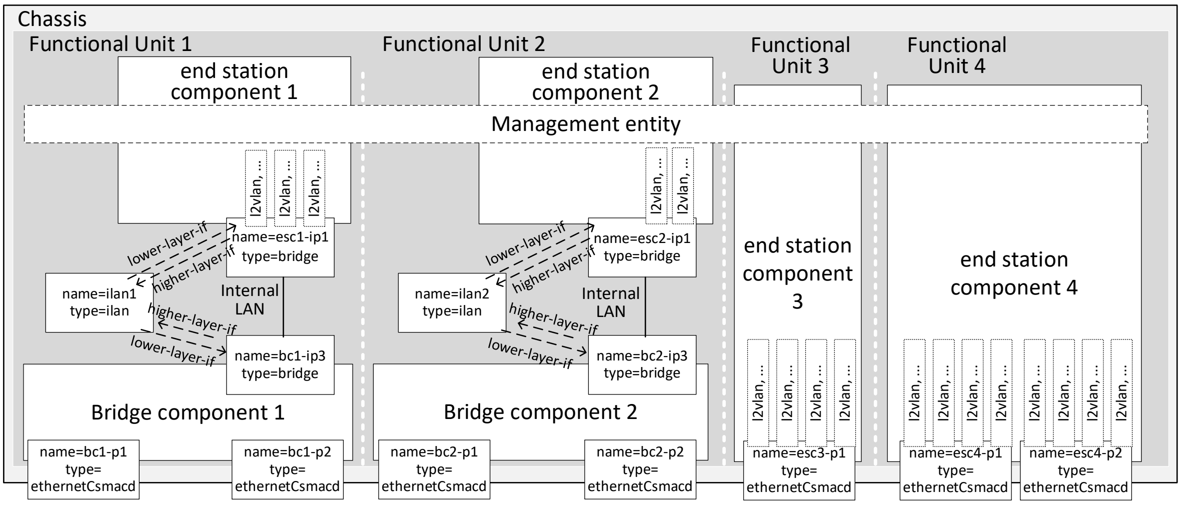


Figure 21 – IA-station example with IETF interfaces

NOTE Figure 21 represents an abstract model and is not intended to imply a particular implementation or partitioning.

Figure 21 also shows the IETF Interfaces of type l2vlan which allow late binding of IA-station applications to the configured VLANs and priorities. The l2vlan interfaces of end station components are described in 6.4.2.5.

6.4.2.4 Spanning Tree, VLAN and TE-MSTID configuration

C-VLAN Bridge components of IA-stations shall support:

- the Common and Internal Spanning Tree (CIST) calculated by the Multiple Spanning Tree Algorithm and Protocol (MSTP), and
- the Traffic Engineering Multiple Spanning Tree Instance Identifier (TE-MSTID) as specified in IEEE Std 802.1Q-2022, 5.5.2.

The MSTP configuration is either default or accomplished by IA-station specific means.

CNCs configure VLANs in the vlan list in the bridge-vlan container of the ieee802-dot1q-bridge YANG module. Ports are assigned to a vlan as static-filtering-entries in a filtering-database.

NOTE vlan, in lowercase, refers to a YANG element.

VLANs are assigned to filtering databases in the vid-to-fid list of the bridge-vlan container. The filtering databases, and in consequence the VLANs, are by default assigned to the MSTP calculated Internal Spanning Tree and can be assigned to the TE-MSTID by management. IA-time-aware streams and IA-streams are assigned to the TE-MSTID.

TE-MSTID assignment is accomplished via the bridge-mst container of the ieee802-dot1q-bridge YANG module.

It is the responsibility of the user to ensure that VLAN names are configured to conform to the scheme defined in 6.4.2.4 to support the required translations for VLAN-ID and PCP values as described in 4.3 and 6.4.2.5. The length of a VLAN name is restricted to a maximum of 32 characters so that a compact name scheme is selected:

VLAN name	60802-[<TrafficTypeCode><PCP>]{1,6}-<VID>[R]

- – <TrafficTypeCode> values are described in the Traffic type code column of Table 7.
- – <PCP> values are in the range of [0..7].
- – <VID> values are in the range of [1..4094].
- – There can be 1 to 6 [<TrafficTypeCode><PCP>] tuples in a VLAN name.

- VLANs with the optional [R] suffix represent VLANs which are used for redundant stream transmission. The VLAN which is associated to a redundant VLAN is identified by the VLAN name without the [R] suffix, with identical <TrafficTypeCode><PCP> tuple values.

VLAN name examples:

– 60802-H6-101	– VID 101 is used for isochronous traffic, which is mapped to PCP 6.
– 60802-H6-102R	– VID 102 is used for the redundant traffic of VID 101.
– 60802-A0B1-100	– VID 100 is used for best effort low traffic applying PCP 0, and best effort high traffic applying PCP 1.

The following example shows the VID/FID/MSTID configuration of an IA-station’s C-VLAN Bridge component, which supports three VLANs in three Forwarding Databases (VID 100 in FID 1, VID 101 in FID 2 and VID 102 in FID 3). FID 2 and FID 3 – and in consequence VID 101 and VID 102 - are assigned to the TE-MSTID. FID 1 – and in consequence VID 100 - is not assigned to a MSTID and thus, is implicitly assigned to the Internal Spanning Tree (IST).

Figure 22 shows the representation of this example configuration in the MST configuration table.

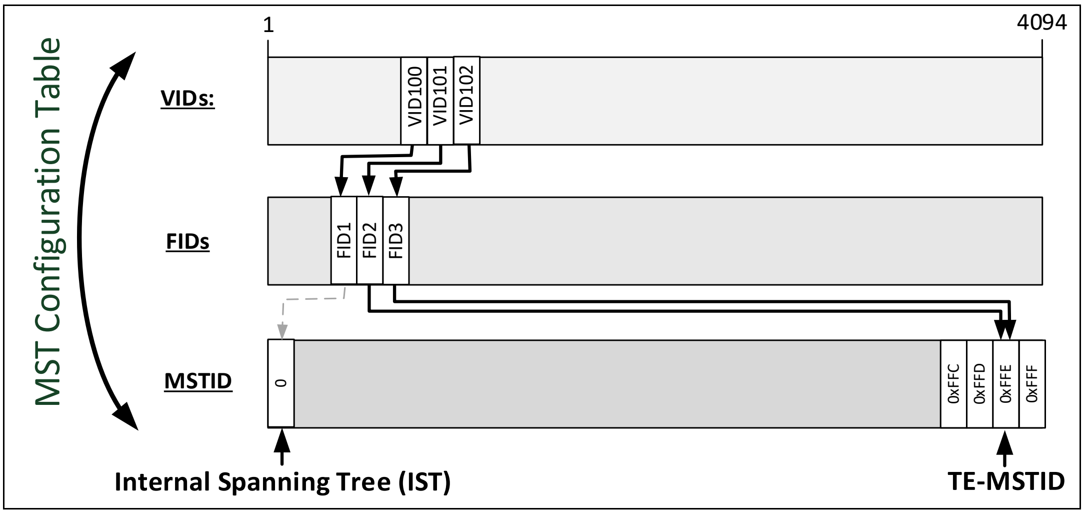


Figure 22 – VID/FID/MSTID example

The YANG-based configuration of this example is shown as YANG instance data snippet of the ieee802-dot1q-bridge YANG module. Herein the MST configuration table is included in component “bridge-component-x”, which is part of bridge “functional-unit-x”.

```
<ieee802-dot1q-bridge xmlns="urn:ietf:params:xml:ns:yang:ieee802-dot1q-bridge">
    <bridges>
        <bridge> <!-- list -->
            <name>functional-unit-x</name>
            ...
            <component> <!-- list -->
                <name>bridge-component-x</name>
                ...
                <bridge-vlan>
                    <version>2</version> <!-- MST supported -->
                    ...
                    <vlan>
                      <vid>100</vid>
                      <name>60802-A0B1-100</name> <!-- best effort high and low -->
                    </vlan>
                    <vlan>
                      <vid>101</vid>
                      <name>60802-H6-101</name> <!-- isochronous -->
                    </vlan>
                    <vlan>
```

```
                      <vid>102</vid>
                      <name>60802-H6-102R</name> <!-- isochronous -->
                    </vlan>
                    ...
                    <vid-to-fid>
                        <vid>100</vid>
                        <fid>1</fid>
                    </vid-to-fid>
                    <vid-to-fid>
                        <vid>101</vid>
                        <fid>2</fid>
                    </vid-to-fid>
                    <vid-to-fid>
                        <vid>102</vid>
                        <fid>3</fid>
                    </vid-to-fid>
                </bridge-vlan>
                ...
                <bridge-mst>
                    ...
                    <fid-to-mstid>  <!-- list -->
                        <!-- fid 1 is implicitly assigned to mstid 0 -->
                        <fid>2</fid>
                        <mstid>4094</mstid>  <!-- TE-MSTID -->
                    </fid-to-mstid>
                    <fid-to-mstid>  <!-- list -->
                        <fid>3</fid>
                        <mstid>4094</mstid>  <!-- TE-MSTID -->
                    </fid-to-mstid>
                </bridge-mst>
                ...
            </component>
        </bridge>
    </bridges>
</ieee802-dot1q-bridge>
```

6.4.2.5 l2vlan type interfaces

Figure 21 shows the IETF Interfaces of type l2vlan (see IETF RFC 7224) in the end station components, which allow late binding of IA-station middleware components and applications to the configured VLANs and priorities.

The CNC/NPE configures the VLANs using the Bridge Component YANG module (ieee802-dot1q-bridge) as shown in 6.4.2.4 with VLAN names describing the usage of PCP/VID values for various traffic types.

The CNC/NPE configures additionally for every member port of the VLAN the l2vlan interfaces with names composed of the VLAN names appended with the port interface name. The lower-layer-if reference can be set by the IA-stations internally to the end station component port interface if required by the end station component.

NOTE The CNC cannot configure the lower-layer-if reference because it is defined read-only in the ietf-interfaces YANG module.

The l2vlan interface names shall conform to the scheme defined in 6.4.2.5 to allow the required translations for VLAN-ID and PCP values as described in 4.6.

VLAN name	as defined in 6.4.2.4
l2vlan interface name	<VLAN name>-<PortIfName>

<PortIfName> is the name of the end station component Port interface in the interface table.

l2vlan name examples:

60802-H6-101-ESC1-IP1	Isochronous traffic on interface ESC1-IP1 is mapped to PCP 6 and VID 101.

60802-H6-102R-ESC1-IP1	Redundant isochronous traffic on interface ESC1-IP1 is mapped to PCP 6 and VID 102.
60802-A0B1-100-ESC1-IP1	Best effort low traffic applying PCP 0, and best effort high traffic applying PCP 1 are both mapped to VID 100 on interface ESC1-IP1.

6.4.3 Discovery of IA-station internal structure

LLDP provides information about the external connectivity of IA-stations. To identify the internal structure of complex IA-stations (see 4.3) the IEEE 802.1Q management model (see 6.4.2.2) and the IETF Interface management model are applied:

- The functional units of an IA-station are represented as bridge entries in the bridge-list.
- The components of a functional unit are represented as component entries in the associated bridge entry’s component-list.
- Internal LAN connections between components of a functional unit are identified by I-LAN entries in the IETF interface list (6.4.2.3).

6.4.4 Network engineering model

To understand the requirements for network configuration, deployment and management, an engineering model covering industrial use cases is required. The “fully centralized model” described in IEEE Std 802.1Q-2022, 46.1.3.3 includes two functional entities: the CUC and the CNC. The relationship between user and network configuration is described in IEEE Std 802.1Q-2022, Clause 46. This document further elaborates this relationship to address uses cases for industrial automation. A conceptual block diagram of a CNC is shown in Figure 23, which adds further details to the CNC specified in IEEE Std 802.1Q-2022 to serve the industrial automation use case. The following functional entities are introduced.

a) The Topology Discovery Entity (TDE)
The topology discovery entity is responsible for the topology discovery (i.e., Bridge component and end station component discovery). The TDE also performs a topology verification in cases where an expected topology is provided by the engineering tool. The resulting topology information is used by the CNC. The TDE detects added or removed IA-stations, including internal structure and connectivity. Thus, the CNC becomes aware of them. Overall, the TDE discovers and maintains an inventory of the devices, including their capabilities and the topology they form.

b) The Path Entity (PE)
The PE computes, establishes and maintains the forwarding paths for the IA time-aware stream and IA stream traffic type categories according to 4.7.3.

c) The Sync Tree Entity (STE)
The STE computes, establishes and maintains the sync trees. For example, for Working Clock and Global Time.

d) The Resource Allocation Entity (RAE)
The RAE is responsible for the allocation of the resources that are necessary for all traffic type categories, according to 4.7.3, to meet their requirements via their forwarding paths. For example, frame buffers at egress ports and FDB entries.

e) The Network Provisioning Entity (NPE)
The NPE applies a network policy provided by the Engineering Tool to the IA-stations within the Configuration Domain. It uses the information discovered by the TDE to create a network configuration based upon this policy which is then applied to all IA-stations. The CNC uses the chosen network configuration together with the discovered IA-stations and their capabilities as input for its stream calculation and deployment.

A CNC includes these functional entities. The implementation of these functional entities and the CNC can vary. The means of communication among these functional entities is implementation dependent.

If there are multiple CNCs in one Configuration Domain, then, by some means not addressed by this document, only a single CNC is in charge at any time in the given Configuration Domain.

The CNC can be in a dedicated station or integrated into any IA-controller or IA-device. Generally, its engineering tool interface is user-specific and can only work with the compatible engineering tools. The definition of this interface is not addressed in this document.

The CUC can be in a dedicated station or integrated into any IA-controller or IA-device. Generally, the CUC is user-specific. In industrial automation use cases, an IA-controller integrated CUC is very likely.

For stream establishment, the UNI of the CNC component is exposed.

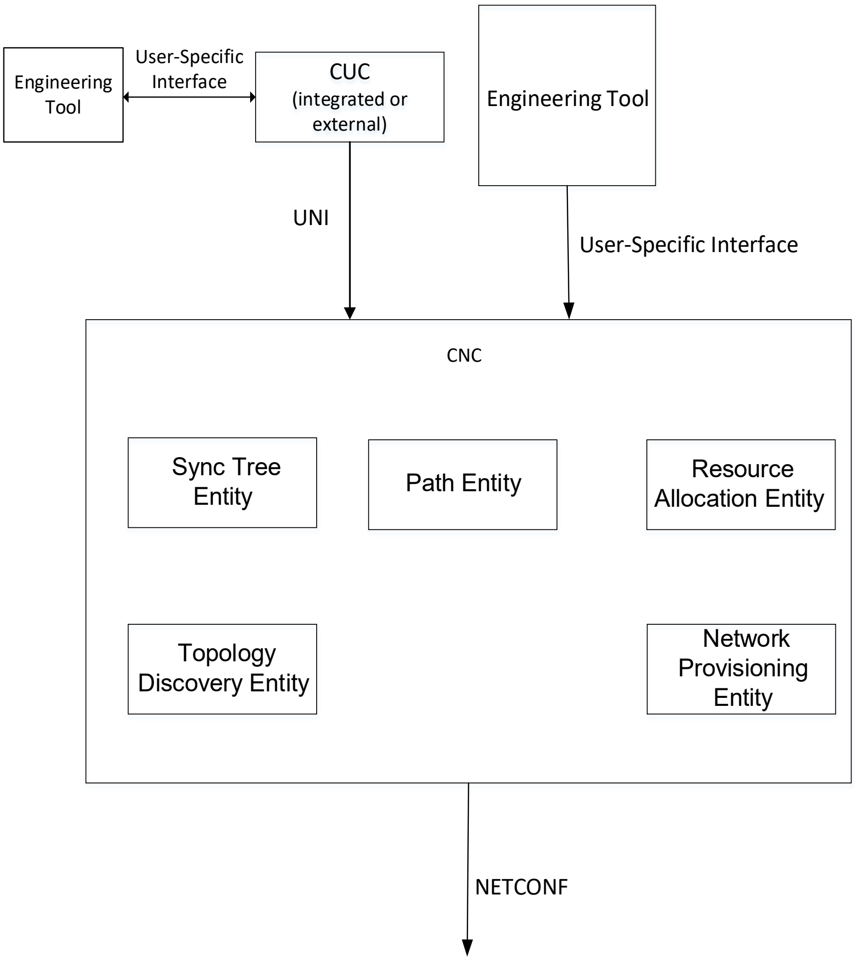


Figure 23 – Structure and interfaces of a CNC

Figure 24 shows an example of the structure of an IA-station which the CNC might discover and manage.

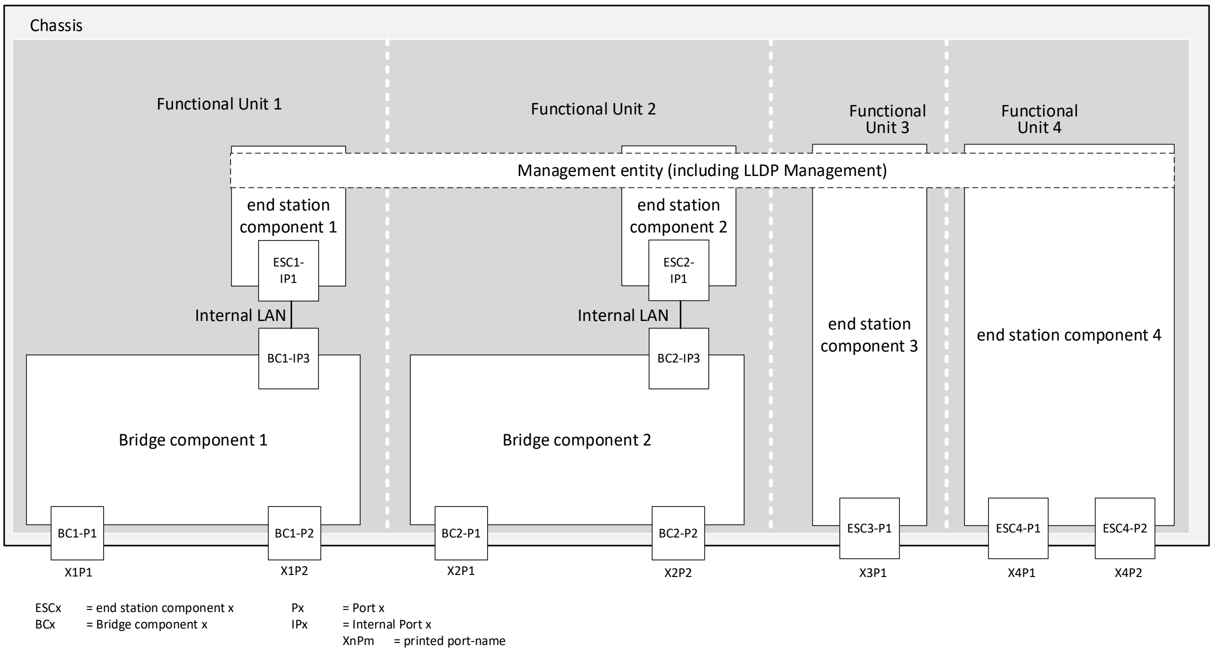


Figure 24 – IA-station structure example

Figure 25 shows the interaction of IA-stations with the CNC.

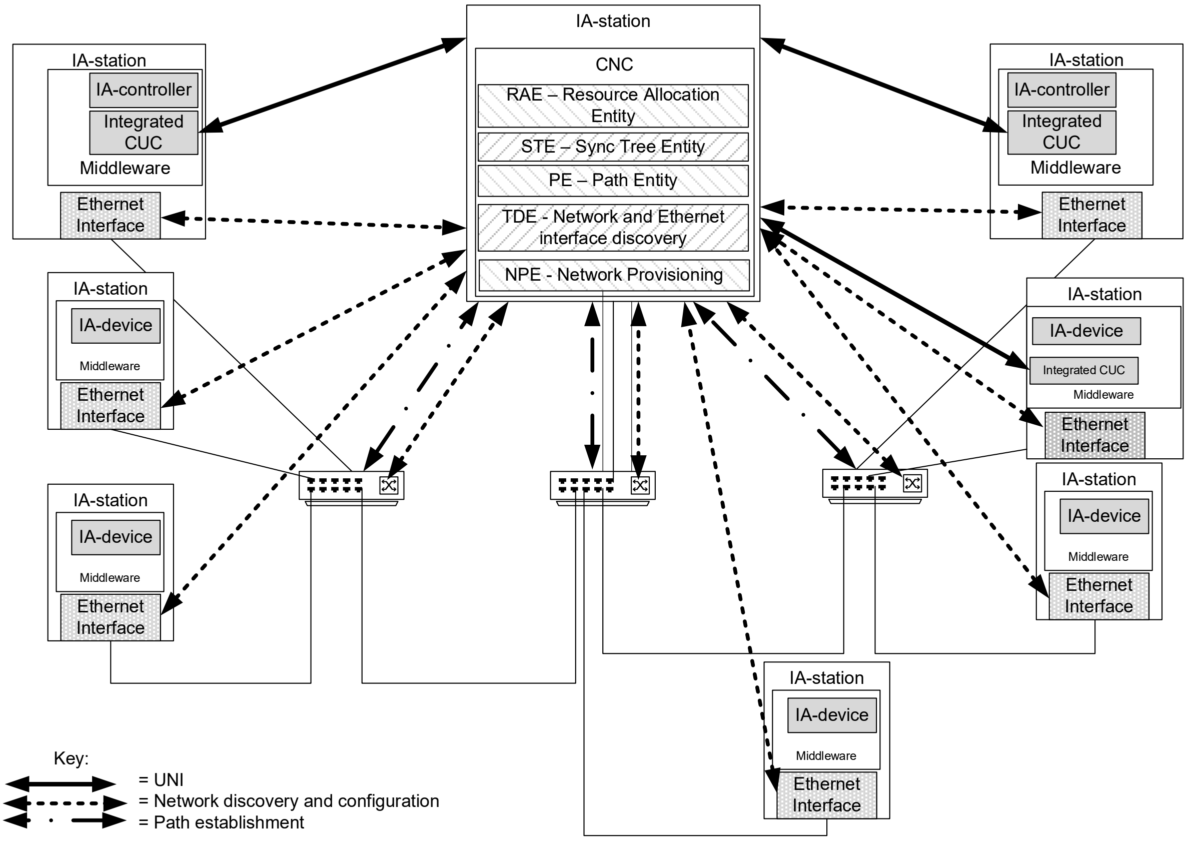


Figure 25 – CNC interaction

6.4.5 Operation

6.4.5.1 General

A representative model for network configuration is shown in Figure 26. This diagram maintains the traditional role of the IA-controller and the IA-device in an industrial automation network. IA-devices and IA-controllers require configuration from engineering tools (refer to engineering tools A, B, D, and E). These tools and associated interfaces are not addressed by this document. In this example, engineering tool C communicates directly with the CNC to provide traffic requirements for the network. The protocols that the engineering tool uses for communication with end stations are specific to the user application.

The UNI is the interface to the CNC which is serviced by NETCONF over TLS. The UNI service recognizes that industrial automation communications are typically connection oriented. There is a communication initiator, typically in an IA-controller, which is responsible for establishing those connections, determining what data is of interest and providing the required update rate. So, while an application/middleware of an IA-station (for example a Drive) understands what information it can produce and the maximum rate at which that information can be provided, until an IA-controller establishes a connection with that device, it does not know where that information goes and what update rate is required to close the control loop. The IA-controller gets this information from its engineering tool. There can be multiple IA-controllers in each Configuration Domain. The CNC uses the topology, the device capabilities, the device configuration, and the traffic specifications from the user to calculate a path for each Talker/Listener pair. The UNI then provides stream identification (VLAN, DMAC, etc.) to the Middleware.

The operational management model, see Figure 26, reflects the current and traditional model used in industrial automation. Figure 26 shows an active CNC managing multiple IA-stations. Each station can wholly incorporate a CUC and interact with the CNC directly.

Security requirements (see 6.3) are an important consideration for these networks and are integrated into the design, configuration, and deployment of any management model.

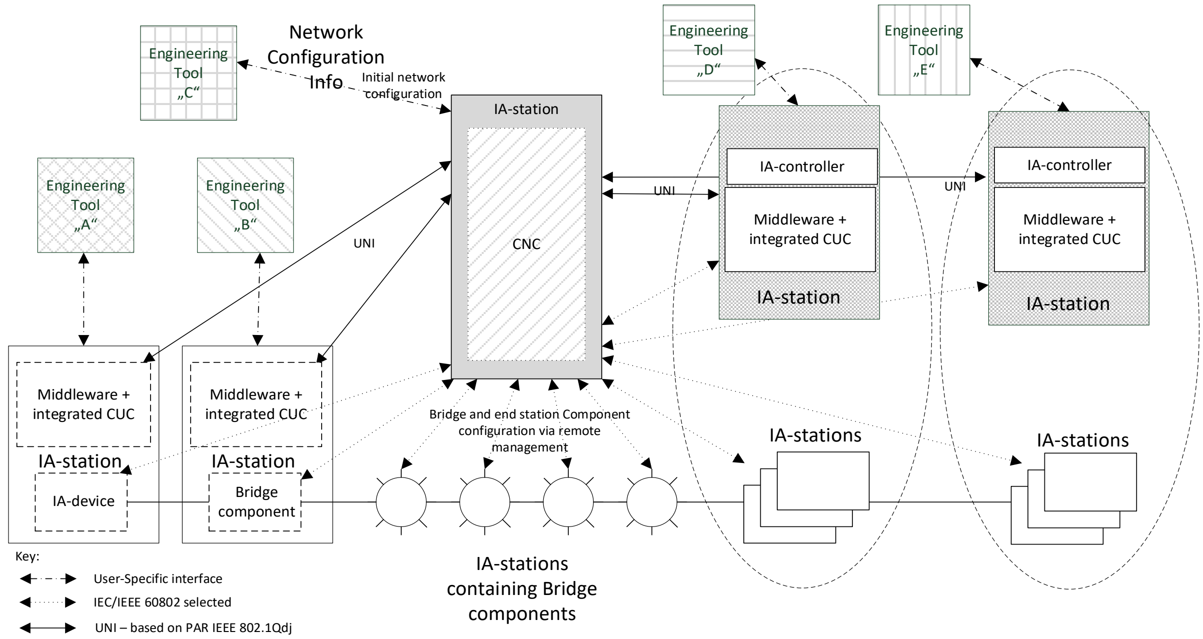
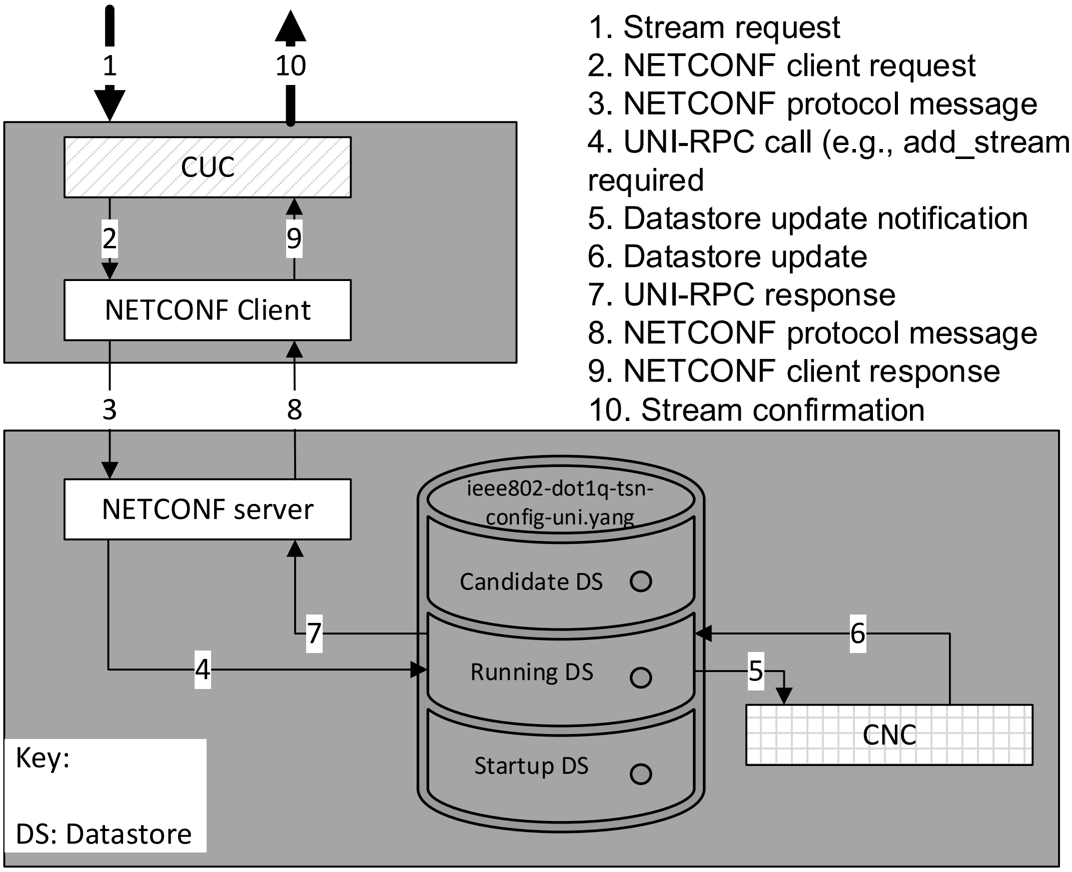

Engineering Tool „C“
Network Configuration Info
Initial network configuration
IA-station
CNC
Engineering Tool „D“
Engineering Tool „E“
IA-controller
Middleware + integrated CUC
IA-station
UNI
UNI
IA-controller
Middleware + integrated CUC
IA-station
Engineering Tool „A“
Engineering Tool „B“
UNI
Middleware + integrated CUC
IA-station
IA-device
Middleware + integrated CUC
IA-station
Bridge component
Bridge and end station Component configuration via remote management
IA-stations
IA-stations
IA-stations containing Bridge components
Key:
User-Specific interface
IEC/IEEE 60802 selected
UNI – based on PAR IEEE 802.1Qdj

Figure 26 – Operational management model

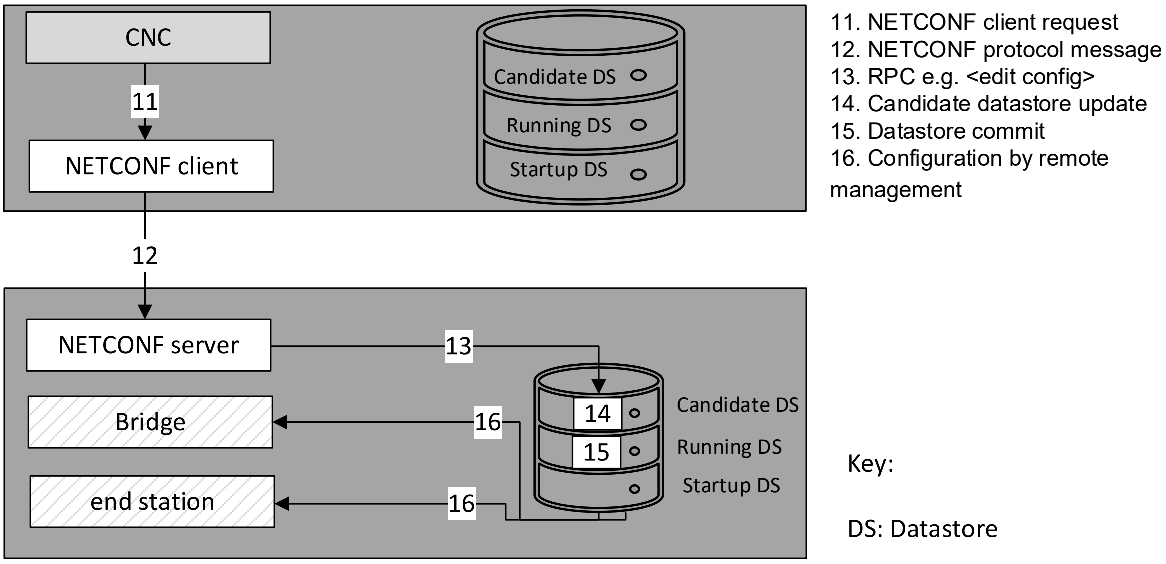
Figure 27 shows the steps that are typically performed in the scope of the CUC-CNC interaction.



1. Stream request
2. NETCONF client request
3. NETCONF protocol message
4. UNI-RPC call (e.g., add_stream), if required
5. Datastore update notification
6. Datastore update
7. UNI-RPC response
8. NETCONF protocol message
9. NETCONF client response
10. Stream confirmation

Figure 27 – UNI service model

After the computation of the paths and the scheduling and/or shaping configuration have been done, the CNC configures the IA-stations via NETCONF client. The typical steps that are performed in this process are shown in Figure 28 below.



11. NETCONF client request
12. NETCONF protocol message
13. RPC e.g. <edit config>
14. Candidate datastore update
15. Datastore commit
16. Configuration by remote management

Figure 28 – CNC southbound

Instances of NETCONF servers and clients within a Configuration Domain are shown in Figure 29. IA-stations that contain a CNC and/or CUC entity contain both a NETCONF server and a NETCONF client. A NETCONF client at the CUC side is needed for the UNI. A NETCONF server at the CNC side is needed to accommodate the UNI as well as remote network management of the end stations and bridges that are contained in the same chassis as the CNC entity. The NETCONF client on the CNC side is needed for the southbound interface of the CNC i.e., for the remote management of the bridges and end stations in the scope of stream configuration. All IA-stations have a NETCONF server to make remote management possible. The NETCONF server used by the CNC serves multiple NETCONF Clients (CUCs) within a single Configuration Domain whose requests clients can occur simultaneously.

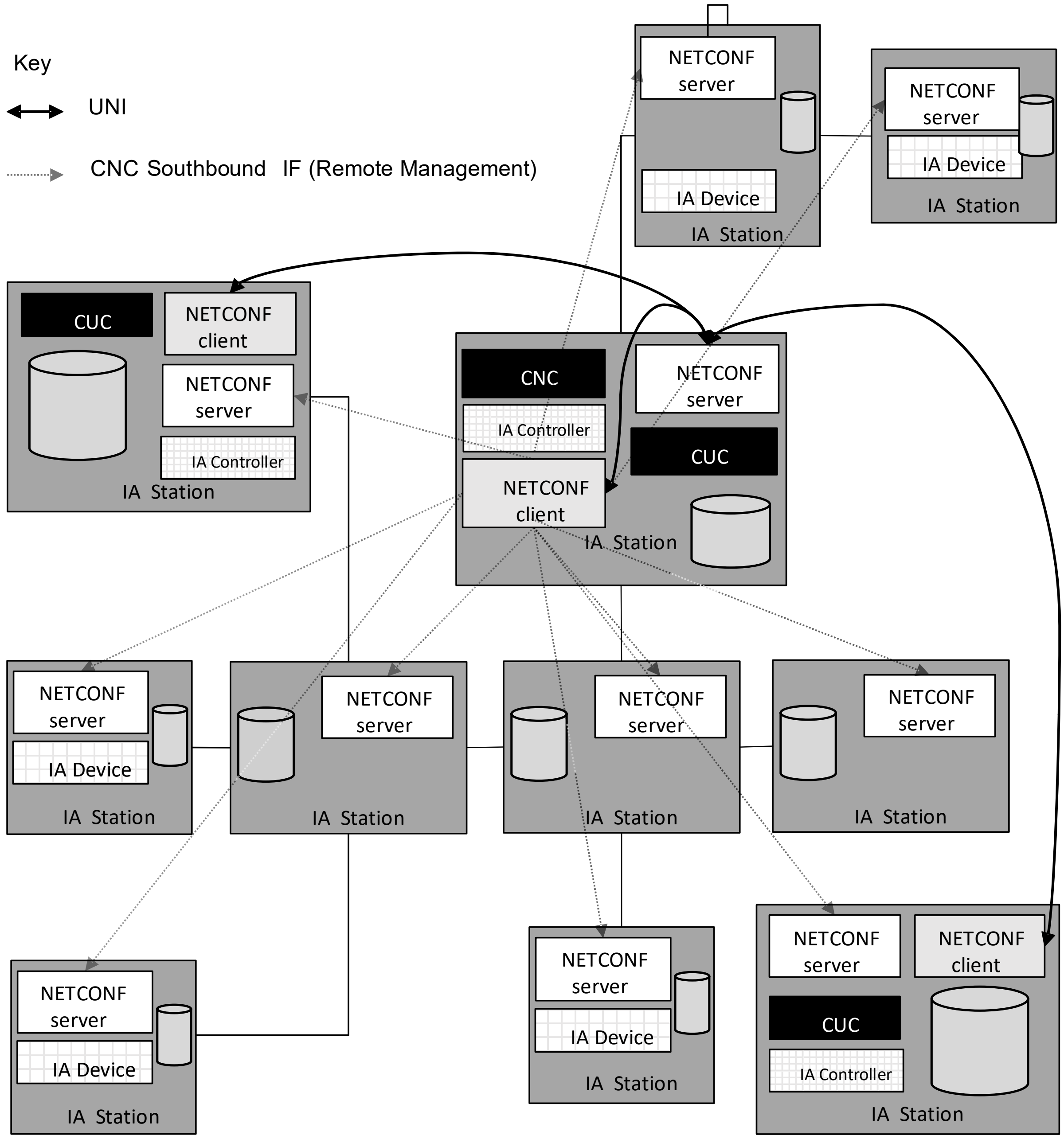


Figure 29 – NETCONF usage in a configuration domain

6.4.5.2 Domain port states

A CNC manages available network resources and assigns them to the IA-stations. Management of the network resources is only possible if the CNC owns these resources. Thus, no connected station is allowed to make use of network resources that are not granted by the CNC. The security configuration of a connected station allows remote access for the CNC.

Protection of the network resources is done by managing the ports (see Figure 30) at the boundary of the Configuration Domain. The state of any newly connected station is unknown. The CNC is responsible for determining if the newly connected station is added to the Configuration Domain and configuring the IA-station appropriately.

This port state model avoids any assumptions about configuration of added stations or network portions.

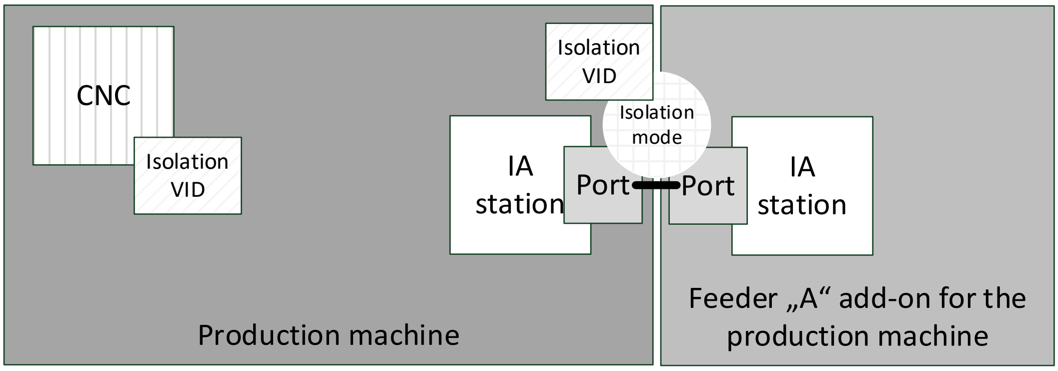


Figure 30 – Boundary port model

Ports of an IA-station that is a member of a Configuration Domain have different states:

- Isolated – a station connected via this port can only exchange information with a CNC. The CNC is responsible for establishing an isolation VID and for on boarding the station. In the isolated state:
  - the port gets to or remains in isolated state in case of a link down event, e.g., when nothing is connected, or no link is established;
  - the port gets to or remains in isolated state in case of a link up event;
  - the port stays in isolated state as long as the neighbor is unknown, not able to enter Boundary state.
- Boundary – a station connected via this port is not part of the Configuration Domain, but is allowed to access devices inside the Configuration Domain and to pass traffic through the Configuration Domain
- Inside – a station connected via this port is part of the Configuration Domain

The determination of whether a given port of an IA-station remains in the Isolated state or transitions to the Boundary or Inside state is performed by the CNC using remote management. A port acts as a domain boundary if it is in the Isolated or Boundary state.

For example, a port could be configured as follows:

- Isolated state
  - Port is IST boundary
  - Port is not part of a sync tree
  - Port uses VLAN stripping for egress
  - Port uses VLAN assignment and priority regeneration to assign all traffic to an isolated VLAN
  - Port uses an ingress rate limiter to control the amount of traffic for the Configuration Domain

- Boundary state
  - Port is part of IST
  - Port is part of a sync tree
  - Port uses VLAN stripping for egress
  - Port uses VLAN assignment and priority regeneration to assign all traffic to a default VLAN
  - Port uses an ingress rate limiter to control the amount of traffic for the Configuration Domain
- Inside state
  - Port is part of IST
  - Port is part of a sync tree
  - Port is part of the active topology for stream and non-stream traffic

An example workflow includes the following steps executed by the CNC:

a) Topology discovery
   1) Case A: Link down / Port not connected
      i) Set port to isolated state
      ii) Configure a NETCONF subscription “on data change” to the port state leaf
   2) Case B: Neighbor is not a Configuration Domain member
      i) Set port to boundary state
      ii) Configure a NETCONF subscription “on data change” to the port state leaf
   3) Case C: Neighbor is not a Configuration Domain member – but part of expected topology
      i) Set port to boundary state
      ii) Configure the neighbor station as Configuration Domain member
      iii) Set port to inside state
b) NETCONF subscription trigger
   Issued to the CNC upon change of subscribed YANG data.

6.4.5.3 Engineered network

For an offline engineered (based on the available digital data sheets of the used IA-stations) centralized approach with fixed topology, fixed stations and fixed paths, the user provides traffic requirements, path information, topology information and expected network configuration to the CNC. The CNC then uses the TDE, RAE and the NPE to perform the calculation of paths, resources, and stream schedules necessary to meet the specified traffic requirements and deploys the result of these calculations via remote management. The CNC also provides the relevant results to the CUC via the UNI. The CUC then configures the end stations using the User-to-User interface (see Figure 3).

The workflow for this example consists of the following steps:

a) The user determines:
   1) the expected network topology
   2) the expected stations and its capabilities, value ranges and quantities
   3) the expected paths and resources
   4) the required streams
   5) the requirements for IA non-stream traffic.

This step focuses on network capabilities including the Ethernet interface of the end stations. For example, if the end station is a sensor, the user needs to consider the Ethernet interface capabilities of the sensor as they apply to the physical world.

b) Engineering Tool provides this information to the CNC via a user-specific interface.

Although the communication between the CNC and any Engineering Tool is user-specific, the CNC needs to obtain all information needed by the integrated TDE and NPE.

c) The CNC uses the TDE to discover the topology and checks it against the expected topology. The NPE is used to configure the IA-stations of the Configuration Domain.
d) The CNC uses STE and NPE to setup, validate, and monitor synchronization configuration in the Configuration Domain.
e) The CNC uses the information from engineering item a), steps 1 to 5, above to respond to requests from Middleware (with integrated CUC) using UNI. These requests are handled using the already established communication paths received from the user.

If the CNC is not required after commissioning, then the CNC can be removed after setting up the IA-stations. That requires that all IA-stations have a persistent storage for the data provided by the CNC.

6.4.5.4 Dynamic topology

6.4.5.4.1 General

For a centralized approach with a dynamic topology and dynamic paths, the user provides the network policy to the CNC. The TDE performs topology discovery including IA-station capabilities (YANG representation of the digital data sheet) and the NPE performs network configuration for the CNC. IA-stations then provide traffic requirements via the Middleware to the CNC via the UNI. The CNC then uses the TDE, RAE, and NPE to perform the calculation of paths, resources, and stream schedules necessary to meet the specified traffic requirements and deploys the result of these calculations via remote management. The CNC also provides the relevant results to the CUC via the UNI. The CUC then configures the end stations using the User-to-User interface (see Figure 3).

The workflow for this example consists of the following steps:

a) The user determines the network policy and provides it to the CNC.
b) The TDE continuously discovers the physical network topology and station capabilities of each station using remote management.
c) The NPE uses the information gathered in steps a) to b) to configure the IA-stations in the Configuration Domain.
d) The CNC uses STE and NPE to setup, validate and monitor time synchronization configuration in the Configuration Domain.

The CNC uses the information from steps a) to d) to respond to requests from Middleware using UNI. The CNC establishes streams in the bridges via a remote management protocol.

6.4.5.4.2 Adding an IA-station

Each IA-station added to the Configuration Domain is discovered by the TDE and receive the network configuration from the NPE. After this, the Middleware can request stream establishment.

When an IA-station is added to the network, it is isolated until the CNC determines that its traffic requirements can be accommodated without disrupting other traffic (see 6.4.5.2).

6.4.5.4.3 Removing an IA-station

Each IA-station removed from the Configuration Domain is discovered by the TDE. A neighboring station can receive an updated network configuration by the NPE. After this, the removed IA-station is no longer part of the Configuration Domain.

6.4.5.4.4 Replacing an IA-station

In the simplest case, replacing an IA-station is simply the sequence of removing an IA-station (6.4.5.4.3) and adding an IA-station (6.4.5.4.2). In more complex cases, other precautions or user actions can be needed following deployment.

6.4.5.5 Engineered network extended by dynamic topology

The engineered and dynamic topology workflows can be used together. For instance, modular machines, robot tool changers or more general plug & produce can add or remove modules. The basic machine is handled as an engineered network. Additional modules or removed modules are handled dynamically.

6.4.6 Engineered time-synchronization spanning tree

6.4.6.1 General

Engineered time-synchronization spanning tree (sync tree) for a given gPTP domain refers to the usage of external port configuration instead of BMCA for the construction of a desired sync tree with the Grandmaster PTP Instance as the root (see IEEE Std 802.1AS-2020, 10.3.1). The Grandmaster PTP Instance can reside in a dedicated grandmaster-capable IA-station.

One of the advantages of engineered sync trees is to enable a planned, deterministic, and stable configuration of the IEEE Std 802.1AS-2020 sync tree for a given gPTP domain. For example, this approach prevents sync tree changes in case of IA-station addition or removal from the network. Working Clock (see 3.3.17) and hot standby (see IEEE P802.1ASdm) are use cases of engineered sync tree.

6.4.6.2 Sync tree requirements

Sync tree requirements for all participating PTP Instances in a gPTP domain are specified in 5.5.3. In addition, 5.6.2 item b) is required for all participating PTP Relay Instances.

6.4.6.3 STE phases

6.4.6.3.1 General

The STE should follow the logical sequence described in 6.4.6.3 if an engineered sync tree is utilized in a gPTP domain. Each STE phase describes an externally observable behavior of the participating PTP Instances in a gPTP domain.

6.4.6.3.2 Discovery phase

In discovery phase, STE utilizes the topology discovered by the TDE to verify the capabilities and status of participating IA-stations via a diagnostics entity (see 6.4.7.1) by reading the following managed objects:

- The status of oper-status parameter is up (see IETF RFC 8343) for all participating Ethernet links.
- The status of isMeasuringDelay (see IEEE Std 802.1AS-2020, 14.16.4) is TRUE for all PTP Ports.
- The status of asCapable (see IEEE Std 802.1AS-2020, 14.8.7) is TRUE for all PTP Ports.
- The status of asCapableAcrossDomains (see IEEE Std 802.1AS-2020, 14.16.5) is TRUE for all LinkPorts.
- The status of gmCapable (see IEEE Std 802.1AS-2020, 14.2.7) is TRUE, only applicable to the Grandmaster PTP Instance.

STE should use the information collected via managed objects and the discovered topology to verify the constraints on the gPTP domain, for example:

- Verify that the number of PTP Relay Instances (hops) between the Grandmaster PTP Instance and any given Slave PTP End Instance is within the limit prescribed by for example, CNC.
- Verify per PTP link that the value of meanLinkDelay (see IEEE Std 802.1AS-2020, 14.16.6) is less than or equal to meanLinkDelayThresh (see IEEE Std 802.1AS-2020, 14.16.7 and IEEE Std 802.1AS-2020, Table 11-1) value to detect for example, anomaly in propagation delay.

NOTE Even if neighboring PTP Instances do report asCapable, it can be that a link between asCapable neighboring PTP Instances is not asCapable due to for example, wrong setting of meanLinkDelayThresh value. The meanLinkDelayThresh value reflects estimated propagation delay of the installed link.

6.4.6.3.3 Provisioning phase

In provisioning phase, STE should apply the desired configuration to all participating PTP Instances, for example:

- The desiredState of all PTP ports of the Grandmaster PTP Instance is set to MasterPort.
- The desiredState of exactly one PTP port of all the other PTP Instances is set to SlavePort.
- The desiredState of remaining PTP ports that are part of sync tree in non-Grandmaster PTP Relay Instances is set to MasterPort.
- The desiredState of all other PTP ports is set to PassivePort.

Then STE should validate, for example, the syncLocked (see IEEE Std 802.1AS-2020, 14.8.52) parameter is TRUE for all PTP ports of PTP Relay Instances that are in MasterPort state.

6.4.6.3.4 Monitoring phase

6.4.6.3.4.1 General

In monitoring phase, STE in combination with for example, TDE and diagnostics entity (see 6.4.7.1) should monitor the status and the performance of the gPTP domain by reading the relevant managed objects.

6.4.6.3.4.2 Status monitoring

The STE in combination with for example, TDE and diagnostics entity (see 6.4.7.1) should monitor the status of the gPTP domain by reading the following managed objects:

- The status of oper-status parameter is up (see IETF RFC 8343) for all participating Ethernet links.
- Verify the existence of at least a single Grandmaster PTP Instance across gPTP domain, i.e., grandmasterIdentity (see IEEE Std 802.1AS-2020, 14.4.4).
- Detect each addition (see 6.7.7.4) and removal (see 6.7.7.5) of a PTP Instance.
- Verify that the number of PTP Relay Instances (hops) between the Grandmaster PTP Instance and any given Slave PTP End Instance is within the limit prescribed by for example, CNC.

6.4.6.3.4.3 Performance monitoring

The STE in combination with the TDE detects the change of status of the gPTP instances within the Configuration Domain by monitoring the following managed objects:

- Verify that the PTP Instances are in SYNCED state (see IEEE P802.1ASdm), i.e., time is synchronized according to the requirements of this document.
- Verify that the clockQuality of Grandmaster PTP Instance (see - IEEE Std 802.1AS-2020, 14.2.4) is within the requirements of this document.
- Detect any change in phase or frequency of the Grandmaster PTP Instance, i.e., lastGmPhaseChange (IEEE Std 802.1AS-2020, 14.3.4), lastGmFreqChange (IEEE Std 802.1AS-2020, 14.3.5).

- Verify per PTP link that the value of meanLinkDelay (see IEEE Std 802.1AS-2020, 14.16.6) is less than or equal to meanLinkDelayThresh (see IEEE Std 802.1AS-2020, 14.16.7 and IEEE Std 802.1AS-2020, Table 11-1) value to detect for example, anomaly in propagation delay.
- Verify that the PTP messages timeout events, syncReceiptTimeoutCount (see IEEE Std 802.1AS-2020, 14.10.10) and announceReceiptTimeoutCount (see IEEE Std 802.1AS-2020, 14.10.11) are within user-defined limits.
- Verify that the RateRatio value (see 6.2.3) is within the expected range (see Table 11 and Table 12) per PTP link.

Any deviation detected by a PTP Instance can be conveyed to the STE via, for example, notification.

6.4.6.4 Adding an IA-station

Each IA-station added to the gPTP domain is discovered by STE via TDE. It is the responsibility of the CNC to on-board this newly added station. IA-stations can receive an updated gPTP configuration via STE.

A newly installed IA-station can disrupt the operation of a gPTP domain. The extent of disruption is dependent on the location of the IA-station in the gPTP domain and the type of PTP Instance running on that IA-station. For example, if PTP Instances are arranged in a daisy-chain formation and if a IA-station with a non-Grandmaster Relay Instance is installed in the middle of a daisy-chain then this change will disrupt for example, the operation of downstream PTP Instances.

6.4.6.5 Removing an IA-station

The removal of a station from the gPTP domain is detected by STE via TDE. IA-stations can receive an updated gPTP configuration via STE.

Removing an IA-station can disrupt the operation of a gPTP domain. It is the responsibility of the CNC to take the steps necessary for the removal of the station without disrupting the network. For example, if PTP Instances are arranged in a daisy-chain formation and if a IA-station that is running a non-Grandmaster Relay Instance is removed from the middle of a daisy-chain then this change disrupts for example, the operation of downstream PTP Instances.

6.4.6.6 Replacing an IA-station

An IA-station replacement follows the sequence of removing a IA-station according to 6.4.6.5 and adding a IA-station according to 6.4.6.4.

6.4.7 Diagnostics

6.4.7.1 General

Diagnosis for an IA-station is done by monitoring YANG representation of the IA-station’s local database.

A vendor can implement an observer in a diagnostics entity, which could reside in the CNC. This diagnostics entity uses the information provided by remote management to define the monitored objects and set up fitting notifications.

6.4.7.2 Observer model

A diagnostic entity can select any objects described via YANG and observe them via NETCONF. The NETCONF binding is specified in RFC 8640, and the subscription model in RFC 8641. NETCONF messages can be pipelined, i.e., a client can invoke multiple RPCs without having to wait for RPC result messages first. RPC messages are defined in RFC 6241 and notification messages are defined in RFC 5277. To reduce the load on the diagnostic entity when many stations are providing notifications, the diagnostic objects can be monitored and notifications can be retrieved from individual IA-Stations.

Figure 31 shows the model of a diagnostic observer.

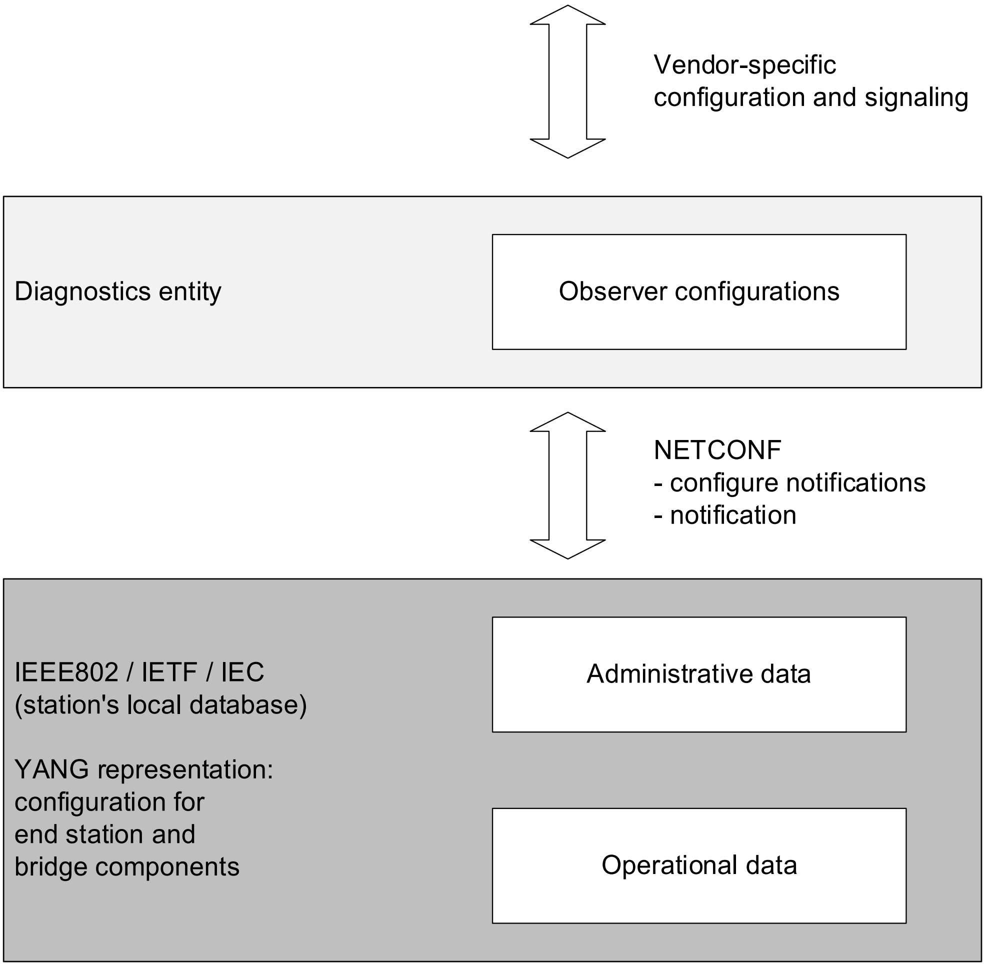


Figure 31 – Observer model

6.4.7.3 Usage of YANG Push

For diagnostics, an IA-station shall support YANG-Push subscriptions according to IETF RFC 8641 (YANG Push) and IETF RFC 8639 (Subscribed Notifications).

IA-stations shall support the “subtree” selection filter as defined in IETF RFC 8041, 3.6

6.4.7.4 Mandatory RPCs

An IA-station shall support following RPCs as defined in IETF RFC 8641:

a) establish-subscription
b) modify-subscription
c) delete-subscription
d) kill-subscription

6.4.7.5 Mandatory notifications

An IA-station shall support following notifications as defined in IETF RFC 8641:

a) subscription-resumed
b) subscription-modified
c) subscription-terminated
d) subscription-suspended
e) push-update
f) push-change-update

6.4.7.6 Mandatory diagnostics data nodes

An IA-station shall provide following data nodes for diagnostic purpose:

a) Change of link-status per Ethernet port:

`/ietf-interfaces/interfaces-state/interface/oper-status`

b) Change of MAU-type per Ethernet port:

`/ieee802-ethernet-lldp/lldp/port/ operational-mau-type`

c) Change of sync-status

   1) per PTP Instance

   – `/dot1as-hs/ptp/instances/instance/ptp-instance-sync-ds/ptp-instance-state`

   – if Grandmaster PTP Instance: `/iecieee60802-ptp/instances/instance/default-ds/clock-source/clock-state`

   – for every application-clock: `/iecieee60802-bridge/bridges/bridge/component/clock/is-synced`

   2) per hot standby Instance

   `/dot1as-hs/ptp/common-services/hss/hot-standby-system-list/hot-standby-system-ds/hot-standby-system-state`

d) Data to be provided as periodic time-aligned subscriptions:

   1) Dropped frames statistic counters per Ethernet interface

   – `/ietf-interfaces/interface/statistics/in-octets`

   – `/ietf-interfaces/interface/statistics/in-discards`

   – `/ietf-interfaces/interface/statistics/in-errors`

   – `/ietf-interfaces/interface/statistics/out-octets`

   – `/ietf-interfaces/interface/statistics/out-discards`

   – `/ietf-interfaces/interface/statistics/out-errors`

   2) VLAN specific counters per Ethernet Interface and VLAN ID

   – `/ieee802-dot1q-bridge/interfaces/interface/bridge-port/statistics/octets-rx`

   – `/ieee802-dot1q-bridge/interfaces/interface/bridge-port/statistics/octets-tx`

   – `/ieee802-dot1q-bridge/interfaces/interface/bridge-port/statistics/forward-outbound`

– `/ieee802-dot1q-bridge/interfaces/interface/bridge-port/statistics/discard-inbound`

6.4.7.7 Usage of NETCONF notifications

IA-stations shall implement the binding of a stream of events according to IETF RFC 8640 (NETCONF Notifications) using the "encode-xml" feature and the "NETCONF" event stream of IETF RFC 8639.

An IA-station shall support dynamic subscriptions as defined in IETF RFC 8640 Clauses 5, 6 and 7.

6.4.8 Data sheet

6.4.8.1 General

The user requires data sheets containing the capabilities, value ranges and quantities of IA-stations. See Annex B for example quantities in a representative Configuration Domain. Data sheets need to be available for offline and online engineering.

Online datasheets are modeled using YANG. YANG modeling can also be used for the offline data sheet to keep the offline (6.4.5.3) and online (6.4.5.4) format the same.

6.4.8.2 Digital data sheet of an IA-station

Both engineering models, offline via an engineering tool and online with plug & produce by the CNC, require information about the capabilities of an IA-station, for example, states, configurations, supported features, etc. An example depicting the creation of a digital datasheet is provided in Figure 32.

This data is extracted from the implemented YANG modules, which are available in the local database of the IA-station.

The data from the implemented YANG modules is also available offline in the form of a digital data sheet of an IA-station as a digital data sheet file.

The digital data sheet of an IA-station provides a collection of all instantiated data nodes of all YANG modules that are required by this document (see 6.4.9). A manufacturer may reduce the instance data set by removing statistical config-false YANG nodes.

The digital data sheet does not contain any additional information that is not modeled by the YANG modules that exist in the local database of the IA-station.

The data sheet contains a single instance data set. It carries complete configuration and state data of each YANG module that is present in the local database of the IA-station.

The identity of the datastore with which the instance data set is associated is reported as defined in IETF RFC 9195. The format of the YANG instance data set is defined in IETF RFC 9195. The file format is based on the XML encoding. It is created by applying the respective XML encoding rules for the YANG structure of all YANG modules included in the digital data sheet.

A user uses the information from the digital data sheet to understand the quantities and capabilities of an IA-station, which is required for successful offline engineering of the network.

The features of a CNC need to be available for offline and online engineering or diagnostics. For this purpose, YANG modules are used that allow structured access to the local database of the CNC according to 6.4.9.2.5.11.

Any IA-station can include a CNC entity in which case the collection of YANG modules of such IA-station includes all CNC specific YANG modules for example, the ieee802-dot1q-tsn-config-uni YANG module. Since all IA-stations meet the requirements from 5.5.4, the CNC related YANG instance data is automatically included in the digital data sheet of the IA-station that hosts the CNC as described in 6.4.9.2.

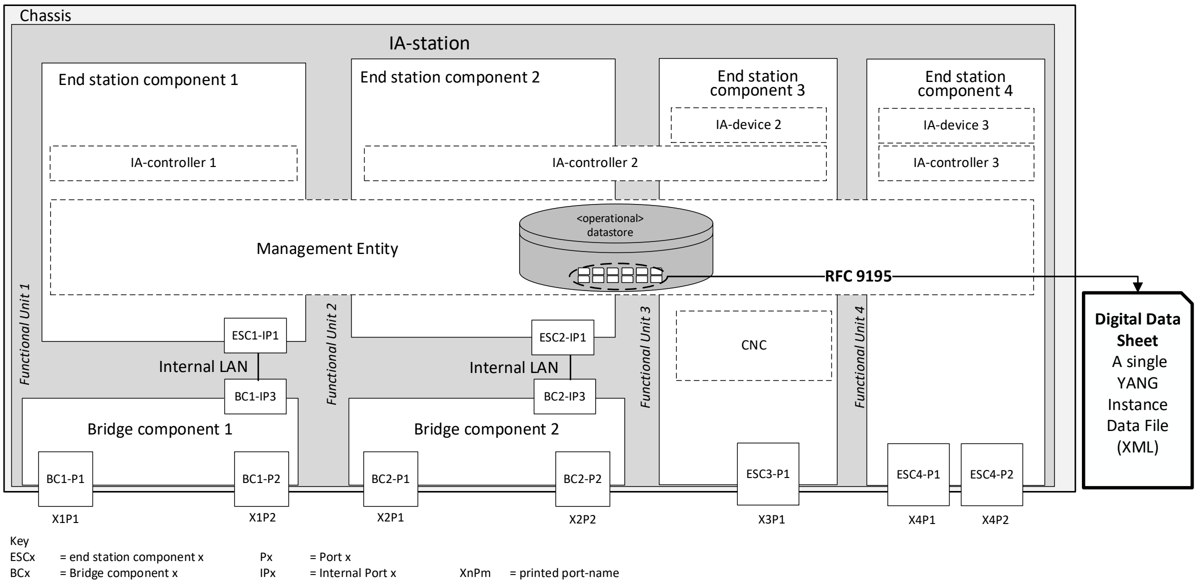


Figure 32 – Creation of the digital data sheet of an IA-station

6.4.9 YANG representation of managed objects and nodes[5][6]

6.4.9.1 General

All managed objects shall be represented in the YANG 1.1 format as described in IETF RFC 7950.

6.4.9.2 Common YANG modules, features, and nodes

6.4.9.2.1 IEEE standard for Ethernet

IA-stations shall support the ieee802-ethernet-interface YANG module according to IEEE Std 802.3.2-2019 with the following nodes:

- `[o] /ietf-interfaces/interface/ethernet/duplex`
- `[o] /ietf-interfaces/interface/ethernet/speed`
- `[o] /ietf-interfaces/interface/ethernet/flow-control/pause/direction (if the feature "ethernet-pause" is supported))`

6.4.9.2.2 Station and media access control connectivity discovery

IA-stations shall support the following nodes from the ieee802-dot1ab-lldp YANG module according to IEEE Std 802.1ABcu-2021 with values and value ranges according to 6.5.

- `[o] /ieee802-dot1ab-lldp/lldp/message-fast-tx`
- `[o] /ieee802-dot1ab-lldp/lldp/message-tx-hold-multiplier`
- `[o] /ieee802-dot1ab-lldp/lldp/message-tx-interval`
- `[o] /ieee802-dot1ab-lldp/lldp/reinit-delay`
- `[o] /ieee802-dot1ab-lldp/lldp/tx-credit-max`

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

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

- [o] /ieee802-dot1ab-lldp/lldp/tx-fast-init
- [o] /ieee802-dot1ab-lldp/lldp/notification-interval
- /ieee802-dot1ab-lldp/lldp/remote-statistics
- [m] /ieee802-dot1ab-lldp/lldp/local-system-data
- /ieee802-dot1ab-lldp/lldp/port
- [o] /ieee802-dot1ab-lldp/lldp/remote-statistics/last-change-time
- [o] /ieee802-dot1ab-lldp/lldp/remote-statistics/remote-inserts
- [o] /ieee802-dot1ab-lldp/lldp/remote-statistics/remote-deletes
- [o] /ieee802-dot1ab-lldp/lldp/remote-statistics/remote-drops
- [o] /ieee802-dot1ab-lldp/lldp/remote-statistics/remote-ageouts
- [o] /ieee802-dot1ab-lldp/lldp/local-system-data/chassis-id-subtype
- [o] /ieee802-dot1ab-lldp/lldp/local-system-data/chassis-id
- [o] /ieee802-dot1ab-lldp/lldp/local-system-data/system-name
- [o] /ieee802-dot1ab-lldp/lldp/local-system-data/system-description
- [m] /ieee802-dot1ab-lldp/lldp/local-system-data/system-capabilities-supported
- [o] /ieee802-dot1ab-lldp/lldp/local-system-data/system-capabilities-enabled
- [o] /ieee802-dot1ab-lldp/lldp/port/name
- [o] /ieee802-dot1ab-lldp/lldp/port/dest-mac-address
- [o] /ieee802-dot1ab-lldp/lldp/port/admin-status
- [o] /ieee802-dot1ab-lldp/lldp/port/notification-enable
- [o] /ieee802-dot1ab-lldp/lldp/port/tlvs-tx-enable
- [o] /ieee802-dot1ab-lldp/lldp/port/message-fast-tx
- [o] /ieee802-dot1ab-lldp/lldp/port/message-tx-hold-multiplier
- [o] /ieee802-dot1ab-lldp/lldp/port/message-tx-interval
- [o] /ieee802-dot1ab-lldp/lldp/port/reinit-delay
- [o] /ieee802-dot1ab-lldp/lldp/port/tx-credit-max
- [o] /ieee802-dot1ab-lldp/lldp/port/tx-fast-init
- [o] /ieee802-dot1ab-lldp/lldp/port/notification-interval
- [o] /ieee802-dot1ab-lldp/lldp/port/management-address-tx-port
- [o] /ieee802-dot1ab-lldp/lldp/port/port-id-subtype
- [o] /ieee802-dot1ab-lldp/lldp/port/port-id
- [o] /ieee802-dot1ab-lldp/lldp/port/port-desc
- [o] /ieee802-dot1ab-lldp/lldp/port/remote-systems-data

6.4.9.2.3 Synchronization

6.4.9.2.3.1 Timesync

IA-stations shall support the ieee1588-ptp YANG module according to IEEE P1588e with the following features:

- cmlds (Common Mean Link Delay Service)

- external-port-config

IA-stations shall support the ieee1588-ptp YANG module according to IEEE P1588e with the following nodes:

- `[o] /ieee1588-ptp/ptp/instances/instance/instance-index`
- `[o] /ieee1588-ptp/ptp/instances/instance/default-ds/clock-identity`
- `[o] /ieee1588-ptp/ptp/instances/instance/default-ds/number-ports`
- `[o] /ieee1588-ptp/ptp/instances/instance/default-ds/priority1`
- `[o] /ieee1588-ptp/ptp/instances/instance/default-ds/domain-number`
- `[o] /ieee1588-ptp/ptp/instances/instance/default-ds/slave-only`
- `[o] /ieee1588-ptp/ptp/instances/instance/default-ds/sdo-id`
- `[o] /ieee1588-ptp/ptp/instances/instance/default-ds/instance-enable`
- `[o] /ieee1588-ptp/ptp/instances/instance/default-ds/external-port-config-enable`
- `[o] /ieee1588-ptp/ptp/instances/instance/default-ds/instance-type`
- `[o] /ieee1588-ptp/ptp/instances/instance/description-ds/user-description`
- `[o] /ieee1588-ptp/ptp/instances/ports/port/port-index`
- `[o] /ieee1588-ptp/ptp/instances/ports/port/underlying-interface`
- `[o] /ieee1588-ptp/ptp/instances/ports/port/port-ds/port-state`
- `[o] /ieee1588-ptp/ptp/instances/ports/port/port-ds/delay-mechanism`
- `[o] /ieee1588-ptp/ptp/instances/ports/port/port-ds/port-enable`
- `[o] /ieee1588-ptp/ptp/instances/ports/port/external-port-config-port-ds/desired-state`
- `[o] /ieee1588-ptp/ptp/common-services/cmlds/default-ds/clock-identity`
- `[o] /ieee1588-ptp/ptp/common-services/cmlds/default-ds/number-link-ports`
- `[o] /ieee1588-ptp/ptp/common-services/cmlds/ports/port/port-index`
- `[o] /ieee1588-ptp/ptp/common-services/cmlds/ports/port/underlying-interface`
- `[o] /ieee1588-ptp/ptp/common-services/cmlds/ports/port/link-port-ds/port-identity/clock-identity`
- `[o] /ieee1588-ptp/ptp/common-services/cmlds/ports/port/link-port-ds/port-identity/port-number`
- `[o] /ieee1588-ptp/ptp/common-services/cmlds/ports/port/link-port-ds/domain-number`
- `[o] /ieee1588-ptp/ptp/common-services/cmlds/ports/port/link-port-ds/service-measurement-valid`
- `[o] /ieee1588-ptp/ptp/common-services/cmlds/ports/port/link-port-ds/mean-link-delay`
- `[o] /ieee1588-ptp/ptp/common-services/cmlds/ports/port/link-port-ds/scaled-neighbor-rate-ratio`
- `[o] /ieee1588-ptp/ptp/common-services/cmlds/ports/port/link-port-ds/log-min-pdelay-req-interval`

- [m] /ieee1588-ptp/ptp/common-services/cmlds/ports/port/link-port-ds/version-number
- [m] /ieee1588-ptp/ptp/common-services/cmlds/ports/port/link-port-ds/minor-version-number
- [o] /ieee1588-ptp/ptp/common-services/cmlds/ports/port/link-port-ds/delay-asymetry

6.4.9.2.3.2 Timesync (draft ieee802-dot1as-ptp)

IA-stations shall support the ieee802-dot1as-ptp YANG module according to IEEE P802.1ASdn with the following nodes:

- [o] /ieee802-dot1as-ptp/ptp/instances/instance/default-ds/gm-capable
- [o] /ieee802-dot1as-ptp/ptp/instances/instance/default-ds/current-utc-offset-valid
- [o] /ieee802-dot1as-ptp/ptp/instances/instance/default-ds/ptp-timescale
- [o] /ieee802-dot1as-ptp/ptp/instances/ports/port/port-ds/sync-receipt-timeout
- [o] /ieee802-dot1as-ptp/ptp/instances/ports/port/port-ds/current-one-step-tx-oper
- [o] /ieee802-dot1as-ptp/ptp/instances/ports/port/port-ds/use-mgt-one-step-tx-oper
- [o] /ieee802-dot1as-ptp/ptp/instances/ports/port/port-ds/mgt-one-step-tx-oper
- [o] /ieee802-dot1as-ptp/ptp/instances/ports/port/port-ds/sync-locked
- [o] /ieee1588-ptp/ptp/common-services/cmlds/ports/port/link-port-ds/cmlds-link-port-enabled
- [o] /ieee1588-ptp/ptp/common-services/cmlds/ports/port/link-port-ds/is-measuring-delay
- [o] /ieee1588-ptp/ptp/common-services/cmlds/ports/port/link-port-ds/as-capable-across-domains
- [o] /ieee1588-ptp/ptp/common-services/cmlds/ports/port/link-port-ds/mean-link-delay-thresh
- [o] /ieee1588-ptp/ptp/common-services/cmlds/ports/port/link-port-ds/current-log-pdelay-req-interval
- [o] /ieee1588-ptp/ptp/common-services/cmlds/ports/port/link-port-ds/use-mgt-log-pdelay-req-interval
- [o] /ieee1588-ptp/ptp/common-services/cmlds/ports/port/link-port-ds/mgt-log-pdelay-req-interval
- [o] /ieee1588-ptp/ptp/common-services/cmlds/ports/port/link-port-ds/current-compute-rate-ratio
- [o] /ieee1588-ptp/ptp/common-services/cmlds/ports/port/link-port-ds/use-mgt-compute-rate-ratio
- [o] /ieee1588-ptp/ptp/common-services/cmlds/ports/port/link-port-ds/mgt-compute-rate-ratio
- [o] /ieee1588-ptp/ptp/common-services/cmlds/ports/port/link-port-ds/current-compute-mean-link-delay
- [o] /ieee1588-ptp/ptp/common-services/cmlds/ports/port/link-port-ds/use-mgt-compute-mean-link-delay

- [o] /ieee1588-ptp/ptp/common-services/cmlds/ports/port/link-port-ds/mgt-compute-mean-link-delay
- [o] /ieee1588-ptp/ptp/common-services/cmlds/ports/port/link-port-ds/allowed-lost-responses
- [o] /ieee1588-ptp/ptp/common-services/cmlds/ports/port/link-port-ds/allowed-faults

6.4.9.2.3.3 Timesync (ieee802-dot1as-hs)

IA-stations shall support the ieee802-dot1as-ptp YANG module according to IEEE P802.1ASdn with the following nodes:

- [o] /ieee1588-ptp/ptp/instances/instance/ptp-instance-sync-ds/ptp-instance-state

6.4.9.2.4 Security configuration modules

6.4.9.2.4.1 YANG module for a keystore

IA-stations shall support the ietf-keystore YANG module according to draft-ietf-netconf-keystore-2x with the following features:

- central-keystore-supported
- asymmetric-keys

IA-stations shall support the ietf-keystore YANG module according to draft-ietf-netconf-keystore-2x with the following nodes:

- [o] /ietf-keystore/keystore/asymmetric-keys/asymmetric-key/name
- [o] /ietf-keystore/keystore/asymmetric-keys/asymmetric-key/public-key-format
- [o] /ietf-keystore/keystore/asymmetric-keys/asymmetric-key/public-key
- [o] /ietf-keystore/keystore/asymmetric-keys/asymmetric-key/private-key-format
- [o] /ietf-keystore/keystore/asymmetric-keys/asymmetric-key/hidden-private-key
- [o] /ietf-keystore/certificates/certificate/name
- [o] /ietf-keystore/certificates/certificate/cert-data
- [o] /ietf-keystore/certificates/certificate/expiration-date
- [o] /ietf-keystore/certificates/certificate/csr-info
- [o] /ietf-keystore/certificates/certificate/certificate-signing-request

6.4.9.2.4.2 Network configuration access control

IA-stations shall support the ietf-netconf-acm YANG module according to IETF RFC 8341 with the following nodes:

- [o] /ietf-netconf-acm/nacm/enable-nacm
- [o] /ietf-netconf-acm/nacm/read-default
- [o] /ietf-netconf-acm/nacm/write-default

- [o] /ietf-netconf-acm/nacm/exec-default
- [o] /ietf-netconf-acm/nacm/enable-external-groups
- [o] /ietf-netconf-acm/nacm/groups
- [o] /ietf-netconf-acm/nacm/rule-list

6.4.9.2.4.3 A YANG data module for a truststore

IA-stations shall support the ietf-truststore YANG module according to draft-ietf-netconf-trust-anchors-2x with the following features:

- central-keystore-supported
- certificates

IA-stations shall support the ietf-truststore YANG module according to draft-ietf-netconf-trust-anchors-12x with the following nodes:

- [o] /ietf-truststore/truststore/certificate-bags/certificate-bag/name
- [o] /ietf-truststore/truststore/certificate-bags/certificate-bag/certificate/name
- [o] /ietf-truststore/truststore/certificate-bags/certificate-bag/certificate/cert-data
- [o] /ietf-truststore/truststore/certificate-bags/certificate-bag/certificate/expiration-date

6.4.9.2.5 IA-station management

6.4.9.2.5.1 System capabilities

IA-stations shall support the ietf-system-capabilities YANG module according to IETF RFC 9196 with the following nodes:

- [m] /ietf-system-capabilities/datastore-capabilities/datastore
- [m] /ietf-system-capabilities/datastore-capabilities/per-node-capabilities
- [m] /ietf-system-capabilities/subscription-capabilities/on-change-supported

6.4.9.2.5.2 YANG library

IA-stations shall support the ietf-yang-library YANG module according to IETF RFC 8525 with the following nodes:

- [m] /ietf-yang-library/yang-library/module-set [list]
- [m] /ietf-yang-library/yang-library/schema [list]
- [m] /ietf-yang-library/yang-library/datastore [list]
- [m] /ietf-yang-library/yang-library/content-id

6.4.9.2.5.3 YANG push

IA-stations shall support the ietf-yang-push YANG module according to IETF RFC 8641 with the on-change feature.

IA-stations shall support the ietf-yang-push YANG module according to IETF RFC 8641 with the following nodes:

- [o] /ietf-subscribed-notifications/filters/selection-filter
- [o] /ietf-subscribed-notifications/subscription/target/datastore
- [o] /ietf-subscribed-notifications/subscription/update-trigger

6.4.9.2.5.4 YANG notification capabilities

IA-stations shall support the ietf-notification-capabilities YANG module according to IETF RFC 9196 with the following nodes:

- [m] /ietf-notification-capabilities/system-capabilities/subscription-capabilities
- [m] /ietf-notification-capabilities/system-capabilities/datastore-capabilities/per-node-capabilities/subscription-capabilities

6.4.9.2.5.5 YANG notifications

IA-stations shall support the ietf-subscribed-notifications YANG module according to IETF RFC 8639 with the following features:

- configured
- encode-xml
- subtree

IA-stations shall support the ietf-subscribed-notifications YANG module according to IETF RFC 8639 with the following nodes:

- [o] /ietf-subscribed-notifications/streams/stream/name
- [o] /ietf-subscribed-notifications/streams/stream/description
- [o] /ietf-subscribed-notifications/streams/stream/replay-support
- [o] /ietf-subscribed-notifications/streams/stream/replay-log-creation-time
- [o] /ietf-subscribed-notifications/streams/stream/replay-log-aged-time
- [o] /ietf-subscribed-notifications/filters/stream-filter/name
- [o] /ietf-subscribed-notifications/filters/stream-filter/filter-spec
- [o] /ietf-subscribed-notifications/subscriptions/subscription/id
- [o] /ietf-subscribed-notifications/subscriptions/subscription/target
- [o] /ietf-subscribed-notifications/subscriptions/subscription/stop-time
- [o] /ietf-subscribed-notifications/subscriptions/subscription/dscp
- [o] /ietf-subscribed-notifications/subscriptions/subscription/weighting
- [o] /ietf-subscribed-notifications/subscriptions/subscription/dependency
- [o] /ietf-subscribed-notifications/subscriptions/subscription/transport
- [o] /ietf-subscribed-notifications/subscriptions/subscription/encoding

- [o] /ietf-subscribed-notifications/subscriptions/subscription/purpose
- [o] /ietf-subscribed-notifications/subscriptions/subscription/notification-message-origin
- [o] /ietf-subscribed-notifications/subscriptions/subscription/configured-subscription-state
- [o] /ietf-subscribed-notifications/subscriptions/subscription/receivers

6.4.9.2.5.6 NETCONF monitoring

IA-stations shall support the ietf-netconf-monitoring YANG module according to IETF RFC 6022 with the following nodes:

- [m] /ietf-netconf-monitoring/netconf-state/capabilities
- [m] /ietf-netconf-monitoring/netconf-state/datastores
- [m] /ietf-netconf-monitoring/netconf-state/schemas

6.4.9.2.5.7 System management

IA-stations shall support the ietf-system YANG module according to IETF RFC 7317 with the following nodes:

- [o] /ietf-system/system/contact
- [o] /ietf-system/system/hostname
- [o] /ietf-system/system/location

6.4.9.2.5.8 Hardware management

IA-stations shall support the ietf-hardware YANG module according to IETF RFC 8348 with the following nodes:

- [m] /ietf-hardware/component/name
- [m] /ietf-hardware/component/class
- [m] /ietf-hardware/component/description
- [m] /ietf-hardware/component/hardware-rev
- [m] /ietf-hardware/component/software-rev
- [o] /ietf-hardware/component/serial-num
- [m] /ietf-hardware/component/mfg-name
- [m] /ietf-hardware/component/model-name

An IA-station shall provide exactly one /ietf-hardware/component with class “chassis” and may provide further components with other classes.

The following nodes of the “chassis” component shall be used for verifiable IA-station identity (see 6.3.3.2.2):

- /ietf-hardware/component/description
- /ietf-hardware/component/hardware-rev

- `/ietf-hardware/component/serial-num`
- `/ietf-hardware/component/mfg-name`
- `/ietf-hardware/component/model-name`

6.4.9.2.5.9 Interface management

IA-stations shall support the ietf-interfaces YANG module according to IETF RFC 8343 with the following nodes:

- `[m] /ietf-interfaces/interface/name`
- `[m] /ietf-interfaces/interface/description`
- `[m] /ietf-interfaces/interface/type`
- `[o] /ietf-interfaces/interface/enabled`
- `[o] /ietf-interfaces/interface/oper-status`
- `[o] /ietf-interfaces/interface/phys-address`
- `[o] /ietf-interfaces/interface/higher-layer-if`
- `[o] /ietf-interfaces/interface/lower-layer-if`
- `[o] /ietf-interfaces/interface/speed`
- `[o] /ietf-interfaces/interface/statistics/discontinuity-time`
- `[o] /ietf-interfaces/interface/statistics/in-octets`
- `[o] /ietf-interfaces/interface/statistics/in-discards`
- `[o] /ietf-interfaces/interface/statistics/in-errors`
- `[o] /ietf-interfaces/interface/statistics/out-octets`
- `[o] /ietf-interfaces/interface/statistics/out-discards`
- `[o] /ietf-interfaces/interface/statistics/out-errors`

6.4.9.2.5.10 Bridge component

IA-stations shall support the ieee802-dot1q-bridge YANG module according to IEEE Std 802.1Q-2022-2018, Clause 48, as amended by IEEE P802.1Qcw with the following feature: ingress-filtering.

IA-stations shall support the ieee802-dot1q-bridge YANG module according to IEEE Std 802.1Q-2022-2018, Clause 48, as amended by IEEE P802.1Qcw with the following nodes:

- `[m] /ietf-interfaces/interfaces/interface/bridge-port/bridge-name`
- `[m] /ietf-interfaces/interfaces/interface/bridge-port/component-name`
- `[m] /ietf-interfaces/interfaces/interface/bridge-port/port-type`
- `[o] /ietf-interfaces/interfaces/interface/bridge-port/pvid`
- `[o] /ietf-interfaces/interfaces/interface/bridge-port/default-priority`
- `[m] /ietf-interfaces/interfaces/interface/bridge-port/traffic-class`
- `[o] /ietf-interfaces/interfaces/interface/bridge-port/statistics`
- `[m] /ieee802-dot1q-bridge/bridges/bridge/name`
- `[o] /ieee802-dot1q-bridge/bridges/bridge/address`

- [m] /ieee802-dot1q-bridge/bridges/bridge/bridge-type
- [m] /ieee802-dot1q-bridge/bridges/bridge/ports
- [m] /ieee802-dot1q-bridge/bridges/bridge/components
- [m] /ieee802-dot1q-bridge/bridges/bridge/component/name
- [o] /ieee802-dot1q-bridge/bridges/bridge/component/id
- [m] /ieee802-dot1q-bridge/bridges/bridge/component/type
- [o] /ieee802-dot1q-bridge/bridges/bridge/component/traffic-class-enabled
- [m] /ieee802-dot1q-bridge/bridges/bridge/component/ports
- [o] /ieee802-dot1q-bridge/bridges/bridge/component/bridge-port
- [m] /ieee802-dot1q-bridge/bridges/bridge/component/capabilities
- [o] /ieee802-dot1q-bridge/bridges/bridge/component/bridge-mst
- [m] /ieee802-dot1q-bridge/bridges/bridge/component/capabilities/extended-filtering
- [m] /ieee802-dot1q-bridge/bridges/bridge/component/capabilities/traffic-classes
- [m] /ieee802-dot1q-bridge/bridges/bridge/component/capabilities/static-entry-individual-port
- [m] /ieee802-dot1q-bridge/bridges/bridge/component/capabilities/ivl-capable
- [m] /ieee802-dot1q-bridge/bridges/bridge/component/capabilities/svl-capable
- [m] /ieee802-dot1q-bridge/bridges/bridge/component/capabilities/hybrid-capable
- [m] /ieee802-dot1q-bridge/bridges/bridge/component/capabilities/configurable-pvid-tagging
- [m] /ieee802-dot1q-bridge/bridges/bridge/component/capabilities/local-vlan-capable
- [o] /ieee802-dot1q-bridge/bridges/bridge/component/filtering-database/aging-time
- [m] /ieee802-dot1q-bridge/bridges/bridge/component/filtering-database/size
- [o] /ieee802-dot1q-bridge/bridges/bridge/component/filtering-database/static-entries
- [o] /ieee802-dot1q-bridge/bridges/bridge/component/filtering-database/dynamic-entries
- [o] /ieee802-dot1q-bridge/bridges/bridge/component/filtering-database/static-vlan-registration-entries
- [o] /ieee802-dot1q-bridge/bridges/bridge/component/filtering-database/dynamic-vlan-registration-entries
- [o] /ieee802-dot1q-bridge/bridges/bridge/component/filtering-database/mac-address-registration-entries
- [o] /ieee802-dot1q-bridge/bridges/bridge/component/filtering-database/filtering-entry

- [o] /ieee802-dot1q-bridge/bridges/bridge/component/filtering-database/vlan-registration-entry
- [m] /ieee802-dot1q-bridge/bridges/bridge/component/permanent-database/size
- [o] /ieee802-dot1q-bridge/bridges/bridge/component/permanent-database/static-entries
- [o] /ieee802-dot1q-bridge/bridges/bridge/component/permanent-database/static-vlan-registration-entries
- [o] /ieee802-dot1q-bridge/bridges/bridge/component/permanent-database/filtering-entry
- [m] /ieee802-dot1q-bridge/bridges/bridge/component/bridge-vlan/version
- [m] /ieee802-dot1q-bridge/bridges/bridge/component/bridge-vlan/max-vids
- [o] /ieee802-dot1q-bridge/bridges/bridge/component/bridge-vlan/override-default-pvid
- [m] /ieee802-dot1q-bridge/bridges/bridge/component/bridge-vlan/max-msti
- [o] /ieee802-dot1q-bridge/bridges/bridge/component/bridge-vlan/vlan
- [o] /ieee802-dot1q-bridge/bridges/bridge/component/bridge-vlan/vid-to-fid-allocation
- [o] /ieee802-dot1q-bridge/bridges/bridge/component/bridge-vlan/fid-to-vid-allocation
- [o] /ieee802-dot1q-bridge/bridges/bridge/component/bridge-vlan/vid-to-fid
- [o] /ieee802-dot1q-bridge/bridges/bridge/component/bridge-mst/mstid
- [o] /ieee802-dot1q-bridge/bridges/bridge/component/bridge-mst/fid-to-mstid
- [o] /ieee802-dot1q-bridge/bridges/bridge/component/bridge-mst/fid-to-mstid-allocation

6.4.9.2.5.11 IEC/IEEE 60802 YANG module

IA-stations shall support the iecieee60802-ethernet-interface YANG module according to this document with the following nodes:

- [m] /iecieee60802-ethernet-interface/interfaces/interface/ethernet/supported-mau-types/mau-type
- [m] /iecieee60802-ethernet-interface/interfaces/interface/ethernet/supported-mau-types/preemption-supported

IA-stations shall support the iecieee60802-bridge YANG module according to this document with the following nodes:

- [m] /iecieee60802-bridge/interfaces/interface/bridge-port/min-interpacket-gap
- [m] /iecieee60802-bridge /interfaces/interface/bridge-port/max-burst-frames

- [m] /iecieee60802-bridge /interfaces/interface/bridge-port/max-burst-bytes
- [m] /iecieee60802-bridge /interfaces/interface/bridge-port/committed-data-rates
- [m] /iecieee60802-bridge /interfaces/interface/bridge-port/transmission-selection-algorithm
- [m] /iecieee60802/interfaces/interface/bridge-port/supported-resource-pools
- [m] /iecieee60802-bridge /bridges/bridge/component/frer-supported
- [m] /iecieee60802-bridge /bridges/bridge/component/max-redundant-streams
- [m] /iecieee60802-bridge /bridges/bridge/component/max-fids
- [m] /iecieee60802-bridge /bridges/bridge/component/max-fdb-entries
- [m] /iecieee60802-bridge /bridges/bridge/component/delay-variance
- [m] /iecieee60802-bridge /bridges/bridge/component/max-ptp-instances
- [m] /iecieee60802-bridge /bridges/bridge/component/max-hot-standby-systems

[m] /iecieee60802-bridge /bridges/bridge/component/clock

6.4.9.2.5.12 NETCONF server

IA-stations shall support the ietf-netconf-server YANG module according to draft-ietf-netconf-client-server with the following features:

- tls-call-home
- central-netconf-server-supported

IA-stations shall support the ietf-netconf-server YANG module according to draft-ietf-netconf-client-server with the following nodes:

- [o] /ietf-netconf-server/netconf-server/listen/idle-timeout
- [o] /ietf-netconf-server/netconf-server/listen/endpoint/name
- [o] /ietf-netconf-server/netconf-server/listen/endpoint/transport/tls/netconf-server-parameters
- [o] /ietf-netconf-server/netconf-server/listen/endpoint/transport/tls/tls-server-parameters
- [o] /ietf-netconf-server/netconf-server/call-home/netconf-client/name
- [o] /ietf-netconf-server/netconf-server/call-home/netconf-client/endpoints/endpoint/name
- [o] /ietf-netconf-server/netconf-server/call-home/netconf-client/endpoints/endpoint/transport/tls/netconf-server-parameters
- [o] /ietf-netconf-server/netconf-server/call-home/netconf-client/endpoints/endpoint/transport/tls/tls-server-parameters

6.4.9.2.5.13 Subscribed Notifications

IA-stations shall support the ietf-subscribed-notifications YANG module according to RFC 8639 with the following nodes:

- [o] /ietf-subscribed-notifications/streams/stream/name
- [o] /ietf-subscribed-notifications/streams/stream/description
- [o] /ietf-subscribed-notifications/filters/stream-filter/name
- [o] /ietf-subscribed-notifications/filters/stream-filter/filter-spec
- [o] /ietf-subscribed-notifications/subscriptions/subscription/id
- [o] /ietf-subscribed-notifications/subscriptions/subscription/targe
- [o] /ietf-subscribed-notifications/subscriptions/subscription/receivers

IA-stations shall support the iecieee60802-subscribed-notifications YANG module according to this document with the following nodes:

- [m] /iecieee60802-subscribed-notifications/subscriptions/max-subscriptions
- [m] /iecieee60802-subscribed-notifications/subscriptions/max-on-change-subscription-leaves
- [m] /iecieee60802-subscribed-notifications/subscriptions/max-periodic-subscription-leaves
- [m] /iecieee60802-subscribed-notifications/subscriptions/max-periodic-subscription-interval

6.4.9.2.5.14 Per Stream Filtering and Policing

IA-stations shall support the ieee802-dot1q-psfp-bridge YANG module according to 802.1Qcw with the following nodes:

- [o] /ieee802-dot1q-psfp-bridge/flow-meters/flow-meter-instance-table/flow-meter-instance-id
- [o] /ieee802-dot1q-psfp-bridge/flow-meters/flow-meter-instance-table/committed-information-rate
- [o] /ieee802-dot1q-psfp-bridge/flow-meters/flow-meter-instance-table/committed-burst-size
- [o] /ieee802-dot1q-psfp-bridge/flow-meters/flow-meter-instance-table/excess-information-rate
- [o] /ieee802-dot1q-psfp-bridge/flow-meters/flow-meter-instance-table/excess-burst-size
- [o] /ieee802-dot1q-psfp-bridge/flow-meters/flow-meter-instance-table/coupling-flag
- [o] /ieee802-dot1q-psfp-bridge/flow-meters/flow-meter-instance-table/color-mode
- [o] /ieee802-dot1q-psfp-bridge/flow-meters/flow-meter-instance-table/drop-on-yellow

6.4.9.2.6 YANG Module for IA station capabilities

IA-stations shall support the iecieee60802-ia-station YANG module according to this document with the following nodes:

- [o] ietf-interfaces/interface/bridge-port/gate-parameter-table/queue-max-sdu-table

- [o] ietf-interfaces/interface/bridge-port/gate-parameter-table/gate-enabled
- [o] ietf-interfaces/interface/bridge-port/gate-parameter-table/admin-gate-states
- [o] ietf-interfaces/interface/bridge-port/gate-parameter-table/oper-gate-states
- [o] ietf-interfaces/interface/bridge-port/gate-parameter-table/admin-control-list
- [o] ietf-interfaces/interface/bridge-port/gate-parameter-table/oper-control-list
- [o] ietf-interfaces/interface/bridge-port/gate-parameter-table/admin-cycle-time
- [o] ietf-interfaces/interface/bridge-port/gate-parameter-table/oper-cycle-time
- [o] ietf-interfaces/interface/bridge-port/gate-parameter-table/admin-cycle-time-extension
- [o] ietf-interfaces/interface/bridge-port/gate-parameter-table/oper-cycle-time-extension
- [o] ietf-interfaces/interface/bridge-port/gate-parameter-table/admin-base-time
- [o] ietf-interfaces/interface/bridge-port/gate-parameter-table/oper-base-time
- [o] ietf-interfaces/interface/bridge-port/gate-parameter-table/config-change
- [o] ietf-interfaces/interface/bridge-port/gate-parameter-table/config-change-time
- [o] ietf-interfaces/interface/bridge-port/gate-parameter-table/tick-granularity
- [o] ietf-interfaces/interface/bridge-port/gate-parameter-table/current-time
- [o] ietf-interfaces/interface/bridge-port/gate-parameter-table/config-pending
- [o] ietf-interfaces/interface/bridge-port/gate-parameter-table/config-change-error
- [c] ietf-interfaces/interface/bridge-port/gate-parameter-table/supported-list-max
- [c] ietf-interfaces/interface/bridge-port/gate-parameter-table/supported-cycle-max
- [c] ietf-interfaces/interface/bridge-port/gate-parameter-table/supported-interval-max

6.4.9.3 Optional YANG data models, features, and nodes

6.4.9.3.1 General

The following YANG modules, features and leaves shall be supported by IA-stations if the functionality they describe is included.

6.4.9.3.2 Scheduled traffic

IA-stations supporting the enhancements for scheduled traffic shall support the ieee802-dot1q-sched YANG module according to IEEE P802.1Qcw with the following feature: scheduled-traffic.

IA-stations supporting the enhancements for scheduled traffic shall support the ieee802-dot1q-sched YANG module according to IEEE P802.1Qcw with the following nodes:

- `[o] ietf-interfaces/interface/bridge-port/gate-parameter-table/queue-max-sdu-table`
- `[o] ietf-interfaces/interface/bridge-port/gate-parameter-table/gate-enabled`
- `[o] ietf-interfaces/interface/bridge-port/gate-parameter-table/admin-gate-states`
- `[o] ietf-interfaces/interface/bridge-port/gate-parameter-table/oper-gate-states`
- `[o] ietf-interfaces/interface/bridge-port/gate-parameter-table/admin-control-list`
- `[o] ietf-interfaces/interface/bridge-port/gate-parameter-table/oper-control-list`
- `[o] ietf-interfaces/interface/bridge-port/gate-parameter-table/admin-cycle-time`
- `[o] ietf-interfaces/interface/bridge-port/gate-parameter-table/oper-cycle-time`
- `[o] ietf-interfaces/interface/bridge-port/gate-parameter-table/admin-cycle-time-extension`
- `[o] ietf-interfaces/interface/bridge-port/gate-parameter-table/oper-cycle-time-extension`
- `[o] ietf-interfaces/interface/bridge-port/gate-parameter-table/admin-base-time`
- `[o] ietf-interfaces/interface/bridge-port/gate-parameter-table/oper-base-time`
- `[o] ietf-interfaces/interface/bridge-port/gate-parameter-table/config-change`
- `[o] ietf-interfaces/interface/bridge-port/gate-parameter-table/config-change-time`
- `[o] ietf-interfaces/interface/bridge-port/gate-parameter-table/tick-granularity`
- `[o] ietf-interfaces/interface/bridge-port/gate-parameter-table/current-time`
- `[o] ietf-interfaces/interface/bridge-port/gate-parameter-table/config-pending`
- `[o] ietf-interfaces/interface/bridge-port/gate-parameter-table/config-change-error`
- `[c] ietf-interfaces/interface/bridge-port/gate-parameter-table/supported-list-max`
- `[c] ietf-interfaces/interface/bridge-port/gate-parameter-table/supported-cycle-max`

- [c] ietf-interfaces/interface/bridge-port/gate-parameter-table/supported-interval-max

6.4.9.3.3 IEC/IEEE 60802 YANG modules

IA-stations that support enhancements for scheduled traffic shall support the iecieee60802-sched-bridge YANG module according to this document with the following nodes:

- [c] /iecieee60802-sched-bridge/interfaces/interface/bridge-port/gate-parameter-table/min-gating-times

6.4.9.3.4 Frame preemption

IA-stations supporting frame preemption according to IEEE Std 802.1Q-2022, 5.4.1 ad), shall support the ieee802-dot1q-preemption YANG module according to IEEE P802.1Qcw with the following feature: frame-preemption.

IA-stations supporting frame preemption according to IEEE Std 802.1Q-2022, 5.4.1 ad), shall support the ieee802-dot1q-preemption YANG module according to IEEE P802.1Qcw with the following nodes:

- [o] /ietf-interfaces/interface/bridge-port/frame-preemption-parameters/frame-preemption-status-table
- [o] /ietf-interfaces/interface/bridge-port/frame-preemption-parameters/preemption-active

6.4.9.3.5 Credit-based shaper

IA-stations supporting the credit-based shaper according to IEEE Std 8021.Q-2022, 8.6.8.2, shall support the <ieee-cbs> YANG module according to IEEE P802.1Qdx.

6.4.9.3.6 FRER

IA-stations supporting FRER according to 5.10.1 item d) or item e), shall support the ieee802-dot1cb-stream-identification and ieee802-dot1cb-frer YANG modules according to IEEE 802.1CBcv-2021 with the following nodes:

- [o] /ieee802-dot1cb-stream-identification/stream-identity/index
- [o] /ieee802-dot1cb-stream-identification/stream-identity/handle
- [o] /ieee802 dot1cb stream identification/stream identity/out facing/input-port
- [o] /ieee802 dot1cb stream identification/stream identity/out facing/output-port
- [o] /ieee802 dot1cb stream identification/stream identity/parameters/null-stream-identification
- [o] /ieee802-dot1cb-frer/frer/sequence-generation/index
- [o] /ieee802-dot1cb-frer/frer/sequence-generation/stream
- [o] /ieee802-dot1cb-frer/frer/sequence-generation/direction-out-facing
- [o] /ieee802-dot1cb-frer/frer/sequence-recovery/index
- [o] /ieee802-dot1cb-frer/frer/sequence-recovery/stream

- `[o] /ieee802-dot1cb-frer/frer/sequence-recovery/port`
- `[o] /ieee802-dot1cb-frer/frer/sequence-recovery/direction-out-facing`
- `[o] /ieee802-dot1cb-frer/frer/sequence-recovery/algorithm/vector`
- `[o] /ieee802-dot1cb-frer/frer/sequence-identification/port`
- `[o] /ieee802-dot1cb-frer/frer/sequence-identification/direction-out-facing`
- `[o] /ieee802-dot1cb-frer/frer/sequence-identification/stream`
- `[o] /ieee802-dot1cb-frer/frer/sequence-identification/encapsulation/r-tag`
- `[o] /ieee802-dot1cb-frer/frer/stream-split`

6.4.9.4 CUC/CNC YANG

6.4.9.4.1 NETCONF Client

IA-stations with CNC and/or CUC functionality shall support the ietf-netconf-client YANG module according to draft-ietf-netconf-client-server with the following features:

- tls-initiate,
- tls-listen,
- central-netconf-client-supported.

6.4.9.4.1 YANG Module for TSN UNI

IA-stations with CNC and/or CUC functionality shall support the ieee802-dot1q-tsn-config-uni YANG module according to P802.1Qdj with the following node: `[o] /ieee802-dot1q-tsn-config/tsn-uni`.

6.4.10 YANG Data Model[7][8]

6.4.10.1 General

Subclause 6.4.10 specifies the YANG data model for IA-Stations. YANG (IETF RFC 7950) is a data modeling language used to model configuration data and state data for remote network management protocols. The selected YANG-based remote network management protocol is NETCONF (IETF RFC 6241). A YANG module specifies the organization and rules for the 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., IA-Stations).

6.4.10.2 YANG framework

The core of the YANG module for IEC/IEEE 60802 IA Stations consists of YANG “augment” statements, used to add members to the tree of existing YANG modules plus one new module for IEC/IEEE 60802 specific objects.

[7] Copyright release for YANG: Users of this document may freely reproduce the YANG modules contained in this document 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 document and can also be obtained from the IEEE 802.1 Website at https://1.ieee802.org/yang-modules/.

6.4.10.3 IEC/IEEE 60802 Specific Managed Objects

6.4.10.3.1 General

Subclause 6.4.10.3 defines the set of managed objects, and their functionality, that provides additional information about an IA-station that is required by a CNC to calculate network configurations.

IEC/IEEE 60802 specific managed objects are defined:

- per ethernet interface, i.e., external port, in 6.4.10.3.2,
- per end station component internal or external port in 6.4.10.3.3 and 6.4.10.3.4,
- per bridge component internal or external port in 6.4.10.3.4,
- per end station component in 6.4.10.3.5 and 6.4.10.3.7,
- per bridge component in 6.4.10.3.6 and 6.4.10.3.7, and
- per IA-station in 6.4.10.3.8.

IEC/IEEE 60802 specific managed objects for CNC entities are defined in 6.4.10.3.9.

6.4.10.3.2 IEC/IEEE 60802 managed objects per ethernet interface

6.4.10.3.2.1 supportedMauTypes

The list of supported MAU Types including the data:

a) mauType

The value is the supported Mau Type derived from the list position of the corresponding dot3MauType as listed in IETF RFC 4836, Clause 5.

b) preemptionSupported

The Boolean value indicates if preemption is supported by the MAU Type.

NOTE The operational MAU Type of an ethernet interface is provided as leaf operational-mau-type of the ieee802-ethernet-lldp YANG module. The operational MAU Type is included in the supportedMauTypes list.

6.4.10.3.3 IEC/IEEE 60802 managed objects per end station component port

6.4.10.3.3.1 minInterpacketGap

The value is the minimum gap in bits between two consecutive frames.

6.4.10.3.3.2 maxBurstFrames

The value is the maximum number of frames per gating cycle.

6.4.10.3.3.3 maxBurstBytes

The value is the maximum number of octets per gating cycle.

6.4.10.3.3.4 committedDataRates

The list of committed data rates per traffic class and supported line speed including the data:

a) committedInformationRate

The value is the bandwidth limit in kbit/s.

b) committedBurstSize

The value is the burst size limit in bytes.

6.4.10.3.4 IEC/IEEE 60802 managed objects per bridge or end station component port

6.4.10.3.4.1 transmissionSelectionAlgorithm

The list of supported transmission selection algorithms according to IEEE Std 802.1Q 8.6.8 per traffic class.

6.4.10.3.4.2 supportedResourcePools

The list of supported buffer resource pools including the data:

a) resourcePoolName

The value is the name of a resource pool.

b) coveredTimeInterval

The value specifies the covered buffering time given as rational number of seconds for the highest supported link speed.

c) resourcePoolTrafficClasses

The list of the traffic classes to be served by the resource pool.

6.4.10.3.4.3 minGatingTimes

The list of minimum gating times per supported line speed including the data:

a) minCycleTime

The value is the minimum value supported by this port of the AdminCycleTime and OperCycleTime parameters given as rational number of seconds.

b) minIntervalTime

The value is the minimum value supported by this port of the TimeIntervalValue parameter in nanoseconds.

6.4.10.3.5 IEC/IEEE 60802 managed objects per end station component.

6.4.10.3.5.1 frerSupported

The value indicates if FRER is supported.

6.4.10.3.5.2 maxRedundantStreams

The value is the maximum number of supported redundant streams.

6.4.10.3.6 IEC/IEEE 60802 managed objects per bridge component.

6.4.10.3.6.1 delayVariance

The value indicates if the bridge component is as single chip architecture (i.e., without countable internal communication delay times) or cascaded chip architecture (i.e., with countable internal communication delay times).

6.4.10.3.6.2 delayTimes

The list of minimum and maximum frame length independent and frame length dependent delay time values of frames as they pass through a bridge component. These values are given:

- per supported MAU Type pair and traffic class, if delayVariance is singleValue, or
- per port pair with supported MAU Types and traffic class, if delayVariance is multipleValues.

The list includes the data:

a) independentDelayMin

The value is the minimum delay portion that is independent of frame length according to IEEE 802.1Q-2022, 12.32.1.1.

b) independentDelayMax

The value is the maximum delay portion that is independent of frame length according to IEEE 802.1Q-2022, 12.32.1.1.

c) dependentDelayMin

The value is the minimum delay portion that is dependent on frame length according to IEEE 802.1Q-2022, 12.32.1.2.

d) dependentDelayMax

The value is the maximum delay portion that is dependent on frame length according to IEEE 802.1Q-2022, 12.32.1.2.

6.4.10.3.7 IEC/IEEE 60802 managed objects per bridge or end station component

6.4.10.3.7.1 maxFids

The value is the maximum number of supported FDBs.

6.4.10.3.7.2 maxFdbEntries

The list of the maximum number of static (6.4.10.3.7.3) and dynamic (6.4.10.3.7.4) FDB entries per FDB.

6.4.10.3.7.3 maxStaticFdbEntries

The value is the maximum number of static FDB entries.

6.4.10.3.7.4 maxDynamicFdbEntries

The value is the maximum number of dynamic FDB entries.

6.4.10.3.7.5 maxPtpInstances

The value is the maximum number of supported PTP Instances.

6.4.10.3.7.6 maxHotStandbySystems

The value is the maximum number of supported hot standby systems.

6.4.10.3.7.7 clockList

The list of supported application clock entities including the data:

a) clockIdentity

The clock identity of the application clock.

b) clockTarget

The Boolean value indicates if the application clock is a clock target (TRUE) or clock source (FALSE).

c) arbSupported

The Boolean value indicates if the application clock supports the ARB timescale.

d) ptpSupported

The Boolean value indicates if the application clock supports the PTP timescale.

e) hotStandbySupported

The Boolean value indicates if the application clock supports the hot standby.

f) attachedPtpInstance

The value is a reference to the PTP or hot standby Instance, that is attached to the application clock.

g) isSynced

The Boolean value indicates if the application clock is either synchronized to the attached PTP Instance (TRUE) or to an internal/external ClockSource (FALSE).

6.4.10.3.8 IEC/IEEE 60802 managed objects per IA-station

6.4.10.3.8.1 maxSubscriptions

The value is the maximum number of supported NETCONF Server subscriptions.

6.4.10.3.8.2 maxOnChangeSubscriptionLeaves

The value is the maximum number of supported leaves for NETCONF Server on-change subscriptions according to IETF RFC 8641.

6.4.10.3.8.3 maxPeriodicSubscriptionLeaves

The value is the maximum number of supported leaves for NETCONF Server periodic subscriptions according to IETF RFC 8641.

6.4.10.3.8.4 minPeriodicSubscriptionInterval

The value is the minimum periodic subscription interval in centiseconds (0.01 seconds) for NETCONF Server periodic subscriptions according to IETF RFC 8641.

6.4.10.3.8.5 capabilityLLDP

This Boolean value indicates if LLDP is supported.

6.4.10.3.8.6 capabilityTimesync

This Boolean value indicates if Timesync is supported.

6.4.10.3.8.7 capabilityKeystore

This Boolean value indicates if Keystore is supported.

6.4.10.3.8.8 capabilityNACM

This Boolean value indicates if NACM is supported.

6.4.10.3.8.9 capabilityTruststore

This Boolean value indicates if Truststore is supported.

6.4.10.3.8.10 capabilityYangLibrary

This Boolean value indicates if YANG library is supported.

6.4.10.3.8.11 capabilityYangPush

This Boolean value indicates if Yang Push is supported.

6.4.10.3.8.12 capabilityYangNotifications

This Boolean value indicates if YANG notifications is supported.

6.4.10.3.8.13 capabilityNetcofMonitoring

This Boolean value indicates if NETCONF Monitoring is supported.

6.4.10.3.8.14 capabilityNetconfClient

This Boolean value indicates if NETCONF client is supported.

6.4.10.3.8.15 capabilityTsnUni

This Boolean value indicates if TSN Uni is supported.

6.4.10.3.8.16 capabilitySchedTraffic

This Boolean value indicates if scheduled traffic is supported.

6.4.10.3.8.17 capabilityFramePreemption

This Boolean value indicates if frame preemption is supported.

6.4.10.3.9 IEC/IEEE 60802 managed objects for CNC entities

6.4.10.3.9.1 maxConfigurationDomains

The value is the maximum number of supported configuration domains.

6.4.10.3.9.2 maxCUCs

The value is the maximum number of supported CUC entities.

6.4.10.3.9.3 maxIAstations

The value is the maximum number of supported IA-stations.

6.4.10.3.9.4 maxNetworkDiameter

The value is the maximum supported network diameter.

6.4.10.3.9.5 maxStreams

The value is the maximum number of supported streams.

6.4.10.3.9.6 maxNumSeamlessTrees

The value is the maximum number of trees supported for seamless redundancy of a stream.

6.4.10.3.9.7 hotStandbySupported

The Boolean value indicates if PTP hot standby is supported.

6.4.10.4 IEC/IEEE 60802 Specific RPCs and Actions

6.4.10.4.1 RPC iecieee60802-factory-default

This RPC is similar to the RPC factory-default which is defined in RFC 8808 with the following description: “The server resets all datastores to their factory default contents and any nonvolatile storage back to factory condition, deleting all dynamically generated files, including those containing keys, certificates, logs, and other temporary files.

Depending on the factory default configuration, after being reset, the device may become unreachable on the network.”

In contrast to the original factory-reset RPC in RFC 8808, this RPC puts the device into a state where a subsequent configuration by a CNC component results in a functioning IEC/IEEE 60802 IA-station.

6.4.10.4.1.1 Input

None.

6.4.10.4.1.2 Output

None.

6.4.10.4.2 Action add-streams

6.4.10.4.2.1 General

This Action requests a CNC to add a list of streams.

6.4.10.4.2.2 Input

a) CucId - The ID of the CUC for which the streams are to be added.

b) StreamId - The Stream ID is a unique identifier of a Stream request and corresponding configuration.

c) Container Talker - The Talker container contains:

- Talker’s behavior for Stream (how/when transmitted),
- Talker’s requirements from the network, and
- TSN capabilities of the Talker’s interface(s).

d) List Listener - Each Listener list entry contains:
   - Listener’s requirements from the network, and
   - TSN capabilities of the Listener’s interface(s).

6.4.10.4.2.3 Output

Result - Status information indicating if Stream addition has been successful.

6.4.10.4.3 Action remove-listener

6.4.10.4.3.1 General

This Action removes listeners from a stream.

6.4.10.4.3.2 Input

List Listener - A list of indices of listeners to be removed from a stream.

6.4.10.4.3.3 Output

Result - Status information indicating if Stream addition has been successful.

6.4.10.5 IEC/IEEE 60802 YANG data models

A UML representation is used to provide an overview of the hierarchy of the IEC/IEEE 60802 YANG data model.

A UML-like representation of the management model is provided in Figure 33 through Figure 37. The purpose of a UML-like diagram is to express the model design in a concise manner. The structure of the UML-like representation shows the name of the object followed by a list of properties for the object. The properties indicate their type and accessibility. It should be noted that the UML-like representation is meant to express simplified semantics for the properties. It is not meant to provide the specific datatype used to encode the object in either MIB or YANG. In the UML-like representation, a box with a white background represents information that comes from sources outside of this document. A box with a gray background represents objects that are defined by this document.

NOTE OMG UML 2.5 [B49] conventions together with C++ language constructs are used as a representation to convey model structure and relationships.

For all UML figures, data that is imported from original modules is shown in white, and data in augments of IEC/IEEE 60802 is shown in grey.

Figure 33 through Figure 37 provide an overview of the IEC/IEEE 60802 augmentations.

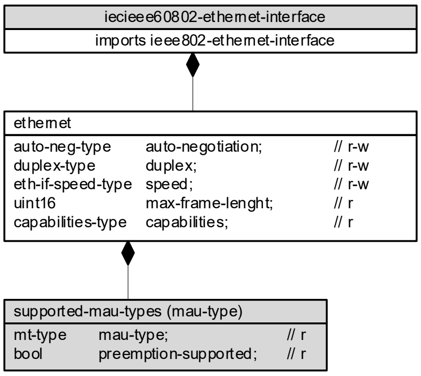

iecieee60802-ethernet-interface
imports ieee802-ethernet-interface
ethernet
auto-neg-type auto-negotiation; // r-w
duplex-type duplex; // r-w
eth-if-speed-type speed; // r-w
uint16 max-frame-lenght; // r
capabilities-type capabilities; // r
supported-mau-types (mau-type)
mt-type mau-type; // r
bool preemption-supported; // r

Figure 33 – Module iecieee60802-ethernet-interface

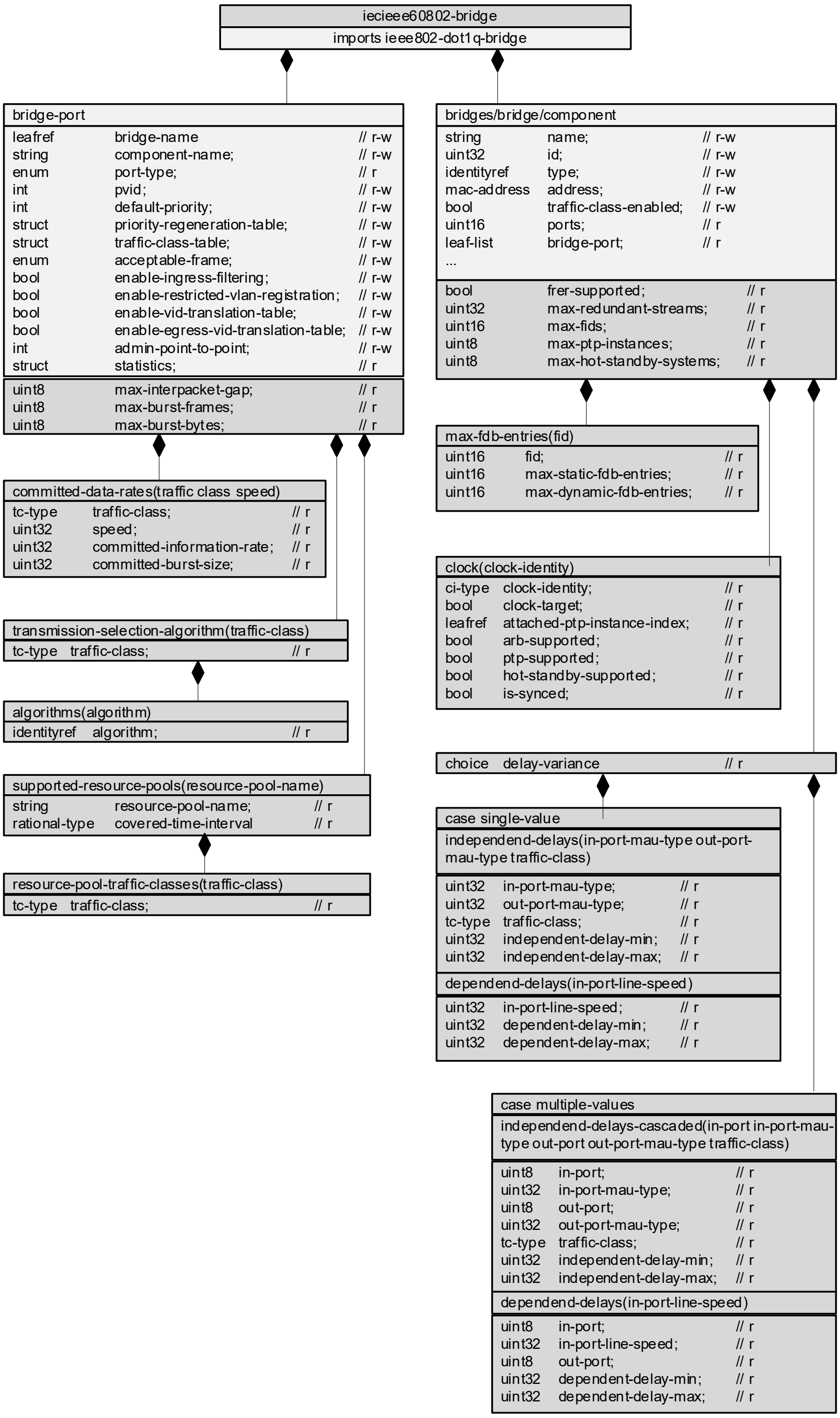

iecieee60802-bridge
imports ieee802-dot1q-bridge
bridge-port
leafref bridge-name // r-w
string component-name; // r-w
enum port-type; // r
int pvid; // r-w
int default-priority; // r-w
struct priority-regeneration-table; // r-w
struct traffic-class-table; // r-w
enum acceptable-frame; // r-w
bool enable-ingress-filtering; // r-w
bool enable-restricted-vlan-registration; // r-w
bool enable-vid-translation-table; // r-w
bool enable-egress-vid-translation-table; // r-w
int admin-point-to-point; // r-w
struct statistics; // r
uint8 max-interpacket-gap; // r
uint8 max-burst-frames; // r
uint8 max-burst-bytes; // r
committed-data-rates(traffic class speed)
tc-type traffic-class; // r
uint32 speed; // r
uint32 committed-information-rate; // r
uint32 committed-burst-size; // r
transmission-selection-algorithm(traffic-class)
tc-type traffic-class; // r
algorithms(algorithm)
identityref algorithm; // r
supported-resource-pools(resource-pool-name)
string resource-pool-name; // r
rational-type covered-time-interval // r
resource-pool-traffic-classes(traffic-class)
tc-type traffic-class; // r
bridges/bridge/component
string name; // r-w
uint32 id; // r-w
identityref type; // r-w
mac-address address; // r-w
bool traffic-class-enabled; // r-w
uint16 ports; // r
leaf-list bridge-port; // r
...
bool frer-supported; // r
uint32 max-redundant-streams; // r
uint16 max-fids; // r
uint8 max-ptp-instances; // r
uint8 max-hot-standby-systems; // r
max-fdb-entries(fid)
uint16 fid; // r
uint16 max-static-fdb-entries; // r
uint16 max-dynamic-fdb-entries; // r
clock(clock-identity)
ci-type clock-identity; // r
bool clock-target; // r
leafref attached-ptp-instance-index; // r
bool arb-supported; // r
bool ptp-supported; // r
bool hot-standby-supported; // r
bool is-synced; // r
choice delay-variance // r
case single-value
independend-delays(in-port-mau-type out-port-mau-type traffic-class)
uint32 in-port-mau-type; // r
uint32 out-port-mau-type; // r
tc-type traffic-class; // r
uint32 independent-delay-min; // r
uint32 independent-delay-max; // r
dependend-delays(in-port-line-speed)
uint32 in-port-line-speed; // r
uint32 dependent-delay-min; // r
uint32 dependent-delay-max; // r
case multiple-values
independend-delays-cascaded(in-port in-port-mau-type out-port out-port-mau-type traffic-class)
uint8 in-port; // r
uint32 in-port-mau-type; // r
uint8 out-port; // r
uint32 out-port-mau-type; // r
tc-type traffic-class; // r
uint32 independent-delay-min; // r
uint32 independent-delay-max; // r
dependend-delays(in-port-line-speed)
uint8 in-port; // r
uint32 in-port-line-speed; // r
uint8 out-port; // r
uint32 dependent-delay-min; // r
uint32 dependent-delay-max; // r

Figure 34 – Module iecieee60802-bridge

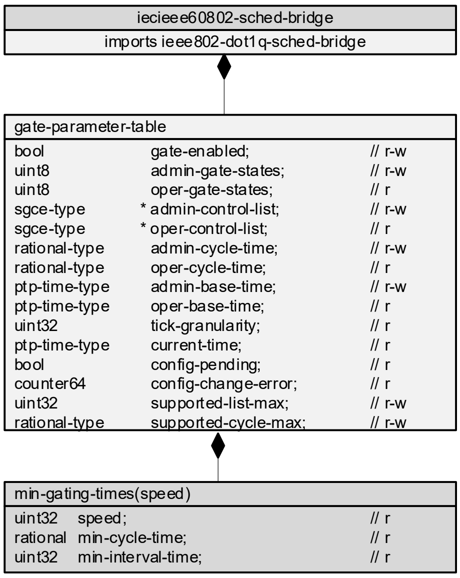

iecieee60802-sched-bridge
imports ieee802-dot1q-sched-bridge
gate-parameter-table
bool gate-enabled; // r-w
uint8 admin-gate-states; // r-w
uint8 oper-gate-states; // r
sgce-type * admin-control-list; // r-w
sgce-type * oper-control-list; // r
rational-type admin-cycle-time; // r-w
rational-type oper-cycle-time; // r
ptp-time-type admin-base-time; // r-w
ptp-time-type oper-base-time; // r
uint32 tick-granularity; // r
ptp-time-type current-time; // r
bool config-pending; // r
counter64 config-change-error; // r
uint32 supported-list-max; // r-w
rational-type supported-cycle-max; // r-w
min-gating-times(speed)
uint32 speed; // r
rational min-cycle-time; // r
uint32 min-interval-time; // r

Figure 35 – Module iecieee60802-dot1-sched-bridge

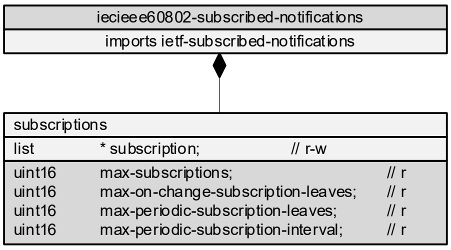

iecieee60802-subscribed-notifications
imports ietf-subscribed-notifications
subscriptions
list * subscription; // r-w
uint16 max-subscriptions; // r
uint16 max-on-change-subscription-leaves; // r
uint16 max-periodic-subscription-leaves; // r
uint16 max-periodic-subscription-interval; // r

Figure 36 – Module iecieee60802-subscribed-notifications

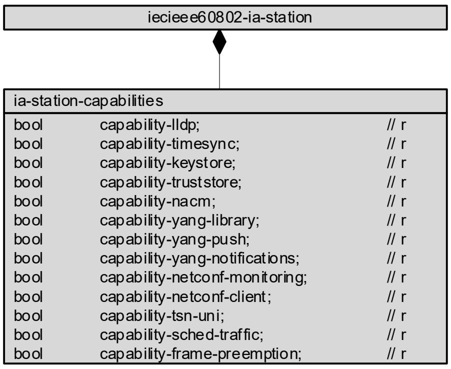

iecieee60802-ia-station
ia-station-capabilities
bool capability-lldp; // r
bool capability-timesync; // r
bool capability-keystore; // r
bool capability-truststore; // r
bool capability-nacm; // r
bool capability-yang-library; // r
bool capability-yang-push; // r
bool capability-yang-notifications; // r
bool capability-netconf-monitoring; // r
bool capability-netconf-client; // r
bool capability-tsn-uni; // r
bool capability-sched-traffic; // r
bool capability-frame-preemption; // r

Figure 37 – Module iecieee60802-ia-station

6.4.10.6 Structure of IEC/IEEE 60802 YANG data models

The YANG data models specified by this standard use the YANG modules are summarized in Table 16.

In the YANG module definitions, if any discrepancy between the “description” text and the corresponding definition in any other part of this standard occur, the definitions outside Clause 6 take precedence.

Table 16 – Summary of the YANG modules

Module	Description
ieee802-ethernet-interface	This module contains YANG definitions for configuring IEEE Std 802.3 Ethernet Interfaces.
ietf-interfaces	This module contains a collection of YANG definitions for managing network interfaces.
iecieee60802-ethernet-interface	This module augments ieee802-ethernet-interface.
ieee802-types	This module contains a collection of generally useful derived data types for IEEE YANG data models.
ieee802-dot1q-bridge	This module describes the bridge configuration model for IEEE 802.1Q Bridges.
ieee802-dot1q-types	This module contains common types used within dot1Q-bridge modules.
iecieee60802-bridge	This module augments ieee802-dot1q-bridge.
ieee802-dot1q-sched-bridge	This module provides for management of IEEE Std 802.1Q Bridges that support Scheduled Traffic Enhancements.
iecieee60802-dot1q-sched-bridge	This module augments ieee802-dot1q-sched-bridge.
ieee802-dot1cb-frer	This module provides management objects that control the frame replication and elimination from IEEE Std 802.1CB-2017.
ieee1588-ptp	This module defines a data model for the configuration and state of IEEE Std 1588 clocks.
ietf-netconf-acm	This module provides management for the Network Configuration Access Control Model.
ieee802-dot1q-tsn-config-uni	This module provides the Time-Sensitive Networking (TSN) User/Network Interface (UNI) for the exchange of information between CUC and CNC that are required to configure TSN Streams in a TSN network.
iecieee60802-tsn-config-uni	This module augments ieee802-dot1q-tsn-config-uni.
iecieee60802-ia-station	This module provides read-only information about the capabilities and RPCs for IEC/IEEE 60802 IA-stations.
Ietf-subscribed-notifications	This module defines a YANG data model for subscribing to event records and receiving matching content in notification messages.
Iecieee60802-subscribed-notifications	This module augments ietf-subscribed-notifications.

6.4.10.7 YANG schema tree definitions

6.4.10.7.1 General

The schema tree is provided as an overview of the YANG modules. The symbols and their meaning are specified in YANG Tree Diagrams (IETF RFC 8340).

6.4.10.7.2 Module iecieee60802-ethernet-interface

```
module: iecieee60802-ethernet-interface

  augment /if:interfaces/if:interface/eth-if:ethernet:
    +--ro supported-mau-types* [mau-type]
       +--ro mau-type                uint32
       +--ro preemption-supported?   Boolean
```

6.4.10.7.3 Module iecieee60802-bridge

```
module: iecieee60802-bridge
```

```
  augment /if:interfaces/if:interface/bridge:bridge-port:
    +--ro min-interpacket-gap?                uint8
    +--ro max-burst-frames?                   uint8
    +--ro max-burst-bytes?                    uint8
    +--ro committed-data-rates* [traffic-class speed]
    |  +--ro traffic-class                 dot1q-types:traffic-class-type
    |  +--ro speed                         uint32
    |  +--ro committed-information-rate?   uint32
    |  +--ro committed-burst-size?         uint32
    +--ro transmission-selection-algorithm* [traffic-class]
    |  +--ro traffic-class    dot1q-types:traffic-class-type
    |  +--ro algorithms* [algorithm]
    |     +--ro algorithm    identityref
    +--ro supported-resource-pools* [resource-pool-name]
       +--ro resource-pool-name               string
       +--ro covered-time-interval
       |  +---u ieee802:rational-grouping
       +--ro resource-pool-traffic-classes* [traffic-class]
          +--ro traffic-class    dot1q-types:traffic-class-type
  augment /bridge:bridges/bridge:bridge/bridge:component:
    +--ro frer-supported?                      boolean
    +--ro max-redundant-streams?               uint32
    +--ro max-fids?                            uint16
    +--ro max-fdb-entries* [fid]
    |  +--ro fid                        uint16
    |  +--ro max-static-fdb-entries?    uint16
    |  +--ro max-dynamic-fdb-entries?   uint16
    +--ro (delay-variance)?
    |  +--:(single-value)
    |  |  +--ro independent-delays* [in-port-mau-type out-port-mau-type
traffic-class]
    |  |  |  +--ro in-port-mau-type         uint32
    |  |  |  +--ro out-port-mau-type        uint32
    |  |  |  +--ro traffic-class            dot1q-types:traffic-class-type
    |  |  |  +--ro independent-delay-min?   uint32
    |  |  |  +--ro independent-delay-max?   uint32
    |  |  +--ro dependent-delays* [in-port-line-speed]
    |  |     +--ro in-port-line-speed     uint32
    |  |     +--ro dependent-delay-min?   uint32
    |  |     +--ro dependent-delay-max?   uint32
    |  +--:(multiple-values)
    |     +--ro independent-delays-cascaded* [in-port in-port-mau-type out-
port out-port-mau-type traffic-class]
    |     |  +--ro in-port                  uint8
    |     |  +--ro in-port-mau-type         uint32
    |     |  +--ro out-port                 uint8
    |     |  +--ro out-port-mau-type        uint32
    |     |  +--ro traffic-class            dot1q-types:traffic-class-type
    |     |  +--ro independent-delay-min?   uint32
    |     |  +--ro independent-delay-max?   uint32
    |     +--ro dependent-delays-cascaded* [in-port in-port-line-speed out-
port]
    |        +--ro in-port                uint8
    |        +--ro in-port-line-speed     uint32
    |        +--ro out-port               uint8
    |        +--ro dependent-delay-min?   uint32
    |        +--ro dependent-delay-max?   uint32
    +--ro max-ptp-instances?                   uint8
    +--ro max-hot-standby-systems?             uint8
    +--ro clock* [clock-identity]
       +--ro clock-identity                 ptp:clock-identity
       +--ro clock-target?                  boolean
       +--ro attached-ptp-instance-index?   ->
/ptp:ptp/instances/instance/instance-index
       +--ro arb-supported?                 boolean
       +--ro ptp-supported?                 boolean
       +--ro hot-standby-supported?         boolean
       +--ro is-synced?                     Boolean
```

6.4.10.7.4 Module iecieee60802-sched-bridge

```
module: iecieee60802-sched-bridge

  augment /if:interfaces/if:interface/bridge:bridge-port/sched-bridge:gate-
parameter-table:
    +--ro min-gating-times* [speed]
       +--ro speed                uint32
       +--ro min-cycle-time
       |  +---u ieee802:rational-grouping
       +--ro min-interval-time?   uint32
```

6.4.10.7.5 Module iecieee60802-tsn-config-uni

```
module: iecieee60802-tsn-config-uni

  augment /tsn:tsn-uni:
    +--ro max-config-domains?      uint8
    +--ro max-cucs?                uint8
    +--ro max-ia-stations?         uint16
    +--ro max-network-diameter?    uint8
    +--ro max-streams?             uint16
    +--ro num-seamless-trees?      uint8
    +--ro hot-standby-supported?   uint8
    +---x add_streams
       +---w input
       |  +---w cuc-id?        string
       |  +---w stream-list* [stream-id]
       |     +---w stream-id    tsn-types:stream-id-type
       |     +---w talker
       |     |  +---w tsn-types:group-talker
       |     +---w listener* [index]
       |        +---w index                       uint32
       |        +---w tsn-types:group-listener
       +--rw output
          +--rw result?   boolean
  augment /tsn:tsn-uni/tsn:domain/tsn:cuc/tsn:stream:
    +---x remove_listener
       +---w input
       |  +---w listener* [index]
       |     +---w index    uint32
       +--rw output
          +--rw result?   Boolean
```

6.4.10.7.6 Module iecieee60802-ia-station

```
module: iecieee60802-ia-station
  +--ro ia-station-capabilities
     +--ro capability-lldp?                 boolean
     +--ro capability-timesync?             boolean
     +--ro capability-keystore?             boolean
     +--ro capability-truststore?           boolean
     +--ro capability-nacm?                 boolean
     +--ro capability-yang-library?          boolean
     +--ro capability-yang-push?             boolean
     +--ro capability-yang-notifications?    boolean
     +--ro capability-netconf-monitoring?    boolean
     +--ro capability-netconf-client?        boolean
     +--ro capability-tsn-uni?               boolean
     +--ro capability-sched-traffic?         boolean
     +--ro capability-frame-preemption?      boolean

  rpcs:
    +---x ia-factory-reset
```

6.4.10.7.7 Module iecieee60802-subscribed-notifications

```
module: iecieee60802-subscribed-notifications

  augment /sn:subscriptions:
    +--ro max-subscriptions?                     uint16
    +--ro max-on-change-subscription-leaves?     uint16
    +--ro max-periodic-subscription-leaves?      uint16
    +--ro max-periodic-subscription-interval?    uint16
```

6.4.10.8 YANG modules

6.4.10.8.1 Module iecieee60802-ethernet-interface

```
module iecieee60802-ethernet-interface {
  yang-version 1.1;
  namespace "urn:ieee:std:60802:yang:iecieee60802-ethernet-interface";
  prefix ia-eth-if;

  import ieee802-ethernet-interface {
    prefix eth-if;
  }
  import ietf-interfaces {
    prefix if;
  }

  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
    "Management objects that provide information about IEC/IEEE 60802 IA-
Stations as specified in IEC/IEEE 60802.

     Copyright (C) IEC/IEEE (2023).
     This version of this YANG module is part of IEC/IEEE 60802;
     see the standard itself for full legal notices.";

  revision 2023-09-08 {
    description
      "Initial version.";
    reference
```

```
      "IEC/IEEE 60802 - YANG Data Model";
  }

  augment "/if:interfaces/if:interface/eth-if:ethernet" {
    description
      "Augment IEEE Std 802.3 ethernet.";
    list supported-mau-types {
      description
        "Contains a list of supported mau parameters.";
      key "mau-type";
      config false;
      leaf mau-type {
        type uint32;
        // the type of this leaf should be a type defined by IEEE P802.3 in
future
        config false;
        description
          "The value is the supported Mau Type derived from the list
position of the corresponding dot3MauType as listed in IETF RFC 4836, Clause
5.";
        reference
          "IEC/IEEE 60802 6.4.10.2.2.1 a)";
      }
      leaf preemption-supported {
        type boolean;
        // the type of this leaf should be a type defined by IEEE P802.3 in
future
        config false;
        description
          "The Boolean value indicates if preemption is supported by the MAU
Type.";
        reference
          "IEC/IEEE 60802 6.4.10.2.2.1 b)";
      }
    }
  }
}
```

6.4.10.8.2 Module iecieee6802-bridge

```
module iecieee60802-bridge {
  yang-version 1.1;
  namespace "urn:ieee:std:60802:yang:iecieee60802-bridge";
  prefix ia-bridge;

  import ieee802-types {
    prefix ieee802;
  }
  import ieee802-dot1q-bridge {
    prefix bridge;
  }
  import ietf-interfaces {
    prefix if;
  }
  import ieee802-dot1q-types {
    prefix dot1q-types;
  }
  import ieee1588-ptp {
    prefix ptp;
  }

  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
    "Management objects that provide information about IEC/IEEE 60802 IA-
Stations as specified in IEC/IEEE 60802.

     Copyright (C) IEC/IEEE (2023).
     This version of this YANG module is part of IEC/IEEE 60802;
     see the standard itself for full legal notices.";

  revision 2023-09-08 {
    description
      "Initial version.";
    reference
      "IEC/IEEE 60802 - YANG Data Model";
  }

  augment "/if:interfaces/if:interface/bridge:bridge-port" {
    description
      "Augment IEEE Std 802.1 bridge.";
    leaf min-interpacket-gap {
      type uint8;
      config false;
      description
        "The value is the minimum gap in bits between two consecutive
frames.";
      reference
        "IEC/IEEE 60802 6.4.10.2.3.1";
    }
    leaf max-burst-frames {
      type uint8;
      config false;
      description
        "The value is the maximum number of frames per gating cycle.";
      reference
        "IEC/IEEE 60802 6.4.10.2.3.2";
    }
    leaf max-burst-bytes {
      type uint8;
      config false;
      description
        "The value of the maximum number of octets per gating cycle.";
      reference
        "IEC/IEEE 60802 6.4.10.2.3.3";
    }
    list committed-data-rates {
      description
        "The list of committed data rates per traffic class and supported
line speed.";
      key "traffic-class speed";
      config false;
      leaf traffic-class {
        type dot1q-types:traffic-class-type;
        description
          "The traffic class of the entry (0..7).";
        reference
          "8.6.6 of IEEE Std 802.1Q";
      }
      leaf speed {
        type uint32;
        description
          "This value is the line speed in Mbps.";
      }
      leaf committed-information-rate {
        type uint32;
        config false;
        description
          "The value is the bandwidth limit in kbit/s.";
        reference
          "IEC/IEEE 60802 6.4.10.2.3.4 a)";
      }
      leaf committed-burst-size {
        type uint32;
        config false;
        description
          "The value is the burst size limit in bytes.";
        reference
          "IEC/IEEE 60802 6.4.10.2.3.4 b)";
      }
    }
    list transmission-selection-algorithm {
      description
        "The list of supported transmission selection algorithms according
to IEEE Std 802.1Q 8.6.8 per traffic class.";
      key "traffic-class";
      config false;
      leaf traffic-class {
        type dot1q-types:traffic-class-type;
        config false;
        description
          "Traffic class. (0..7)";
        reference
          "IEEE Std 802.1Q 8.6.6";
      }
      list algorithms {
        description
          "The list of supported transmission selection algorithms according
to IEEE Std 802.1Q 8.6.8 for this traffic class.";
        key "algorithm";
        config false;
        leaf algorithm {
          type identityref {
            base dot1q-types:transmission-selection-algorithm;
          }
          config false;
          description
            "Transmission selection algorithm";
          reference
            "8.6.8, Table 8-6 of IEEE Std 802.1Q";
        }
      }
    }
    list supported-resource-pools {
      description
        "The list of supported buffer resource pools.";
      key "resource-pool-name";
      config false;
      leaf resource-pool-name {
        type string;
        config false;
        description
          "The value is the name of the resource pool.";
        reference
          "6.4.10.2.4.2 a) of IEC/IEEE 60802";
      }
      container covered-time-interval {
        config false;
        uses ieee802:rational-grouping;
        description
          "The value is the covered buffering time given as rational number
of seconds for the highest supported link speed.";
        reference
          "6.4.10.2.4.2 b) of IEC/IEEE 60802";
      }
      list resource-pool-traffic-classes {
        description
          "The list of the traffic classes to be served by the resource
pool.";
        reference
          "6.4.10.2.4.2 c) of IEC/IEEE 60802";
        key "traffic-class";
        config false;
        leaf traffic-class {
          type dot1q-types:traffic-class-type;
          description
            "The traffic class of the entry.";
          reference
            "8.6.6 of IEEE Std 802.1Q";
        }
      }
    }
  }

  augment "/bridge:bridges/bridge:bridge/bridge:component" {
    description
      "Augment IEEE Std 802.1 bridge component.";
    leaf frer-supported {
      type boolean;
      config false;
      description
        "The Boolean value indicates if FRER is supported.";
      reference
        "IEC/IEEE 60802 6.4.10.2.5.1";
    }
    leaf max-redundant-streams {
      type uint32;
      config false;
      description
        "The value is the maximum number of supported redundant streams.";
      reference
        "IEC/IEEE 60802 6.4.10.2.5.2";
    }
    leaf max-fids {
      type uint16;
      config false;
      description
        "The value is the maximum number of supported FIDs.";
      reference
```

```
        "IEC/IEEE 60802 6.4.10.2.7.1";
    }
    list max-fdb-entries {
      config false;
      description
        "The list of the maximum number of static and dynamic FDB entries
per FID.";
      reference
        "IEC/IEEE 60802 6.4.10.2.7.2";
      key "fid";
      leaf fid {
        type uint16;
        config false;
        description
          "The FID number";
      }
      leaf max-static-fdb-entries {
        type uint16;
        config false;
        description
          "The value is the maximum number of static FDB entries.";
        reference
          "IEC/IEEE 60802 6.4.10.2.7.3";
      }
      leaf max-dynamic-fdb-entries {
        type uint16;
        config false;
        description
          "The value is the maximum number of dynamic FDB entries.";
        reference
          "IEC/IEEE 60802 6.4.10.2.7.4";
      }
    }
    choice delay-variance {
      config false;
      description
        "The value indicates if the bridge component is as single chip
architecture (i.e., without countable internal communication delay times) or
cascaded chip architecture (i.e., with countable internal communication
delay times).";
      reference
        "6.4.10.2.6.1 of IEC/IEEE 60802";
      case single-value {
        list independent-delays {
          description
            "The list of minimum and maximum frame length independent delay
time values of frames as they pass through a bridge component.";
          reference
            "6.4.10.2.6.2 of IEC/IEEE 60802";
          key "in-port-mau-type out-port-mau-type traffic-class";
          config false;
          leaf in-port-mau-type {
            type uint32;
            config false;
            description
              "The MAU type of the input port";
          }
          leaf out-port-mau-type {
            type uint32;
            config false;
            description
              "The MAU type of the input port";
          }
          leaf traffic-class {
            type dot1q-types:traffic-class-type;
            config false;
            description
              "The traffic class of the entry.";
            reference
              "8.6.6 of IEEE Std 802.1Q";
          }
          leaf independent-delay-min {
            type uint32;
            config false;
            description
              "The value is the minimum delay portion that is independent of
frame length according to IEEE 802.1Q-2022, 12.32.1.1.";
            reference
              "6.4.10.2.6.2 a) of IEC/IEEE 60802";
          }
          leaf independent-delay-max {
            type uint32;
            config false;
            description
              "The value is the maximum delay portion that is independent of
frame length according to IEEE 802.1Q-2022, 12.32.1.1.";
            reference
              "6.4.10.2.6.2 b) of IEC/IEEE 60802";
          }
        }
        list dependent-delays {
          description
            "The list of minimum and maximum frame length dependent delay
time values of frames as they pass through a bridge component";
          reference
            "6.4.10.2.6.2 of IEC/IEEE 60802";
          key "in-port-line-speed";
          config false;
          leaf in-port-line-speed {
            type uint32;
            config false;
            description
              "This value is the line speed in Mbps.";
          }
          leaf dependent-delay-min {
            type uint32;
            config false;
            description
              "The value is the minimum delay portion that is dependent on
frame length according to IEEE 802.1Q-2022, 12.32.1.2.";
            reference
              "6.4.10.2.6.2 c) of IEC/IEEE 60802";
          }
          leaf dependent-delay-max {
            type uint32;
            config false;
            description
              "The value is the maximum delay portion that is dependent on
frame length according to IEEE 802.1Q-2022, 12.32.1.2.";
            reference
              "6.4.10.2.6.2 d) of IEC/IEEE 60802";
          }
        }
      }
      case multiple-values {
        list independent-delays-cascaded {
          description
            "The list of minimum and maximum frame length independent delay
time values of frames as they pass through a bridge component.";
          reference
            "6.4.10.2.6.2 of IEC/IEEE 60802";
          key "in-port in-port-mau-type out-port out-port-mau-type traffic-
class";
          config false;
          leaf in-port {
            type uint8;
            config false;
            description
              "The port number of the input port";
          }
          leaf in-port-mau-type {
            type uint32;
            config false;
            description
              "The MAU type of the input port";
          }
          leaf out-port {
            type uint8;
            config false;
            description
              "The port number of the output port";
          }
          leaf out-port-mau-type {
            type uint32;
            config false;
            description
              "The MAU type of the input port";
          }
          leaf traffic-class {
            type dot1q-types:traffic-class-type;
            config false;
            description
              "The traffic class of the entry.";
            reference
              "8.6.6 of IEEE Std 802.1Q";
          }
          leaf independent-delay-min {
            type uint32;
            config false;
            description
              "The value is the minimum delay portion that is independent of
frame length according to IEEE 802.1Q-2022, 12.32.1.1.";
            reference
              "6.4.10.2.6.2 a) of IEC/IEEE 60802";
          }
          leaf independent-delay-max {
            type uint32;
            config false;
            description
              "The value is the maximum delay portion that is independent of
frame length according to IEEE 802.1Q-2022, 12.32.1.1.";
            reference
              "6.4.10.2.6.2 b) of IEC/IEEE 60802";
          }
        }
        list dependent-delays-cascaded {
          description
            "The list of minimum and maximum frame length dependent delay
time values of frames as they pass through a bridge component";
```

```
          reference
            "6.4.10.2.6.2 of IEC/IEEE 60802";
          key "in-port in-port-line-speed out-port";
          config false;
          leaf in-port {
            type uint8;
            config false;
            description
              "The port number of the input port";
          }
          leaf in-port-line-speed {
            type uint32;
            config false;
            description
              "This value is the line speed in Mbps.";
          }
          leaf out-port {
            type uint8;
            config false;
            description
              "The port number of the output port";
          }
          leaf dependent-delay-min {
            type uint32;
            config false;
            description
              "The value is the minimum delay portion that is dependent on
frame length according to IEEE 802.1Q-2022, 12.32.1.2.";
            reference
              "6.4.10.2.6.2 c) of IEC/IEEE 60802";
          }
          leaf dependent-delay-max {
            type uint32;
            config false;
            description
              "The value is the maximum delay portion that is dependent on
frame length according to IEEE 802.1Q-2022, 12.32.1.2.";
            reference
              "6.4.10.2.6.2 d) of IEC/IEEE 60802";
          }
        }
      }
    }
    leaf max-ptp-instances {
      type uint8;
      config false;
      description
        "The value is the maximum number of supported PTP Instances.";
      reference
        "IEC/IEEE 60802 6.4.10.2.7.5";
    }
    leaf max-hot-standby-systems {
      type uint8;
      config false;
      description
        "The value is the maximum number of supported hot standby systems.";
      reference
        "IEC/IEEE 60802 6.4.10.2.7.6";
    }
    list clock {
      description
        "The list of supported application clock entities.";
      reference
```

```
          "6.4.10.2.7.7 of IEC/IEEE 60802";
        key "clock-identity";
        config false;
        leaf clock-identity {
          type ptp:clock-identity;
          config false;
          description
            "The clock identity of the application clock.";
          reference
            "6.4.10.2.7.7 a) of IEC/IEEE 60802";
        }
        leaf clock-target {
          type boolean;
          config false;
          description
            "The Boolean value indicates if the application clock is a clock
target (TRUE) or clock source (FALSE).";
          reference
            "6.4.10.2.7.7 b) of IEC/IEEE 60802";
        }
        leaf attached-ptp-instance-index {
          type leafref {
            path "/ptp:ptp/ptp:instances/ptp:instance/ptp:instance-index";
          }
          config false;
          description
            "The value is a reference to the index of the PTP or hot standby
Instance, that is attached to the application clock.";
          reference
            "6.4.10.2.7.7 f) of IEC/IEEE 60802";
        }
        leaf arb-supported {
          type boolean;
          config false;
          description
            "The Boolean value indicates if the application clock supports the
ARB timescale.";
          reference
            "6.4.10.2.7.7 c) of IEC/IEEE 60802";
        }
        leaf ptp-supported {
          type boolean;
          config false;
          description
            "The Boolean value indicates if the application clock supports the
PTP timescale.";
          reference
            "6.4.10.2.7.7 d) of IEC/IEEE 60802";
        }
        leaf hot-standby-supported {
          type boolean;
          config false;
          description
            "The Boolean value indicates if the application clock supports the
hot standby.";
          reference
            "6.4.10.2.7.7 e) of IEC/IEEE 60802";
        }
        leaf is-synced {
          type boolean;
          config false;
          description
          "The Boolean value indicates if the application clock is either
synchronized to the attached PTP Instance (TRUE) or to an internal/external
ClockSource (FALSE).";
        reference
          "6.4.10.2.7.7 g) of IEC/IEEE 60802";
      }
    }
  }
}
```

6.4.10.8.3 Module iecieee60802-sched-bridge

```
module iecieee60802-sched-bridge {
  yang-version 1.1;
  namespace "urn:ieee:std:60802:yang:iecieee60802-sched-bridge";
  prefix ia-sched-bridge;

  import ieee802-types {
    prefix ieee802;
  }
  import ieee802-dot1q-bridge {
    prefix bridge;
  }
  import ieee802-dot1q-sched-bridge {
    prefix sched-bridge;
  }
  import ietf-interfaces {
    prefix if;
  }

  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
    "Management objects that provide information about IEC/IEEE 60802 IA-
Stations as specified in IEC/IEEE 60802.

     Copyright (C) IEC/IEEE (2023).
     This version of this YANG module is part of IEC/IEEE 60802;
     see the standard itself for full legal notices.";

  revision 2023-09-08 {
    description
      "Initial version.";
    reference
      "IEC/IEEE 60802 - YANG Data Model";
  }

  augment "/if:interfaces/if:interface/bridge:bridge-port/sched-bridge:gate-
parameter-table" {
    description
```

```
        "Augment IEEE Std 802.1 bridge/gate-parameter-table.";
      list min-gating-times {
        description
          "The list of minimum gating times per supported line speed.";
        reference
          "6.4.10.2.4.3 of IEC/IEEE 60802";
        key "speed";
        config false;
        leaf speed {
          type uint32;
          config false;
          description
            "This value is the line speed in Mbps.";
        }
        container min-cycle-time {
          uses ieee802:rational-grouping;
          description
            "The value is the minimum value supported by this port of the
AdminCycleTime and OperCycleTime parameters given as rational number of
seconds.";
          reference
            "6.4.10.2.4.3 a) of IEC/IEEE 60802";
        }
        leaf min-interval-time {
          type uint32;
          description
            "The value is the minimum value supported by this port of the
TimeIntervalValue parameter in nanoseconds.";
          reference
            "6.4.10.2.4.3 b) of IEC/IEEE 60802";
        }
      }
    }
}
```

6.4.10.8.4 Module iecieee60802-tsn-config-uni

```
module iecieee60802-tsn-config-uni {
  yang-version 1.1;
  namespace "urn:ieee:std:60802:yang:iecieee60802-tsn-config-uni";
  prefix ia-tsn;

  import ieee802-dot1q-tsn-config-uni {
    prefix tsn;
  }
  import ieee802-dot1q-tsn-types {
    prefix tsn-types;
  }

  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
    "Management objects that provide information about IEC/IEEE 60802 IA-
Stations as specified in IEC/IEEE 60802.

     Copyright (C) IEC/IEEE (2023).
     This version of this YANG module is part of IEC/IEEE 60802;
     see the standard itself for full legal notices.";

  revision 2023-09-08 {
    description
      "Initial version.";
    reference
      "IEC/IEEE 60802 - YANG Data Model";
  }

  augment "/tsn:tsn-uni" {
    description
      "Augment main container in tsc-config-uni.";
    leaf max-config-domains {
      type uint8;
      config false;
      description
        "The value is the maximum number of supported configuration
domains.";
      reference
        "6.4.10.2.9.1 of IEC/IEEE 60802";
    }
    leaf max-cucs {
      type uint8;
      config false;
      description
        "The value is the maximum number of supported CUC entities.";
      reference
        "6.4.10.2.9.2 of IEC/IEEE 60802";
    }
    leaf max-ia-stations {
      type uint16;
      config false;
      description
        "The value is the maximum number of supported IA-stations.";
      reference
        "6.4.10.2.9.3 of IEC/IEEE 60802";
    }
    leaf max-network-diameter {
      type uint8;
      config false;
      description
        "The value is the maximum supported network diameter.";
      reference
        "6.4.10.2.9.4 of IEC/IEEE 60802";
    }
    leaf max-streams {
      type uint16;
      config false;
      description
        "The value is the maximum number of supported streams.";
      reference
        "6.4.10.2.9.5 of IEC/IEEE 60802";
    }
    leaf num-seamless-trees {
      type uint8;
      config false;
      description
        "The value is the maximum number of trees supported for seamless
redundancy of a stream.";
      reference
        "6.4.10.2.9.6 of IEC/IEEE 60802";
    }
    leaf hot-standby-supported {
      type uint8;
      config false;
      description
        "The Boolean value indicates if PTP hot standby is supported.";
      reference
        "6.4.10.2.9.7 of IEC/IEEE 60802";
    }
    action add_streams {
      description
        "This Action requests a CNC to add a list of streams.";
      input {
        leaf cuc-id {
          type string;
          description
            "The CUC ID where the streams are to be added";
        }
        list stream-list {
          key "stream-id";
          description
            "List of Streams that should be added.";
          leaf stream-id {
            type tsn-types:stream-id-type;
            description
              "The Stream ID is a unique identifier of a Stream request
               and corresponding configuration. It is used to associate a
               CUC’s Stream request with a CNC’s corresponding response.";
          }
          container talker {
            description
              "The Talker container contains: - Talker’s behavior for
               Stream (how/when transmitted) - Talker’s requirements from
               the network - TSN capabilities of the Talker’s
               interface(s).";
            uses tsn-types:group-talker;
          }
          list listener {
            key "index";
            description
              "Each Listener list entry contains: - Listener’s
               requirements from the network - TSN capabilities of the
               Listener’s interface(s).";
            leaf index {
              type uint32;
              description
                "This index is provided in order to provide a unique key
                 per list entry.";
            }
            uses tsn-types:group-listener;
          }
        }
      }
      output {
        leaf result {
          type boolean;
          description
            "Returns status information indicating if Stream addition
```

```
             has been successful.";
        }
      }
    }
  }

  augment "/tsn:tsn-uni/tsn:domain/tsn:cuc/tsn:stream" {
    description
      "Augment stream list in tsc-config-uni.";
    action remove_listener {
      description
        "This Action removes listeners from a stream.";
      input {
        list listener {
          key "index";
          description
            "Each Listener list entry contains: - Listener’s
             requirements from the network - TSN capabilities of the
             Listener’s interface(s).";
          leaf index {
            type uint32;
            description
              "This index is provided in order to provide a unique key
               per list entry.";
          }
        }
      }
      output {
        leaf result {
          type boolean;
          description
            "Returns status information indicating if listene removal
             has been successful.";
        }
      }
    }
  }
}
```

6.4.10.8.5 Module iecieee60802-ia-station

```
module iecieee60802-ia-station {
  yang-version 1.1;
  namespace "urn:ieee:std:60802:yang:iecieee60802-ia-station";
  prefix ias;

  import ietf-datastores {
    prefix ds;
    reference
      "RFC 8342: Network Management Datastore Architecture
       (NMDA)";
  }
  import ietf-netconf-acm {
    prefix nacm;
    reference
      "RFC 8341: Network Configuration Access Control Model";
  }

  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
    "Capability information and reset to factory defaults functionality for
IEC/IEEE 60802 IA-Stations as specified in IEC/IEEE 60802 IEC/IEEE 60802.

     Copyright (C) IEC/IEEE (2023).
     This version of this YANG module is part of IEC/IEEE 60802;
     see the standard itself for full legal notices.";

  revision 2023-09-08 {
    description
      "Initial version.";
    reference
      "IEC/IEEE 60802 - YANG Data Model";
  }

  feature ia-factory-default-datastore {
    description
      "Indicates that the factory default configuration is
       available as a datastore.";
  }

  identity ia-factory-default {
    if-feature "ia-factory-default-datastore";
    base ds:datastore;
    description
      "This read-only datastore contains the factory default
       configuration for the device that will be used to replace
       the contents of the read-write conventional configuration
       datastores during a 'ia-factory-reset' RPC operation.";
  }

  container ia-station-capabilities {
    description
      "This container provides read only information about an ia-station's
capabilities.";
    reference
      "IEC/IEEE 60802 - YANG Data Model";
    config false;
    leaf capability-lldp {
      type boolean;
      config false;
      description
        "The value is true if the device supports LLDP.";
      reference
        "6.4.10.2.8.5 of IEC/IEEE 60802";
    }
    leaf capability-timesync {
      type boolean;
      config false;
      description
        "The value is true if the device supports Timesync.";
      reference
        "6.4.10.2.8.6 of IEC/IEEE 60802";
    }
```

```
      leaf capability-keystore {
        type boolean;
        config false;
        description
          "The value is true if the device supports Keystore.";
        reference
          "6.4.10.2.8.7 of IEC/IEEE 60802";
      }
      leaf capability-truststore {
        type boolean;
        config false;
        description
          "The value is true if the device supports Truststore.";
        reference
          "6.4.10.2.8.9 of IEC/IEEE 60802";
      }
      leaf capability-nacm {
        type boolean;
        config false;
        description
          "The value is true if the device supports NACM.";
        reference
          "6.4.10.2.8.8 of IEC/IEEE 60802";
      }
      leaf capability-yang-library {
        type boolean;
        config false;
        description
          "The value is true if the device supports YANG library.";
        reference
          "6.4.10.2.8.10 of IEC/IEEE 60802";
      }
      leaf capability-yang-push {
        type boolean;
        config false;
        description
          "The value is true if the device supports YANG push.";
        reference
          "6.4.10.2.8.11 of IEC/IEEE 60802";
      }
      leaf capability-yang-notifications {
        type boolean;
        config false;
        description
          "The value is true if the device supports YANG notifications.";
        reference
          "6.4.10.2.8.12 of IEC/IEEE 60802";
      }
      leaf capability-netconf-monitoring {
        type boolean;
        config false;
        description
          "The value is true if the device supports NETCONF monitoring.";
        reference
          "6.4.10.2.8.13 of IEC/IEEE 60802";
      }
      leaf capability-netconf-client {
        type boolean;
        config false;
        description
          "The value is true if the device supports NETCONF client.";
        reference
          "6.4.10.2.8.14 of IEC/IEEE 60802";
```

```
    }
    leaf capability-tsn-uni {
      type boolean;
      config false;
      description
        "The value is true if the device supports TSN uni.";
      reference
        "6.4.10.2.8.15 of IEC/IEEE 60802";
    }
    leaf capability-sched-traffic {
      type boolean;
      config false;
      description
        "The value is true if the device supports scheduled traffic.";
      reference
        "6.4.10.2.8.16 of IEC/IEEE 60802";
    }
    leaf capability-frame-preemption {
      type boolean;
      config false;
      description
        "The value is true if the device supports frame preemption.";
      reference
        "6.4.10.2.8.17 of IEC/IEEE 60802";
    }
  }

  rpc ia-factory-reset {
    nacm:default-deny-all;
    description
      "The server resets all datastores to their factory
       default contents and any nonvolatile storage back to
       factory condition, deleting all dynamically
       generated files, including those containing keys,
       certificates, logs, and other temporary files.

       Depending on the factory default configuration, after
       being reset, the device may become unreachable on the
       network.

       In contrast to the original factory-reset RPC in RFC 8808,
       this RPC puts the device into a state where a subsequent
       configuration by a CNC component results in a funcioning
       60802 IA-station";
  }
}
```

6.4.10.8.6 Module iecieee60802-subscribed-notifications

```
module iecieee60802-subscribed-notifications {
  yang-version 1.1;
  namespace "urn:ieee:std:60802:yang:iecieee60802-subscribed-notifications";
  prefix ia-sn;

  import ietf-subscribed-notifications {
    prefix sn;
  }

  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
    "Management objects that provide information about IEC/IEEE 60802 IA-
Stations as specified in IEC/IEEE 60802.

     Copyright (C) IEC/IEEE (2023).
     This version of this YANG module is part of IEC/IEEE 60802;
     see the standard itself for full legal notices.";

  revision 2023-09-08 {
    description
      "Initial version.";
    reference
      "IEC/IEEE 60802 - YANG Data Model";
  }

  augment "/sn:subscriptions" {
    description
      "Augment subscriptions in ietf-subscribed-notifications.";
    leaf max-subscriptions {
      type uint16;
      config false;
      description
        "The value is the maximum number of supported NETCONF Server
subscriptions.";
      reference
        "6.4.10.2.8.1 of IEC/IEEE 60802";
    }
    leaf max-on-change-subscription-leaves {
      type uint16;
      config false;
      description
        "The value is the maximum number of supported leaves for NETCONF
Server on-change subscriptions according to IETF RFC 8641.";
      reference
        "6.4.10.2.8.2 of IEC/IEEE 60802";
    }
    leaf max-periodic-subscription-leaves {
      type uint16;
      config false;
      description
        "The value is the maximum number of supported leaves for NETCONF
Server periodic subscriptions according to IETF RFC 8641.";
      reference
        "6.4.10.2.8.3 of IEC/IEEE 60802";
    }
    leaf max-periodic-subscription-interval {
      type uint16;
      config false;
      description
        "The value is the minimum periodic subscription interval in
centiseconds (0.01 seconds) for NETCONF Server periodic subscriptions
according to IETF RFC 8641.";
      reference
        "6.4.10.2.8.4 of IEC/IEEE 60802";
        }
    }
}
```

6.5 Topology discovery and verification

6.5.1 Topology discovery and verification requirements

Electrical engineering of machines with multiple IA-stations includes the definition of the machine internal network topology (i.e., the engineered topology).

The machine internal network topology includes type specific data of IA-stations (for example model name or manufacturer name) as well as instance specific data (for example IP addresses or DNS names).

The electrical engineering data of the network topology is used:

- During commissioning so that machine planning and installation are identical.
- By the TDE during operation to verify that the actual topology of the Configuration Domain matches the engineered topology.
- By maintenance staff during repair to easily identify failed IA-stations, ports, or links to be replaced.

Repair and replacement of an IA-station do not require verification of the updated engineered topology so that the TDE does not produce a verification error.

IA-stations do not need to be pre-configured when they are repaired or replaced. IA-stations report type and instance data as described in 6.5.3.

6.5.2 Topology discovery overview

6.5.2.1 General

LLDP enables the discovery of IA-stations, their external ports, and their external connectivity. A Topology Discovery Entity can query LLDP data by remote management to derive the physical network topology.

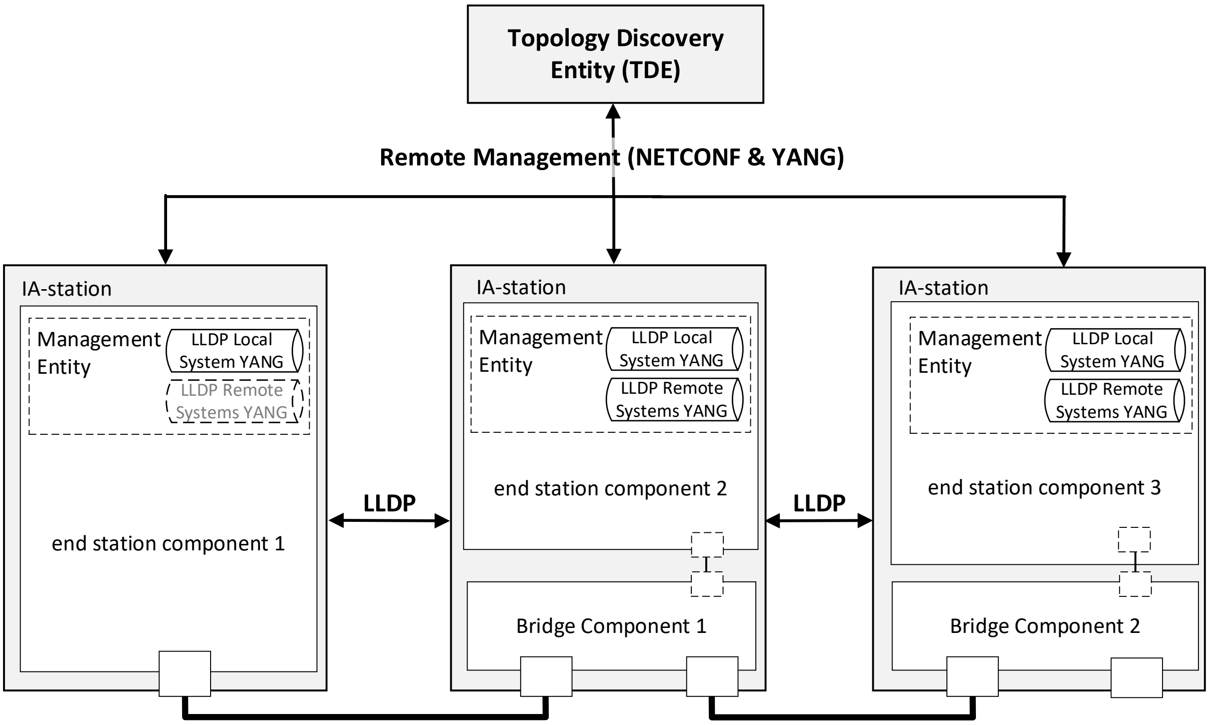


Figure 38 – Usage example of LLDP

Figure 38 illustrates a network showing the LLDP agent implementations in an IA-station consisting of a single end station component and two IA-stations with end station and Bridge components (see 4.3). The LLDP protocol is used to convey neighborhood information among peers, and NETCONF is used between the TDE and the IA-stations to query this neighborhood information from the IA-stations. This information allows the TDE to discover IA-stations and the physical network topology.

NOTE A Topology Discovery Entity (TDE) can be run from anywhere in the network with reachability to the to-be-discovered devices.

IA-stations announce themselves via LLDP to support discovery by the TDE. Announcements contain the management address (see 6.5.2.4.6) and system capabilities (see 6.5.2.4.5) for the discovery operation. The announced system capabilities information enables the TDE to identify IA-stations with multiple end station and Bridge components. The TDE can use the definitions in 6.4.3 for the discovery of the internal structure of such IA-stations.

To allow for operational behavior and exchanged information, IA-stations support the local system YANG (see 6.4.9.2.2). IA-stations that include a Bridge component additionally support the processing of received LLDP messages and support the remote systems YANG (see 6.4.9.2.2).

6.5.2.2 LLDP operational control parameters

LLDP defines several operational parameters that control the protocol behavior (see IEEE Std 802.1AB-2016, 10.5.1). These parameter definitions apply to all external ports of an IA-station.

NOTE According to IEEE Std 802.1AB-2016, 9.1.1 c), changes to the local system that impact information exchanged via LLDP immediately trigger the transmission of an LLDPDU to communicate the local changes as quickly as possible to any neighboring systems.

An IA-station shall support LLDP transmit mode (adminStatus enabledTxOnly) on an external end station component port and may support transmit and receive mode (adminStatus enabledRxTx) on that port (see IEEE Std 802.1AB-2016, 10.5.1).

An IA-station shall support LLDP transmit and receive mode (adminStatus enabledRxTx) on an external Bridge component port (see IEEE Std 802.1AB-2016, 10.5.1).

6.5.2.3 LLDPDU transmission, reception, and addressing

The destination address to be used for LLDPDU transmission (dest-mac-address) shall be the nearest bridge group MAC address, i.e., 01-80-C2-00-00-0E, on all ports to limit the scope of LLDPDU propagation to a single physical link (see IEEE Std 802.1AB-2016, 7.1 item a).

NOTE IEEE Std 802.1AB-2016 defines LLDPDUs to be transmitted untagged, i.e., frames do not carry priority information for traffic class selection. At the same time, IEEE Std 802.1AB-2016 neither specifies a well-defined device-internal priority nor management capabilities for the configuration of the traffic class to be used for the transmission of LLDPDUs. It is the user's responsibility to prevent LLDPDUs from interfering with the transmission of time-critical control data.

6.5.2.4 LLDP TLV selection

6.5.2.4.1 General

An IA-station transmitting LLDPDUs shall include the LLDP TLVs selected in 6.5.2.4 and may include additional TLVs (tlvs-tx-enable). An IA-station receiving LLDPDUs shall process LLDPDUs.

Each LLDPDU shall contain the following LLDP TLVs specified in IEEE Std 802.1AB-2016, 8.5:

- Exactly one Chassis ID TLV according to 6.5.2.4.2,
- Exactly one Port ID TLV according to 6.5.2.4.3,
- Exactly one Time To Live TLV according to 6.5.2.4.4,
- Exactly one System Capabilities TLV according to 6.5.2.4.5, and
- One or more Management Address TLVs according to 6.5.2.4.6.

NOTE The concatenation of the Chassis ID and Port ID fields enables the recipient of an LLDPDU to identify the sending LLDP agent/port.

6.5.2.4.2 Chassis ID TLV

The Chassis ID field shall contain the same value for all transmitted LLDPDUs independent from the transmitting port of the IA-station, i.e., be a non-volatile identifier which is unique within the context of the administrative domain.

The Chassis ID subtype field (chassis-id-subtype) should contain subtype 4, indicating that the Chassis ID field (chassis-id) contains a MAC address to achieve the Chassis ID's desired uniqueness. For IA-stations with multiple unique MAC addresses, any one of the IA-station's MAC addresses may be used and shall be the same for all external ports of that IA-station.

6.5.2.4.3 Port ID TLV

The Port ID field shall contain the same value for all transmitted LLDPDUs for a given external port, i.e., be a non-volatile, IA-station-unique identifier of the LLDPDU-transmitting port.

The Port ID subtype field (port-id-subtype) should contain subtype 5, indicating that the Port ID field contains the port interface name (name) according to IETF RFC 8343.

IA-stations should restrict the system-defined port ID to read-only access and a maximum name length of 255 characters. The names should match the imprinted port names on the chassis.

6.5.2.4.4 Time To Live TLV

The Time To Live value shall be set according to IEEE Std 802.1AB-2016, 8.5.4 (message-tx-interval * message-tx-hold-multiplier + 1).

6.5.2.4.5 System capabilities TLV

An IA-station consisting of a single end station component shall set the system capabilities and enabled capabilities fields (system-capabilities-supported, system-capabilities-enabled) to Station Only (i.e., bit 8 set to 1) for all transmitted LLDPDUs.

An IA-station consisting of at least one end station component and at least one Bridge component shall set the system capabilities and enabled capabilities fields to Station Only (i.e., bit 8 set to “1”) and C-VLAN component (i.e., bit 9 set to “1”) for all transmitted LLDPDUs.

NOTE The combination of the Station Only and C-VLAN component flags is used as a marker indicating to the TDE that the internal structure of the IA-station consists of multiple components. This is a deliberate deviation from IEEE Std 802.1AB-2016, Table 8-4, which states in a footnote: "The Station Only capability is intended for devices that implement only an end station capability, and for which none of the other capabilities in the table apply. Bit 8 should therefore not be set in conjunction with any other bits."

6.5.2.4.6 Management address TLV

An IA-station shall announce at least one IPv4 address by which its Management entity (see 4.3) can be reached (management-address-tx-port).

6.5.2.5 LLDP remote systems data

An IA-station supporting the remote systems YANG shall be able to store information from at least one neighbor per external port.

Receiving LLDPDUs from more neighbors than supported on a given port shall result in the last one received being saved to the remote systems YANG as described in IEEE Std 802.1AB-2016, 9.2.7.7.5.

6.5.3 Topology verification overview

Topology verification checks discovered topologies against engineered topologies. Topology verification data includes for every IA-station:

- model name,
- manufacturer name,
- management address.

Topology verification data includes for every external port of an IA-station:

- port name,
- remote connection (i.e., management address and port name of connected IA-station).

To support topology verification IA-stations shall support LLDP YANG data as defined in 6.4.9.2.2 and Hardware Management YANG data as defined in 6.4.9.2.5.8.

IA-station hardware instance specific data like MAC addresses or serial numbers are not considered for topology verification. This kind of data changes after a repair and replacement operation and thus, induces a topology verification error.

6.6 CNC

6.6.1 General

Subclause 6.6 describes stream destination MAC address handling at the CNC.

6.6.2 Stream destination MAC address range

A CNC manages the destination MAC address for requested streams. This destination MAC address together with the VID identifies the path used for these streams. Thus, a stream destination MAC address is unique together with the VID in a configuration domain.

Preferably, a CNC uses a contiguous address range for managing the stream addresses to support hardware optimization.

Figure 39 shows the possible selections of a CNC for a contiguous address range. The CNC selects an OUI and an offset of the address range for the stream destination MAC addresses.

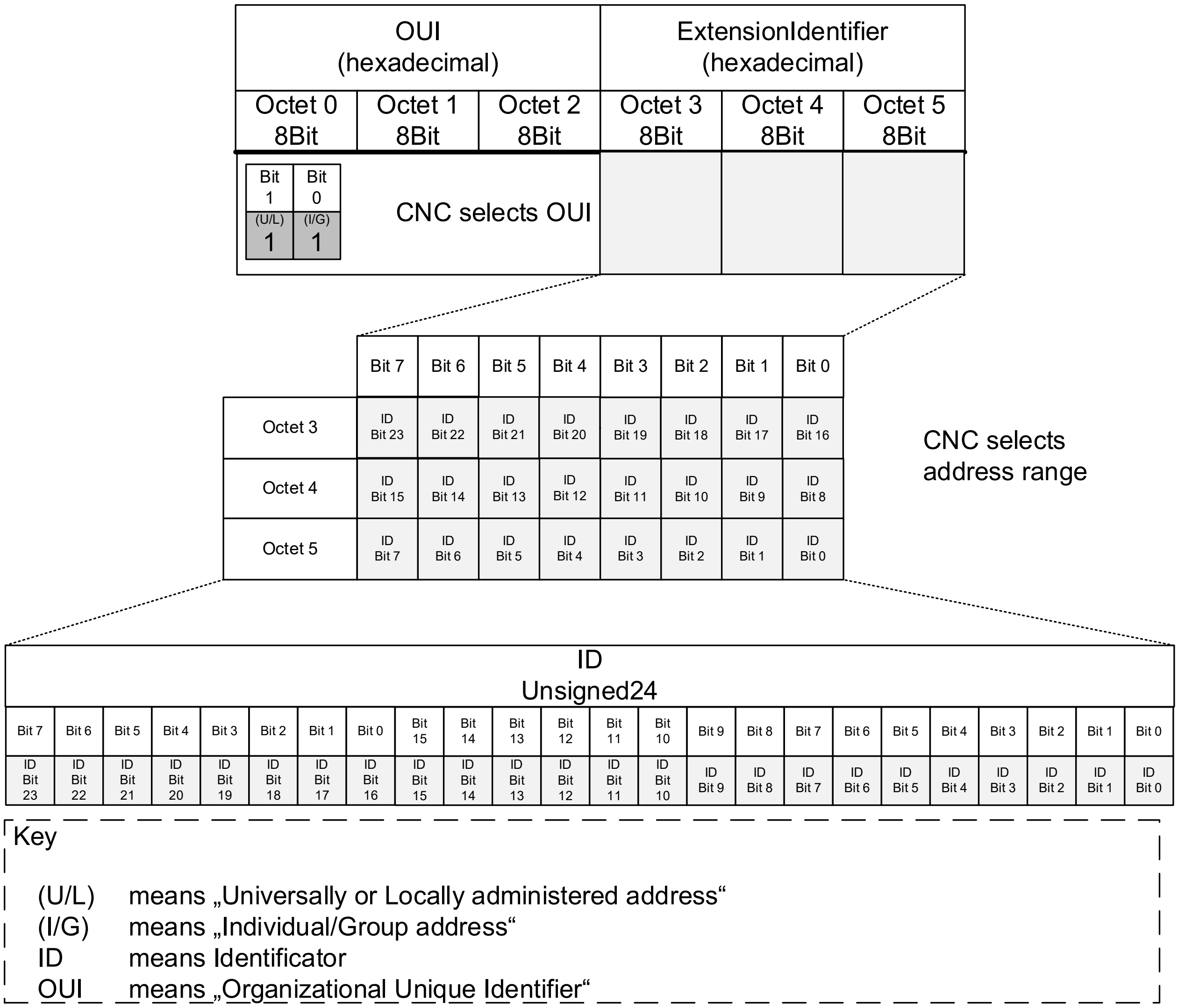
An address range of 2048 stream destination MAC addresses allows together with a VID the usage of 2048 streams. Each additional VID used for streams allows an additional 2048 streams.

EXAMPLE

CNC selected OUI := 00-80-C2

CNC selected address range := 0..2047

CNC selected VID := 101



Key

(U/L) means „Universally or Locally administered address“
(I/G) means „Individual/Group address“
ID means Identificator
OUI means „Organizational Unique Identifier“

Figure 39 – Stream Destination MAC Address

Annex A
(normative)

PCS proforma – Time-sensitive networking profile for industrial automation

A.1 General[9]

The supplier of an implementation that is claimed to conform to the profile specified in this document shall complete the corresponding Profile Conformance Statement (PCS) proforma, which is presented in a tabular format based on the format used for Protocol Implementation Conformance Statement (PICS) proformas.

The tables do not contain an exhaustive list of all requirements that are stated in the referenced standards; for example, if a row in a table asks whether the implementation is conformant to Standard X, and the answer “Yes” is chosen, then it is assumed that it is possible, for that implementation, to fill out the PCS proforma defined in Standard X to show that the implementation is conformant; however, the tables in this document will only further refine those elements of conformance to Standard X where particular answers are required for the profiles specified here.

A completed PCS proforma is the PCS for the implementation in question. The PCS is a statement of which capabilities and options of the protocol have been implemented. The PCS can have several uses, including use by the following.

a) Protocol implementer, as a checklist to reduce the risk of failure to conform to the document through oversight.
b) Supplier and acquirer, or potential acquirer, of the implementation, as a detailed indication of the capabilities of the implementation, stated relative to the common basis for understanding provided by the standard PCS proforma.
c) User, or potential user, of the implementation, as a basis for initially checking the possibility of interworking with another implementation.

NOTE While interworking can never be guaranteed, failure to interwork can often be predicted from incompatible PCS.

d) Protocol tester, as the basis for selecting appropriate tests against which to assess the claim for conformance of the implementation.
e) The user, to verify whether the IA-station, as described by the PCS, fulfills use-case requirements.

A.2 Abbreviations and special symbols

A.2.1 Status symbols

M: mandatory

O: optional

O.n: optional, but support of at least one of the group of options labeled by the same numeral n is required

X: prohibited

pred: conditional-item symbol, including predicate identification: see A.3.4

¬: logical negation, applied to a conditional item’s predicate

[9] Copyright release for the PCS: Users of this document may freely reproduce the PCS contained in this document so that they can be used for their intended purpose.

A.2.2 General abbreviations

N/A: not applicable

PCS: Profile Conformance Statement

A.3 Instructions for completing the PCS proforma

A.3.1 General structure of the PCS proforma

The first part of the PCS proforma, implementation identification and protocol summary, is to be completed as indicated with the information necessary to identify fully both the supplier and the implementation.

The main part of the PCS proforma is a fixed-format questionnaire, divided into several subclauses, each containing a number of individual items. Answers to the questionnaire items are to be provided in the rightmost column, either by simply marking an answer to indicate a restricted choice (usually Yes or No) or by entering a value or a set or range of values. There are some items where two or more choices from a set of possible answers can apply; all relevant choices are to be marked. Each item is identified by an item reference in the first column. The second column contains the question to be answered; the third column records the status of the item—whether support is mandatory, optional, or conditional; see also A.3.4. The fourth column contains the reference or references to the material that specifies the item in the main body of this document, and the fifth column provides the space for the answers.

The PCS indicates support of one of the conformance classes, ccA or ccB, per bridge and end-station component, specified in this profile.

A single IA-station can incorporate the functionality of one or more of the functions listed in this PCS. For example, an IA-station could have both an end station component and a Bridge component.

A supplier can also provide (or be required to provide) further information, categorized as either additional information (see A.3.2) or exception information (see A.3.3). When present, each kind of further information is to be provided in a further subclause of items labeled Ai or Xi, respectively, for cross-referencing purposes, where (i) is any unambiguous identification for the item (for example, simply a numeral). There are no other restrictions on its format and presentation.

A completed PCS proforma, including any Additional Information and Exception Information, is the Protocol Implementation Conformation Statement for the implementation in question.

NOTE Where an implementation is capable of being configured in more than one way, a single PCS can be used to describe all such configurations. However, the supplier has the choice of providing more than one PCS, each covering some subset of the implementation’s configuration capabilities, in case that makes for easier and clearer presentation of the information.

A.3.2 Additional information

Items of Additional Information allow a supplier to provide further information intended to assist the interpretation of the PCS. It is not intended or expected that a large quantity will be supplied, and a PCS can be considered complete without any such information. Examples might be an outline of the ways in which a (single) implementation can be set up to operate in a variety of environments and configurations, or information about aspects of the implementation that are outside the scope of this document but that have a bearing on the answers to some items.

References to items of Additional Information can be entered next to any answer in the questionnaire and can be included in items of Exception Information.

A.3.3 Exception information

It can occasionally happen that a supplier will wish to answer an item with mandatory status (after any conditions have been applied) in a way that conflicts with the indicated requirement. No preprinted answer will be found in the Support column for this item. Instead, the supplier shall write the missing answer into the Support column, together with an Xi reference to an item of Exception Information and shall provide the appropriate rationale in the Exception item itself.

An implementation for which an Exception item is required in this way does not conform to this document.

NOTE A possible reason for the situation described previously is that a defect in this document has been reported, a correction for which is expected to change the requirement not met by the implementation.

A.3.4 Conditional status

A.3.4.1 Conditional items

The PCS proforma contains a number of conditional items. These are items for which both the applicability of the item itself, and its status if it does apply (mandatory or optional) are dependent on whether certain other items are supported.

Where a group of items is subject to the same condition for applicability, a separate preliminary question about the condition appears at the head of the group, with an instruction to skip to a later point in the questionnaire if the “Not Applicable” (N/A) answer is selected. Otherwise, individual conditional items are indicated by a conditional symbol in the Status column.

A conditional symbol is of the form “pred: S” where pred is a predicate as described in A.3.4.2, and S is a status symbol, M or O.

If the value of the predicate is true (see A.3.4.2), the conditional item is applicable, and its status is indicated by the status symbol following the predicate: The answer column is to be marked in the usual way. If the value of the predicate is false, the “Not Applicable” (N/A) answer is to be marked.

A.3.4.2 Predicates

A predicate is one of the following:

a) An item-reference for an item in the PCS proforma: The value of the predicate is true if the item is marked as supported and is false otherwise.

   1) A predicate-name, for a predicate defined as a Boolean expression constructed by combining item-references using the Boolean operator OR: The value of the predicate is true if one or more of the items is marked as supported.

   2) The logical negation symbol “¬” prefixed to an item-reference or predicate-name: The value of the predicate is true if the value of the predicate formed by omitting the “¬” symbol is false, and vice versa.

Each item whose reference is used in a predicate or predicate definition, or in a preliminary question for grouped conditional items, is indicated by an asterisk in the Item column.

A.3.4.3 References to other standards

The following shorthand notation is used in the References columns of the profile tables:

<standard abbreviation>:<Clause-number/sub-clause-number>

where standard abbreviation is one of the following:

- RFC5246: IETF RFC 5246
- RFC5280: IETF RFC 5280
- RFC5289: IETF RFC 5289
- RFC6241: IETF RFC 6241
- RFC7589: IETF RFC 7589
- RFC7905: IETF RFC 7905
- AB: IEEE Std 802.1AB-2016
- AS: IEEE Std 802.1AS-2020
- ASdm: IEEE P802.1ASdm
- CB: IEEE Std 802.1CB-2017,

- CBdb: IEEE Std 802.1CBdb-2021,
- CBdv: IEEE Std 802.1CBcv-2021
- Dot3: IEEE Std 802.3-2022
- Q: IEEE Std 802.1Q-2022
- TS: IEEE Std 1588-2019

Hence, a reference to “IEEE Std 802.1Q-2022, 5.4.2” would be abbreviated to “Q:5.4.2”.

A.3.5 Electronic datasheet

A provider of a device shall provide the PCS values in a standardized electronic format as data sheet of the product.

A.4 Common requirements

A.4.1 Instructions

One instance of Clause A.4 shall be filled out per IA-station.

A.4.2 Implementation identification

The entire PCS pro forma is a form that shall be filled out by a supplier according to Table A.1.

Table A.1 – Implementation identification template

Supplier	
Contact point for queries about the PCS	
Implementation Name(s) and Version(s)	
Other information necessary for full identification, for example, name(s) and version(s) of machines and/or operating system names	

Only the first three items are required for all implementations; other information can be completed as appropriate in meeting the requirement for full identification. The terms “Name” and “Version” should be interpreted appropriately to correspond with a supplier’s terminology (for example, Type, Series, Model).

A.4.3 Profile summary, IEC/IEEE 60802

Table A.2 shows the profile summary template.

Table A.2 – Profile summary template

Identification of profile specification	**IEC/IEEE 60802 - Time-Sensitive Networking profile for industrial automation**
Identification of amendments (Amd) and corrigenda (Corr) to the PCS proforma that have been completed as part of the PCS	Amd. : Corr. : Amd. : Corr. :
Have any Exception items been required? (See A.3.3: the answer “Yes” means that the implementation does not conform to IEC/IEEE 60802)	No [ ] Yes [ ]
Date of Statement	

A.4.4 Implementation summary

The form in Table A.3 is used to indicate the type of system that the PCS describes.

Table A.3 – Implementation type

Item	Feature	Status	References	Support
BC-CCA-N	State the number of Conformance Class A bridge components implemented by the IA-station.	O:1	5.7.2, 5.8.2	Number _____
BC-CCB-N	State the number of Conformance Class B bridge components implemented by the IA-station.	O:1	5.7.3, 5.8.3	Number _____
ESC-CCA-N	State the number of Conformance Class A end station components implemented by the IA-station.	O:1	5.9.2, 5.10.2	Number _____
ESC-CCB-N	State the number of Conformance Class B end station components implemented by the IA-station.	O:1	5.9.3, 5.10.3	Number _____
CNC	Does the IA-station include a CNC?	O	5.11	Yes [ ] No [ ]
CUC	Does the IA-station include a CUC?	O	5.13	Yes [ ] No [ ]

A.5 IA-station Requirements and Options

A.5.1 Instructions

One instance of Clause A.5 shall be filled out for an IA-station.

A.5.2 IA-station requirements

The form in Table A.4 is used to indicate the IA-station requirements.

Table A.4 – IA-station requirements

Item	Feature	Status	References	Support
IASTA-1	Does the IA-Station support PHY and MAC requirements for external ports?	M	5.5.1, Dot3	Yes [ ]
IASTA-2	Does the IA-Station support topology discovery requirements?	M	5.5.2, AB	Yes [ ]
IASTA-3	Does the IA-Station support requirements for time synchronization?	M	5.5.3, AS, ASdm, TS	Yes [ ]
IASTA-4	Does the IA-Station support requirements for Secure management exchanges?	M	5.5.4.2	Yes [ ]
IASTA-5	Does the IA-Station support management YANG modules?	M	5.5.4.3	Yes [ ]
IASTA-6	Does the IA-Station provide a data sheet?	M	5.5.4.4	Yes [ ]

A.5.3 IA-station prohibited management features

The form in Table A.5 is used to indicate prohibited management features for an IA-station.

Table A.5 – IA-station prohibited management features

Item	Feature	Status	References	Support
SECMGMT-1	Is NETCONF-over-SSH used to configure the end-station?	X	6.3.2.1.1	No [ ]
SECMGMT-2	Does the IA-station implement TLS_RSA_WITH_AES_128_CBC_SHA?	X	6.3.2.1.2	No [ ]
SECMGMT-3	Does the IA-station implement TLS extensions in IETF RFC 6066 and IETF RFC 6961?	X	6.3.2.1.2	No [ ]

SECMGMT-4	Does the IA-station mark the id-60802-pe-roles as critical?	X	6.3.2.1.4	No [ ]

A.5.4 IA-station PHY and MAC options for external ports

The form in Table A.6 is used to indicate PHY and MAC options for external ports.

Table A.6 – IA-station PHY and MAC options

Item	Feature	Status	References	Support
DOT3-1	Does the IA-station support PoE over 2 pairs?	O	5.6.1:a), dot3:33	Yes [ ] No [ ] N/A [ ]
DOT3-2	Does the IA-Station support Power Interfaces?	O	5.6.1:b), dot3:104	Yes [ ] No [ ] N/A [ ]
DOT3-3	Does the IA-Station support PoE?	O	5.6.1:c), dot3:145	Yes [ ] No [ ] N/A [ ]

A.5.5 IA-station options for time synchronization

The form in Table A.7 is used to indicate options for time synchronization.

Table A.7 – IA-station time synchronization options

Item	Feature	Status	References	Support
PTP-1	Does the IA-station support media-independent master capability according to IEEE Std 802.1AS-2020, 5.4.2 item b)?	O	5.6.2:a), AS:5.4.2	Yes [ ] No [ ]
PTP-2	Does the IA-station support Grandmaster PTP Instance capability according to IEEE Std 802.1AS-2020, 5.4.2 item c)?	O	5.6.2:b), AS:5.4.2	Yes [ ] No [ ]
PTP-3	Does the IA-station support more than one PTP port as a PTP Relay Instance according to IEEE Std 802.1AS-2020, 5.4.2 item d)?	O	5.6.2:c), AS:5.4.2	Yes [ ] No [ ]
PTP-4	Does the IA-station support transmit of the Signaling message according to IEEE Std 802.1AS-2020, 5.4.2 item e)?	O	5.6.2:d), AS:5.4.2	Yes [ ] No [ ]
PTP-5	Does the IA-station support more than 1 PTP Instance according to IEEE Std 802.1AS-2020, 5.4.2 item f)?	O	5.6.2:e), AS:5.4.2	Yes [ ] No [ ]
PTP-6	Does the IA-station support the SyncIntervalSetting state machine according to IEEE Std 802.1AS-2020, 5.4.2 item h)?	O	5.6.2:f), AS:5.4.2	Yes [ ] No [ ]
PTP-7	Does the IA-station support one or more application interfaces according to IEEE Std 802.1AS-2020, 5.4.2 item i)?	O	5.6.2:g), AS:5.4.2	Yes [ ] No [ ]
PTP-8	Does the IA-station support hot standby redundancy requirements?	O	5.6.2:h), ASdm:5.4.2	Yes [ ] No [ ]

A.5.6 IA-station secure management exchange options

The form in Table A.8 is used to indicate options for secure management exchange.

Table A.8 – IA-station secure management exchange options

Item	Feature	Status	References	Support
SECMGMT-5	Does the IA-station support Writable-Running capability?	O	5.6.3:a), RFC6241:8.2	Yes [ ] No [ ]
SECMGMT-6	Does the IA-station support Confirmed Commit capability?	O	5.6.3:b), RFC6241:8.4	Yes [ ] No [ ]

SECMGMT-7	Does the IA-station support Distinct Startup capability?	O	5.6.3:c), RFC6241:8.7	Yes [ ] No [ ]
SECMGMT-8	Does the IA-station support URL capability?	O	5.6.3:d), RFC6241:8.8	Yes [ ] No [ ]
SECMGMT-9	Does the IA-station support XPath capability?	O	5.6.3:e), RFC6241:8.9	Yes [ ] No [ ]
SECMGMT-10	Does the IA-station support NETCONF-over-TLS server with the TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 cypher suite?	O	5.6.3:f), RFC7589, RFC5289:3.2, RFC5289:5	Yes [ ] No [ ]
SECMGMT-11	Does the IA-station support NETCONF-over-TLS server with the TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 cypher suite?	O	5.6.3:f), RFC7589, RFC7905:2, RFC7905:3	Yes [ ] No [ ]
SECMGMT-12	Does the IA-station support TLS with the Curve P-521 elliptic curve?	O	5.6.3:g), 6.3.2.1.2	Yes [ ] No [ ]
SECMGMT-13	Does the IA-station support TLS with the Curve25519 elliptic curve?	O	5.6.3:g), 6.3.2.1.2	Yes [ ] No [ ]
SECMGMT-14	Does the IA-station support TLS with the Curve448 elliptic curve?	O	5.6.3:g), 6.3.2.1.2	Yes [ ] No [ ]
SECMGMT-15	Does the IA-station support the YANG features and leaves of the ietf-keystore?	O	5.6.3:h), 6.3.4.3	Yes [ ] No [ ]
SECMGMT-16	Does the IA-station support PKIX?	O	5.6.3:i), RFC5280,	Yes [ ] No [ ]
SECMGMT-17	Does the IA-station support internal key generation?	O	5.6.3, 6.3.4.3.2	Yes [ ] No [ ]

A.5.7 CNC Requirements

The form in Table A.9 is used to indicate requirements for CNCs.

Table A.9 – CNC Requirements

Item	Feature	Status	References	Support
CNC-1	Does the IA-station support CNC requirements?	CNC:M	5.11	Yes [ ] N/A [ ]

A.5.8 CUC Requirements

The form in Table A.10 is used to indicate requirements for CUCs.

Table A.10 – CUC Requirements

Item	Feature	Status	References	Support
CUC-1	Does the IA-Station support CUC requirements?	CUC:M	5.13	Yes [ ] N/A [ ]

A.6 Bridge Component

A.6.1 Instructions

One instance of Clause A.6 shall be filled out per bridge component implemented by an IA-station.

A.6.2 Bridge Component Requirements

The form in Table A.11 is used to indicate bridge component requirements.

Table A.11 –Bridge Component Requirements

Item	Feature	Status	References	Support
BC-1	Does the bridge component support the common bridge component requirements?	M	5.7.1, Q	Yes [ ]
BC-2	Does the bridge component support ccA bridge component requirements?	O:2	5.7.2, Q	Yes [ ] No [ ]
BC-3	Does the bridge component support ccB bridge component requirements?	O:2	5.7.3, Q	Yes [ ] No [ ]

A.6.3 Common Bridge Component Options

The form in Table A.12 is used to indicate bridge component options.

Table A.12 – Common Bridge Component Options

Item	Feature	Status	References	Support
BC-4	Does the bridge component support the operation of the credit-based shaper algorithm?	O	5.8.1, 6.4.9.3.5, Q:8.6.8.2	Yes [ ] No [ ]

A.6.4 ccA Bridge Component Options

The form in Table A.13 is used to indicate options for bridge components conforming to conformance class A.

Table A.13 – ccA Bridge Component Options

Item	Feature	Status	References	Support
CCA-BC-1	Does the bridge component support any of the common bridge component options?	O	5.8.2:a), 5.8.1	Yes [ ] No [ ] N/A [ ]
CCA-BC-2	Does the bridge component support more than 2 PTP instances?	O	5.8.2:b), 5.5.3	Yes [ ] No [ ] N/A [ ]
CCA-BC-3	State the number of PTP instances supported by the bridge component.	CCA-BC-2:M	5.8.2:b), 5.5.3	Number ____
CCA-BC-4	Does the bridge component support enhancements for scheduled traffic for the 10 Mb/s, 2,5 Gb/s, 5 Gb/s, or 10 Gb/s data rates?	O	5.8.2:c), Q:5.4.1:ab), ac)	Yes [ ] No [ ] N/A [ ]
CCA-BC-5	Does the bridge component support frame preemption for the 10Mb/s, 2,5 Gb/s, 5Gb/s, or 10Gb/s data rates?	O	5.8.2:d), Q:5.4.1:ad)	Yes [ ] No [ ] N/A [ ]

A.6.5 ccB Bridge Component Options

The form in Table A.14 is used to indicate options for bridge components conforming to conformance class B.

Table A.14 – ccB Bridge Component Options

Item	Feature	Status	References	Support
CCB-BC-1	Does the bridge component support any of the common bridge component options?	O	5.8.3:a), 5.8.1	Yes [ ] No [ ] N/A [ ]
CCB-BC-2	Does the bridge component support more than 4 but not more than 8 egress queues?	O	5.8.3:b), Q:8.6.6	Yes [ ] No [ ] N/A [ ]
CCB-BC-3	State the number of egress queues supported by the bridge component.	CCB-BC-2:M	5.8.3:b)	Number ____
CCB-BC-4	Does the bridge component support more than 1 PTP instance?	O	5.8.3:c), 5.5.3	Yes [ ] No [ ] N/A [ ]
CCB-BC-5	State the number of PTP instances supported by the bridge component.	CCB-BC-4:M	5.8.3:c), 5.5.3	Number ____
CCB-BC-6	Does the bridge component support enhancements for scheduled traffic?	O	5.8.3:d), Q:5.4.1:ab), ac)	Yes [ ] No [ ] N/A [ ]
CCB-BC-7	Does the bridge component support frame preemption?	O	5.8.3:e), Q:5.4.1:ad)	Yes [ ] No [ ] N/A [ ]

A.7 End Station Component

A.7.1 Instructions

One instance of Clause A.7 shall be filled out per end station component implemented by an IA-station.

A.7.2 Common End Station Component Requirements

The form in Table A.15 is used to indicate common requirements for end stations.

Table A.15 – Common End Station Component Requirements

Item	Feature	Status	References	Support
ESC-1	Does the end station component support the common end station component requirements?	M	5.9.1	Yes [ ]
ESC-2	Does the end station component support the ccA end station component requirements?	O:3	5.9.2, Q	Yes [ ] No [ ]
ESC-3	Does the end station component support the ccB end station component requirements?	O:3	5.9.3	Yes [ ] No [ ]

A.7.3 Common End Station Component Options

The form in Table A.16 is used to indicate options for end stations.

Table A.16 – Common End Station Component Options

Item	Feature	Status	References	Support
ESC-4	Does the end station component support the operation of the credit-based shaper?	O	5.10.1:a), Q:8.6.8.2	Yes [ ] No [ ]
ESC-5	Does the end station component support talker end system behaviors?	O	5.10.1:b), CB, CBdb, CBcv	Yes [ ] No [ ]
ESC-6	Does the end station component support listener end system behaviors?	O	5.10.1:c), CB, CBdb, CBcv	Yes [ ] No [ ]

A.7.4 ccA End Station Component Options

The form in Table A.17 is used to indicate options for end stations conforming to conformance class A.

Table A.17 – ccA End Station Component Options

Item	Feature	Status	References	Support
CCA-ESC-1	Does the end station component support any of the common end station component options?	O	5.10.2:a), 5.10.1	Yes [ ] No [ ] N/A [ ]
CCA-ESC-2	Does the end station component support more than 2 PTP instances?	O	5.10.2:b), 5.5.3	Yes [ ] No [ ] N/A [ ]
CCA-ESC-3	State the number of PTP instances supported by the end-station component.	CCA-ESC-2:M	5.10.2:b), 5.5.3	Number ____
CCA-ESC-4	Does the end station component support enhancements for scheduled traffic for data rates 10 Mb/s, 2,5 Gb/s, 5 Gb/s, or 10 Gb/s?	O	5.10.2:c), Q:5.4.1:ab), ac)	Yes [ ] No [ ] N/A [ ]
CCA-ESC-5	Does the end station component support requirements for frame pre-emption for data rates 10 Mb/s, 2,5 Gb/s, 5 Gb/s, or 10 Gb/s?	O	5.10.2:d), Q:5.4.1:ad)	Yes [ ] No [ ] N/A [ ]

A.7.5 ccB End Station Component Options

The form in Table A.18 is used to indicate options for end stations conforming to conformance class B.

Table A.18 – ccB End Station Component Options

Item	Feature	Status	References	Support
CCB-ESC-1	Does the end station component support any of the common end station component options?	O	5.10.3:a), 5.10.1	Yes [ ] No [ ] N/A [ ]
CCB-ESC-2	Does the end station component support more than one PTP instance?	O	5.10.3:b), 5.5.3	Yes [ ] No [ ] N/A [ ]
CCB-ESC-3	State the number of PTP instances supported by the end-station component.	CCB-ESC-2:M	5.10.3:b), 5.5.3	Number ____
CCB-ESC-4	Does the end station component support enhancements for scheduled traffic?	O	5.10.3:c), Q:5.4.1:ab), ac)	Yes [ ] No [ ] N/A [ ]
CCB-ESC-5	Does the end station component support requirements for frame preemption?	O	5.10.3:d), Q:5.4.1:ad	Yes [ ] No [ ] N/A [ ]

Annex B
(informative)

Representative Configuration Domain

The following quantities are representative of what could be supported in a single Configuration Domain:

IA-stations: 1 024

Network diameter: 64

Streams per IA-Controller for IA-Controller to IA-device (C2D) communication:

- 512 Talker and >= 512 Listener streams.
- 1 024 Talker and >= 1 024 Listener streams in case of seamless redundancy.

Streams per IA-Controller for IA-Controller to IA-Controller (C2C) communication:

- 64 Talker and >= 64 Listener streams.
- 128 Talker and >= 128 Listener streams in case of seamless redundancy.

Streams per IA-device for IA-device-to-IA-device (D2D) communication:

- 2 Talker and 2 Listener streams.
- 4 Talker and 4 Listener streams in case of seamless redundancy.

Example calculation of data flow quantities for eight PLCs – without seamless redundancy:

- 8 x 512 x 2 = 8 192 streams for C2D communication, plus
- 8 x 64 x 2 = 1 024 streams for C2C communication
- (8 192 + 1 024) * 2 000 = 18 432 000 Bytes data of all streams

Annex C
(informative)

Description of Clock Control System

C.1 Clock control system introduction

Annex C provides an introductory discussion of a basic clock control system. For more detailed information, see the Bibliography References for Annex C.

Figure C.1 shows a basic control system model that uses a proportional plus integral (PI) controller. This is meant to be reference model, i.e., it is not meant to specify an implementation. Requirements for the clock control system can be expressed using parameters (e.g., 3dB bandwidth, gain peaking, frequency response) that are based on this reference model. Any implementation whose parameters are within the requirements is considered to be acceptable. For example, the model of Figure C.1 is expressed in the analog domain (i.e., s-domain), and will be shown shortly to be second order. An actual implementation can be digital, and can be higher order, as long as it meets the respective requirements.

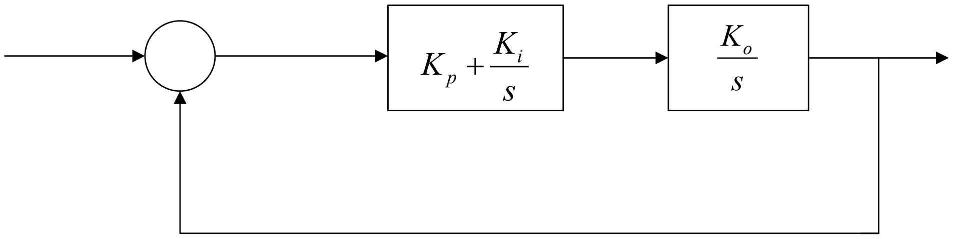


Figure C.1 – Reference model for clock control system

In Figure C.1, the plant, i.e., the entity being controlled, represents the clock oscillator. It is desired that the phase output, *y(t)* of the oscillator follows the phase input, *u(t)*, as closely as possible (the signals are shown in the frequency domain (i.e., as Laplace Transforms) in Figure C.1; however, they can equivalently be expressed in the time domain, with *t* representing time). Because of this behavior, this control system is also referred to as a phase-locked loop (PLL). The parameter K_o is the oscillator gain; the oscillator frequency is equal to the oscillator input multiplied by K_o. In some implementations the input signal to the oscillator is a voltage, and the oscillator is referred to as a voltage-controlled oscillator (VCO). However, other implementations are possible, e.g., digital implementations, where the oscillator is a digital controlled oscillator (DCO). Since the input to the oscillator depends on the implementation, it is not labeled in Figure C.1.

The control system of Figure C.1 uses negative feedback to enable the phase output to follow the phase input. The phase detector computes the difference between the input and output signals to produce the error signal *e(t)*. The error signal is then filtered by the PI filter to produce the input to the oscillator. The filter is referred to as a PI filter because its output is the sum of the proportional gain, K_p, multiplied by the error signal and the integral gain, K_i, multiplied by the integral of the error signal. The gains K_o, K_p, and K_i must be chosen such that the performance of the control system is acceptable, i.e., the time-domain behavior of the output with respect to the input is acceptable. However, an alternative set of parameters, which are more convenient, can be defined in terms of K_o, K_p, and K_i; this is done in Clause C.2.

C.2 Transfer function for control system

From the block diagram of Figure C.1, the input and output are related by:

$$Y(s)=\left(K_p+\frac{K_i}{s}\right)\left(\frac{K_o}{s}\right)\left(U(s)-Y(s)\right) \tag{C.1}$$

or

$$Y(s)=\frac{\left(K_p+\frac{K_i}{s}\right)\left(\frac{K_o}{s}\right)}{1+\left(K_p+\frac{K_i}{s}\right)\left(\frac{K_o}{s}\right)}U(s) \tag{C.2}$$

This can be simplified by multiplying the numerator and denominator by s^2 to produce:

$$Y(s)=H(s)U(s) \tag{C.3}$$

where the transfer function $H(s)$ is given by:

$$H(s)=\frac{K_pK_os+K_iK_o}{s^2+K_pK_os+K_iK_o} \tag{C.4}$$

In equation (C.4), the parameter K_o does not appear independently of K_p and K_i; rather, only the products K_pK_o and K_iK_o appear. The plant and PI filter could have been combined in the model of Figure C.1; this is consistent with the fact that the exact nature of the signal between the PI filter and plant is unimportant in this reference model. The units of K_pK_o are (time)$^{-1}$ and the units of K_iK_o are (time)$^{-2}$. The frequency units need to be the same as the units of s, e.g., if s has units rad/s, then K_pK_o has units rad/s and K_iK_o has units (rad/s)^2. The integration operation in the plant results in the transfer function being dimensionless, which is consistent with the fact that the input and output of the control system both have units of phase.

The transfer function can be expressed in an equivalent form by defining the undamped natural frequency, ω_n, and damping ratio, ζ:

$$H(s)=\frac{2\varsigma\omega_ns+\omega_n^2}{s^2+2\varsigma\omega_ns+\omega_n^2} \tag{C.5}$$

where:

$$\omega_n = \sqrt{K_i K_o}$$
$$\varsigma = \frac{K_p K_o}{2\sqrt{K_i K_o}} = \frac{K_p}{2}\sqrt{\frac{K_i}{K_o}} \tag{C.6}$$

In the equation for ζ, the first form shows explicitly that ζ depends only on the products K_pK_o and K_iK_o.

C.3 Frequency response for control system

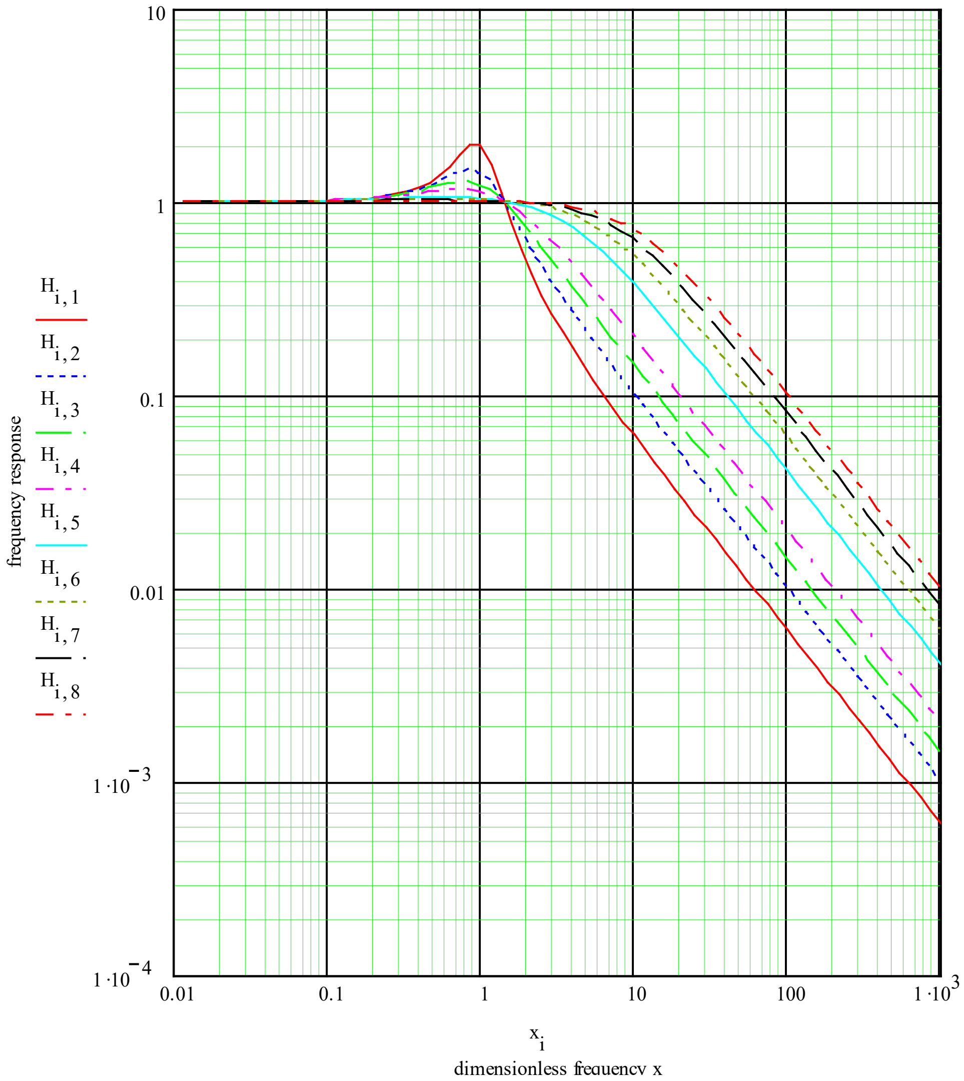
The frequency response is obtained by setting $s = j\omega$ in equation (C.5) and taking the absolute value (here j rather than i is used for $\sqrt{-1}$ to avoid confusion with other uses of i), where ω is the frequency in rad/s. The result is:

$$\left|H(j\omega)\right| = \left|\frac{2\varsigma\omega_n\omega j + \omega_n^2}{-\omega^2 + \omega_n^2 + 2\varsigma\omega_n\omega j}\right| = \left(\frac{4\varsigma^2\omega_n^2\omega^2 + \omega_n^4}{\left(\omega_n^2 - \omega^2\right)^2 + 4\varsigma^2\omega_n^2\omega^2}\right)^{1/2} \tag{C.7}$$

Dividing the numerator and denominator of equation (C.7) by $\omega_n{}^4$ and defining the dimensionless frequency $x = \omega/\omega_n$ produces:

$$\left|H(j\omega)\right| = \left(\frac{4\varsigma^2 x^2 + 1}{\left(1 - x^2\right)^2 + 4\varsigma^2 x^2}\right)^{1/2} \tag{C.8}$$

Figure C.2 contains plots of frequency response (equation (C.8)) versus dimensionless frequency x, on a log-log scale, for damping ratio ζ equal to 0,3, 0,5, 0,707, 1,0, 2,0, 3,0, 4,0, and 5,0. It is seen that the frequency response is very close to 1 for values of dimensionless frequency much less than 1 (i.e., for $\omega << \omega_n$). The frequency response increases as the frequency approaches the undamped natural frequency (i.e., as dimensionless frequency approaches 1) and reaches a peak for dimensionless frequency slightly less than 1. The frequency response then decreases, eventually having a slope (i.e., roll-off) of 20 dB/decade (i.e., frequency response decreases by a factor of 10 for every factor of 10 increase in x for $x >> 1$). Figure C.3 shows the detail of frequency response for x in the range 0,1 to 10.



6034

6035 **Figure C.2 – Frequency response for the control system of Figure C.1**

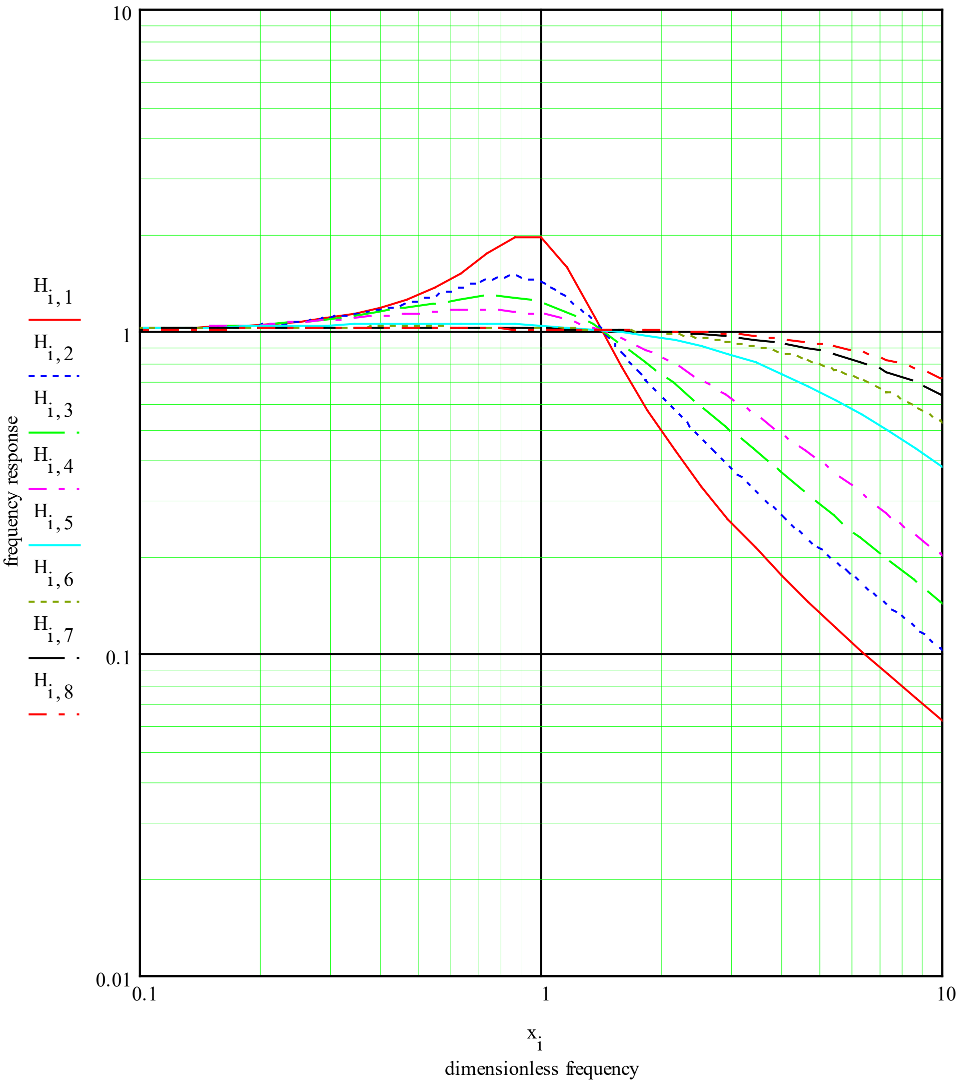


Figure C.3 – Detail of frequency response for the control system of Figure C.1 for dimensionless frequency in the range 0,1 to 10

In addition to undamped natural frequency ω_n and damping ratio ζ, the parameters 3dB bandwidth and gain peaking are often used when specifying clock performance. The 3dB bandwidth is defined as the value of frequency for which the frequency response is equal to –3dB. Since dB is given by 10 multiplied by the logarithm to base 10 of the power ratio, which is 20 multiplied by the logarithm to base 10 of the amplitude ratio, –3dB corresponds to the value $10^{-3/20}$. The 3dB bandwidth can be computed by setting equation (C.8) equal to $10^{-3/20}$ and solving for x in terms of ζ. This is equivalent to setting the quantity in parentheses (i.e., inside the square root) in equation (C.8) equal to $10^{-3/10}$ and solving for x. Now, $10^{-3/10}$ is approximately equal to 0,5012, i.e., it is very close to ½. Then the 3dB bandwidth can be obtained by solving the following equation for x in terms of ζ:

$$\frac{4\varsigma^2x^2+1}{\left(1-x^2\right)^2+4\varsigma^2x^2}=\frac{1}{2} \tag{C.9}$$

or

$$x^4 - 2\left(2\varsigma^2 + 1\right)x^2 - 1 = 0 \tag{C.10}$$

The result is:

$$x = \left[2\varsigma^2 + 1 + \sqrt{(2\varsigma^2 + 1)^2 + 1}\right]^{1/2} \tag{C.11}$$

or

$$\omega_{3\text{dB}} = \omega_n \left[2\varsigma^2 + 1 + \sqrt{(2\varsigma^2 + 1)^2 + 1}\right]^{1/2} \tag{C.12}$$

The gain peaking is the maximum value of the frequency response, in dB. It is computed by differentiating equation (C.8) with respect to x, setting the result to zero, solving for x, and then substituting this value of x into equation (C.8) to obtain the maximum. The result is:

$$H_p = \left[1 - 2\alpha - 2\alpha^2 + 2\alpha\left(2\alpha + \alpha^2\right)^{1/2}\right]^{-1/2} \tag{C.13}$$

where α is related to damping ratio by:

$$\alpha = \frac{1}{4\varsigma^2} \tag{C.14}$$

and H_p is the gain peaking expressed as a pure fraction. The gain peaking in dB is equal to 20·log$_{10}$ H_p. In some cases, it is necessary to compute damping ratio from gain peaking. The result for this is:

$$\alpha = \frac{\left(1 - q\right)\left(1 + \sqrt{1 - q}\right)}{2q} \tag{C.15}$$

where

$$q = \frac{1}{H_p^2} \tag{C.16}$$

Damping ratio is obtained from α using equation (C.14).

If 3dB bandwidth and gain peaking are given, damping ratio can be obtained using equations (C.14) through (C.16). Undamped natural frequency can then be obtained using equation (C.12).

Figure C.4 shows gain peaking, expressed as a pure fraction, as a function of damping ratio. Figure C.5 shows gain peaking in dB as a function of damping ratio.

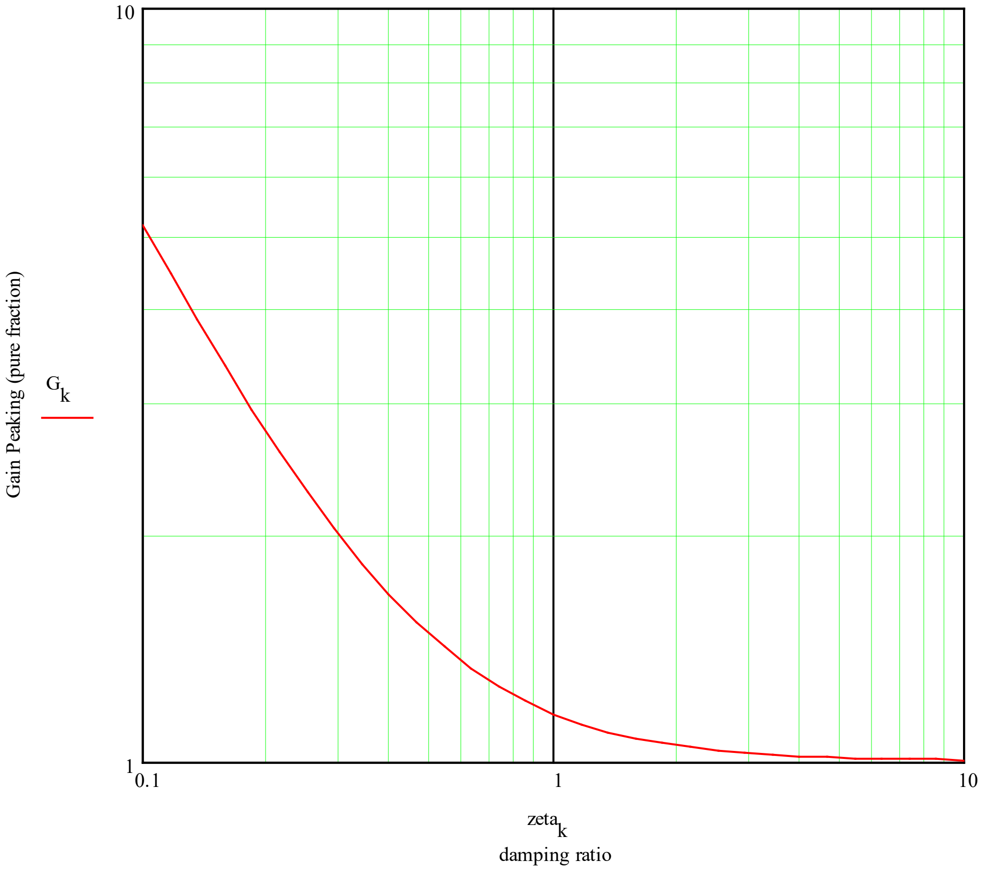


Figure C.4 – Gain peaking (pure fraction) as a function of damping ratio

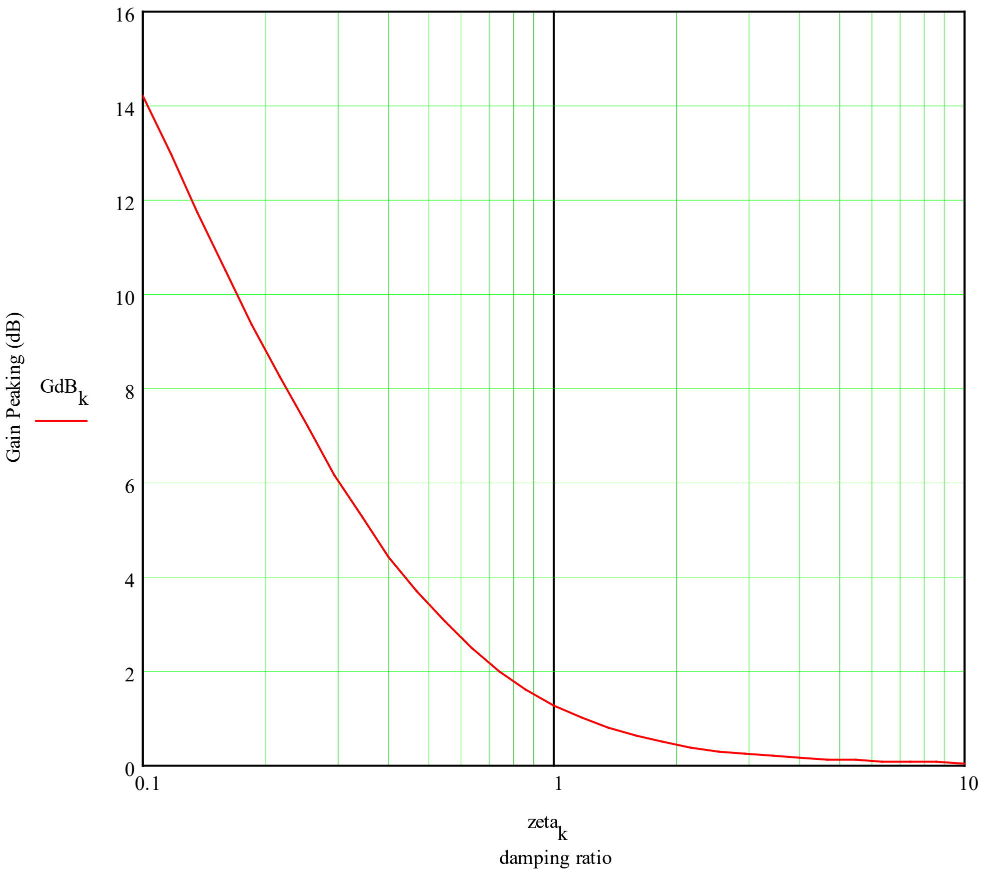


Figure C.5 – Gain peaking in dB as a function of damping ratio

The performance requirements for the clock can be specified using the frequency response. Specifically, the requirement can be stated as:

a) Maximum 3dB bandwidth in Hz,

b) Maximum gain peaking in dB, and

c) Frequency response plot (mask) corresponding to (a) and (b) that is not to be exceeded.

C.4 Example

Consider a clock control system with K_pK_o = 4.23 rad/s and K_iK_o = 9.62 (rad/s)^2. The undamped natural frequency and damping ratio are:

$$\begin{aligned} \omega_n &= \sqrt{K_iK_o} = \sqrt{9.62\ (\text{rad/s})^2} = 3.10\ \text{rad/s} \\ \varsigma &= \frac{K_pK_o}{2\sqrt{K_iK_o}} = \frac{4.23\ \text{rad/s}}{2\sqrt{9.62\ (\text{rad/s})^2}} = 0.682 \end{aligned} \tag{C.17}$$

The gain peaking is obtained from:

$$\alpha = \frac{1}{4(0.682)^2} = 0.537$$

$$H_p \text{ (purefraction)} = \left[1 - 2(0.537) - 2(0.537)^2 + 2(0.537)\sqrt{2(0.537) + (0.537)^2}\right]^{-1/2} = 1.28803 \qquad \text{(C.18)}$$

$$H_p \text{ (dB)} = 20\log_{10}(1.29) \text{ dB} = 2.2 \text{ dB}$$

The 3dB bandwidth is:

$$\begin{aligned} f_{3\text{dB}} \text{ (Hz)} &= \frac{\omega_n}{2\pi}\left[1 + 2\varsigma^2 + \sqrt{\left(1 + 2\varsigma^2\right)^2 + 1}\right]^{1/2} \\ &= \frac{3.10}{2\pi}\left[1 + 2\left(0.682\right)^2 + \sqrt{\left(1 + 2\left(0.682\right)^2\right)^2 + 1}\right]^{1/2} \\ &= 1.0 \text{ Hz} \end{aligned} \qquad \text{(C.19)}$$

The frequency response is shown in Figure D.8.

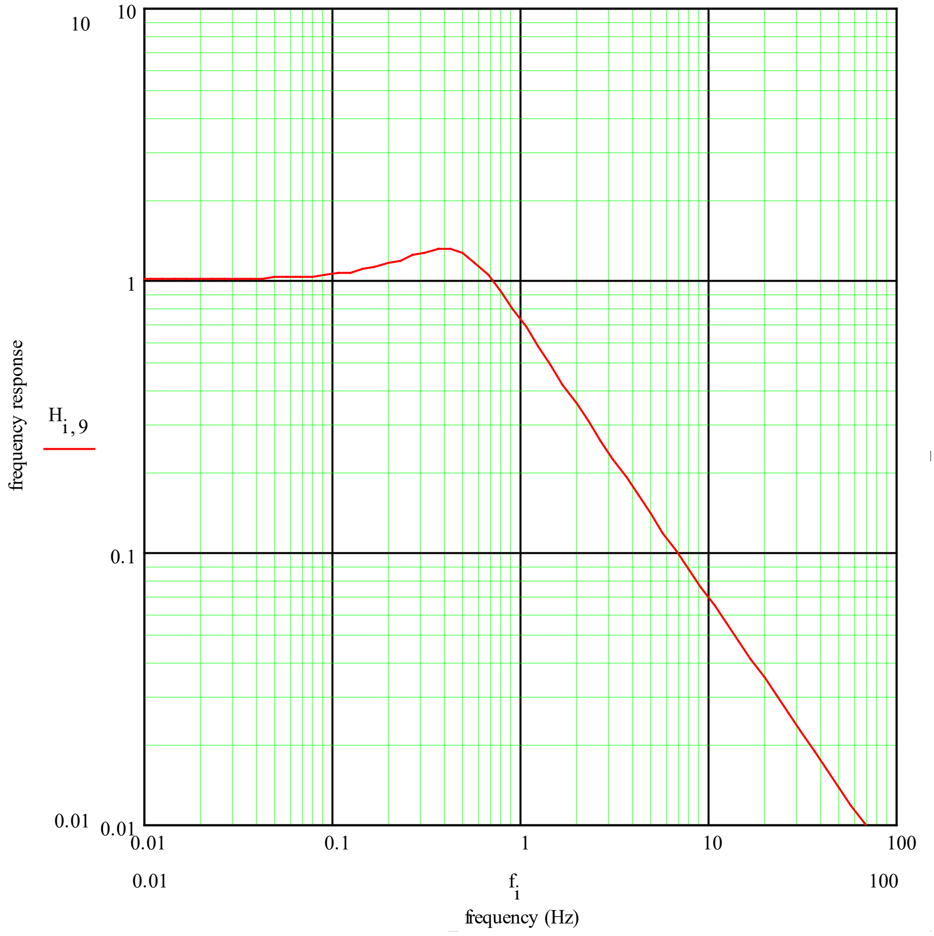


Figure C.6 – Example Frequency response

Annex D
(informative)

Time Synchronization informative Annex

D.1 Overview

This document describes how a network of compliant devices can achieve a time synchronization accuracy, at the application level, of ±1 µs, relative to the Clock Source at the Grandmaster, over 100 network hops. To achieve this, it allocates the overall error budget of 1 000 ns as described in Table D.1.

Table D.1 – Time Synchronisation Error Budget

Network Aspect	Error Type	Network-Level Error Budget (ns)	Normative or Informative?
All PTP Instances	Constant Time Error	200	Normative
	Dynamic Time Error	600	
All PTP Links	Constant Time Error	200	Informative
	Dynamic Time Error	Negligible	

A chain of 1 Grandmaster PTP Instance, 99 PTP Relay Instances and 1 PTP End Instance (100 network hops) that all comply with the normative requirements of 6.2.2 and 6.2.3 will generate a network-level Time Error at or below the Error Budget for All PTP Instances.

Subclause D.2 describes the principles of operation this document assumes.

Subclause D.3 provides additional information on specific normative requirements.

The principles of operation include the use of crystal oscillators (XOs) as opposed to more accurate, stable, and costly options such as temperature-compensated crystal oscillators (TCXOs).

Clause D.4 describes a potential approach to testing the normative requirements. It is not a test specification but rather a high-level overview one potential approach that might be adopted by a full test specification.

The use of XOs means that some of the normative requirements are difficult or impossible to meet without employing algorithms that track Neighbor Rate Ratio drift and Rate Ratio drift and compensate for consequent errors in calculating Rate Ratio and Correction Field.

Clause D.5 of Annex D provides examples of algorithms that can be used for this purpose, and which have been shown to enable compliance with the normative requirements.

Clock drift in neighboring PTP Instances that use XOs means implementations that employ TCXOs or other more accurate, stable oscillators can still find some of the normative requirements difficult or impossible to meet without employing algorithms to track and compensate for errors due to clock drift.

There is no normative requirement to use the algorithms described in Clause D.5; an implementation can employ alternative algorithms provided the normative requirements are met. Clause D.5 describes the potential risks of deploying a network whose instances employ a mix of different algorithms. It is the responsibility of implementers to mitigate the risks and ensure alternative algorithms deliver the network-level performance.

This document does not include normative requirements for PTP Links. Annex D.2.3 describes PTP Link characteristics that influence achieving 1 µs time synchronization accuracy. It includes some examples using common PTP Link characteristics.

This document's normative requirements regarding instance-level error generation are necessitated by the need to ensure not just an overall level of dTE generation at each node, but also the performance of drift tracking and error compensation algorithms and the amount of dTE generation due to timestamp error verses clock drift. The algorithms are employed to mitigate errors due to clock drift but cannot mitigate timestamp errors. Clause D.5 describes an example approach to testing the normative requirements. It is not a test specification nor the only viable approach.

D.2 Principles of Operation

D.2.1 General

Achieving ±1 µs time synchronization accuracy across 100 network hops involves managing the accumulation of errors in the Precise Origin Timestamp + Correction Field and the Rate Ratio as they are passed, via Sync or Follow_Up messages, down the chain of PTP instances and are then used by the PTP End Instance to keep its ClockTarget in line with the ClockSource at the Grandmaster PTP Instance. The majority of significant errors can ultimately be traced back to one of three sources: timestamp error, clock drift, or path delay asymmetry. The selection of PTP protocol parameters often involves trading off one source of error against the other. This document requires specific PTP protocol configurations, and assumes the use of mechanisms (algorithms), that reduce dTE due to timestamp error but would also – without additional measures – increase dTE due to clock drift to the point where the latter exceeds the allocated error budget. However, this document also assumes additional measures to minimize some sources of dTE due to clock drift and mechanisms and to track and compensate for errors from other sources to a sufficient degree that the error budget is not exceeded.

The specific protocol configurations and other measures, along with their intended effects, are described in Table D.2.

Table D.2 – Protocol configurations & other measures to achieve dTE budget

Configuration or Measure	Description and Intended Effect(s)
Sync Interval 125 ms	Effects: 1. Calibrate the balance between dTE from timestamp error vs error due to clock drift. Larger intervals lead to less timestamp error and more error due to clock drift. 2. Keep below acceptable limits the impact of errors in Rate Ratio and Rate Ratio Drift estimation when keeping ClockTarget in line with ClockSource between arrival of Sync messages. Larger intervals increase the impact of any errors.
Drift_Tracking TLV - syncEgressTimestamp	Effect: Enables calculation of NRR using Sync message timestamps, which eliminates error due to NRR clock drift that would otherwise occur between calculation of NRR using Pdelay_Resp messages and use during Sync message processing (i.e. calculation of Rate Ratio and output Correction Field values)
NRR Smoothing	Description: Algorithm to use timestamps from multiple past Sync messages when estimating NRR. Effect: Reduce the amount of error in the estimate of NRR due to timestamp error while increasing the amount of error due to clock drift.
NRR Drift Tracking & Compensation	Description: Algorithm to use timestamps from multiple past Sync messages to estimate NRR drift then apply compensation to correct for consequent errors in NRR Smoothing calculation. Effect: Mitigate the effect of errors due to clock drift when calculating and using the estimated NRR.

Drift_Tracking TLV – rateRatioDrift	Description: Carries estimate of Rate Ratio drift rate from one node to the next. Effect: Allows each node to estimate its own Rate Ratio drift rate by combining the incoming Rate Ratio drift rate with the local estimate of NRR drift rate.
RR Drift Compensation	Description: Algorithm that uses the estimate of RR drift rate to compensate for that drift, adjusting the estimated RR over time according to the drift rate. Effect: For PTP Relay Instances, minimises errors in the Correction Field caused by Rate Ratio drift. For PTP End Instances, a similar approach can reduce errors in keeping ClockTarget in line with ClockSource between arrival of Sync messages, but is outside the scope of this document.
Pdelay Interval Consistency	Description: This document requires tighter control of the interval between Pdelay messages generated at the Grandmaster PTP Instance than the defaults in IEEE Std 802.1AS-2020. Effect: This document requires the use of Sync messages to calculate NRR (see above). However, when a sufficient number of Sync messages are not available, for example on startup or after a reconfiguration, Pdelay_Resp messages may be used instead. In such cases, errors due to clock drift at Relay Instances have a tendency to cancel out. A clock drift which generates a positive error in NRR measurement on receipt of a Pdelay_Resp message generates a negative error in NRR measurement at the next node. The degree of cancellation depends on the consistency of the intervals over which NRR is measured at neighboring nodes. Tighter control of the Pdelay Interval increases the consistency of the measurement interval and thus decreases the amount of error.
Mean Residence Time	Description: This document defines a mean Residence Time requirement, which is significantly lower than the default maximum Residence Time in IEEE Std 802.1AS-2020. Effect: The amount of error in the Correction Field at the PTP End Instance due to clock drift is proportional to the cumulative meanLinkDelay and residenceTime experienced by a Sync message during transit from the Grandmaster PTP Instance to the PTP End Instance. Specifying a lower mean residenceTime reduces this source of error.

D.2.2 Grandmaster PTP Instance Implementation

Depending on implementation, a Grandmaster PTP Instance can:

a) Contain a single oscillator used for both Local Clock and Clock Source,

b) Contain separate oscillators for Local Clock and Clock Source, or

c) Contain only an oscillator for Local Clock and accept an external input for Clock Source.

In some cases, a Grandmaster PTP instance can support more than one mode of operation and transition between them depending on changes in network configuration (see Splitting, Joining and Aligning Time Domains).

In the first case the rateRatio and rateRatioDrift fields transmitted by the Grandmaster PTP Instance will be zero, reflecting the fact there is no difference between the Local Clock and Clock Source frequencies.

In the second and third cases there can be differences between the Local Clock and Clock Source frequencies. Any differences will be reflected in the rateRatio and rateRatioDrift fields transmitted by the Grandmaster PTP Instance. This means that Grandmaster PTP instances will track rateRatio over time in order to calculate rateRatioDrift, similarly to PTP Relay Instances and PTP End Instances. The exact implementation can vary.

D.2.3 Splitting, Joining and Aligning Time Domains

Modular machines or production cells can allow the splitting and combining of machines if this is required by the production process. When separate, the ClockSources of two machines run separately, each with its own time domain. If both ClockSources are traceable to the same PTP timescale, the difference between the ClockSources may be relatively small. If traceable to different timescales, especially if one or both are ARB timescales, there can be a very large difference between the ClockSources.

When two machines are joined, the first machine’s time domain remains unaffected, and it can continue operation without disruption. There are two typical approaches to how the second machine behaves. In the first case, the second machine’s time domain ceases to exist, with its PTP Instances becoming part of the first machine’s time domain. In the second case, the second machine’s time domain is gradually aligned with the first machine’s time domain such that control loop cycles are coordinated.

D.2.3.1 Joining Machines with Single Time Domain

In the first case, where the second machine’s time domain ceases to exist, a discontinuity in timing for the second machine’s PTP Instances can occur, as they switch to use the first machine’s Grandmaster. Some implementations implement measures to limit such timing discontinuities, but these measures are outside the scope of this document. Typically, in this case, the second machine is not operational while it is joined to the first. It resumes operation once its PTP Instances have synchronized with the first machine’s Grandmaster.

D.2.3.2 Joining Machines with Multiple Coordinated Time Domains

In the second case, where the second machine’s time domain is gradually aligned with the first machine’s time domain, this typically requires both machines to be implementing the same control loop cycle time. The goal is that, once coordinated, each control loop cycle of the first machine will be aligned with the start of a control loop cycle of the second machine, even though the two machines maintain separate time domains and there can be a large difference between their Clock Sources.

In this case, after being joined together, the first machine effectively drives the second machine’s Clock Source faster or slower, during an alignment period, until coordination is achieved. During the alignment period, this drive from the first machine can result in the second machines’ Clock Source temporarily exceeding the usual normative requirement on range of fractional frequency offset relative to the nominal frequency of ± 50 ppm. The usual normative requirement on range of rate of change of fractional frequency offset of ± 1 ppm/s, applicable when split (i.e. independent) or coordinated (i.e. joined and stable, after the alignment period), may also be temporarily exceeded. However, if the value stays with the range ± 1.5 ppm/s, the network-level performance of 1 µs time synchronization accuracy can be maintained. For this reason, this document specifies a separate normative requirement for temporary, externally driven, rate of rate of change of fractional frequency offset.

Since the second machine experiences no time discontinuities and the network-level performance is maintained the second machine can, if desired, continue operation during the alignment period.

Once coordinated, the first machine continues to drive the Clock Source of the second machine to maintain coordination. In this stable, coordinated mode of operation the normal range of ± 1 ppm/s is not exceeded.

The mechanism by which the first machine drives the Clock Source of the second machine is not addressed in this document.

D.2.3.3 Splitting Machines

In the first case, where the second machine’s time domain ceased to exist while joined to the first, splitting machines means that the second machine must create it’s own time domain again. The second machine’s Clock Source typically starts at the PTP Grandmaster Instance’s last, best estimate of the first machine's Clock Source. It can take some time to before the time synchronization accuracy of all the second machine's PTP Instances relative to its Grandmaster can be relied upon. For this reason, the second machine is typically not operational during the split. Hot Standby can be employed to mitigate this transition time, but the details of how to do so are out of scope for this document.

In the second case, where the second machine maintains its time domain while joined to the first, splitting machines means that the first machine ceases driving the second machine's Clock Source to maintain coordination of control loop cycle times. Without this drive, the two-time domains can drift relative to each other. Time synchronization performance within the second machine is maintained during the split and the second machine can, if desired, continue operation throughout the process.

D.2.4 PTP Link Characteristics

A vast majority of time synchronization error due PTP link characteristics is cTE due to asymmetrical path delay in one direction verses the other. The mechanism to measure path delay assumes the link is symmetrical and cannot detect asymmetry, thus asymmetry causes an error. The potential maximum asymmetry and thus error typically scales linearly with physical path length.

The error budget for cTE due to PTP link characteristics for an entire network is 200 ns. In any specific network this budget can be allocated as required with some links allocated a higher budget (typically longer length) than others.

A typical specified maximum delay skew for Category 6 Ethernet cables is 50 ns per 100 m. If such cables are used, a maximum total cable length between Clock Source and Clock Target with 99 PTP Relay Instances between them (i.e., 100 network hops) is 400 m. Extending the cable length beyond 400 m without jeopardizing network-level performance would require the use of cables with less delay skew or asymmetry compensation for delay skew.

It is possible for the delay skew in one section of cable to cancel all or part of a delay skew in the opposite direction from prior section but, depending on how cables are manufactured and deployed, it is feasible for the delay skews of every cable segment between a Grandmaster PTP Instance and a PTP End Instance to be additive.

D.3 Notes on Normative Requirements

D.3.1 Oscillator Requirements

Clock drift at the Grandmaster PTP Instance causes greater dTE than the same amount of clock drift at a PTP Relay Instance or the PTP End Instance. This document therefore requires tighter limits on maximum fractional frequency offset for an oscillator at the Grandmaster PTP Instance than at other instances.

This document does not place requirements on operational temperature range or other environmental factors. The required oscillator behavior is delivered for the operational conditions across which a device claims it is compliant. These conditions typically include temperature range but can also include rate of change of ambient temperature, supply voltage stability, amount of vibration and others.

D.3.2 Timestamp Granularity Error

Timestamp Granularity Error (TSGE) is the error in timestamping each incoming and outgoing message due to the maximum timestamp resolution of which an implementation is capable. It is typically directly related to an implementation's clock rate. For example, a clock rate of 125 MHz typically results in a maximum resolution of 8 ns while a clock rate of 500 MHz typically results in a maximum resolution of 2 ns.

It is assumed that TSGE varies between - 4 ns and + 4 ns with an average of 0 ns. Lower levels of TSGE are better. Implementations where TSGE is higher will find some of the normative requirements difficult or impossible to meet.

D.3.3 Dynamic Timestamp Error

Dynamic Timestamp Error (DTSE) is the, effectively random, error in timestamping each incoming and outgoing event message due to an implementation's inherent inaccuracies, excluding TSGE. It is assumed to vary between a minimum of -6 ns and a maximum of + 6 ns with an average of 0 ns. Lower levels of DTSE are better.

If an implementation timestamps an incoming or outgoing message at a point other than the PHY, any variability in delay between that point and the PHY (PHY delay) will translate to DTSE. Some common implementations were not designed to limit this variability. If care is not taken to avoid implementations with high variability, the assumed DTSE range is easily exceeded. Such implementations will find some of the normative requirements difficult or impossible to meet.

D.3.4 Grandmaster PTP Instance Error Generation Requirements

Table 12 sets normative requirements for error generation at a Grandmaster PTP Instance that ensure the relevant fields in the Sync and Sync_Followup messages it transmits are sufficiently accurate to deliver the network-level performance. Table D.3 describes how the normative requirements align with major sources of error.

Table D.3 – Protocol configurations & other measures to achieve dTE budget

	Normative Requirement	Main Sources of Error
1	preciseOriginTimestamp + correctionField vs Direct measurement of Working Clock at Grandmaster (acting as a Clock Source)	Timestamp Error relative to Clock Source plus accuracy measuring any internal delay between generation of the preciseOriginTimestamp and Sync message transmission.
2	rateRatio vs Direct measurement of Rate Ratio of Clock Source vs Local Clock	Accuracy of internal mechanism to measure Rate Ratio of Clock Source vs. Local Clock, potentially including algorithms that track RateRatioDrift and modify Rate Ratio accordingly[a]
3	syncEgressTimestamp vs Direct measurement of Local Clock	Timestamp Error relative to Local Clock

[a] Only applicable if Clock Source and Local Clock are not locked to the same frequency by the implementation. If they are locked, then rateRatio will be 0 ppm and rateRatioDrift will be 0 ppm/s.

Limits on error generation due to Clock Drift are defined via normative requirements in Table 9.

D.3.5 PTP Relay Instance

Table 13 sets normative requirements for error generation at a PTP Relay Instance that ensure the relevant fields in the Sync and Sync_Followup messages it transmits as part of Sync processing are sufficiently accurate to deliver the network-level time sync performance. The requirements include the ability to mitigate errors in rateRatio and rateRatio drift that would otherwise occur due to clock drift at the current PTP Relay Instance, an adjacent PTP Relay Instance, or the Grandmaster PTP Instance. Table D.4 describes how the normative requirements align with major sources of error.

Table D.4 – Protocol configurations & other measures to achieve dTE budget

	Normative Requirement	Clock Drifts	Main Sources of Error
1	preciseOriginTimestamp + correctionField vs Direct measurement of Clock Source at Grandmaster PTP Instance	None	Timestamp Errors relative to Local Clock when measuring Residence Time, i.e. Sync message ingress and egress. Accuracy of meanLinkDelay measurement. Errors in Rate Ratio used when translating Residence Time measured in terms of Local Clock to Residence time

			in terms of Clock Source, although these are typicaly orders of magnitude smaller than those from Timestamp Errors.
2	rateRatio vs Direct measurement of Rate Ratio of Clock Source vs Local Clock	None	Timestamp Error affecting measurement of NRR when there is no NRR Drift. The effect should be low. This normative requirement is a baseline for the next two requirements.
3		Clock Source (RR Drift)	Accuracy of measurement of NRR when there is no NRR Drift (as above). Accuracy of calculation of rateRatio, including algorithms for RR Drift tracking & error compensation.
4		Clock Source and Local Clock at previous PTP Instance (RR Drift & NRR drift)	Accuracy of measurement of NRR when there is NRR Drift, including algorithms for NRR Drift tracking & error compensation Accuracy of calculation of rateRatio, including algorithms for RR Drift tracking & error compensation. Combined with test 3 this effectively requires a level of performance regarding NRR Drift tracking & error compensation, whether the source of the NRR drift is the Local Clock of the current PTP Instance or the previous PTP Instance.
5	rateRatioDrift vs Direct measurement of Rate Ratio Drift of Clock Source vs Local Clock	None	Timestamp Error affecting measurement of NRR Drift when there is no NRR Drift. The effect should be low. This normative requirement is a baseline for the next two requirements.
6		Clock Source (RR Drift)	Accuracy of measurement of NRR Drift when there is no NRR Drift (as above). Accuracy of calculation of rateRatioDrift, including algorithms for RR Drift tracking & error compensation.
7		Clock Source and Local Clock at previous PTP Instance (RR Drift & NRR drift)	Accuracy of measurement of NRR Drift when there is NRR Drift, including algorithms for NRR Drift tracking & error compensation. Accuracy of calculation of rateRatioDrift, including algorithms for RR Drift tracking & error compensation. Combined with test 6 this effectively requires a level of performance regarding NRR Drift tracking & error compensation, whether the source of the NRR drift is the Local Clock of the current PTP Instance or the previous PTP Instance.
8	syncEgressTimestamp vs	None	Timestamp Error relative to Local Clock

	Direct measurement of Local Clock		

Limits on error generation due to Clock Drift are defined via normative requirements in Table 9.

D.3.6 PTP End Instance

Table 14 sets normative requirements for error generation at a PTP End Instance that ensure the ClockTarget it generates from incoming Sync and Sync_Followup messages is sufficiently accurate to deliver the network-level time sync performance. Table D.5 describes how the normative requirements align with major sources of error.

Table D.5 – Protocol configurations & other measures to achieve dTE budget

	Normative Requirement	Clock Drifts	Main Sources of Error
1	ClockTarget vs ClockSource	None	Timestamp Error affecting measurement of NRR Drift when there is no NRR Drift. The effect should be low. This normative requirement is a baseline for the next two tests.
2		Clock Source (RR Drift)	Accuracy of measurement of NRR Drift when there is no NRR Drift (as above). Accuracy of calculation of rateRatioDrift, including algorithms for RR Drift tracking & error compensation.
3		Clock Source and Local Clock at previous PTP Instance (RR Drift & NRR drift)	Accuracy of measurement of NRR Drift when there is NRR Drift, including algorithms for NRR Drift tracking & error compensation. Accuracy of calculation of rateRatioDrift, including algorithms for RR Drift tracking & error compensation. Combined with test 2 this effectively requires a level of performance regarding NRR Drift tracking & error compensation, whether the source of the NRR drift is the Local Clock of the current PTP Instance or the previous PTP Instance.

Limits on error generation due to Clock Drift are defined via normative requirements in Table 9.

D.4 Approach to Testing Normative Requirements

D.4.1 General

This document does not specify tests to ensure conformance with the normative requirements. However, it is important that the normative requirements are, in principle, testable. Clause D.4 describes, at a high level, approaches a test specification might take to testing conformance with some of the normative requirements related to time synchronization.

It is assumed that test equipment can precisely measure the output of the ClockSource (at a Grandmaster PTP Instance), ClockTarget (at a PTP End Instance) and Local Clock (at any PTP Instance) to ensure conformance with frequency offset and frequency offset drift requirements.

This might be via a Pulse per Second (PPS) plus Time-of-Day information or another mechanism.

It is also assumed that test equipment can generate sequences of PTP messages with precise timing and content (for testing PTP Relay Instances and PTP End Instances) and receive, log, and process sequences of PTP messages with precise timing measurement, e.g. of message arrival.

D.4.2 Testing Grandmaster PTP Instance

Figure D.1 illustrates an approach to testing the three normative requirements discussed in D.3.4.

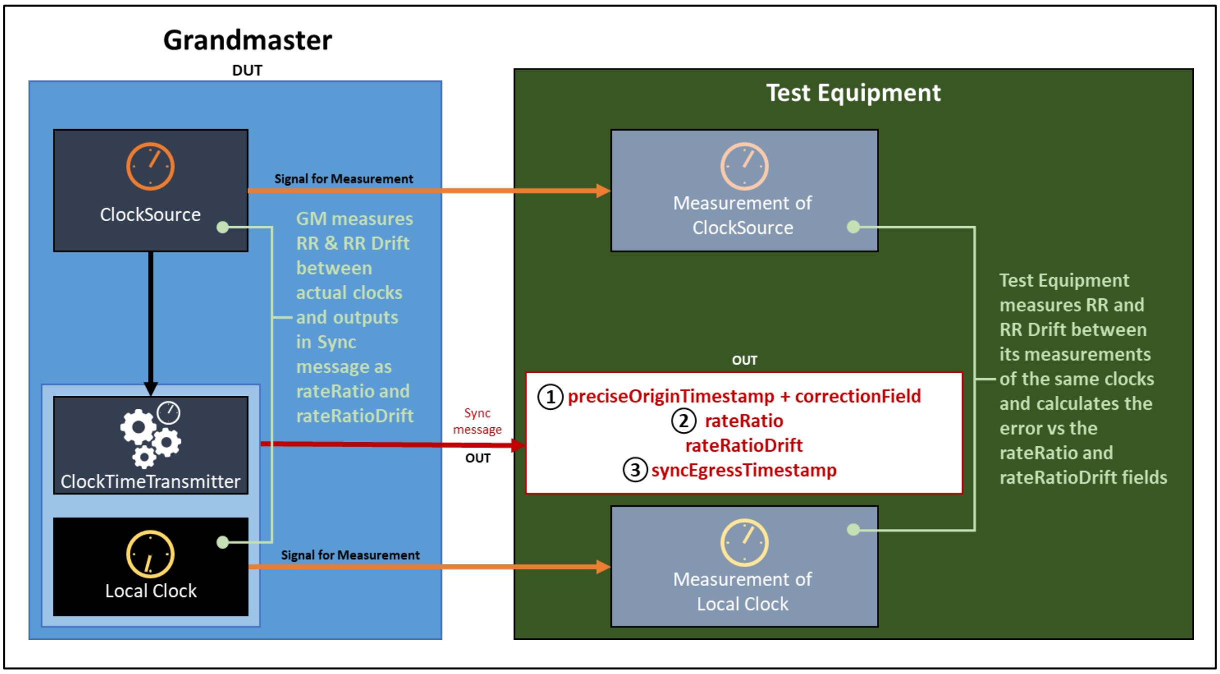

Grandmaster
DUT
ClockSource
Signal for Measurement
GM measures RR & RR Drift between actual clocks and outputs in Sync message as rateRatio and rateRatioDrift
ClockTimeTransmitter
Sync message
OUT
Local Clock
Signal for Measurement
Test Equipment
Measurement of ClockSource
OUT
① preciseOriginTimestamp + correctionField
② rateRatio
rateRatioDrift
③ syncEgressTimestamp
Test Equipment measures RR and RR Drift between its measurements of the same clocks and calculates the error vs the rateRatio and rateRatioDrift fields
Measurement of Local Clock

Figure D.1 – Approach to Testing Normative Requirements for Grandmaster PTP Instance

The test equipment can calculate the time the Sync message is output at the DUT by subtracting the link delay from the measured arrival time at the test equipment.

For test 1, the test equipment compares the value of the preciseOriginTimestamp + correctionField against its measurement of the ClockSource.

For tests 2, the test equipment compares the value in the rateRatio field with its calculation of the equivalent value based on its measurement of the ClockSource.

For test 3, the test equipment compares the value of the syncEgressTimestamp against its measurement of the Local Clock.

D.4.3 Testing PTP Relay Instance

Figure D.8 illustrates an approach to testing normative requirements 1, 2, 5 and 8 discussed in D.3.5.

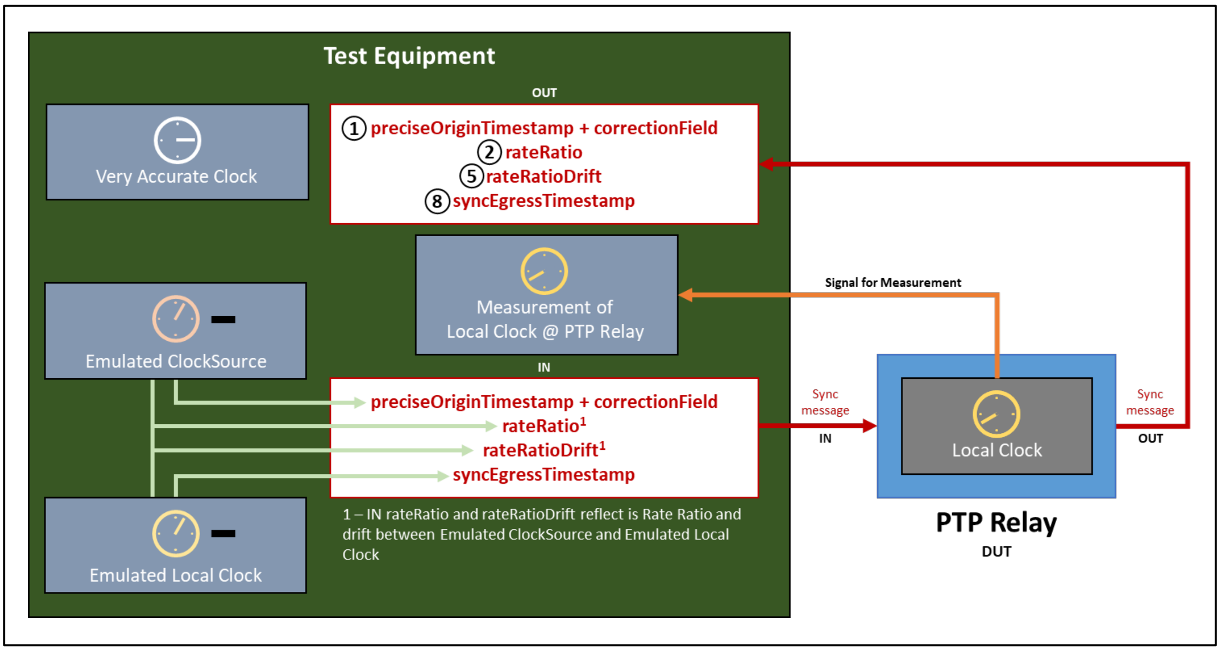


Figure D.2 – Approach to Testing Normative Requirements for PTP Relay Instance - 1

The test equipment can compare the DUT's output Sync message to the expected result given the measurement of the Local Clock and the timing of the input Sync message transmission and output Sync message reception.

For these four tests, the Emulated ClockSource and Emulated Local Clock are stable and in sync. In practice, both can be equal to the test equipment's Very Accurate Clock. In the input Sync message, rateRatio will be 0 ppm and rateRatioDrift will be 0 ppm/s. If the Local Clock of the PTP Relay Instance is also stable, it will measure NRR of 0 ppm and NRR Drift of 0 ppm/s.

The test equipment can calculate the time the output Sync message is output at the DUT by subtracting the link delay from the measured arrival time at the test equipment.

For test 1, the test equipment can compare the increase in the value of the correctionField to the measured meanLinkDelay (from the test equipment to the DUT) plus residenceTime. The test equipment will need to account for the additional delay between the PTP Relay Instance's transmission of the input Sync message and its reception by the test equipment.

For tests 2 and 5, the test equipment can compare the rateRatio and rateRatioDrift fields in the output Sync message with the equivalent calculated values between the measured Local Clock and the Emulated ClockSource.

For test 8, the test equipment can compare syncEgressTimestamp value in the ouput Sync message with its measurement of the Local Clock.

Figure D.8 illustrates an approach to testing normative requirements 3 and 6 discussed in D.3.5.

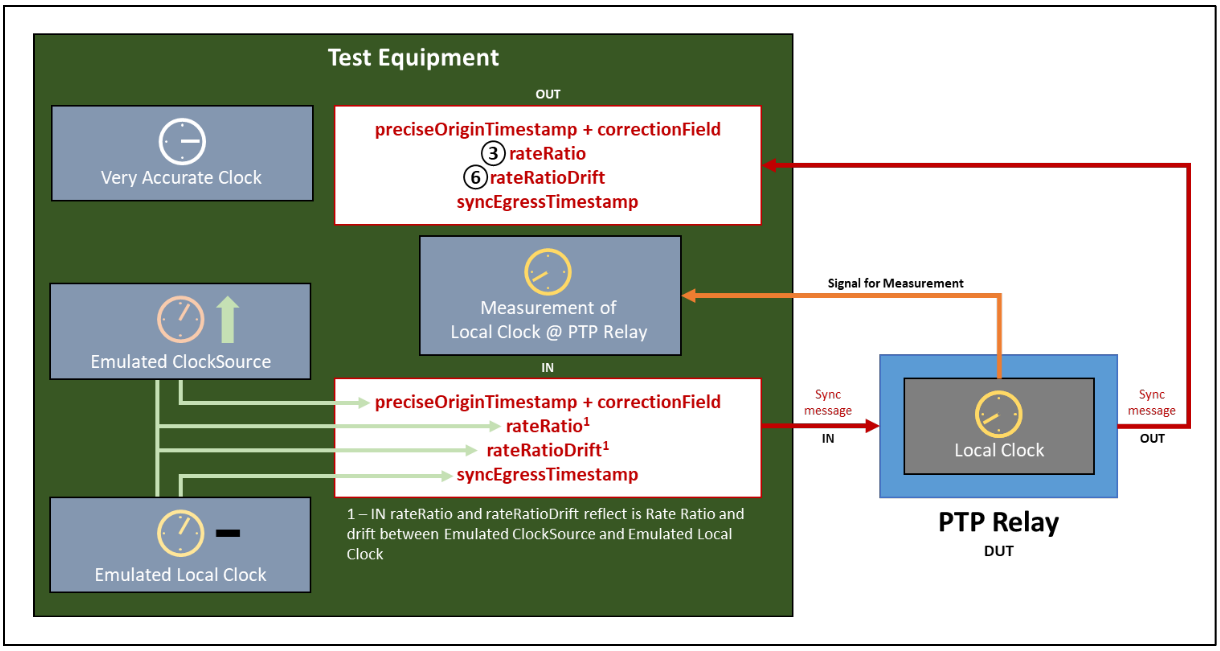


Figure D.3 – Approach to Testing Normative Requirements for PTP Relay Instance - 2

For these two tests, the fractional frequency offset of the Emulated ClockSource is increasing at a defined ppm/s rate relative to the Very Accurate Clock. The Emulated Local Clock is stable; in practice, it can be equal to the test equipment's Very Accurate Clock. In the output Sync message, the rateRatio field will increase over time, and the rateRatioDrift field will maintain a matching positive value. If the Local Clock of the PTP Relay Instance is also stable, it will measure NRR of 0 ppm and NRR Drift of 0 ppm/s.

For tests 3 and 6, the test equipment can compare the rateRatio and rateRatioDrift fields in the output Sync message with the equivalent calculated values between the measured Local Clock and the Emulated ClockSource.

Figure D.8 illustrates an approach to testing normative requirements 4 and 7 discussed in D.3.5.

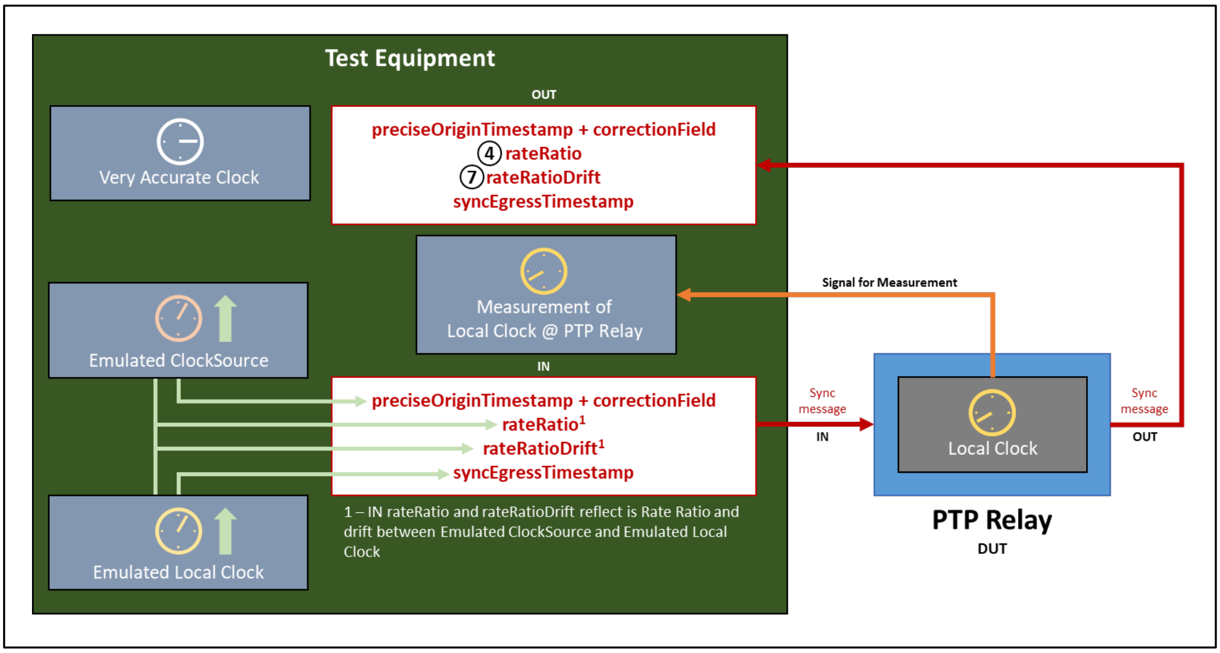


Figure D.4 – Approach to Testing Normative Requirements for PTP Relay Instance - 3

For these two tests, the fractional frequency offsets of the Emulated ClockSource and the Emulated Local Clock are equal and increasing at a defined ppm/s rate relative to the Very Accurate Clock. In the output Sync message, the rateRatio field will be 0 ppm, and the rateRatioDrift field will be 0 ppm/s. If the Local Clock of the PTP Relay Instance is stable, the NRR it measures will increase over time and the NRR Drift it measures will maintain a matching positive value.

For tests 4 and 7, the test equipment can compare the rateRatio and rateRatioDrift fields in the output Sync message with the equivalent calculated values between the measured Local Clock and the Emulated ClockSource.

D.4.4 Testing PTP End Instance

Figure D.8 illustrates an approach to testing the three normative requirements discussed in D.3.6.

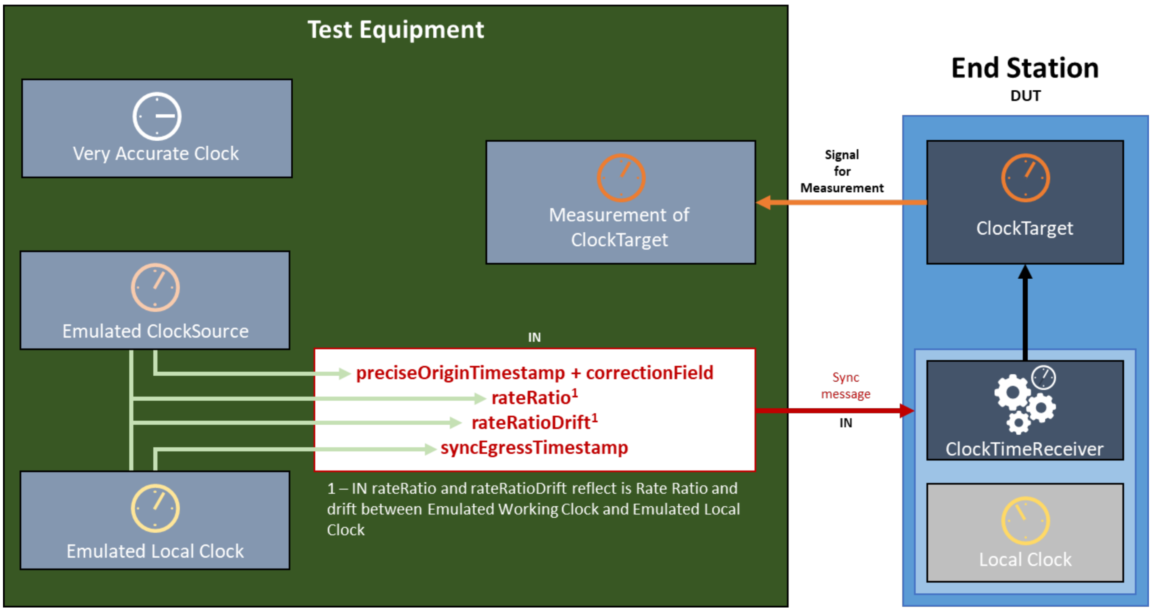


Figure D.5 – Approach to Testing Normative Requirements for PTP End Instance

The test equipment can compare its measurement of the DUT's ClockTarget to the Emulated ClockSource. It will need to account for the additional delay between its transmission of the input Sync message and the reception of the message by the DUT.

For test 1, the Emulated ClockSource and Emulated Local Clock are stable and in sync. In practice, both can be equal to the test equipment's Very Accurate Clock. In the input Sync message, rateRatio will be 0 ppm and rateRatioDrift will be 0 ppm/s. If the Local Clock of the PTP End Instance is also stable, it will measure NRR of 0 ppm and NRR Drift of 0 ppm/s.

For test 2, the fractional frequency offset of the Emulated ClockSource is increasing at a defined ppm/s rate relative to the Very Accurate Clock. The Emulated Local Clock is stable; in practice, it can be equal to the test equipment's Very Accurate Clock. In the output Sync message, the rateRatio field will increase over time, and the rateRatioDrift field will maintain a matching positive value. If the Local Clock of the PTP Relay Instance is also stable, it will measure NRR of 0 ppm and NRR Drift of 0 ppm/s.

For test 3, the fractional frequency offsets of the Emulated ClockSource and the Emulated Local Clock are equal and increasing at a defined ppm/s rate relative to the Very Accurate Clock. In the output Sync message, the rateRatio field will be 0 ppm, and the rateRatioDrift field will be 0 ppm/s. If the Local Clock of the PTP Relay Instance is stable, the NRR it measures will increase over time and the NRR Drift it measures will maintain a matching positive value.

D.5 Example Algorithms

D.5.1 General

This document does not place normative requirements on the use of specific algorithms. However, the normative requirements assume the use of algorithms to reduce the effect of errors in meanLinkDelay and to track clock drift and compensate for consequent errors. PTP instances that do not implement algorithms will find it difficult or impossible to meet the normative requirements.

D.5 provides examples of algorithms that can be used for:

- Tracking NRR drift.
- Correcting for errors in measured NRR (mNRR) due to NRR drift.
- Calculating RR drift.
- Correcting for errors in measured RR (mRR) due to RR drift.
- Reducing the effect of errrors in meanLinkDelay

For measured NRR, measured RR, and meanLinkDelay, an example for how startup behavior can be handled is provided.

NRR Drift Tracking and Error Correction is carried out for each network hop, i.e. at every node other than the Grandmaster. It is based on pairs of timestamps with each pair associated with a Sync message transmitted from the previous node (n-1) to the current node (n).

- t_{s1outP} – Timestamp of the Sync message egress from the **previous** node (n-1), timestamped by that node’s Local Clock. Unit: **ns**.
- t_{s2in} – Timestamp of the Sync message ingress to the current node (n), timestamped by that node’s Local Clock. Unit: **ns**.

All timestamps are affected by Timestamp Errors.

The algorithm uses information from the 32 most recent Sync messages. However, a node need only keep track of the 9 most recent pairs of timestamps from the most recent (x) to the 9th most recent (x-8) Sync message. The algorithm generates one measurement of NRR using the prior 2 s of Sync message data (on average, based on a nominal Sync Interval of 125 ms), and a second measure based on the 2 s of Sync message data prior to that. It then uses the difference in the two measurements over the interval between the effective measurement points to calculate the NRR drift rate.

On arrival of a new timestamp pair (x), a node executes a NRR calculation:

$$NRRcalc(x) = \left(\frac{t_{s1outP}(x) - t_{s1outP}(x-8)}{t_{s2in}(x) - t_{s2in}(x-8)} - 1\right) \times 10^6 \qquad \textbf{ppm}$$

with an associated effective measurement point:

$$NRRcalcT(x) = \frac{t_{s2in}(x) + t_{s2in}(x-8)}{2} \qquad \textbf{ns}$$

A node keeps track of the 24 most recent NRR calculations and effective measurement points, from the most recent (x) to the 24th most recent (x-23).

After of a new most-recent NNR calculation, a node calculates an NRR drift rate:

$$NRRaverageA = \sum_{i=x-7}^{x} \frac{mNRRcalc(i)}{8} \qquad \textbf{ppm}$$

$$NRRaverageB = \sum_{i=x-23}^{x-16} \frac{mNRRcalc(i)}{8} \qquad \textbf{ppm}$$

$$NRRdriftInterval = \sum_{i=x-7}^{x} \frac{mNRRcalcT(i)}{8} - \sum_{i=x-23}^{x-16} \frac{mNRRcalcT(i)}{8} \qquad \textbf{ns}$$

$$NRRdriftRate(n) = \left(\frac{NRRaverageA - NRRaverageB}{NRRdriftInterval}\right) \times 10^9 \qquad \textbf{ppm/s}$$

where *NRRdriftRate(n)* is the NRR drift rate for the current Node n.

D.5.2 Algorithm to Compensate for Errors in measured NRR due to Clock Drift

The algorithm to measure NRR uses data from the previous 1 s of Sync message data, combined with the NRR drift estimate from the previous step. This smaller amount of data (vs. that used for either of the NRR measurements in the previous step) is employed as it improves responsiveness to sudden changes in NRR drift with minimal loss of accuracy.

On arrival of a new timestamp pair (x), a node executes a NRR calculation:

$$mNRRcalc(x) = \left(\frac{t_{s1outP}(x) - t_{s1outP}(x-4)}{t_{s2in}(x) - t_{s2in}(x-4)} - 1\right) \times 10^6 \qquad \textbf{ppm}$$

with an associated effective measurement point:

$$mNRRcalcT(x) = \frac{t_{s2in}(x) + t_{s2in}(x-4)}{2} \qquad \textbf{ns}$$

A node keeps track of the 4 most recent mNRR calculations and effective measurement points, from the most recent (x) to the 4th most recent (x-3). (The mNRR calculations use information from the 5 most recent Sync messages, but the node is already keeping track of information from the 9 most recent Sync messages for the NRR drift tracking algorithm.)

The node then calculates an error corrected measured NRR value.

$$For\ i = x\ to\ (x-3)$$

$$mNRRcorrected(i) = mNRRcalc(i) + \left(mNRRdriftRate(n) \times \frac{(t_{s2in}(x) - mNRRcalcT(i))}{10^9}\right) \qquad \textbf{ppm}$$

$$mNRR = \sum_{i=x-3}^{x} \frac{mNRRcorrected(i)}{4} \qquad \textbf{ppm}$$

The result is a measured NRR value, error-corrected to the time when the most recent Sync message was received.

D.5.3 Measured NRR Algorithm – Startup Behaviour

NRR is used when calculating meanLinkDelay and output Sync message fields. The first NRR drift calculation will only be available after receipt of 32 Sync messages, i.e. after approximately 4 seconds of operation given the 125 ms Sync Interval. During this time meanLinkDelay and output Sync messages fields must still be calculated, so an alternative must be used, even if it can not deliver the same assurances regarding network-level performance.

If measured NRR from Sync message information is unavailable but equivalent information from Pdelay_Resp messages is available it may be substituted for Sync message information. However, measuring NRR using Pdelay_Resp messages is vulnerable to additional error due to clock drift between the time NRR is measured, on receipt of the latest Pdelay_Resp message, and use of the measurement during Sync message processing. This is the reason using Sync message information is preferable. It also means that a switch to using Sync message information as soon as possible is desirable. It is technically possible to calculate a NRR using a combination of Pdelay_Resp and Sync messages but this can be risky due to the potential for very short intervals between messages and resulting high error due to timestamp errors, so it not recommended.

The following describes potential startup behaviour when using either Sync or Pdelay_Resp message information. It is the responsibility of implementers to decide whether and when to use Pdelay_Resp message information and when to switch to using Sync message information. It is, however, a normative requirement that implementations use Sync message information when information from 32 or more timely Sync messages is available.

The following describes potential startup behaviour applicable to either Sync or Pdelay_Resp message information.

At least two two messages must be received before calculating a NRR value.

Prior to two messages being received, NRR = 1 (i.e., 0 ppm) should be used.

Once two messages have been received, NRR should be calculated using the formula:

2nd message: $mNRR = \left(\left(\frac{t_3(x) - t_3(x-1)}{t_4(x) - t_4(x-1)}\right) - 1\right) \times 10^6$ **ppm**

Where:

- t_3 – Timestamp of the Pdelay_Resp message egress from the previous node (n-1), timestamped by that node’s Local Clock. Unit: **ns**.
- t_4 – Timestamp of the Pdelay_Resp message ingress to the current node (n), timestamped by that node’s Local Clock. Unit: **ns**.

When three to four messages have been received, NRR should be calculated using the following formulae:

3rd message: $mNRR = \left(\left(\frac{t_{1outP}(x) - t_{1outP}(x-2)}{t_{2in}(x) - t_{2in}(x-2)}\right) - 1\right) \times 10^6$ **ppm**

4th message: $mNRR = \left(\left(\frac{t_{1outP}(x) - t_{1outP}(x-3)}{t_{2in}(x) - t_{2in}(x-3)}\right) - 1\right) \times 10^6$ **ppm**

On arrival of the 5th Sync message the first mNRRcalc and mNRRcalcT calculations can take place and should be used for NRR:

$$mNRRcalc(x) = \left(\frac{t_{s1outP}(x) - t_{s1outP}(x-4)}{t_{s2in}(x) - t_{s2in}(x-4)} - 1\right) \times 10^6 \qquad \textbf{ppm}$$

$$mNRRcalcT(x) = \frac{t_{s2in}(x) + t_{s2in}(x-4)}{2} \qquad \textbf{ns}$$

5th Sync message: $mNRR = mNRRcalc(x)$ **ppm**

As the 6th, 7th and 8th messages arrive an average can be taken and used for NRR, so:

6th message: $mNRR = \sum_{i=x-1}^{x} \frac{mNRRcalc(i)}{2}$ **ppm**

7th message: $mNRR = \sum_{i=x-2}^{x} \frac{mNRRcalc(i)}{3}$ **ppm**

8th message: $mNRR = \sum_{i=x-3}^{x} \frac{mNRRcalc(i)}{4}$ **ppm**

For the 9th to the 31st message, the same equation as for the 8th message can be used.

Once the 32nd message arrives, the regular equations with NRR drift tracking and error correction can be used.

D.5.4 Algorithm for Tracking RR Drift

A Sync or Sync_Followup message carries the rateRatio field, which informs each node of the previous node's estimate of its (the previous node's) Rate Ratio. This document also requires support for the Drift_Tracking TLV that carries the rateRatioDrift field, which informs each node of the previous node's estimate of its (the previous node's) Rate Ratio Drift.

If the implementation of the Grandmaster PTP Instance means the ClockSource and Local Clock (at the Grandmaster PTP Instance) are linked such that the two are always operating at the same frequency, the rateRatio field received by the first node (Node 1) will always be 0 ppm and the rateRatioDrift field will always be 0 ppm/s. Thus, at Node 1, RR will equal NRR, RR Drift will equal NRR Drift, and therefore D.5.2 and D.5.3 describe how to calculate RR and RR Drift at Node 1.

If the implementation of the Grandmaster PTP Instance means the ClockSource and Local Clock (at the Grandmaster PTP Instance) can operate at different frequencies, the implementation populates the rateRatio and rateRatioDrift field with values reflecting those differences.

In either case all PTP Instances, other than the Grandmaster PTP Instance, calculate an estimate of the local Rate Ratio Drift when the latest Sync Message is received, based on the received rateRatioDrift field and the local measure of NRR Drift. The Rate Ratio Drift Rate from the previous node is in ppm/s relative to the timebase of its Local Clock (i.e. the "s" in "ppm/s"). For highest precision, this can be converted to the timebase of the current node's Local Clock.

$$rateRatioDrift(n) = \frac{rateRatioDrift(n-1)}{\left(1 + \frac{mNRRc(n)}{10^6}\right)} + NRRdriftRate(n) \qquad \textbf{ppm/s}$$

However, given that adding ppm/s already lacks the precision of multiplying actual ratios, this simplification delivers similarly accurate results.

$$rateRatioDrift(n) = rateRatioDrift(n-1) + NRRdriftRate(n) \qquad \textbf{ppm/s}$$

D.5.5 Algorithm to Compensate for Errors in measured RR due to Clock Drift

On receipt of a Sync or Sync_Folloup message, all PTP Relay Instances estimate a measured RR (mRR) based on the received rateRatio field and the local measure of NRR. An mRR value is used to translate the sum of meanLinkDelay and residenceTime from Local Clock timebase into Grandmaster timebase. An mRR value is also passed in the transmitted Sync or Sync_Followup message's rateRatio field to the next node. Errors in these estimates due to clock drift can be reduced by taking account of RR Drift. Since the optimal point in time for each estimate is different, the amount of applicable RR Drift is different, and hence the estimates will be different.

(For discussion of how different Grandmaster PTP Implementations affect the behaviour of a PTP Relay Instance at Node 1 – or not – see D.5.4.)

A PTP End Instance is similar in that it estimates mRR on receipt of a Sync message, subsequently uses an mRR value, and errors in the latter due to clock drift can be mitigated by taking account of RR Drift. However, unlike a PTP Relay Instance, the mRR value is used to keep the ClockTarget in line with the ClockSource and there is no need to transmit a rateRatio field to a subsequent node.

For a PTP Relay Instance there are three points in time of interest:

- Point A: Receipt of the Sync Message by the current node (Node n)
- Point B: Mid-point between transmission of the Sync message by the previous node (Node $n\text{-}1$) and transmission of the consequent Sync message by the current node (Node n)
- Point C: Transmission of the Sync Message by the current node (Node n)

Figure D.8 illustrates these points and the associated calculations.

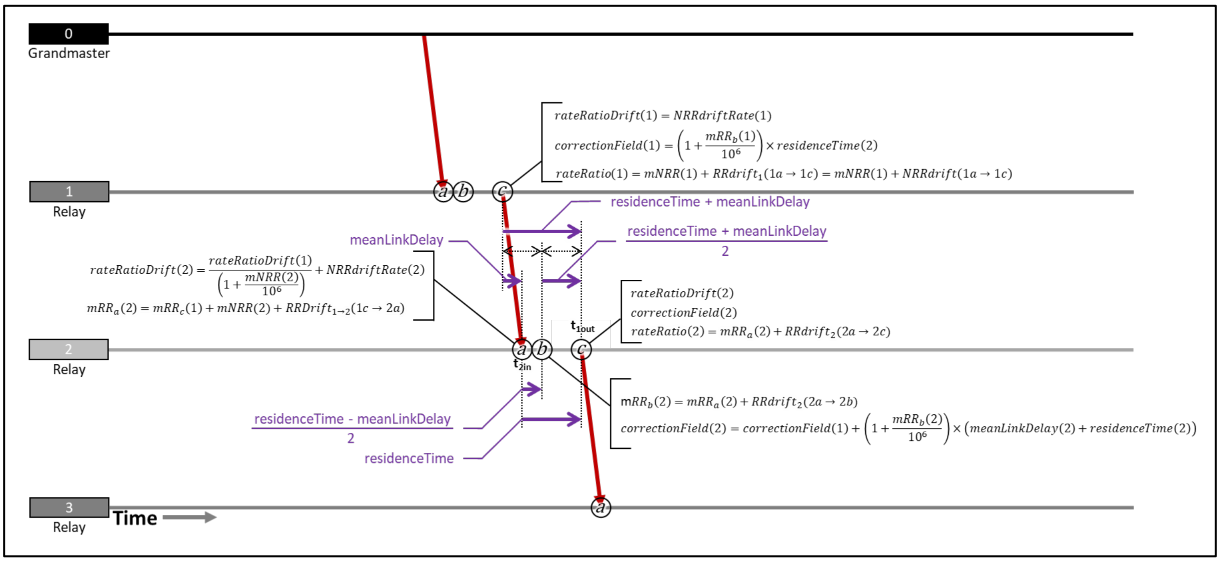

0
Grandmaster
1
Relay
2
Relay
3
Relay
Time
$rateRatioDrift(1) = NRRdriftRate(1)$
$correctionField(1) = \left(1 + \frac{mRR_b(1)}{10^6}\right) \times residenceTime(2)$
$rateRatio(1) = mNRR(1) + RRdrift_1(1a \to 1c) = mNRR(1) + NRRdrift(1a \to 1c)$
a
b
c
residenceTime + meanLinkDelay
meanLinkDelay
residenceTime + meanLinkDelay / 2
$rateRatioDrift(2) = \frac{rateRatioDrift(1)}{\left(1 + \frac{mNRR(2)}{10^6}\right)} + NRRdriftRate(2)$
$mRR_a(2) = mRR_c(1) + mNRR(2) + RRDrift_{1 \to 2}(1c \to 2a)$
$rateRatioDrift(2)$
$correctionField(2)$
$rateRatio(2) = mRR_a(2) + RRdrift_2(2a \to 2c)$
t_{1out}
t_{2in}
$mRR_b(2) = mRR_a(2) + RRdrift_2(2a \to 2b)$
$correctionField(2) = correctionField(1) + \left(1 + \frac{mRR_b(2)}{10^6}\right) \times (meanLinkDelay(2) + residenceTime(2))$
residenceTime - meanLinkDelay / 2
residenceTime

Figure D.6 – RR Drift Tracking and Error Compensation Calculations – PTP Relay Instance

The estimate of RR when the Sync message arrives can be calculated as follows.

$$mRR_a(n) = rateRatio(n-1) + RRdrift_{n-1 \to n}((n-1)_c \to n_a) + mNRR(n) \qquad \textbf{ppm}$$

$$= rateRatio(n-1) + \left(\frac{rateRatioDrift(n-1)}{\left(1 + \frac{mNRRc(n)}{10^6}\right)} \times meanLinkDelay(n) \right) + mNRR(n) \qquad \textbf{ppm}$$

Where $RRdrift_{n-1/n}((n-1)_c \to n_a)$ is amount $rateRatio(n-1)$ drifts between transmission of the Sync message at Node n-1 and reception at Node n. This is $rateRatioDrift(n-1)$ multipled by meanLinkDelay but, since meanLinkDelay is measured in terms of Node n's Local Clock and rateRatioDrift is in terms of Node n-1's Local Clock the latter should be divided by the NRR for the highest accuracy.

However, given that adding ppm/s already lacks the precision of multiplying actual ratios, this simplification delivers similarly accurate results.

$$mRR_a(n) = rateRatio(n-1) + \left(rateRatioDrift(n-1) \times meanLinkDelay(n)\right) + mNRRc(2) \quad \textbf{ppm}$$

Once the time when Node n transmits the consequent Sync message is known, the correctionField value can be calculated.

$$mRR_b(n) = mRR_a(n) + RRdrift_n(a \to b) \quad \textbf{ppm}$$

$$= mRR_a(n) + \left(rateRatioDrift(n) \times \frac{residenceTime(n) - meanLinkDelay(n)}{2}\right) \quad \textbf{ppm}$$

$$correctionField(n) = correctionField(n-1) + \left(1 + \frac{mRR_b(n)}{10^6}\right) \times \left(meanLinkDelay(2) + residenceTime(2)\right) \textbf{ns}$$

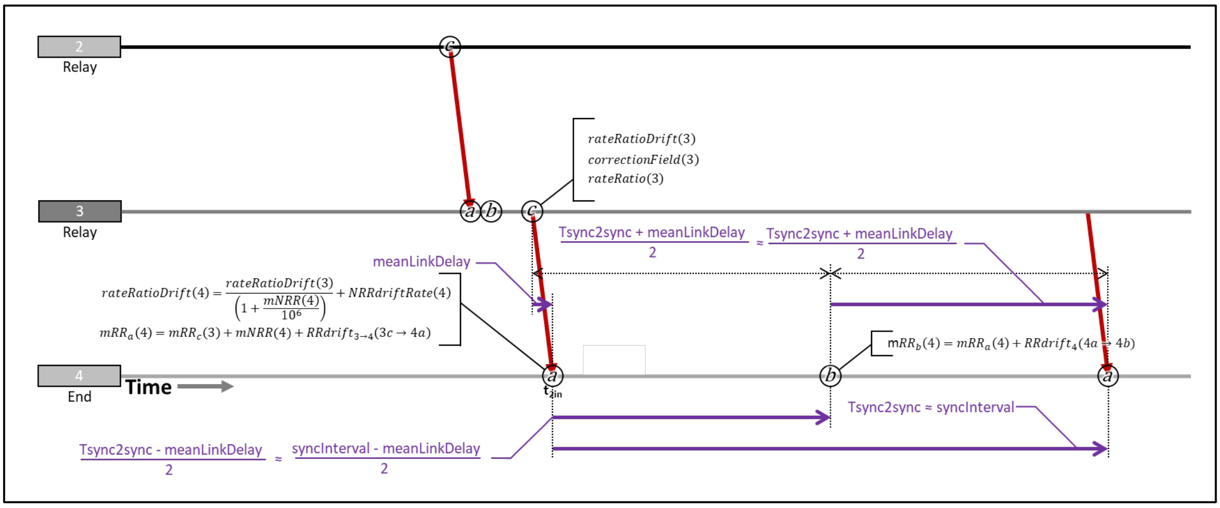
And the rateRatio field.

$$rateRatio(n) = mRR_a(n) + RRdrift_n(a \to c) \quad \textbf{ppm}$$

$$= mRR_a(n) + \left(RRdriftRate(n) \times residenceTime(n)\right) \quad \textbf{ppm}$$

D.5.6 Compensate for Errors in measured RR due to Clock Drift at PTP End Instance

Figure D.8 illustrates a possible approach to applying similar RR drift tracking and error compensation at a PTP End Instance.



2
Relay
c
rateRatioDrift(3)
correctionField(3)
rateRatio(3)
3
Relay
a
b
c
Tsync2sync + meanLinkDelay / 2 ≈ Tsync2sync + meanLinkDelay / 2
meanLinkDelay
$rateRatioDrift(4) = \frac{rateRatioDrift(3)}{\left(1 + \frac{mNRR(4)}{10^6}\right)} + NRRdriftRate(4)$
$mRR_a(4) = mRR_c(3) + mNRR(4) + RRdrift_{3\to4}(3c \to 4a)$
$mRR_b(4) = mRR_a(4) + RRdrift_4(4a \to 4b)$
4
End
Time
a
t_{2in}
b
a
Tsync2sync ≈ syncInterval
Tsync2sync - meanLinkDelay / 2 ≈ syncInterval - meanLinkDelay / 2

Figure D.7 – RR Drift Tracking and Error Compensation Calculations – PTP End Instance

The initial calculations for rateRatioDrift and mRRa are exactly the same as for a PTP Relay Instance. Instead of using RR to translate meanLinkDelay + residenceTime from the Local Clock timebase to the Grandmaster timebase – as is done at a PTP Relay Instance – a PTP End Instance uses RR to keep its ClockTarget in line with the ClockSource. If only one value of RR is used for this purpose, the optimal value is not mRR_a; it is mRR_b, where Point b is halfway between the most recently received Sync message and the next Sync message.

Of course, the exact interval until the next Sync message's arrival ($Tsync2sync$ in Figure D.8) can't be known before it happens, but the Rate Ratio value is required as soon as possible after arrival of the most recent Sync message. The solution is to use the nominal value of the interval, i.e. syncInterval, which is 125 ms.

$$mRR_b(n) = mRR_a(n) + RRdrift_n(a \to b) \quad \textbf{ppm}$$

$$= mRR_a(n) + \left(rateRatioDrift(n) \times \frac{syncInterval}{2}\right) \quad \textbf{ppm}$$

$$= mRR_a(n) + (rateRatioDrift(n) \times 0.0625) \quad \textbf{ppm}$$

It is also possible to use more complex algorithms that repeatedly or continuously adjust the mRR value between Sync messages, but such an approach is not addressed this document.

D.5.7 Mean Link Delay Averaging

The actual Path Delay from one node to the next – for a wired connection – is very stable and errors measuring it due to Timestamp Error average to zero. Thus, taking a long average or applying a low-pass filter with a low bandwidth is an effective way to reduce error in meanLinkDelay. Care needs to be taken during system startup or after any other initialisation of the algorithm, to quickly converge on a stable value.

The basic Pdelay calculation, used by the Common Mean Link Delay service, remains the same. Figure D.8 illustrates it.

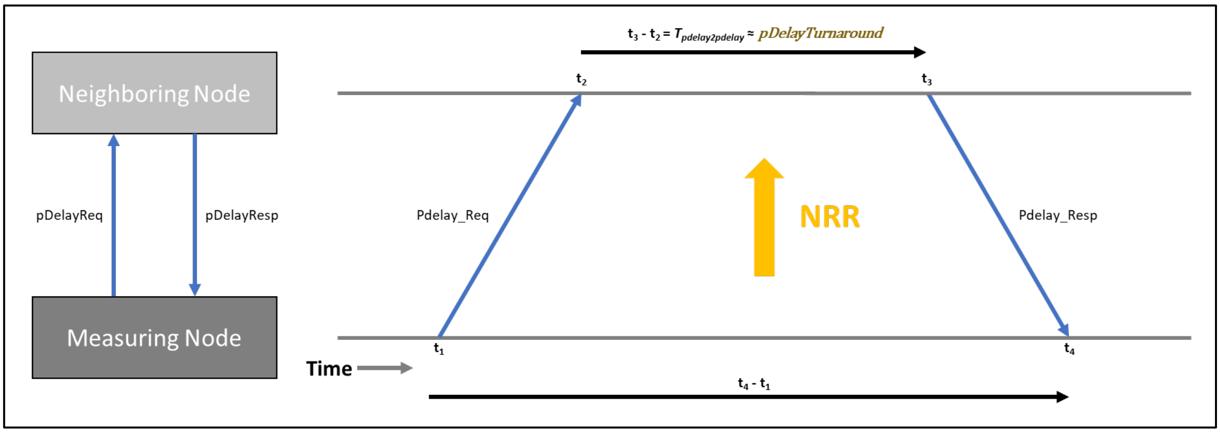


Figure D.8 – Signals and timestamps to measure path delay

Following each Pdelay_Req – Pdelay_Resp exchange, the measured path delay ($mPathDelay$) is calculated.

For the x^{th} message after initialisation…

$$mPathDelay(x) = \frac{(t_4 - t_1) - \frac{(t_3 - t_2)}{NRR}}{2} \qquad \textbf{ns}$$

The meanLinkDelay is then updated via an IIR (Infinite Impulse Response) filter. For the first couple of minutes after initialization the filter is in initialization mode.

If $x < 1000$ then $f = x$ else $f = 1000$

$$meanLinkDelay(x) = \frac{(meanLinkDelay(x-1) \times (f-1)) + pDelay(x)}{f} \qquad \textbf{ns}$$

For example:

$$meanLinkDelay(100) = \frac{(meanLinkDelay(99) \times (99)) + pDelay(x)}{100} \qquad \textbf{ns}$$

$$meanLinkDelay(5836) = \frac{(meanLinkDelay(5835) \times (999)) + pDelay(x)}{1000} \qquad \textbf{ns}$$

It is possible to automatically reinitialise the algorithm if an $mPathDelay$ value, or series of values, deviates too much from the $meanLinkDelay$, but the details are not addressed in this document.

Bibliography

- Best, Roland E., Phase-Locked Loops, Design, Simulation, and Applications, Fifth Edition, 2003.
- Gardner, Floyd M., Phaselock Techniques, Second Edition, 1979.
- IEC 61784-2 (all parts), *Industrial networks - Profiles - Part 2: Additional real-time fieldbus profiles based on ISO/IEC/IEEE 8802-3*
- IEEE Std 1588-2019, *IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*
- IEEE Std 802-2014, *IEEE Standard for Local and Metropolitan Area Networks: Overview and Architecture*
- IEEE Std 802c-2017, *IEEE Standard for Local and Metropolitan Area Networks: Overview and Architecture Amendment 2: Local Medium Access Control (MAC) Address Usage*
- IETF RFC 6020, Bjorklund, M., YANG: *A Data Modeling Language for the Network Configuration Protocol (NETCONF)*, October 2010, available at https://www.rfc-editor.org/info/rfc6020
- IETF RFC 7224, Bjorklund, M., *IANA Interface Type YANG Module*, May 2014, available at https://www.rfc-editor.org/info/rfc7224
- *)*, October 2010, available at https://www.rfc-editor.org/info/rfc6020
- IETF RFC 8995, Pritikin, M., Richardson, M., Eckert, T., Behringer, M., and Watsen, K., *Bootstrapping Remote Secure Key Infrastructure (BRSKI)*, May 2021, available at https://www.rfc-editor.org/info/rfc8995
- ITU-T Recommendation G.8260, *Definitions and terminology for synchronization in packet networks*
- ITU-T Series G Supplement 65, Simulations of transport of time over packet networks, Geneva, October 2018.
- Ogata, Katsuhiko, Modern Control Engineering, Second Edition, Prentice Hall, 1990.
- Rogers, John, Plett, Calvin, Dai, Foster, Integrated Circuit Design for High-Speed Frequency Synthesis, Artech House, 2006.
- Wolaver, Dan H., Phase-Locked Loop Circuit Design, Prentice Hall, 1991.