P1588e™/D1.1 Draft Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems Amendment: MIB and YANG Modules

Sponsor

IEEE Instrumentation and Measurement Society/ TC9 - Sensor Technology (IM/ST)
of the
IEEE Instrumentation and Measurement Society

Approved <XX MONTH 20XX>

IEEE SA Standards Board

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

Abstract:
This draft standard is an amendment of IEEE Std 1588™-2019. This amendment identifies the structure and content of the IEEE 1588 MIB and YANG modules.

Keywords:
PTP, MIB, YANG, Management

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

PDF: ISBN 978-0-XXXX-XXXX-X STDXXXXX
Print: ISBN 978-0-XXXX-XXXX-X STDPDXXXXX

Important Notices and Disclaimers Concerning IEEE Standards Documents

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

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

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

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

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

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

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

Translations

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

Official statements

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

Comments on standards

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

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

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

Laws and regulations

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

Data privacy

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

Copyrights

Photocopies

Updating of IEEE Standards documents

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

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

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

Errata

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

Patents

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

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

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

IMPORTANT NOTICE

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

Participants

At the time this draft standard was completed, the Precise Networked Clock Synchronization Working Group (IM/ST/PNCS) Working Group had the following membership:

Douglas Arnold, *Chair*
Rodney Cummings, *Vice Chair*
Silvana Rodrigues, Secretary
John MacKay, Editor

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

Participant1
Participant2
Participant3
Participant4
Participant5
Participant6
Participant7
Participant8
Participant9

The following members of the <individual/entity> Standards Association balloting group voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

[To be supplied by IEEE]

Balloter1
Balloter2
Balloter3
Balloter4
Balloter5
Balloter6
Balloter7
Balloter8
Balloter9

When the IEEE SA Standards Board approved this standard on **Error! No document variable supplied.**, it had the following membership:

[To be supplied by IEEE]

<Name>, *Chair*
<Name>, *Vice Chair*
<Name>, *Past Chair*
<Name>, *Secretary*

SBMember1
SBMember2
SBMember3
SBMember4
SBMember5
SBMember6
SBMember7
SBMember8
SBMember9

*Member Emeritus

Introduction

This introduction is not part of IEEE P1588e/D1.1, Draft Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems.

IEEE Std 1588™-2019 defines the Precision Time Protocol and the identification of entity and interface names and other terminology used in the protocol. IEEE 1588e identifies the structure and content of the IEEE 1588 MIB and YANG modules.

Instructions for Ballot Comments

The .MIB and .YANG modules are a formal part of this IEEE P1588e draft. The two modules are attached to the PDF of the ballot's draft, for review by balloters. To review each module, please detach the file from the PDF, and view the file in a text editor that shows line numbers (e.g. Notepad++).

The IEEE SA ballot format for comments requires a subclause, page, and line number for each comment. These entries are clear for the actual draft PDF, but voters will also need to submit comments on the .MIB and .YANG modules. Use the following values for a comment on a module:

- Line number: Use the line number in the module.
- Page: This does not apply, but enter "1" to provide a valid number for ballot processing tools.
- Subclause: Enter "100" for .MIB module. Enter "200" for .YANG module.

P1588e™/D1.1 Draft Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems Amendment: MIB and YANG Modules

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

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

Change the first paragraph of 8.2.2.3 as shown:

8.2.2.3 currentDS.offsetFromMaster

The value of currentDS.offsetFromMaster shall be the current value of the time difference between a Master ~~PTP Instance~~Clock and a Slave Clock as computed by the Slave PTP Instance; that is, <offsetFromMaster> = <Time ~~on~~ of the Slave ~~PTP Instance~~Clock> − <Time ~~on~~ of the Master ~~PTP Instance~~Clock> (see 11.2).

Change title of 8.3.3.2 and 8.3.3.3 as shown:

8.3.3.2 Static members of the ~~portDS~~transparentClockPortDS data set

8.3.3.3 Dynamic members of the ~~portDS~~transparentClockPortDS data set

Change the title of Clause 15 from "PTP management messages (optional)" to "Management" as follows:

15. ~~PTP management messages (optional)~~ Management

Change the title of 15.1 and text as follows:

15.1 Introduction

This clause specifies management mechanisms, protocols used for configuring and/or monitoring PTP Nodes and PTP Instances (see 3.1.30).

The specified management mechanisms are:

— 15.2: PTP management messages, used with PTP's built-in messaging.
— 15.3: MIB, used with the SNMP management protocol.
— 15.4: YANG, used with management protocols like NETCONF and RESTCONF.

Change the title of 15.2 and text as follows:

15.2 PTP management messages (optional)

Insert content of clause "15. PTP management messages (optional) from IEEE Std 1588™-2019" into subclause 15.2; update subclause numbers as appropriate. In addition, update subclause 15.2.5.2.3 (after renumbering) as described below:

15.2.5.2.3 managementId (Enumeration16)

Insert a new row 'SLAVE_EVENT_MONITORING' into Table 59, and change the subsequent Reserved row from 2004 to 2005 as follows:

Table 59—managementId values

managementId name	**managementId value** (hex)	**Allowed actions**	**Applies to**
Applicable to all PTP Instance types	0000 – 1FFF		
NULL_PTP_MANAGEMENT	0000	GET, SET, COMMAND	PTP Port
CLOCK_DESCRIPTION	0001	GET	PTP Port
USER_DESCRIPTION	0002	GET, SET	PTP Instance
SAVE_IN_NON_VOLATILE_STORAGE	0003	COMMAND	PTP Instance
RESET_NON_VOLATILE_STORAGE	0004	COMMAND	PTP Instance
INITIALIZE	0005	COMMAND	PTP Instance
FAULT_LOG	0006	GET	PTP Instance
FAULT_LOG_RESET	0007	COMMAND	PTP Instance
Reserved	0008 – 1FFF	—	—
Applicable to Ordinary Clock and Boundary Clocks	2000 – 2FFF	—	—
DEFAULT_DATA_SET	2000	GET	PTP Instance
CURRENT_DATA_SET	2001	GET	PTP Instance
PARENT_DATA_SET	2002	GET	PTP Instance
TIME_PROPERTIES_DATA_SET	2003	GET	PTP Instance
PORT_DATA_SET	2004	GET	PTP Port
PRIORITY1	2005	GET, SET	PTP Instance
PRIORITY2	2006	GET, SET	PTP Instance
DOMAIN	2007	GET, SET	PTP Instance
SLAVE_ONLY	2008	GET, SET	PTP Instance

managementId name	**managementId value** (hex)	**Allowed actions**	**Applies to**
LOG_ANNOUNCE_INTERVAL	2009	GET, SET	PTP Port
ANNOUNCE_RECEIPT_TIMEOUT	200A	GET, SET	PTP Port
LOG_SYNC_INTERVAL	200B	GET, SET	PTP Port
VERSION_NUMBER	200C	GET, SET	PTP Port
ENABLE_PORT	200D	COMMAND	PTP Port
DISABLE_PORT	200E	COMMAND	PTP Port
TIME	200F	GET, SET	PTP Instance
CLOCK_ACCURACY	2010	GET, SET	PTP Instance
UTC_PROPERTIES	2011	GET, SET	PTP Instance
TRACEABILITY_PROPERTIES	2012	GET, SET	PTP Instance
TIMESCALE_PROPERTIES	2013	GET, SET	PTP Instance
UNICAST_NEGOTIATION_ENABLE	2014	GET, SET	PTP Port
PATH_TRACE_LIST	2015	GET	PTP Instance
PATH_TRACE_ENABLE	2016	GET, SET	PTP Instance
GRANDMASTER_CLUSTER_TABLE	2017	GET, SET	PTP Instance
UNICAST_MASTER_TABLE	2018	GET, SET	PTP Port
UNICAST_MASTER_MAX_TABLE_SIZE	2019	GET	PTP Port
ACCEPTABLE_MASTER_TABLE	201A	GET, SET	PTP Instance
ACCEPTABLE_MASTER_TABLE_ENABLED	201B	GET, SET	PTP Port
ACCEPTABLE_MASTER_MAX_TABLE_SIZE	201C	GET	PTP Instance
ALTERNATE_MASTER	201D	GET, SET	PTP Port
ALTERNATE_TIME_OFFSET_ENABLE	201E	GET, SET	PTP Instance
ALTERNATE_TIME_OFFSET_NAME	201F	GET, SET	PTP Instance
ALTERNATE_TIME_OFFSET_MAX_KEY	2020	GET	PTP Instance
ALTERNATE_TIME_OFFSET_PROPERTIES	2021	GET, SET	PTP Instance
Reserved	2022 – 2FFF	—	—
Optional PTP management messages applicable to Ordinary Clock and Boundary Clocks	3000 – 3FFF		
EXTERNAL_PORT_CONFIGURATION_ENABLED	3000	GET, SET (see 17.6)	PTP Instance
MASTER_ONLY	3001	GET, SET	PTP Port
HOLDOVER_UPGRADE_ENABLE	3002	GET, SET	PTP Instance
EXT_PORT_CONFIG_PORT_DATA_SET	3003	GET, SET	PTP Port
SLAVE_EVENT_MONITORING	3004	GET, SET	PTP Port
Reserved	~~3004~~3005 – 3FFF	—	—
Applicable to Transparent Clocks	4000 to 4FFF	—	—
TRANSPARENT_CLOCK_DEFAULT_DATA_SET	4000	GET	PTP Instance
TRANSPARENT_CLOCK_PORT_DATA_SET	4001	GET	PTP Port
PRIMARY_DOMAIN	4002	GET, SET	PTP Instance
Reserved	4003 – 4FFF	—	—
Optional PTP management messages applicable to	5000 – 5FFF		

managementId name	managementId value (hex)	Allowed actions	Applies to
Transparent Clocks			
Reserved	5000 – 5FFF	—	—
Applicable to Ordinary Clocks, Boundary Clocks, and Transparent Clocks	6000 – 7FFF	—	—
DELAY_MECHANISM	6000	GET, SET	PTP Port
LOG_MIN_ PDELAY_REQ_INTERVAL	6001	GET, SET	PTP Port
Reserved	6002 – BFFF	—	—
This range is to be used for implementation-specific identifiers	C000 – DFFF	—	—
This range is to be assigned by an alternate PTP Profile	E000 – FFFE	—	—
Reserved	FFFF	—	—

Insert new subclauses 15.3 and 15.4 as follows:

15.3. Management Information Base (MIB)

15.3.1 Introduction

A management information base (MIB) is a virtual information store that allows accessing managed objects, typically through the Simple Network Management Protocol (SNMP), see 15.3.2.

This clause describes the IEEE1588 MIB module that allows accessing all the data set members defined in IEEE Std 1588™-2019, including optional data sets and their members. The IEEE1588 MIB is intended for the Default PTP Profiles described in IEEE Std 1588™-2019. Subclause 15.3.3 provides design considerations when implementing the IEEE1588 MIB and information about its extension for the purpose of PTP Profiles defined outside of this standard, see 15.3.3.15.

The IEEE1588 MIB is placed under iso(1) org(3) ieee(111) standards-association-numberedseries-standards(2) ieee1588(1588).

15.3.2 Internet Standard Management Framework

For a detailed overview of the documents that describe the current Internet Standard Management Framework, refer to section 7 of IETF RFC 3410 [Bss].

Managed objects are accessed via a virtual information store, termed the Management Information Base (MIB). MIB objects are generally accessed through the Simple Network Management Protocol (SNMP). Objects in the MIB are defined using the mechanisms defined in the Structure of Management Information (SMI). This clause specifies a MIB module that is compliant to the SMIv2, which is described in IETF STD 58, comprising IETF RFC 2578 [B26], IETF RFC 2579 [Bxx], and IETF RFC 2580 [Byy].

15.3.3 Design considerations

15.3.3.1 General

As described in IETF RFC 3444 [Bzz] and subclause 8.1.4.1 of this standard, the data sets specified in clause 8 of this standard serve as the normative specification of the information model for management.

In addition to describing the overall MIB design for IEEE Std 1588™-2019, this subclause describes how the information model is mapped to the MIB data model (see 15.3.4).

15.3.3.2 Structure

The overall structure of the IEEE1588 MIB module follows the information model defined in section 8.1.4.2 and it is outlined below:

- **ieee1588MibV1** - This is a base of IEEE1588 MIB module specified for version 2.1 of PTP (IEEE Std 1588™-2019) and backward compatible with version 2.0 of PTP (IEEE Std 1588™-2008). The V1 in the name of ieee1588MibV1 defines the version of the IEEE1588 MIB subtree, not the PTP Protocol nor the IEEE1588 standard.
    - **ieee1588Notifications** - The IEEE1588 MIB does not implement Traps, as such this object is empty. It is a placeholder for future specifications.
    - **ieee1588Objects -** The core part of IEEE1588 MIB that maps the IEEE1588 information model into MIB data model.
        - **ieee1588Base** - This subtree is specific to MIB, so it is not modeled in the PTP data sets of clause 8.1.4.2.
            - **ieee1588BaseInstanceTable** - A list of available PTP Instances; it can be used to add and remove PTP Instances, see 15.3.3.7.
            - **ieee1588BasePortTable** - Mapping of the PTP Port index of a PTP Instance into a physical interface, see 15.3.3.8.
            - **ieee1588BaseEnum** - Mappings of enumeration values defined in the standard into OIDs, see 15.3.3.9.
        - **ieee1588Instances** - A subtree that follows the structure described in section 8.1.4.2, point a) InstanceList[]. Tables that contain data sets specific to PTP Instance are indexed with a single index ieee1588InstanceIndex. Tables that contain data sets specific for a given PTP Port in a given PTP Instance are indexed with two indices: ieee1588InstanceIndex and ieee1588PortIndex. Tables that contain lists of records specific for a particular data set use one more index to distinguish records.
        - **ieee1588TcDefaultDS** - A subtree that follows the structure described in a section 8.1.4.2, point b) transparentClockDefaultDS.
        - **ieee1588TcPortListDS** - A subtree that follows the structure described in a section 8.1.4.2, point c) transparentClockPortList.
        - **ieee1588CommonServices** - A subtree that follows the structure described in a section 8.1.4.2, point d) commonServices.
        - **ieee1588Profiles** - The base for MIBs that extend the IEE1588 MIB with PTP Profile-specific information. It is intended to serve PTP Profiles defined outside of this standard. See 15.3.3.15.
    - **ieee1588Conformance** - It defines which OIDs are optional or mandatory.

NOTE 1—The IEEE1588 MIB does not include <foreignMasterList> defined in 9.3.2.4 as this list is not a part of the information model defined in clause 8.

NOTE 2—It is expected that the IEEE1588 MIB is extended and updated with the evolution of the IEEE1588 standard by the body developing the standard exclusively. If this evolution provides changes that are backward-compatible with the structure of the subtree defined under ieee1588MibV1, such changes would be made to the subtree under ieee1588MibV1. On the other hand, if the updated standard introduces changes to the structure of data sets, or other changes that cannot be made in a backward-compatible way in the ieee1588MibV1 subtree, a new structure would be defined under ieee1588MibV2. This new structure under ieee1588MibV2 would fully or partially replace the one under ieee1588MibV1. An example of a backward-compatible change is the addition of a new data set member to an existing data set, which would result in adding a new OID with the next available number under the subtree relevant to the updated data set.

15.3.3.3 Description

The text in a MIB "DESCRIPTION" statement is typically displayed in SNMP monitoring software. For the IEEE1588 MIB, the "DESCRIPTION" statement is provided for each OID that maps a data set member of this standard.

The person using SNMP monitoring software often does not have expertise in IEEE1588 implementation. Therefore, the MIB "DESCRIPTION" focuses more on how to use the data set member from a management perspective, and not on how to implement the data set member. In contrast, the technical description of a data set member in this standard includes significant information and requirements for implementation. Since much of that text is not relevant in the context of SNMP monitoring software, the MIB "DESCRIPTION" is not a direct duplication of the standard's text, but a focused summary of its usage for management.

15.3.3.4 Imported type definitions

The IEEE1588 MIB imports the following definitions from other MIBs:

— InterfaceIndexOrZero from IF-MIB (IETF RFC 2863 [Bkk]) used in ieee1588BasePortTable to map a PTP Port of a PTP Instance into the underlying physical interface.

— SnmpAdminString from SNMP-FRAMEWORK-MIB (IETF RFC 3411 [Bvv]) used in ieee1588BaseInstanceTable to store the free-text name of a PTP Instance

15.3.3.5 Naming conventions

For the name of each data set member (see 4.1.1), this standard uses lower case with the initial letter of the second and following words capitalized (e.g., externalPortConfigurationEnabled). This MIB module follows a similar pattern.

Most of OIDs' names are constructed using the following convention. The data set member name is concatenated with the name of the data set it belongs to. The first letter of the data set and the data set member name are capitalized (e.g., AnnounceReceiptTimeout), while the "s" in the "DS" is lower case (e.g., PortDs). Then the result is prefixed with "ieee1588" giving the final form (e.g., "ieee1588PortDsAnnounceReceiptTimeout").

A number of exceptions to the rule described above can be observed:

a) When the final name is longer than 64 characters, some parts are abbreviated to fit into the 64 characters limit. To keep the consistency within the IEEE1588 MIB some names are abbreviated even in OID names that would meet 64 characters limit e.g. "PerformanceMonitoring" in "ieee1588PerfMonDsEnable".
b) "Table" is suffixed after the data set name to table names. Similarly, "Index" is suffixed to index names.
c) Data set members that represent complex types are split into several OIDs, e.g*., portDS.portIdentity* is mapped into *ieee1588PortDsPortIdentityClockIdentity* and *ieee1588PortDsPortIdentityPortNumber*.

Where applicable, OIDs contain UNITS clauses like "seconds", "nanoseconds", "scaled nanoseconds", "log2 seconds", "log2 intervals", "messages", "octets" and "events" to give a hint about the units of a given value.

15.3.3.6 Read/write permissions

For each initialization classification in 8.1.4.4, the following describes its relationship with MIB's "MAX-ACCESS":

— Static and dynamic data set members are typically modeled as read-only, there are a few exceptions in which they are modelled as read-write (see 15.3.3.12).
— Configurable data sets are always modeled as read-write.
— Data set members that can be either configurable or dynamic are modelled as read-write

In addition, OIDs in the following tables are defined as read-create:

— ieee1588BaseInstanceTable
— ieee1588BasePortTable
— ieee1588AlternateTimeScaleOffsetDsListTable
— ieee1588GrandmasterClusterDsPortAddressTable
— ieee1588AcceptableMasterTableDsListTable

— ieee1588UnicastDiscoveryPortDsPortAddressTable

This allows for row creation and deletion in addition to the read and write.

15.3.3.7 Creation and removal of a PTP Instance

The IEEE1588 MIB allows the functionality of creation and removal of a PTP Instance. As such it adds the table ieee1588BaseInstanceTable with ieee1588BaseInstanceIndex. This table allows storing the free-text name of a PTP Instance. This table is specific to MIB and it is not modeled in the PTP data sets of clause 8.

15.3.3.8 Underlying interface

Relative to MIB's data model for interface management (IETF RFC 2863 [Bnn]), each PTP Port is a logical access point (see 3.1.61), and therefore the PTP Port does not reside in the stack of interfaces.

An SNMP management client will often need to know which specific underlying interface is used by a PTP Port. To provide this information, the MIB module adds the table ieee1588BasePortTable with two alternative means to provide a reference to the underlying interface used by the PTP Port:

a) ieee1588BasePortIfIndex,
   1) if implemented, the underlying IETF interface shall be referenced as an index value (ifIndex) of an interface instance in ifTable (defined in IETF RFC 2863 [Bnn]). The IETF interface can reside at any layer in the interface stack, such as layer 2 (e.g, IEEE 802.3 Ethernet), or layer 4 (e.g., UDP on IPv4).
   2) The value of ieee1588BasePortIfIndexOid shall be set to zeroDotZero, if ieee1588BasePortIfIndexOid is implemented.
b) ieee1588BasePortIfIndexOid,
   1) if implemented, it shall be a reference to an index value of an interface instance in a table with the definition of the used interface. The value of ieee1588BasePortIfIndexOid may point to an instance of a vendor specific interface in an enterprise MIB.
   2) The value of ieee1588BasePortIfIndex shall be set to zero, if ieee1588BasePortIfIndex is implemented.

If the underlying interface for a PTP Port is not defined then the ieee1588BasePortIfIndex value shall be zero and the ieee1588BasePortIfIndexOid value shall be zeroDotZero, if the respective objects are implemented.

This table allows row creation and deletion to support dynamic assignment of PTP Ports to interfaces. This table is specific to MIB and it is not modeled in the PTP data sets of clause 8.

15.3.3.9 Enumerations defined in this MIB

The enumerations defined in IEEE Std 1588™ contain reserved values and values designated for assignment by alternate PTP Profiles. The SMIv2 syntax does not allow extending a definition of enumerations. As such, a future PTP Profile is not able to provide a meaning to the enumeration values designated for assignment by alternate PTP Profiles. To overcome this limitation, relevant objects within the IEEE1588 MIB are defined as custom types, which are based on OBJECT IDENTIFIER type. Instead of an integer that can be mapped to an enumeration value on the SNMP manager side, the particular object passes an OID to an Object within a MIB tree (which may be outside of the IEEE1588 MIB), which represents a predefined value. The custom types are used to give a hint to implementers that the particular OID should return an OID value that represents the expected enumeration value. SMIv2 syntax is limited in that it does not allow restriction of the set of allowed OIDs. It is up to the implementation of an agent to provide the correct value.

The allowed Objects for each added custom type are placed under ieee1588BaseEnum subtree. This subtree is divided into the following groups of enumerations:

a) ieee1588BaseEnumClockAccuracy - Clock Accuracy value as defined in 7.6.2.6.
b) ieee1588BaseEnumClockClass - Clock Class value as defined in 7.6.2.5.
c) ieee1588BaseEnumDelayMechanism - Enumeration for the path delay measuring mechanism, as defined in 8.2.15.4.4.
d) Ieee1588IfIndexOid - Shall point to an index value of a table with the instance of the used interface.
e) ieee1588BaseEnumInstanceType - Type indicates the type of PTP Instance, as defined in 8.2.1.5.5.
f) ieee1588BaseEnumL1SyncState - Enumeration for states of an L1Sync port, as defined in L.5.3.5.
g) ieee1588BaseEnumMessageType - The value indicates the type of the PTP message, as defined in 13.3.2.3.
h) ieee1588BaseEnumNetworkProtocol - Enumeration for the network transport protocol, as defined in 7.4.1.
i) ieee1588BaseEnumPortState - Enumeration for the state of the PTP protocol engine associated with the PTP Port, as defined in 8.2.15.3.1 and 9.2.5.
j) ieee1588BaseEnumSeverityCode - Enumeration for the severity of a fault record, as defined in 8.2.6.3.
k) ieee1588BaseEnumTimeSource - Source of time used by Grandmaster PTP Instance, as defined in 7.6.2.8.

Thanks to the mechanism described above, a separate MIB defining PTP Profile-specific extensions (see 15.3.3.15) can assign the enumeration values designated for assignment by alternate PTP Profile. Assignment of reserved enumeration values by the future versions of this standard is also made easier.

15.3.3.10 Textual conventions defined in this MIB

The IEEE1588 MIB introduces the following textual conventions, which are built on top of OBJECT IDENTIFIER type (see 15.3.3.9). Their purpose is to indicate where an OID with a given type points to.

- Ieee1588ClockAccuracy shall point to ieee1588BaseEnumClockAccuracy
- Ieee1588ClockClass shall point to ieee1588BaseEnumClockClass
- Ieee1588DelayMechanism shall point to ieee1588BaseEnumDelayMechanism
- Ieee1588InstanceType shall point to ieee1588BaseEnumInstanceType
- Ieee1588L1SyncState shall point to ieee1588BaseEnumL1SyncState
- Ieee1588MessageType shall point to ieee1588BaseEnumMessageType
- Ieee1588NetworkProtocol shall point to ieee1588BaseEnumNetworkProtocol
- Ieee1588PortState shall point to ieee1588BaseEnumPortState
- Ieee1588SeverityCode shall point to ieee1588BaseEnumSeverityCode
- Ieee1588TimeSource shall point to ieee1588BaseEnumTimeSource

In addition, the IEEE1588 MIB introduces the following textual conventions, which map into IEEE Std 1588™-2019 data types as follows:

- Ieee1588ClockIdentity - Identifies unique entities within a PTP Network, e.g. a PTP Instance or an entity of a common service. The identity is an 8-octet array, as defined in 5.3.4 and 7.5.2.2.
- Ieee1588LogTimeInterval - Time interval expressed as logarithm base 2 of the number of seconds, as defined in 7.7.2.1.
- Ieee1588RelativeDifference - Integer64, relative difference expressed as a dimensionless fraction and multiplied by 2^{62}, with any remaining fractional part truncated, as defined in 5.3.11.
- Ieee1588TimeInterval - Integer64, time intervals in units of 2^{-16} ns, as defined in 5.3.2.
- UInteger48 - 48 bit unsigned integer value, as defined in 5.2.

15.3.3.11 Compliance

As stated in 8.1.4.3, management is optional for PTP, and when management is supported (e.g., using SNMP), any one member can be accessible via management. Therefore, all OIDs are defined as optional in the compliance statements.

15.3.3.12 Departure from the IEEE Std 1588™-2019 information model

This clause describes a number of minor departures of the IEEE1588 MIB from the information model defined in clause 8 of this standard.

The data set member actualTableSize is not implemented in tables ieee1588GrandmasterClusterDsTable, ieee1588UnicastDiscoveryPortDsTable and ieee1588AcceptableMasterTableDsTable. These tables use the MIB-specific functionality to add and remove table's entries which represent lists in specific data sets. As such, the functionality of actualTableSize (intended by IEEE Std 1588™) is provided by MIB-specific mechanism.

The member faultRecordLength of the FaultRecord data set member in the faultLogDS is not implemented. It provides redundant information. If needed, record's length can be calculated based on the size of all objects associated with a specific faultRecord. The value of a faultRecord varies with the length of PTP Text type members.

The following data sets have a member index that is not represented as an object in MIB: performanceMonitoringDs, performanceMonitoringPortDS and cmldsPerformanceMonitoringLinkPortDS in their recordList and recordListPeerDelay. The index value can be extracted from a row number of the following tables:

- ieee1588PerformanceMonitoringDsRecordListTable
- ieee1588PerfMonPortDsRecordListTable
- ieee1588CmldsPerfMonPortDsRecordListTable
- ieee1588PerfMonPortDsRecordListPdTable
- ieee1588CmldsPerfMonPortDsRecordListPdTable.

To allow remote configuration of the clockClass value for a PTP Instance, the following objects mapping dynamic data set members have read-write permissions: ieee1588DefaultDsClockQualityClockClass, ieee1588DefaultDsClockQualityClockAccuracy and ieee1588DefaultDsClockQualityOffsetScaledLogVariance. While the objects are read-write, it is possible that the PTP implementation does not support write of the respective data set members by the management mechanism, and therefore the configuration value will not be applied to the PTP data sets (e.g., defaultDS.clockQuality has a constant value). To communicate this problem back to the MIB management software, the MIB agent should respond to such writes with an error or warning response.

15.3.3.13 Relationship to MIB(s) for IEEE Std 1588™-2008

There exist a number of MIB data models for the IEEE Std 1588™-2008 and its PTP Profiles defined outside of this standard, e.g. MIB for Default PTP Profiles defined in IETF RFC 8173 [Buu], MIB for the PTP Profile defined in IEEE Std 802.1AS, MIB for the Power PTP Profile defined in IEC 62439 [Bww]. The IEEE1588 MIB defined in this standard for IEEE Std 1588™-2019:

- is meant to be used with Default PTP Profiles of IEEE Std 1588™-2019;
- is backward compatible with the Default PTP Profiles of IEEE Std 1588™-2008.
- is not meant to be compatible with the MIB data models defined outside of this standard.
- resides on a different subtree than the MIB data models defined outside of this standard, so they can co-exist. It is advised to use only the MIB data model defined in this standard.

The MIB data model for the Default PTP Profiles of the IEEE Std 1588™-2008 is specified in IETF RFC 8173 [Buu]. Differences between the IETF RFC 8173 and the MIB module of this standard include:

— number of supported data sets: the MIB data model in IETF RFC 8173 does not support all the data sets defined in this standard.

— number and type of indexes: the MIB defined in IETF RFC 8173 uses the clock type and the domain number as indexes, while the MIB defined in this standard uses ieee1588InstanceIndex.

— prefix used for OIDs: IETF RCF 8173 uses “ptpbase”, the MIB data model defined in this standard “ieee1588”

15.3.3.14 Security considerations

SNMP versions prior to SNMPv3 did not include adequate security. Even if the network itself is secure (for example by using IPsec), there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in these MIB module.

It is recommended that implementers consider the security features as provided by the SNMPv3 framework, see section 8 in IETF RFC 3410 [Bss], including full support for the SNMPv3 cryptographic mechanisms (for authentication and privacy).

Further, deployment of SNMP versions prior to SNMPv3 is not recommended. Instead, it is recommended to deploy SNMPv3 and to enable cryptographic security. It is then a customer/operator responsibility that the SNMP entity giving access to an instance of these MIB modules is properly configured to give access to the objects only to the principals (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

A number of management objects defined in the IEEE1588 MIB module have a MAX-ACCESS clause of read-write and/or read-create. Such objects might be considered sensitive or vulnerable in some network environments. The support for SET operations in a non-secure environment without proper protection can have a negative effect on network operations.

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

Some objects in the IEEE1588 MIB can be manipulated to interfere with the operation of time synchronization. This could, for example, be used to force a reinitialization of state machines to cause time synchronization and network instability. Another possibility would be for an attacker to override Grandmaster PTP Instance status to give a user (or an attacker) unauthorized control over the network time.

As an example, improper manipulation of ieee1588DefaultDsPriority1 (or any other clock attribute taken into account by BMCA) writable object could result in an unintended Grandmaster PTP Instance being elected when a system is grandmaster-capable in a PTP domain. It could also be used maliciously to cause frequent Grandmaster PTP Instance changes that could affect network stability.

As another example, improper manipulation of the following writable objects could result in a segmented PTP Network, could compromise the expected accuracy, and could interrupt paths of the PTP domain.

— ieee1588PortDsPortEnable

— ieee1588DefaultDsExternalPortConfigurationEnabled

— ieee1588ExternalPortConfigurationPortDsDesiredState

Unintended access to any of the readable tables or variables in the IEEE1588 MIB alerts the reader that time synchronization in the PTP domain is configured, which values of timing parameters are configured, and which system is the current Grandmaster PTP Instance. This information can suggest to an attacker what applications are being run, and thus suggest application-specific attacks, or can enable the attacker to detect whether their attacks are successful. Values of clockIdentity can reveal the MAC address of a device and its peers. This can help to discover details about the network topology and used devices. It is thus important to control even GET access to these objects and possibly to even encrypt the values of these objects when sending them over the network via SNMP.

15.3.3.15 Extensions by alternate PTP Profiles

It is expected that the IEEE1588 MIB is used and extended by PTP Profiles defined outside of this standard. A PTP Profile defined outside of this standard can use the IEEE1588 MIB defined in this standard as is, if such a PTP Profile does not define any additional data set members or enumeration values. If a PTP Profile defines profile-specific information that needs to be mapped into MIB, it should define a profile-specific MIB that includes only the profile-specific information, and its structure is placed under the following base: ieee1588MibV1.ieee1588Profiles.mibProfileID, where:

- ieee1588MibV1 specifies the IEEE1588 MIB version on which the profile-specific MIB is based, currently only ieee1588MibV1 is defined
- mibProfileID is a single 32 bit value that contains the 4 most significant octets of the profileIdentifier (see 20.3.3), i.e. the most significant 3 octets are either an OUI or a CID owned by the organization creating the profile, and the next octet is the profileNumber.

Such a profile-specific MIB would be typically imported along with the IEEE1588 MIB. MIBs for a number of different profiles can be imported at the same time.

NOTE—An example of profile-specific information that can be defined in a profile-specific MIB is an extension of enumeration for the value designated for assignment by alternate PTP Profiles, see 15.3.3.9.

15.3.4 Tree diagram

This clause contains the Tree diagram of IEEE1588 MIB.

- ieee1588Notifications(0)
- ieee1588Objects(1)
  - ieee1588Base(1)
    - ieee1588BaseInstanceTable(1)
    - ieee1588BasePortTable(2)
    - ieee1588BaseEnum(3)
      - ieee1588BaseEnumClockAccuracy(1)
      - ieee1588BaseEnumClockClass(2)
      - ieee1588BaseEnumDelayMechanism(3)
      - ieee1588BaseEnumInstanceType(4)
      - ieee1588BaseEnumL1SyncState(5)
      - ieee1588BaseEnumMessageType(6)
      - ieee1588BaseEnumNetworkProtocol(7)
      - ieee1588BaseEnumPortState(8)
      - ieee1588BaseEnumSeverityCode(9)
      - ieee1588BaseEnumTimeSource(10)
  - ieee1588Instances(2)
    - ieee1588DefaultDsTable(1)
    - ieee1588CurrentDsTable(2)
    - ieee1588ParentDsTable(3)
    - ieee1588TimePropertiesDsTable(4)

- ieee1588DescriptionDsTable(5)
- ieee1588FaultLogDs(6)
  - ieee1588FaultLogDsGeneralTable(1)
  - ieee1588FaultLogDsFaultRecordListTable(2)
- ieee1588NonVolatileStorageDsTable(7)
- ieee1588PathTraceDs(8)
  - ieee1588PathTraceDsGeneralTable(1)
  - ieee1588PathTraceDsListTable(2)
- ieee1588AlternateTimeScaleOffsetDs(9)
  - ieee1588AlternateTimeScaleOffsetDsTable(1)
  - ieee1588AlternateTimeScaleOffsetDsListTable(2)
- ieee1588HoldoverUpgradeDsTable(10)
- ieee1588GrandmasterClusterDs(11)
  - ieee1588GrandmasterClusterDsTable(1)
  - ieee1588GrandmasterClusterDsPortAddressTable(2)
- ieee1588AcceptableMasterTableDs(12)
  - ieee1588AcceptableMasterTableDsTable(1)
  - ieee1588AcceptableMasterTableDsListTable(2)
- ieee1588PerfMonDs(13)
  - ieee1588PerfMonDsTable(1)
  - ieee1588PerfMonDsRecordListTable(2)
- ieee1588EnhancedSynchronizationAccuracyMetricsDsTable(14)
- ieee1588PortList(15)
  - ieee1588PortDsTable(1)
  - ieee1588TimestampCorrectionPortDsTable(2)
  - ieee1588AsymmetryCorrectionPortDsTable(3)
  - ieee1588DescriptionPortDsTable(4)
  - ieee1588UnicastNegotiationPortDsTable(5)
  - ieee1588AlternateMasterPortDsTable(6)
  - ieee1588UnicastDiscoveryPortDs(7)
    - ieee1588UnicastDiscoveryPortDsTable(1)
    - ieee1588UnicastDiscoveryPortDsPortAddressTable(2)
  - ieee1588AcceptableMasterPortDsTable(8)
  - ieee1588L1SyncBasicPortDsTable(9)
  - ieee1588L1SyncOptParamsPortDsTable(10)
  - ieee1588CommunicationCapabilitiesPortDsTable(11)
  - ieee1588PerfMonPortDs(12)
    - ieee1588PerfMonPortDsRecordListPdTable(1)
    - ieee1588PerfMonPortDsRecordListTable(2)
  - ieee1588SlaveMonitoringPortDsTable(13)
  - ieee1588CommonServicesPortDsTable(14)
  - ieee1588ExternalPortConfigurationPortDsTable(15)

— ieee1588TcDefaultDs(3)
— ieee1588TcPortList(4)
    — ieee1588TcPortDsTable(1)
— ieee1588CommonServices(5)
    — ieee1588CommonMeanLinkDelayService(1)
    — ieee1588CmldsDefaultDs(1)
    — ieee1588CmldsLinkPortList(2)
        — ieee1588CmldsLinkPortDsTable(1)
        — ieee1588CmldsTimestampCorrectionLinkPortDsTable(2)
        — ieee1588CmldsAsymmetryCorrectionLinkPortDsTable(3)
        — ieee1588CmldsPerfMonPortDs(4)
            — ieee1588CmldsPerfMonPortDsRecordListPdTable(1)
            — ieee1588CmldsPerfMonPortDsRecordListTable(2)
— ieee1588Profiles(6)
— ieee1588Conformance(2)

15.3.5 Module

The MIB module specified by IEEE Std 1588™ is the revision "202305091919Z:

Editor's Note: The published standard will replace this editor's note with a URL to the published MIB module. This URL cannot be provided during the draft balloting process. For review of the MIB module for this ballot, the MIB module IEEE1588-MIB-mib is attached to the balloted PDF file.

NOTE—Copyright release for MIBs: Users of this standard may freely reproduce the MIB module defined by this standard so that they can be used for their intended purpose.

15.4. YANG

15.4.1 Introduction

YANG (IETF RFC 7950 [Baa]) is a data modeling language used to model configuration and state data manipulated by network management protocols like NETCONF (IETF RFC 6241 [Bbb]) and RESTCONF (IETF RFC 8040 [Bcc]). In the context of this standard, the YANG-based network management protocols can serve as a "management mechanism" (see 3.1.30).

Some characteristics of YANG-based management mechanisms that are attractive for PTP are:

- Since YANG is not specific to PTP, the NETCONF or RESTCONF server (i.e., PTP Node) can support management of other protocols and features, such as those specified in organizations like IETF, IEEE 802, and ITU-T. All of these features can be managed by client software that is not specific to PTP.
- The mechanisms are designed to operate over the Internet, such that the client can be physically distant from the server.
- Support for modern security standards is required.
- Configuration capabilities such as validation and rollback provide for a comprehensive system of network management.

15.4.2 Design considerations

As described in IETF RFC 3444 [Bzz] and subclause 8.1.4.1 of this standard, the data sets specified in clause 8 of this standard serve as the normative specification of the information model for management.

In addition to describing the overall YANG design for IEEE Std 1588™, this subclause describes how the information model is mapped to the YANG data model (see 15.4.4).

15.4.2.1 Relationship to YANG for IEEE Std 1588™-2008

The YANG data model for IEEE Std 1588™-2008 is specified in IETF RFC 8575 [Bgg].

As described in Annex A of RFC 8575 [Bgg], the specifications in this clause represent the predicted transfer of work from the IETF TICTOC Working Group to the IEEE 1588 Working Group.

Products that support YANG for IEEE Std 1588™-2008 can use RFC 8575 [Bgg], or the YANG module of this standard (see 15.4.4). Products that support YANG for IEEE Std 1588™-2019 or later are expected to use the YANG module of this standard.

15.4.2.2 Underlying interface

Relative to YANG's data model for interface management (IETF RFC 8343), each PTP Port is a logical access point (see 3.1.61), and therefore the PTP Port does not reside in the stack of interfaces.

A YANG management client will often need to know which specific IETF-defined underlying interface is used by a PTP Port. To provide this information, the YANG module in 15.4.4 adds the leaf "underlying-interface" to portDS, to provide a YANG reference to the underlying interface in use by the PTP Port. This leaf is specific to YANG, so it is not modeled in the PTP data sets of clause 8.

The initialization classification (see 8.1.2) of "`underlying-interface`" is dynamic.

15.4.2.3 Read/write permissions

Change the numbering of all figures in this clause as appropriate.

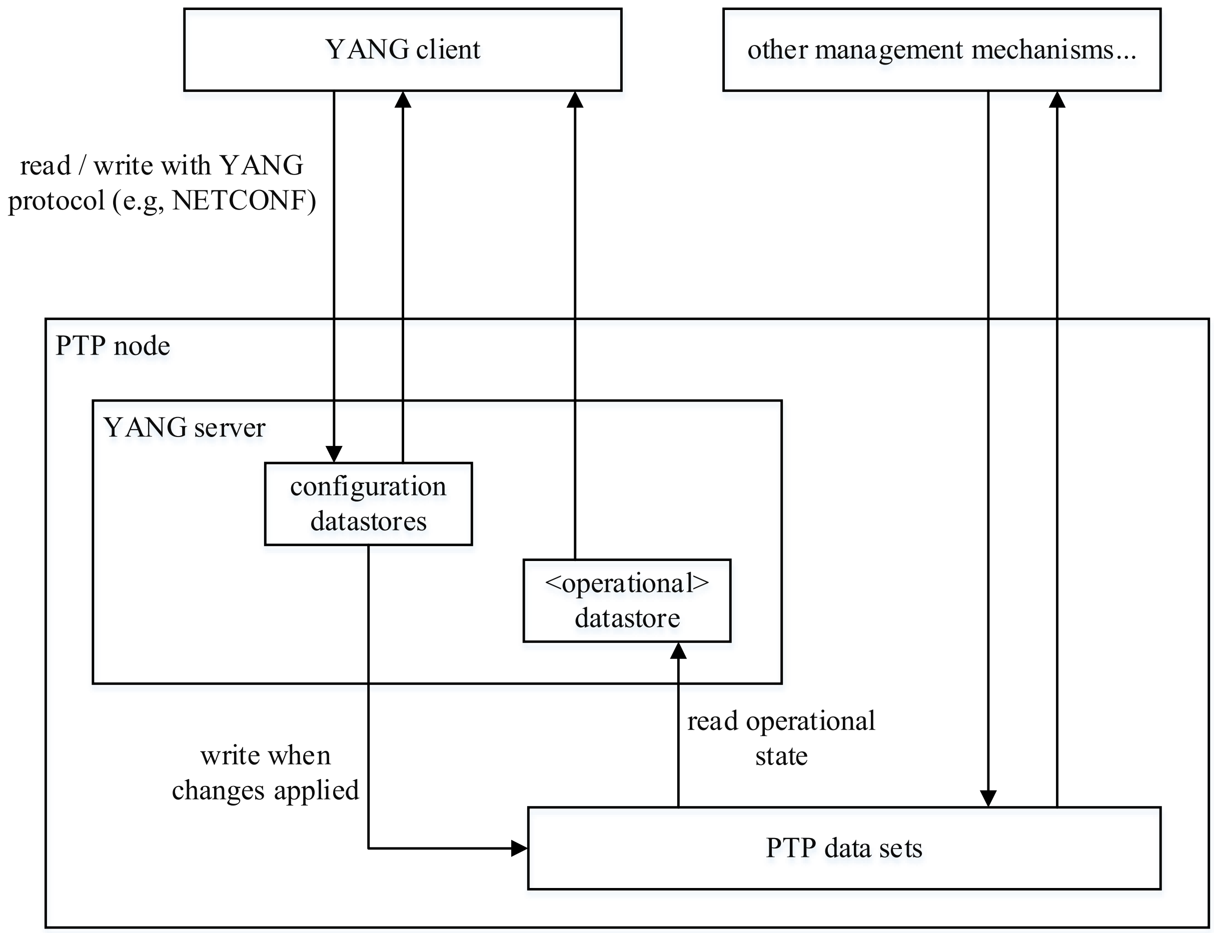

Figure 901 - Architectural model of YANG and PTP

IETF RFC 8342 (Network Management Datastore Architecture, NMDA) [Btt] specifies the organization of data within a YANG server, including its relationship to the system that is managed. In the context of PTP, the managed system is the PTP data sets, which are contained within the implementation of PTP. Figure 901 shows this architectural relationship.

When a YANG client writes data, that data is not necessarily applied to the system immediately. YANG NMDA supports a variety of datastores for configuration. In the context of PTP, when the YANG server applies its configuration as described in YANG NMDA, that translates to an internal write of the configuration values to the PTP data sets.

When a YANG client reads data from the YANG operational state datastore (<operational>), the read returns the value that is currently in use by the system. In the context of PTP, the value in use is contained within the PTP data sets. Therefore, a YANG client read of <operational> is implemented as an internal read of the PTP data sets. Other non-YANG management mechanisms can configure the PTP data sets in parallel to YANG. Therefore, the YANG <operational> datastore is effectively a reflection of the current values of the PTP data sets.

Subclause 8.1.4.4 of this standard specifies permissions for read and write of the PTP data sets, as shown in the lower read and write of Figure 901.

In YANG, the "`config`" statement effectively specifies permissions for read and write by a YANG client (see Figure 901). When "`config`" is true, the data represents configuration, which can be written by a client. When "`config`" is false, the data represents state only (i.e., information that is read-only). If "`config`" is missing in the YANG text, the default is true. For more information, see IETF RFC 8342, Network Management Datastore Architecture (NMDA) [Btt].

For each initialization classification in 8.1.4.4, the following describes its relationship with YANG "`config`":

— Configurable: Configurable data set members are read/write for management, which aligns with YANG's concept of configuration. Therefore, all configurable data set members are modeled as "`config`" true in YANG.

— Static: 8.1.4.4 states that static data set members are modeled as read-only for management. Therefore, all static data set members are modeled as "`config`" false in YANG (i.e., state only).

— Dynamic: For dynamic data set members, 8.1.2.1.2 lists three reasons why the member can change. The first reason is a result of protocol operations, and therefore the member is read-only for management (YANG "`config`" false). The third reason is a change external to PTP, using an example of a GPS system that changes defaultDS.clockQuality. For the third reason, the external system can use a YANG protocol to change the member (GPS system writes defaultDS.clockQuality using NETCONF), and therefore the member is read-write for management (YANG "`config`" true). The second reason is similar, in that an external system can change the member. Due to this inconsistency, dynamic data set members are evaluated on a member-by-member basis. For each member, the value of YANG "`config`" is true if either of the last two reasons can apply, and false if only the first reason applies.

The preceding description means that it is possible for a YANG client to write a leaf that is "`config`" true, but the product implementation does not support write of that member, and therefore the configuration value will not be applied to the PTP data sets (e.g., defaultDS.clockQuality has a constant value). To communicate this problem back to the YANG client, the YANG server should respond to such writes with an error or warning response.

15.4.2.4 Naming conventions

For the name of each data set member (see 4.1.1), this standard uses lower case, with an upper case character to separate words (e.g., externalPortConfigurationEnabled).

IETF RFC 8407 section 4.3.1 [Bdd] recommends that YANG identifiers use the convention of lower case, with dash to separate words (e.g., external-port-configuration-enabled). This naming convention is very common in YANG modules. The YANG module in 15.4.4 uses this naming convention.

15.4.2.5 Description

The text in a YANG "description" statement is typically displayed in YANG client software. For PTP YANG, each PTP data set member has a YANG "description" statement. The person using YANG client software often does not have expertise in PTP implementation. Therefore, the YANG "description" focuses more on how to use the data set member from a management perspective, and not on how to implement the data set member.

In contrast, the technical description of a PTP data set member in this standard includes significant information and requirements for implementation. Since much of that text is not relevant in the context of YANG client software, the YANG "description" is not a direct duplication of the standard's text, but a focused summary of its usage for management.

15.4.2.6 Type, grouping, enumeration

Each PTP data set member specifies use of a data type from clause 5, and that PTP data type is mapped to a corresponding data type in YANG.

A PTP primitive data type (see 5.2) corresponds to a YANG built-in type. A PTP struct (see 5.3) corresponds to a YANG grouping, used by YANG containers within the tree.

A PTP data set (e.g., defaultDS) is represented directly as a YANG container within the tree. Since each PTP data set exists only once in the tree (see 8.1.4.2), re-use does not apply, and therefore YANG grouping is not necessary.

One of the PTP primitive data types is enumeration. In the context of PTP, the numeric value of each enumerated name is used on-the-wire by the protocol.

If a PTP enumeration consists of name/value pairs that are entirely within the scope of this standard (i.e., profile-specific range not allowed), a YANG enumeration is used along with a corresponding YANG typedef. The YANG enumeration prevents addition of new values by YANG modules that import the YANG module of this standard.

If a PTP enumeration supports a range of values that can be specified outside this standard (e.g., by a PTP Profile), a YANG identity is used. The PTP-specified name is the name of the identity, and the numeric value is listed in the YANG description of the identity. YANG identity allows for new values to be specified by YANG modules that import the YANG module of this standard (see 15.4.4)

If a PTP data set member consists of description/value pairs without a specified name for each pair (e.g., clockClass in 7.6.2.5), a YANG unsigned integer is used.

15.4.2.7 Default

The YANG "`default`" statement provides a mechanism for the YANG module to specify a default value when no other value is applicable. In this standard, this concept is specified as the initialization value (see 8.1.3).

Regardless of where it is specified, the initialization value originates in the PTP data sets, not within a YANG datastore. To help explain this, consider an example PTP Node that supports both NETCONF (YANG) and SNMP (MIB). If the initialization value originated from each management mechanism, the potential would exist for contradictory initialization values to be written down from each mechanism to the PTP data sets. When the initialization value originates from the PTP data sets based on specifications, the current value is read upward (e.g., through YANG <operational>).

The PTP initialization value is specified by PTP standard documents that are not related to YANG (i.e., this document and/or a document that specifies a PTP Profile). The PTP initialization value is internal to the PTP Instance. Therefore, the YANG "default" statement is not used in this clause, because the PTP data set implementation already knows this value. The YANG client will read the correct default value even though this value is not in the YANG module.

15.4.2.8 Mandatory

As stated in 8.1.4.3, management is optional for PTP, and when management is supported (e.g., using YANG), any one member can be accessible via management. Therefore, the YANG "`mandatory`" statement is not used in this clause.

15.4.2.9 Optional Features

Some clauses and annexes (e.g. clauses 16 and 17, Annexes J and L) specify many optional features that can be supported by a PTP Node. Each feature often specifies data sets for configuration, including a member to enable/disable the feature.

The YANG "feature" statement provides a way to identify optional features in the YANG model. A YANG server advertises the supported features to any YANG client, so that the client can avoid attempts to configure a feature that is not supported. IETF RFC 8525 (YANG Library) [Bee] returns the list of features from server to client.

The YANG module in this standard uses the YANG "`feature`" statement for each optional feature in clauses relevant clauses and annexes.

A PTP Profile specification does not replace the specifications of this standard, but instead profiles the optional features of this standard. Therefore, it is possible for a PTP Profile to explicitly require or prohibit an optional feature. A PTP Profile requirement means that the corresponding YANG "`feature`" is returned by the server's YANG library while operating the PTP Profile. A PTP Profile prohibition means that the corresponding YANG "`feature`" is not returned by the server's YANG library while operating the PTP Profile. For all other cases (e.g., PTP Profile does not mention the feature), the PTP Node implementer decides whether to support the feature, and the server uses the corresponding YANG "`feature`" to indicate support to the client.

15.4.2.10 Non-volatile Storage

The nonvolatileStorageDS in 8.2.7 is not applicable for YANG, since protocols like NETCONF and RESTCONF specify analogous features for configuration storage (see 8.1.3.5).

15.4.3 Tree diagram

The following YANG tree diagram provides a summary of the YANG module (see 15.4.4). Refer to IETF RFC 8340 [Bff] for the syntax convention of the YANG tree diagram.

```
module: ieee1588-ptp
  +--rw ptp
     +--rw instances
     |  +--rw instance* [instance-index]
     |     +--rw instance-index               uint32
     |     +--rw default-ds
     |     |  x--ro two-step-flag?                 boolean
     |     |  +--ro clock-identity?                clock-identity
     |     |  +--ro number-ports?                  uint16
     |     |  +--rw clock-quality
     |     |  |  +--rw clock-class?                  identityref
     |     |  |  +--rw clock-accuracy?               identityref
     |     |  |  +--rw offset-scaled-log-variance?   uint16
     |     |  +--rw priority1?                     uint8
     |     |  +--rw priority2?                     uint8
     |     |  +--rw domain-number?                 uint8
     |     |  +--rw slave-only?                    boolean
     |     |  +--rw sdo-id?                        uint16
     |     |  +--rw current-time
     |     |  |  +--rw seconds-field?       uint64
     |     |  |  +--rw nanoseconds-field?   uint32
     |     |  +--rw instance-enable?               boolean
     |     |  +--rw external-port-config-enable?   boolean {external-port-config}?
     |     |  +--rw max-steps-removed?             uint8
     |     |  +--rw instance-type?                 instance-type
     |     +--rw current-ds
     |     |  +--ro steps-removed?               uint16
     |     |  +--ro offset-from-master?          time-interval
     |     |  +--ro mean-delay?                  time-interval
     |     |  x--ro mean-path-delay?             time-interval
     |     |  +--ro synchronization-uncertain?   boolean
     |     +--rw parent-ds
     |     |  +--ro parent-port-identity
     |     |  |  +--ro clock-identity?   clock-identity
     |     |  |  +--ro port-number?      uint16
     |     |  +--ro parent-stats?                                boolean
     |     |  +--ro observed-parent-offset-scaled-log-variance?   uint16
     |     |  +--ro observed-parent-clock-phase-change-rate?      int32
     |     |  +--ro grandmaster-identity?                        clock-identity
     |     |  +--ro grandmaster-clock-quality
     |     |  |  +--ro clock-class?                  identityref
     |     |  |  +--ro clock-accuracy?               identityref
     |     |  |  +--ro offset-scaled-log-variance?   uint16
     |     |  +--ro grandmaster-priority1?                       uint8
     |     |  +--ro grandmaster-priority2?                       uint8
     |     |  +--rw protocol-address
```

```
|     |  |  +--rw network-protocol?   identityref
|     |  |  +--rw address-length?     uint16
|     |  |  +--rw address-field?      string
|     |  +--ro synchronization-uncertain?                       boolean
|     +--rw time-properties-ds
|     |  +--rw current-utc-offset?         int16
|     |  +--rw current-utc-offset-valid?   boolean
|     |  +--rw leap59?                     boolean
|     |  +--rw leap61?                     boolean
|     |  +--rw time-traceable?             boolean
|     |  +--rw frequency-traceable?        boolean
|     |  +--rw ptp-timescale?              boolean
|     |  +--rw time-source?                identityref
|     +--rw description-ds
|     |  +--ro manufacturer-identity?   string
|     |  +--ro product-description?     string
|     |  +--ro product-revision?        string
|     |  +--rw user-description?        string
|     +--ro fault-log-ds {fault-log}?
|     |  +--ro number-of-fault-records?   uint16
|     |  +--ro fault-record-list* []
|     |  |  +--ro time
|     |  |  |  +--ro seconds-field?       uint64
|     |  |  |  +--ro nanoseconds-field?   uint32
|     |  |  +--ro severity?      fault-severity
|     |  |  +--ro name?          string
|     |  |  +--ro value?         string
|     |  |  +--ro description?   string
|     |  +---x reset
|     +--rw path-trace-ds {path-trace}?
|     |  +--ro list*     clock-identity
|     |  +--rw enable?   boolean
|     +--rw alternate-timescale-ds {alternate-timescale}?
|     |  +--ro max-key?   uint8
|     |  +--rw list* [key-field]
|     |     +--rw key-field            uint8
|     |     +--rw enable?              boolean
|     |     +--rw current-offset?      int32
|     |     +--rw jump-seconds?        int32
|     |     +--rw time-of-next-jump?   uint64
|     |     +--rw display-name?        string
|     +--rw holdover-upgrade-ds {holdover-upgrade}?
|     |  +--rw enable?   boolean
|     +--rw grandmaster-cluster-ds {grandmaster-cluster}?
|     |  +--ro max-table-size?       uint8
|     |  +--rw log-query-interval?   int8
|     |  +--rw port-address* [index]
|     |     +--rw index               uint16
|     |     +--rw network-protocol?   identityref
|     |     +--rw address-length?     uint16
|     |     +--rw address-field?      string
|     +--rw acceptable-master-ds {acceptable-master}?
|     |  +--ro max-table-size?   uint16
|     |  +--rw list* [index]
|     |     +--rw index                       uint8
|     |     +--rw acceptable-port-identity
|     |     |  +--rw clock-identity?   clock-identity
|     |     |  +--rw port-number?      uint16
|     |     +--rw alternate-priority1?       uint8
|     +--rw performance-monitoring-ds {performance-monitoring}?
|     |  +--rw enable?       boolean
|     |  +--ro record-list* [index]
|     |     +--ro index                         uint16
```

```
     |     |     +--ro measurement-valid?            boolean
     |     |     +--ro period-complete?              boolean
     |     |     +--ro pm-time?                      yang:timestamp
     |     |     +--ro average-master-slave-delay?   time-interval
     |     |     +--ro minimum-master-slave-delay?   time-interval
     |     |     +--ro maximum-master-slave-delay?   time-interval
     |     |     +--ro stddev-master-slave-delay?    time-interval
     |     |     +--ro average-slave-master-delay?   time-interval
     |     |     +--ro minimum-slave-master-delay?   time-interval
     |     |     +--ro maximum-slave-master-delay?   time-interval
     |     |     +--ro stddev-slave-master-delay?    time-interval
     |     |     +--ro average-mean-path-delay?      time-interval
     |     |     +--ro minimum-mean-path-delay?      time-interval
     |     |     +--ro maximum-mean-path-delay?      time-interval
     |     |     +--ro stddev-mean-path-delay?       time-interval
     |     |     +--ro average-offset-from-master?   time-interval
     |     |     +--ro minimum-offset-from-master?   time-interval
     |     |     +--ro maximum-offset-from-master?   time-interval
     |     |     +--ro stddev-offset-from-master?    time-interval
     |     +--rw enhanced-metrics-ds {enhanced-metrics}?
     |     |  +--rw enable?   boolean
     |     +--rw ports
     |        +--rw port* [port-index]
     |           +--rw port-index                        uint16
     |           +--rw underlying-interface?             if:interface-ref
     |           +--rw port-ds
     |           |  +--ro port-identity
     |           |  |  +--ro clock-identity?   clock-identity
     |           |  |  +--ro port-number?      uint16
     |           |  +--ro port-state?                    port-state
     |           |  +--rw log-min-delay-req-interval?    int8
     |           |  +--ro mean-link-delay?               time-interval
     |           |  x--ro peer-mean-path-delay?          time-interval
     |           |  +--rw log-announce-interval?         int8
     |           |  +--rw announce-receipt-timeout?      uint8
     |           |  +--rw log-sync-interval?             int8
     |           |  +--rw delay-mechanism?               delay-mechanism
     |           |  +--rw log-min-pdelay-req-interval?   int8
     |           |  +--rw version-number?                uint8
     |           |  +--rw minor-version-number?          uint8
     |           |  +--rw delay-asymmetry?               time-interval
     |           |  +--rw port-enable?                   boolean
     |           |  +--rw master-only?                   boolean
     |           +--rw timestamp-correction-port-ds {timestamp-correction}?
     |           |  +--rw egress-latency?    time-interval
     |           |  +--rw ingress-latency?   time-interval
     |           +--rw asymmetry-correction-port-ds {asymmetry-correction}?
     |           |  +--rw constant-asymmetry?         time-interval
     |           |  +--rw scaled-delay-coefficient?   relative-difference
     |           |  +--rw enable?                     boolean
     |           +--rw description-port-ds
     |           |  +--ro profile-identifier?   string
     |           |  +--ro protocol-address
     |           |     +--ro network-protocol?   identityref
     |           |     +--ro address-length?     uint16
     |           |     +--ro address-field?      string
     |           +--rw unicast-negotiation-port-ds {unicast-negotiation}?
     |           |  +--rw enable?   boolean
     |           +--rw alternate-master-port-ds {alternate-master}?
     |           |  +--rw number-of-alt-masters?            uint8
     |           |  +--rw tx-alt-multicast-sync?            boolean
     |           |  +--rw log-alt-multicast-sync-interval?   int8
     |           +--rw unicast-discovery-port-ds {unicast-discovery}?
```

```
     |           |  +--ro max-table-size?       uint16
     |           |  +--rw log-query-interval?   int8
     |           |  +--rw port-address* [index]
     |           |     +--rw index               uint16
     |           |     +--rw network-protocol?   identityref
     |           |     +--rw address-length?     uint16
     |           |     +--rw address-field?      string
     |           +--rw acceptable-master-port-ds {acceptable-master}?
     |           |  +--rw enable?   boolean
     |           +--rw l1-sync-basic-port-ds {l1-sync}?
     |           |  +--rw enabled?                        boolean
     |           |  +--rw tx-coherent-is-required?        boolean
     |           |  +--rw rx-coherent-is-required?        boolean
     |           |  +--rw congruent-is-required?          boolean
     |           |  +--rw opt-params-enabled?             boolean
     |           |  +--rw log-l1sync-interval?            int8
     |           |  +--rw l1sync-receipt-timeout?         uint8
     |           |  +--ro link-alive?                     boolean
     |           |  +--ro is-tx-coherent?                 boolean
     |           |  +--ro is-rx-coherent?                 boolean
     |           |  +--ro is-congruent?                   boolean
     |           |  +--ro l1sync-state?                   l1sync-state
     |           |  +--ro peer-tx-coherent-is-required?   boolean
     |           |  +--ro peer-rx-coherent-is-required?   boolean
     |           |  +--ro peer-congruent-is-required?     boolean
     |           |  +--ro peer-is-tx-coherent?            boolean
     |           |  +--ro peer-is-rx-coherent?            boolean
     |           |  +--ro peer-is-congruent?              boolean
     |           +--rw l1-sync-opt-params-port-ds {l1-sync}?
     |           |  +--rw timestamps-corrected-tx?         boolean
     |           |  +--ro phase-offset-tx-valid?           boolean
     |           |  +--ro phase-offset-tx?                 time-interval
     |           |  +--ro phase-offset-tx-timestamp
     |           |  |  +--ro seconds-field?       uint64
     |           |  |  +--ro nanoseconds-field?   uint32
     |           |  +--ro frequency-offset-tx-valid?       boolean
     |           |  +--ro frequency-offset-tx?             time-interval
     |           |  +--ro frequency-offset-tx-timestamp
     |           |     +--ro seconds-field?       uint64
     |           |     +--ro nanoseconds-field?   uint32
     |           +--ro communication-cap-port-ds
     |           |  +--ro sync
     |           |  |  +--ro multicast-capable?              boolean
     |           |  |  +--ro unicast-capable?                boolean
     |           |  |  +--ro unicast-negotiation-capable?    boolean
     |           |  |  +--ro unicast-negotiation-required?   boolean
     |           |  +--ro delay-resp
     |           |     +--ro multicast-capable?              boolean
     |           |     +--ro unicast-capable?                boolean
     |           |     +--ro unicast-negotiation-capable?    boolean
     |           |     +--ro unicast-negotiation-required?   boolean
     |           +--rw performance-monitoring-port-ds {performance-monitoring}?
     |           |  +--ro record-list-peer-delay* [index]
     |           |  |  +--ro index                       uint16
     |           |  |  +--ro pm-time?                    yang:timestamp
     |           |  |  +--ro average-mean-link-delay?    time-interval
     |           |  |  +--ro min-mean-link-delay?        time-interval
     |           |  |  +--ro max-mean-link-delay?        time-interval
     |           |  |  +--ro stddev-mean-link-delay?     time-interval
     |           |  +--ro record-list* [index]
     |           |     +--ro index                        uint16
     |           |     +--ro pm-time?                     yang:timestamp
     |           |     +--ro announce-tx?                 yang:zero-based-counter32
```

```
   |           |     +--ro announce-rx?                yang:zero-based-counter32
   |           |     +--ro announce-foreign-rx?        yang:zero-based-counter32
   |           |     +--ro sync-tx?                    yang:zero-based-counter32
   |           |     +--ro sync-rx?                    yang:zero-based-counter32
   |           |     +--ro follow-up-tx?               yang:zero-based-counter32
   |           |     +--ro follow-up-rx?               yang:zero-based-counter32
   |           |     +--ro delay-req-tx?               yang:zero-based-counter32
   |           |     +--ro delay-req-rx?               yang:zero-based-counter32
   |           |     +--ro delay-resp-tx?              yang:zero-based-counter32
   |           |     +--ro delay-resp-rx?              yang:zero-based-counter32
   |           |     +--ro pdelay-req-tx?              yang:zero-based-counter32
   |           |     +--ro pdelay-req-rx?              yang:zero-based-counter32
   |           |     +--ro pdelay-resp-tx?             yang:zero-based-counter32
   |           |     +--ro pdelay-resp-rx?             yang:zero-based-counter32
   |           |     +--ro pdelay-resp-follow-up-tx?   yang:zero-based-counter32
   |           |     +--ro pdelay-resp-follow-up-rx?   yang:zero-based-counter32
   |           +--rw common-services-port-ds
   |           |  +--ro cmlds-link-port-port-number?   uint16 {cmlds}?
   |           +--rw external-port-config-port-ds {external-port-config}?
   |           |  +--rw desired-state?   port-state
   |           +--rw slave-monitoring-port-ds {slave-monitoring}?
   |              +--rw enable?                            bits
   |              +--rw events-per-rx-sync-timing-tlv?     uint8
   |              +--rw events-per-rx-sync-computed-tlv?   uint8
   |              +--rw events-per-tx-timestamps-tlv?      uint8
   |              +--rw tx-event-type?                     uint8
   |              +--rw rx-sync-timing-tlv-message-m?      uint8
   |              +--rw rx-sync-computed-tlv-message-m?    uint8
   |              +--rw tx-timestamps-tlv-message-m?       uint8
   x--rw transparent-clock-default-ds
   |  x--ro clock-identity?    clock-identity
   |  x--ro number-ports?      uint16
   |  x--rw delay-mechanism?   delay-mechanism
   |  x--rw primary-domain?    uint8
   x--rw transparent-clock-ports
   |  x--rw port* [port-index]
   |     +--rw port-index              uint16
   |     +--rw underlying-interface?   if:interface-ref
   |     +--rw port-ds
   |        x--ro port-identity
   |        |  +--ro clock-identity?   clock-identity
   |        |  +--ro port-number?      uint16
   |        x--rw log-min-pdelay-req-interval?   int8
   |        x--rw faulty-flag?                   boolean
   |        x--ro peer-mean-path-delay?          time-interval
   +--rw common-services
      +--rw cmlds {cmlds}?
         +--rw default-ds
         |  +--ro clock-identity?      clock-identity
         |  +--ro number-link-ports?   uint16
         +--rw ports
            +--rw port* [port-index]
               +--rw port-index                     uint16
               +--rw underlying-interface?          if:interface-ref
               +--rw link-port-ds
               |  +--ro port-identity
               |  |  +--ro clock-identity?   clock-identity
               |  |  +--ro port-number?      uint16
               |  +--ro domain-number?                 uint8
               |  +--ro service-measurement-valid?     boolean
               |  +--ro mean-link-delay?               time-interval
               |  +--ro scaled-neighbor-rate-ratio?    int32
               |  +--rw log-min-pdelay-req-interval?   int8
```

```
|  +--rw version-number?                uint8
|  +--rw minor-version-number?          uint8
|  +--rw delay-asymmetry?               time-interval
+--rw timestamp-correction-port-ds {timestamp-correction}?
|  +--rw egress-latency?    time-interval
|  +--rw ingress-latency?   time-interval
+--rw asymmetry-correction-port-ds {asymmetry-correction}?
|  +--rw enable?                     boolean
|  +--rw constant-asymmetry?         time-interval
|  +--rw scaled-delay-coefficient?   relative-difference
+--rw performance-monitoring-port-ds {performance-monitoring}?
   +--ro record-list-peer-delay* [index]
   |  +--ro index                      uint16
   |  +--ro pm-time?                   yang:timestamp
   |  +--ro average-mean-link-delay?   time-interval
   |  +--ro min-mean-link-delay?       time-interval
   |  +--ro max-mean-link-delay?       time-interval
   |  +--ro stddev-mean-link-delay?    time-interval
   +--ro record-list* [index]
      +--ro index                       uint16
      +--ro pm-time?                    yang:timestamp
      +--ro announce-tx?                yang:zero-based-counter32
      +--ro announce-rx?                yang:zero-based-counter32
      +--ro announce-foreign-rx?        yang:zero-based-counter32
      +--ro sync-tx?                    yang:zero-based-counter32
      +--ro sync-rx?                    yang:zero-based-counter32
      +--ro follow-up-tx?               yang:zero-based-counter32
      +--ro follow-up-rx?               yang:zero-based-counter32
      +--ro delay-req-tx?               yang:zero-based-counter32
      +--ro delay-req-rx?               yang:zero-based-counter32
      +--ro delay-resp-tx?              yang:zero-based-counter32
      +--ro delay-resp-rx?              yang:zero-based-counter32
      +--ro pdelay-req-tx?              yang:zero-based-counter32
      +--ro pdelay-req-rx?              yang:zero-based-counter32
      +--ro pdelay-resp-tx?             yang:zero-based-counter32
      +--ro pdelay-resp-rx?             yang:zero-based-counter32
      +--ro pdelay-resp-follow-up-tx?   yang:zero-based-counter32
      +--ro pdelay-resp-follow-up-rx?   yang:zero-based-counter32
```

15.4.4 Module

The YANG module specified by IEEE Std 1588™ is the revision 2023-05-26, as indicated in the YANG module, located at:

Editor's Note: The published standard will replace this editor's note with a URL to the published YANG module. This URL cannot be provided during the draft balloting process. For review of the YANG module for this ballot, the YANG module ieee1588-ptp.yang is attached to the balloted PDF file.

NOTE—Copyright release for YANG: Users of this standard may freely reproduce the YANG module defined by this standard so that they can be used for their intended purpose.

Insert the following in Annex Q (Informative) Bibliography:

[Bxx] IETF RFC 2579 (STD 58), Textual Conventions for SMIv2, McCloghrie, K., D. Perkins, J. Schoenwaelder, J. Case, M. Rose, and S. Waldbusser, Apr. 1999.

[Byy] IETF RFC 2580 (STD 58), Conformance Statements for SMIv2, McCloghrie, K., D. Perkins, J. Schoenwaelder, J. Case, M. Rose, and S. Waldbusser, Apr. 1999

[Bzz] IETF RFC 3444, On the Difference between Information Models and Data Models, A. Pras, and J. Schoenwaelde, Jan. 2003

[Bkk] IETF RFC 2863, The Interfaces Group MIB, K. McCloghrie and F. Kastenholz, Jun. 2000

[Bvv] IETF RFC 3411, An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks, D. Harrington, R. Presuhn, and B. Wijnen, Dec. 2002

[Bnn] IETF RFC 2863, The Interfaces Group MIB, K. McCloghrie, F. Kastenholz, June 2000.

[Buu] IETF RFC 8173, Precision Time Protocol Version 2 (PTPv2) Management Information Base, V. Shankarkumar, L. Montin, T. Frost, and G. Dowd.

[Bss] IETF RFC 3410, Introduction and Applicability Statements for Internet Standard Management Framework, J. Case, R. Mundy, D. Partain, and B. Stewart, Dec. 2002.

[Btt] IETF RFC 8342, Network Management Datastore Architecture (NMDA), M. Björklund, J. Schönwälder, P.A. Shafer, K. Watsen, and R. Wilton, Mar. 2018.

[Bww] IEC 62439 Industrial communication networks - High availability automation networks.

[Baa] IETF RFC 7950, The YANG 1.1 Data Modeling Language, M. Björklund, J. Schönwälder, Jan. 2020.

[Bbb] IETF RFC 6241, Network Configuration Protocol (NETCONF), R. Enns, M. Björklund, A. Bierman, J. Schönwälder, Jan. 2020.

[Bcc] IETF RFC 8040, RESTCONF Protocol, A. Bierman, M. Björklund, K. Watsen, Jan. 2020.

[Bdd] IETF RFC 8407, Guidelines for Authors and Reviewers of Documents Containing YANG Data Models, A. Bierman, Jan. 2020.

[Bee] IETF RFC 8525, YANG Library, A. Bierman , M. Björklund , J. Schönwälder , K. Watsen , R. Wilton, Mar. 2021.

[Bff] IETF RFC 8340, YANG Tree Diagrams, M. Björklund , L. Berger, Mar. 2018.

[Bgg] IETF RFC 8575 ANG Data Model for the Precision Time Protocol (PTP), Y. Jiang, X. Liu, J. Xu, R. Cummings, May 2019.