Glenn Parsons
IEEE 802.1 Working Group

December 2019
Martin Rostan
EtherCAT Technology Group
EtherCAT
Ostendstraße 196, 90482 Nürnberg, Germany

Dear Martin Rostan:

The following IEEE Standards draft is being submitted by

IEEE 802.1 Working Group
(hereinafter **IEEE 802.1**)
to

EtherCAT
(hereinafter **EtherCAT**)

for coordination with

EtherCAT Technology Group
(hereinafter **EtherCAT Technology Group**):

IEEE P802.1, Draft 8.0, Timing and Synchronization for Time-Sensitive Applications

IEEE is providing access to this IEEE Standards draft via an IEEE website for the following purpose:
[ ] Request for comments from individual participants/reviewers from **Organization Group**
[**x**] Technical coordination between **IEEE 802.1** and
EtherCAT Technology Group *until date* or completion of the project
[ ] Education during regulatory consideration for **<Regulatory Body>** *until date* or completion of the project

IEEE P802.1AS-Rev/D8.0

Draft Standard for Local and Metropolitan Area Networks—

Timing and Synchronization for Time-Sensitive Applications

Sponsor
LAN/MAN Standards Committee of the IEEE Computer Society

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

Abstract: This standard defines a protocol and procedures for the transport of timing over local area networks. It includes the transport of synchronized time, the selection of the timing source (i.e., best master), and the indication of the occurrence and magnitude of timing impairments (i.e., phase and frequency discontinuities).
Keywords: best master, frequency offset, grandmaster, IEEE 802.1AS, phase offset, synchronization, syntonization, time-aware system, PTP Instance, PTP Relay Instance, PTP End Instance.

DRAFT STATUS:

This draft was prepared by the editor for Sponsor ballot.

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

The base standard to which this is a revision, IEEE Std 802.1AS-2011, and the corrigenda, IEEE Std 802.1AS-2011/Cor 1-2013 and IEEE Std 802.1AS-2011/Cor 2-2015, can be obtained free of charge from:

http://standards.ieee.org

Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page, appear in all standards and may be found under the heading "Important Notices and Disclaimers Concerning IEEE Standards Documents." They can also be obtained on request from IEEE or viewed at http://standards.ieee.org/IPR/disclaimers.html.

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

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

IEEE Standards do not guarantee or ensure safety, security, health, or environmental protection, or ensure against interference with or from other devices or networks. 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.

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

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

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

IN NO EVENT SHALL IEEE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR

BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE PUBLICATION, USE OF, OR RELIANCE UPON ANY STANDARD, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

Translations

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

Official statements

A statement, written or oral, that is not processed in accordance with the IEEE-SA Standards Board Operations Manual shall not be considered or inferred to be the official position of IEEE or any of its committees and shall not be considered to be, or be relied 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 his or her views should be considered the personal views of that individual rather than the formal position of IEEE.

Comments on standards

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

Comments on standards should be submitted to the following address:

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

Laws and regulations

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

Copyrights

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

Photocopies

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

Updating of IEEE Standards documents

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

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

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

Errata

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

Patents

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

Essential Patent Claims may exist for which a Letter of Assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims, or determining whether any licensing terms or conditions provided in association 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.

Introduction to IEEE P802.1AS-Rev/D8.0™

(This introduction is not part of P802.1AS-Rev/D8.0, Draft Standard for Local and Metropolitan Area Networks—Timing and Synchronization for Time-Sensitive Applications.)

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

Secretary, IEEE-SA Standards Board
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
USA

Participants

The following is a list of participants in the Time-Sensitive Networking activities of the IEEE 802.1 Working Group during the development of P802.1AS-Rev. Voting members at the time of publication are marked with an asterisk (*).

When the IEEE 802.1 Working Group approved IEEE Std 802.1AS-2019, it had the following membership:

Glenn Parsons, *Chair*
John Messenger, *Vice Chair*
Janos Farkas, *Chair, Time-Sensitive Networking Task Group*
Geoffrey Garner, *Editor*

<<TBA>>

The following members of the balloting committee voted on P802.1AS-Rev. Balloters may have voted for approval, disapproval, or abstention.

<<TBA>>

When the IEEE-SA Standards Board approved this standard on <<TBA>>, it had the following membership:

???, *Chair* **???,** *Vice Chair*
???, *Secretary*

<<TBA>>

Contents

List of figures

List of tables

IEEE Standard for
Local and metropolitan area networks—

Timing and Synchronization for Time-Sensitive Applications

IMPORTANT NOTICE: This standard is not intended to ensure safety, security, health, or environmental protection. Implementers of the standard are responsible for determining appropriate safety, security, environmental, and health practices or regulatory requirements.

This IEEE document is made available for use subject to important notices and legal disclaimers. These notices and disclaimers appear in all publications containing this document and may be found under the heading "Important Notice" or "Important Notices and Disclaimers Concerning IEEE Documents." They can also be obtained on request from IEEE or viewed at https://standards.ieee.org/ipr/disclaimers.html.

1. Overview

1.1 Scope

This standard specifies the protocol and procedures used to ensure that the synchronization requirements are met for time-sensitive applications, such as audio,video, and time-sensitive control, across networks; for example, IEEE 802 and similar media. This includes the maintenance of synchronized time during normal operation and following addition, removal, or failure of network components and network reconfiguration. It specifies the use of IEEE 1588™ specifications where applicable in the context of IEEE Std 802.1Q.[1] Synchronization to an externally provided timing signal (e.g., a recognized timing standard such as UTC or TAI) is not part of this standard but is not precluded.

1.2 Purpose

This standard enables systems to meet the respective jitter, wander, and time-synchronization requirements for time-sensitive applications. This includes applications that involve multiple streams delivered to multiple endpoints. To facilitate the widespread use of packet networks for these applications, synchronization information is one of the components needed at each network element where time-sensitive application data are mapped or demapped or a time-sensitive function is performed. This standard leverages

[1]Information on references can be found in Clause 2.

the work of the IEEE 1588 Working Group by developing the additional specifications needed to address these requirements.

2. Normative references

The following referenced documents are indispensable for the application of this standard (i.e., they must be understood and used, so each referenced document is cited in text and its relationship to this document is explained). For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

IEEE Std 754™-2008, IEEE Standard for Floating-Point Arithmetic.

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.

IEEE Std 802.1AC™-2016, IEEE Standard for Local and metropolitan area networks—Media Access Control (MAC) Service Definition.[2, 3]

IEEE Std 802.1AX™-2014, IEEE Standard for Local and metropolitan area networks—Link Aggregation.

IEEE Std 802.1Q™-2018, IEEE Standard for Local and metropolitan area networks— Bridges and Bridged Networks.

IEEE Std 802.3™-2018, IEEE Standard for Ethernet.

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

IEEE Std 1588™-2019, IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems.

IERS Bulletin C (see https://www.iers.org/IERS/EN/Publications/Bulletins/bulletins.html).

IETF RFC 3410 (December 2002), Introduction and Applicability Statements for Internet Standard Management Framework, Case, J., Mundy, R., Partain, D, and Stewart, B.[4]

ITU-T Recommendation G.9960 (ex. G.hn), Unified high-speed wire-line based home networking transceivers—System architecture and physical layer specification, June 2010.[5]

ITU-T Recommendation G.9961, Data link layer (DLL) for unified high-speed wire-line based home networking transceivers, June 2010.

ITU-T Recommendation G.984.3, Amendment 2 (2009-11) Gigabit-capable Passive Optical Networks (G-PON): Transmission convergence layer specification—Time-of-day distribution and maintenance updates and clarifications, November 2009.

---

[2]IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, Piscataway, NJ 08854, USA (http://standards.ieee.org).

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

[4]IETF RFCs are available from the Internet Engineering Task Force Web site at http://www.ietf.org/rfc.html.

[5]ITU-T publications are available from the International Telecommunications Union, Place des Nations, CH-1211, Geneva 20, Switzerland/Suisse (http://www.itu.int/).

MoCA® MAC/PHY Specification v2.0, MoCA-M/P-SPEC-V2.0-20100507, Multimedia over Coax Alliance (MoCA).[6]

ISO 80000-3:2006, Quantities and units - Part 3: Space and time.

IETF RFC 2863, The Interfaces Group MIB, June 2000.

IETF RFC 3418, Management Information Base (MIB) for the Simple Network Management Protocol (SNMP), December 2002.

[6]MoCA specifications are available from the Multimedia over Coax Alliance at http://www.mocalliance.org/specs.

3. Definitions

For the purposes of this document, the following terms and definitions apply. *The IEEE Standards Dictionary: Glossary of Terms & Definitions* should be consulted for terms not defined in this clause.[7]

3.1 accuracy: The mean of the time or frequency error between the clock under test and a perfect reference clock over an ensemble of measurements.

3.2 Bridge: Either a MAC Bridge or a VLAN-aware Bridge, as specified in Clause 5 of IEEE Std 802.1Q-2018.

3.3 clock: A physical device that is capable of providing a measurement of the passage of time since a defined epoch.

3.4 direct communication: A communication of IEEE 802.1AS information between two PTP Instances with no intervening PTP Instance.

3.5 end station: A device attached to a local area network (LAN) or metropolitan area network (MAN), which acts as a source of, and/or destination for, traffic carried on the LAN or MAN.

NOTE—In this standard, an *end station* is sometimes referred to as a *station*.[8]

3.6 event message: A message that is timestamped on egress from a PTP Instance and ingress to a PTP Instance.

NOTE—See 8.4.3.

3.7 fractional frequency offset: The fractional frequency offset, y, between a measured clock and a reference clock is defined as:

$$y = \frac{f_m - f_r}{f_r}$$

where f_m is the frequency of the measured clock and f_r is the frequency of the reference clock. The measurement units of f_m and f_r are the same.

3.8 general message: A message that is not timestamped.

3.9 gPTP communication path: A segment of a generalized precision time protocol (gPTP) domain that enables direct communication between two PTP Instances.

NOTE—See 8.1.

3.10 grandmaster: The PTP Instance that contains the best clock, as determined by the best master clock algorithm (BMCA), in the generalized precision time protocol (gPTP) domain of the PTP Instance.

3.11 local area network (LAN): A network of devices, whether indoors or outdoors, covering a limited geographic area, e.g., a building or campus.

3.12 message timestamp point: A point within an event message serving as a reference point for when a timestamp is taken.

---

[7]*The IEEE Standards Dictionary: Glossary of Terms & Definitions* is available at http://shop.ieee.org/.

[8]Notes in text, tables, and figures of a standard are given for information only and do not contain requirements needed to implement this standard.

3.13 message type: The message type of a message is the name of the respective message, e.g., Sync, Announce, Timing Measurement Frame.

3.14 precision: A measure of the deviation from the mean of the time or frequency error between the clock under test and a perfect reference clock.

3.15 primary reference: A source of time and/or frequency that is traceable to international standards. *See also:* **traceability.**

3.16 PTP End Instance: A PTP Instance that is capable of acting as the source of synchronized time on the network, or destination of synchronized time using the IEEE 802.1AS protocol, or both.

3.17 PTP Instance: A PTP Relay Instance or a PTP End Instance. A PTP Instance implements those portions of this standard indicated as applicable to a PTP Relay Instance or a PTP End Instance. Each PTP Instance operates in exactly one domain.

3.18 PTP Link: Within a domain, a network segment between two PTP Ports using the peer-to-peer delay mechanism of this standard. The peer-to-peer delay mechanism is designed to measure the propagation time over such a link.

NOTE—A PTP Link between PTP Ports of PTP Instances is also a gPTP Communication Path (see 3.9).

3.19 PTP Relay Instance: A PTP Instance that is capable of communicating synchronized time received on one port to other ports, using the IEEE 802.1AS protocol.

NOTE—A PTP Relay Instance could, for example, be contained in a bridge, a router, or a multi-port end station.

3.20 recognized standard source of time: A recognized standard time source is a source external to IEEE 1588 precision time protocol (PTP) that provides time that is traceable to the international standards laboratories maintaining clocks that form the basis for the *temps atomique international* (international atomic time) (TAI) and coordinated universal time (UTC) timescales. Examples of these are National Institute of Standards and Technology (NIST) timeservers and global navigation satellite systems (GNSSs).

3.21 reference plane: The boundary between a port of a PTP Instance and the network physical medium. Timestamp events occur as frames cross this interface.

3.22 residence time: The duration of the time interval between the receipt of a time-synchronization event message by a PTP Instance, and the sending of the next subsequent time-synchronization event message on another port of that PTP Instance. The residence time can be different for different ports.

NOTE—If a port of a PTP Instance sends a time-synchronization event message without having received a time-synchronization event message, i.e., if sync receipt timeout occurs (see 10.7.3.1), the duration of the interval between the most recently received time-synchronization event message and the sent time-synchronization event message is mathematically equivalent to residence time; however, this interval is not normally referred to as a *residence time*.

3.23 stability: A measure of how the mean of the time or frequency error between the clock under test and a perfect reference clock varies with respect to variables such as time, temperature, etc.

3.24 synchronized time: The synchronized time of an event is the time of that event relative to the grandmaster.

NOTE—If there is a change in the grandmaster or grandmaster time base, the *synchronized time* can experience a phase and/or frequency step.

3.25 synchronized PTP Instances: Two PTP Instances are synchronized to a specified uncertainty if they have the same epoch and their measurements of the time of a single event at an arbitrary time differ by no more than that uncertainty.

NOTE—See 8.2.2.

3.26 syntonized PTP Instances: Two PTP Instances are syntonized if the duration of the second is the same on both, which means the time measured by each advances at the same rate. They can but need not share the same epoch.

3.27 time-aware system: A device that contains one or more PTP Instance and/or PTP services (e.g., Common Mean Link Delay Service (see 11.2.17).

3.28 timestamp measurement plane: The plane at which timestamps are captured. If the timestamp measurement plane is different from the reference plane, the timestamp is corrected for ingressLatency and/or egressLatency. *See:* reference plane.

NOTE—For timestamp on egress and ingress, see 8.4.3.

3.29 traceability: See IEEE Std 1588-2019, 3.1.82.

4. Acronyms and abbreviations

AAI-48	48-bit Administratively Assigned Identifier
AAI-64	64-bit Administratively Assigned Identifier
Ack	acknowledgement
ADEV	Allan deviation
AP	(wireless LAN) access point
ARB	arbitrary
AV	audio/video
AVB	audio/video bridging
AVB network	audio/video bridged network
BC	boundary clock
BMC	best master clock
BMCA	best master clock algorithm
CID	Company ID
CMLDS	Common Mean Link Delay Service
CSN	coordinated shared network
CTC	channel time clock
ELI-48	48-bit Extended Local Identifier
ELI-64	64-bit Extended Local Identifier
EPON	IEEE 802.3 Ethernet passive optical network, as specified in IEEE Std 802.3
ESS	extended service set
EUI-48	48-bit Extended Unique Identifier
EUI-64	64-bit Extended Unique Identifier
FTM	fine timing measurement
G.hn	ITU-T G.9960 and ITU-T G.9961
GM	grandmaster
GMT	Greenwich mean time

GNSS	global navigation satellite system
GPS	global positioning (satellite) system
gPTP	generalized precision time protocol (IEEE Std 802.1AS)
IERS	International Earth Rotation and Reference Systems Service
IP	Internet protocol
IS	integration service
ISO	International Organization for Standardization[9]
ISS	Internal Sublayer Service
LAN	local area network
LCI	location configuration information
LLC	logical link control
MAC	media access control
MACsec	media access control security
MA-L	MAC addresses large
MA-M	MAC addresses medium
MAN	metropolitan area network
MA-S	MAC addresses small
MLME	IEEE 802.11 MAC layer management entity
MPCP	IEEE 802.3 multipoint control protocol
MPDPDU	IEEE 802.3 MPCP data unit
MII	media-independent interface
MD	media-dependent
NTP	network time protocol[10]
NUI-48	48-bit network unique identifier
NUI-64	64-bit network unique identifier
OC	ordinary clock

[9]Information available at www.iso.org.
[10]Information available at www.ietf.org/rfc/rfc1305.txt.

OLT	IEEE 802.3 optical line terminal
ONU	IEEE 802.3 optical network unit
OSSP	organization-specific slow protocol
OUI	Organizationally Unique Identifier
P2P	peer-to-peer
PAR	project authorization request
PICS	Protocol Implementation Conformance Statement
PLL	phased-lock loop
POSIX®	portable operating system interface (see ISO/IEC 9945:2003 [B8][11])
PTP	IEEE 1588 precision time protocol
PTPDEV	PTP deviation
RTT	round-trip time
SAI-48	48-bit Standard Assigned Identifier
SAI-64	64-bit Standard Assigned Identifier
SI	international system of units
SM	state machine
STA	station
TAI	temps atomique international (international atomic time)
TC	transparent clock
TDEV	time deviation
TDM	time division multiplexing
TDMA	time division multiple access
TG	task group
TM	timing measurement
TS	timestamp
UCT	unconditional transfer

[11]The numbers in brackets correspond to those of the bibliography in Annex G.

UTC	coordinated universal time
VLAN	virtual local area network
WG	Working Group
WLAN	wireless local area network

5. Conformance

This clause specifies the mandatory and optional capabilities provided by conformant implementations of this standard.

5.1 Requirements terminology

For consistency with existing IEEE and IEEE 802.1 standards, requirements placed upon conformant implementations of this standard are expressed using the following terminology:

a) ***shall*** is used for mandatory requirements;
b) ***may*** is used to describe implementation or administrative choices ("may" means "is permitted to," and hence, "may" and "may not" mean precisely the same thing);
c) ***should*** is used for recommended choices (the behaviors described by "should" and "should not" are both permissible but not equally desirable choices).

The Protocol Implementation Conformance Statement (PICS) proforma (see Annex A) reflects the occurrences of the words shall, may, and should within the standard.

The standard 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 ***can not***. The word ***allow*** is used as a replacement for the cliché "support the ability for," and the word ***capability*** means "can be configured to."

5.2 Protocol Implementation Conformance Statement (PICS)

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

5.3 Time-aware system requirements

An implementation of a time-aware system shall support at least one PTP Instance.

5.4 PTP Instance requirements

An implementation of a PTP Instance shall:

a) Implement the generalized precision time protocol (gPTP) requirements specified in Clause 8;
b) Support the requirements for time-synchronization state machines (10.1.1, 10.2.1, 10.2.2, a, 10.2.5, 10.2.6);
c) Support at least one port;
d) On each supported port, implement the PortSyncSyncReceive state machine (10.2.8);
e) Implement the ClockSlaveSync state machine (10.2.13);
f) Support the following best master clock algorithm (BMCA) requirements:
    1) Implement the BMCA (10.3.2, 10.3.3, 10.3.4, 10.3.5, 10.3.6, 10.3.8, 10.3.10);
    2) Implement specifications for externalPortConfigurationEnabled value of FALSE (10.3.1);
    3) Implement the PortAnnounceReceive state machine (10.3.11);

4) Implement the PortAnnounceInformation state machine (10.3.12);
5) Implement the PortStateSelection state machine (10.3.13);
6) Have the BMCA as the default mode of operation, with externalPortConfiguration FALSE, on domain 0;
7) Implement at least one of the possiblities for externalPortConfigurationEnabled (i.e., FALSE, meaning BMCA is used, and TRUE, meaning external port configuration is used) on domains other than domain 0;

g) Implement the SiteSyncSync state machine (10.2.7);
h) Implement the state machines related to signaling gPTP protocol capability (10.4);
i) For receipt of all messages, and for transmission of all messages except Announce and Signaling, support the message requirements as specified in 10.5, 10.6, and 10.7;
j) Support the performance requirements in B.1 and B.2.4.

5.4.1 Time-aware system options

An implementation of a PTP Instance should:

a) Support the performance requirements in B.2.2 and B.2.3;

An implementation of a PTP Instance may:

b) Support the following media-independent master capability on at least one port:
   1) Implement the PortSyncSyncSend state machine (10.2.12);
   2) Implement the PortAnnounceTransmit state machine (10.3.16);
   3) Implement the AnnounceIntervalSetting state machine (10.3.17);
   4) For transmit of the Announce message, support the message requirements as specified in 10.5, 10.6, and 10.7;
c) Support the following for grandmaster capability:
   1) Support the media-independent master capability specified in 5.4.1(b);
   2) Support the requirements for a grandmaster-capable PTP Instance (10.1.2);
   3) Implement the ClockMasterSyncSend state machine (10.2.9);
   4) Implement the ClockMasterSyncOffset state machine (10.2.10);
   5) Implement the ClockMasterReceive state machine (10.2.11).
d) Support more than one port as a PTP Relay Instance (5.4.2);
e) Support transmit of the Signaling message according to the message requirements as specified in 10.5 and 10.6;
f) Support more than one PTP Instance, which allows for more than one domain (7.2.3);
g) Support the following external port configuration capability on at least one port:
   1) Implement specifications for externalPortConfigurationEnabled value of true (10.3.1);
   2) Implement the PortAnnounceInformationExt state machine (10.3.14);
   3) Implement the PortStateSettingExt state machine (10.3.15);
h) Implement the SyncIntervalSetting state machine (10.3.18);
i) Implement one or more of the application interfaces specified in clause 9; A PTP Instance that claims to support application interfaces shall state which application interfaces are supported;
j) Support timing and synchronization management as specified in clause 14;
k) Support the use of a remote management protocol. A PTP Instance that claims to support remote management shall:
   1) State which remote management protocol standard(s) or specification(s) are supported;
   2) State which standard(s) or specification(s) for managed object definitions and encodings are supported for use by the remote management protocol;
   3) If the Simple Network Management Protocol (SNMP) is supported as a remote management protocol, support the managed object definitions specified as SMIv2 MIB modules in clause 15;

l) Implement both BMCA and external port configuration on domains other than domain 0; if both possibilities are implemented on domains other than domain 0, the the default value of externalPortConfigurationEnabled shall be FALSE.

5.4.2 PTP Relay Instance requirements

An implementation of a PTP Relay Instance shall:

a) Support more than one port;
b) Support the PTP Instance requirements specified in 5.4;
c) Support the media-independent master capability specified in 5.4.1(b);

5.5 MAC-specific timing and synchronization methods for IEEE 802.3 full-duplex links

An implementation of a time-aware system with IEEE 802.3 MAC services to physical ports shall:

a) Support full-duplex operation, as specified in IEEE Std 802.3-2018, 4.2 and Annex 4A;
b) Support the requirements as specified in clause 11;
c) Implement the SyncIntervalSetting state machine (10.3.18);

An implementation of a PTP Instance with IEEE 802.3 MAC services to physical ports may:

d) Support asymmetry measurement mode as specified in 10.3.12, 10.3.13, 10.3.16, 11.2.14, 11.2.15, and 11.2.19, 14.8.46;
e) Support one-step capability on receive as specified in 11.2.14;
f) Support one-step capability on transmit as specified in 11.2.15;
g) Support propagation delay averaging, as specified in 11.2.19.3.4;

5.6 MAC-specific timing and synchronization methods for IEEE Std 802.11

An implementation of a time-aware system with IEEE 802.11 MAC services to physical ports shall:

a) Support the requirements as specified in clause 12;
b) Support at least one of (i) the media-dependent master state machines (12.5.1) or (ii) the media-dependent slave state machine (12.5.2);

An implementation of a PTP End Instance with IEEE 802.11 MAC services to physical ports shall:

c) Support at least one of TIMINGMSMT as specified in IEEE Std 802.11 or FINETIMINGMSMT as specified in IEEE Std 802.11-2016

An implementation of a PTP Relay Instance with IEEE 802.11 MAC services to physical ports shall:

d) Support the requirements of TIMINGMSMT as specified in IEEE Std 802.11;

An implementation of a PTP Relay Instance with IEEE 802.11 MAC services to physical ports should:

e) Support FINETIMINGMSMT as specified in IEEE Std 802.11-2016;

NOTE—In order to maintain backward compatibility with existing TM-based end stations, the bridge is required to support TM. End stations are allowed to support either TM or FTM, or both. This allows new IEEE 802.1AS compliant end stations to only implement the newer FTM standard, which obviously requires a bridge that supports FTM.

5.7 MAC-specific timing and synchronization methods for IEEE 802.3 EPON

An implementation of a time-aware system with IEEE 802.3 EPON MAC services to physical ports shall:

a) Support the requirements as specified in IEEE Std 802.3-2018, Multipoint MAC Control (64.2 and 64.3) and Multipoint PCS and PMA extensions (65);
b) Support the requirements as specified in clause 13;

5.8 MAC-specific timing and synchronization methods for coordinated shared network (CSN)

An implementation of a time-aware system with CSN MAC services to physical ports shall:

a) Support the requirements as specified in clause 16;
b) Support at least one MoCA port (16.6.1) or ITU-T G.hn port (16.6.2);

6. Conventions

6.1 General

This clause defines various conventions and notation used in the standard, i.e., naming conventions, service specification method and notation, and data type definitions.

6.2 Service specification method and notation

The method and notation for specifying service interfaces is described in Clause 7 of IEEE Std 802.1AC-2012.

6.3 Lexical form syntax

A lexical form refers to:

a) A name
b) A data type

The conventions illustrated in the following list regarding lexical forms are used in this standard:

1) Type names: e.g., ClockQuality (no word separation, initial letter of each word capitalized);
2) Enumeration members and global constants: e.g., ATOMIC_CLOCK (underscore word separation, all letters capitalized);
3) Fields within PTP messages, instances of structures, and variables: e.g., secondsField, clockQuality, clockIdentity (two-word field names at a minimum, no word separation, initial word not capitalized, initial letter capitalization on subsequent words);
4) Members of a structure: e.g., clockQuality.clockClass (structure name followed by a period followed by the member name);
5) Data set name: e.g., defaultDS, parentDS, portDS, currentDS, timePropertiesDS (no word separation, initial word not capitalized, initial letter capitalization on subsequent words, followed by the letters DS);
6) Data set members: e.g., defaultDS.clockQuality.clockClass (Data set name followed by a period followed by a member name followed by a period followed by a member name); and
7) PTP message names: e.g., Sync, Pdelay_Req (underscore word separation, initial letter of each word capitalized.

When a lexical form appears in text, as opposed to in a type, or a format definition, the form is to be interpreted as singular, plural, or possessive as appropriate to the context of the text.

6.4 Data types and on-the-wire formats

6.4.1 General

The data types specified for the various variables and message fields define logical properties that are necessary for correct operation of the protocol or interpretation of IEEE 1588 precision time protocol (PTP) or IEEE802.11 message content.

NOTE—Implementations are free to use any internal representation of data types if the internal representation does not change the semantics of any quantity visible via communications using the IEEE 802.1AS protocol or in the specified operations of the protocol.

6.4.2 Primitive data types specifications

All non-primitive data types are derived from the primitive types in Table 6-1. Signed integers are represented in two's complement form.

Table 6-1—Primitive data types

Data type	Definition
Boolean	TRUE or FALSE
EnumerationN	N-bit enumerated value
UIntegerN	N-bit unsigned integer
IntegerN	N-bit signed integer
Nibble	4-bit field not interpreted as a number
Octet	8-bit field not interpreted as a number
OctetN	N-octet field not interpreted as a number, with N > 1
Float64	IEEE Std 754 binary64 (64-bit double-precision floating-point format)

NOTE—The Float64 data type was called Double in the 2011 edition of this standard. The semantics of the data type has not changed.

6.4.3 Derived data type specifications

6.4.3.1 ScaledNs

The ScaledNs type represents signed values of time and time interval in units of 2^{-16} ns.

typedef Integer96 ScaledNs;

For example: –2.5 ns is expressed as:

0xFFFF FFFF FFFF FFFF FFFD 8000

Positive or negative values of time or time interval outside the maximum range of this data type are encoded as the largest positive or negative value of the data type, respectively.

6.4.3.2 UScaledNs

The UScaledNs type represents unsigned values of time and time interval in units of 2^{-16} ns.

typedef UInteger96 UScaledNs;

For example: 2.5 ns is expressed as:

0x0000 0000 0000 0000 0002 8000

Values of time or time interval greater than the maximum value of this data type are encoded as the largest positive value of the data type.

6.4.3.3 TimeInterval

The TimeInterval type represents time intervals, in units of 2^{-16} ns.

```
struct TimeInterval
{
        Integer64 scaledNanoseconds;
};
```

For example: 2.5 ns is expressed as:

0x0000 0000 0002 8000

Positive or negative time intervals outside the maximum range of this data type are encoded as the largest positive and negative values of the data type, respectively.

6.4.3.4 Timestamp

The Timestamp type represents a positive time with respect to the epoch.

```
struct Timestamp
{
        UInteger48 seconds;
        UInteger32 nanoseconds;
};
```

The seconds member is the integer portion of the timestamp in units of seconds.

The nanoseconds member is the fractional portion of the timestamp in units of nanoseconds.

The nanoseconds member is always less than 10^9.

For example:

+2.000000001 seconds is represented by seconds = 0x0000 0000 0002 and nanoseconds= 0x0000 0001

6.4.3.5 ExtendedTimestamp

The ExtendedTimestamp type represents a positive time with respect to the epoch.

```
struct ExtendedTimestamp
{
        UInteger48 seconds;
        UInteger48 fractionalNanoseconds;
};
```

The seconds member is the integer portion of the timestamp in units of seconds.

The fractionalNanoseconds member is the fractional portion of the timestamp in units of 2^{-16} ns.

The fractionalNanoseconds member is always less than $(2^{16})(10^9)$.

For example:

+2.000000001 seconds is represented by seconds = 0x0000 0000 0002 and fractionalNanoseconds = 0x0000 0001 0000

6.4.3.6 ClockIdentity

The ClockIdentity type identifies a PTP Instance.

```
typedef Octet8 ClockIdentity;
```

6.4.3.7 PortIdentity

The PortIdentity type identifies a port of a PTP Instance.

```
struct PortIdentity
{
        ClockIdentity clockIdentity;
        UInteger16 portNumber;
};
```

6.4.3.8 ClockQuality

The ClockQuality represents the quality of a clock.

```
struct ClockQuality
{
        UInteger8 clockClass;
        Enumeration8 clockAccuracy;
        UInteger16 offsetScaledLogVariance;
};
```

6.4.4 Protocol data unit (PDU) formats

6.4.4.1 General

The data types defined in 6.4.2 and 6.4.3 shall be mapped onto the wire according to the mapping rules for the respective medium, e.g., IEEE Std 802.3 and IEEE Std 802.11, and the terms of 6.4.4.

IEEE 802.1AS PDUs consist of the messages defined or referenced in Clause 10, Clause 11, Clause 12, and Clause 13, based on the data types defined in 6.4.2 and 6.4.3. The internal ordering of the fields of the IEEE 802.1AS PDUs is specified in 6.4.4.3 to 6.4.4.5.

6.4.4.2 Numbering of bits within an octet

Bits are numbered with the most significant bit being 7 and the least significant bit being 0.

NOTE—The numbering and ordering of bits within an octet of a PDU, described here, is independent of and unrelated to the order of transmission of the bits on the underlying physical layer.

6.4.4.3 Primitive data types

Numeric primitive data types defined in 6.4.2 shall be formatted with the most significant octet nearest to the beginning of the PDU followed in order by octets of decreasing significance.

The Boolean data type TRUE shall be formatted as a single bit equal to 1 and FALSE as a single bit equal to 0.

Enumerations of whatever length shall be formatted as though the assigned values are unsigned integers of the same length, e.g., Enumeration16 shall be formatted as though the value had type UInteger16.

6.4.4.4 Arrays of primitive types

All arrays shall be formatted with the member having the lowest numerical index nearest to the beginning of the PDU followed by successively higher numbered members, without any padding. In octet arrays, the octet with the lowest numerical index is termed the most significant octet.

When a field containing more than one octet is used to represent a numeric value, the most significant octet shall be nearest to the beginning of the PDU, followed by successively less significant octets.

When a single octet contains multiple fields of primitive data types, the bit positions within the octet of each of the primitive types as defined in the message field specification shall be preserved. For example, the first field of the header of PTP messages is a single octet composed of two fields, one of type Nibble bits 4–7, and one of type Enumeration4 bits 0–3 (see 11.4.2 and 10.6.2).

6.4.4.5 Derived data types

Derived data types defined as structs shall be formatted with the first member of the struct nearest to the beginning of the PDU followed by each succeeding member, without any padding. Each member shall be formatted according to its data type.

Derived data types defined as typedefs shall be formatted according to its referenced data type.

7. Time-synchronization model for a packet network

7.1 General

This clause provides a model for understanding the operation of the generalized precision time protocol (gPTP), which specifies the operation of time-aware systems on a packet network. Although this standard is based on the precision time protocol (PTP) described in IEEE Std 1588-2019 (and, indeed, is a proper profile of IEEE Std 1588 in particular configurations) there are differences, which are summarized in 7.5.

Although this standard has been written as a stand-alone document, it is useful to understand the IEEE 1588 architecture as described in Clause 6 of IEEE Std 1588-2019.

7.2 Architecture of a time-aware network

7.2.1 General

A time-aware network consists of a number of interconnected time-aware systems that support the gPTP defined within this standard. These time-aware systems can be any networking device, including, for example, bridges, routers, and end stations. A set of time-aware systems that are interconnected by gPTP-capable network elements is called a *gPTP network.* Each instance of gPTP that the time-aware systems support is in one *gPTP domain*, and the instances of gPTP are said to be part of that gPTP domain. A time-aware system can support, and therefore be part of, more than one gPTP domain. The entity of a single time-aware system that executes gPTP in one gPTP domain is called a PTP Instance. A time-aware system can contain multiple PTP Instances, which are each associated with a different gPTP domain. There are two types of PTP Instances, as follows:

a) PTP End Instance, which if not grandmaster, is a recipient of time information, and
b) PTP Relay Instance, which if not grandmaster, receives time information from the grandmaster (perhaps indirectly through other PTP Relay Instances), applies corrections to compensate for delays in the LAN and the PTP Relay Instance itself, and retransmits the corrected information.

This standard defines mechanisms for delay measurements using standard-based procedures for the following:

c) IEEE 802.3 Ethernet using full-duplex point-to-point links (Clause 11)
d) IEEE 802.3 Ethernet using passive optical network (EPON) links (Clause 13)
e) IEEE 802.11 wireless (Clause 12)
f) Generic coordinated shared networks (CSNs, e.g., MoCA and G.hn) (Clause 16)

7.2.2 Time-aware network consisting of a single gPTP domain

Figure 7-1 illustrates an example time-aware network consisting of a single gPTP domain, using all theabove network technologies (i.e., (c) - (f) of 7.2.1), where end stations on several local networks are connected to a grandmaster on a backbone network via an EPON access network.

Any PTP Instance with clock sourcing capabilities can be a potential grandmaster, so there is a selection method (the *best master clock algorithm*, or BMCA) that ensures that all of the PTP Instances in a gPTP domain use the same grandmaster.[12] The BMCA is largely identical to that used in IEEE Std 1588-2019, but somewhat simplified. In Figure 7-1 the BMCA process has resulted in the grandmaster being on the

[12] There are, however, short periods during network reconfiguration when more than one grandmaster might be active while the BMCA process is taking place.

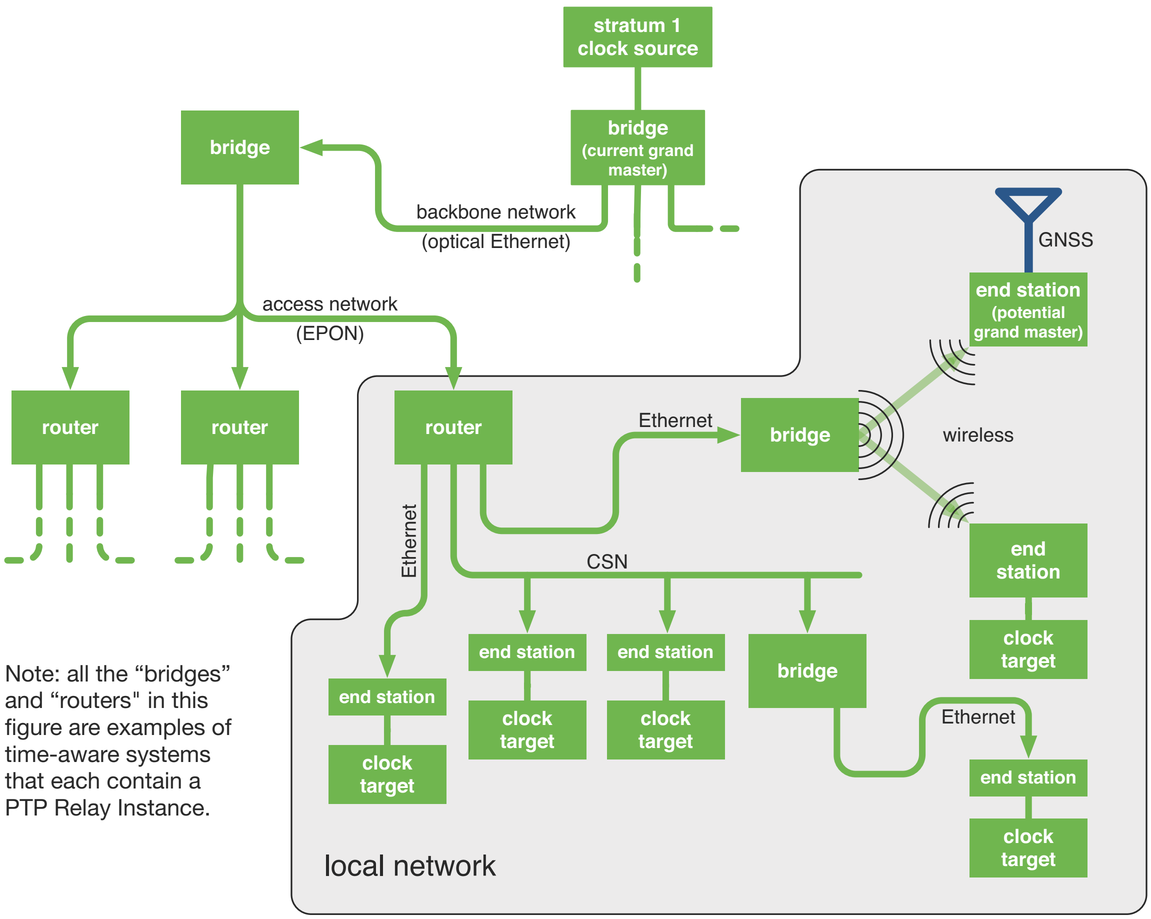


Figure 7-1—Time-aware network example

network backbone. If, however, the access network fails, the systems on a local network automatically switch over to one of the potential grandmasters on the local network that is as least as “good” as any other. For example, in Figure 7-2, the access network link has failed, so a potential grandmaster that has a GNSS reference source has become the active grandmaster, and there are now two gPTP domains where there used to be one. Finally, note that in the case where a time-aware system supports more than one domain, one of the domains supported must be domain 0 for backward compatibility with the 2011 edition of this standard, though domain 0 is not necessarily active in a time-aware system.

7.2.3 Time-aware network consisting of multiple gPTP domains

Figure 7-3 illustrates an example time-aware network consisting of multiple gPTP domains that could be used in an industrial application. Specifically, in this example the network has two timescales/domains, where domain 0 uses the PTP timescale and domain 1 uses the ARB timescale (see 8.2). Notice that not all PTP Instances in domain 1 (within the blue shorter-dashed area) have domain 0 active in this example, even though every time-aware system supports domain 0 for backward compatibility with the 2011 edition of this standard. In addition, it is required that all PTP Instances belonging to the same domain have direct connections among them in their physical topology (e.g., time cannot be transported from one PTP Instance in domain 0 to another PTP Instance in domain 0 via a PTP Instance in domain 1 if the PTP Instance in domain 1 does not also have domain 0 active) . In addition, those time-aware systems for which both

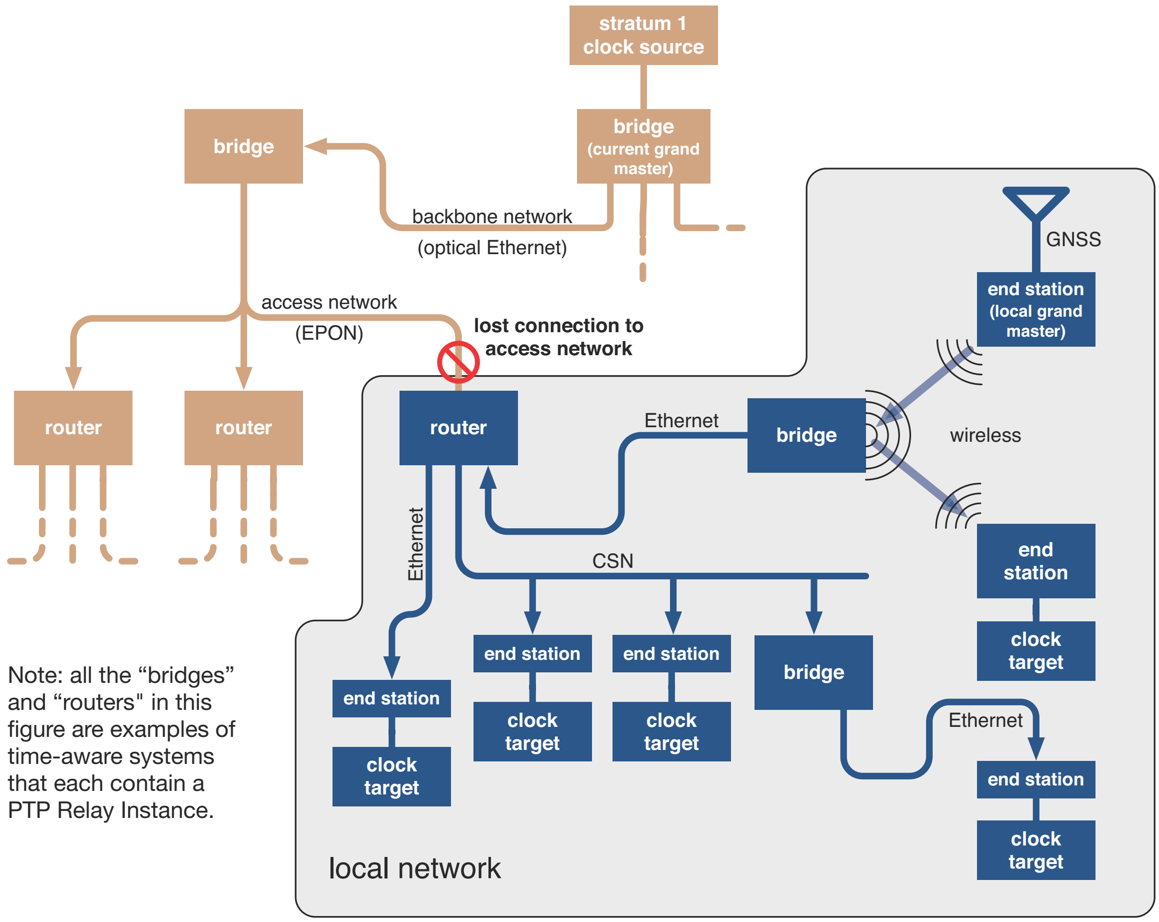


Figure 7-2—Time-aware network of Figure 7-1 after an access network link failure

domains are active are depicted by slanted internal hatching, representing two independent, active PTP Instances.

As in the single-domain case, any of the network technologies of 7.2.1 can be used. The grandmaster of each domain is selected by the BMCA; now, a separate, independent instance of the BMCA is invoked in each domain.

7.2.4 Time-aware networks with redundant grandmasters and/or redundant paths

Redundancy has many levels of sophistication, performance, and cost. Therefore, the appropriate level and/or amount of redundancy required in a time-aware network can be very different for each application. Nonetheless, all solutions for redundancy consist of a detection component, a correction component, and an action component. The detection component detects that something is not working correctly. The correction component determines the appropriate corrective action. The action component performs the required action(s) to fix the detected problem.

This standard provides a basic level of redundancy as follows:

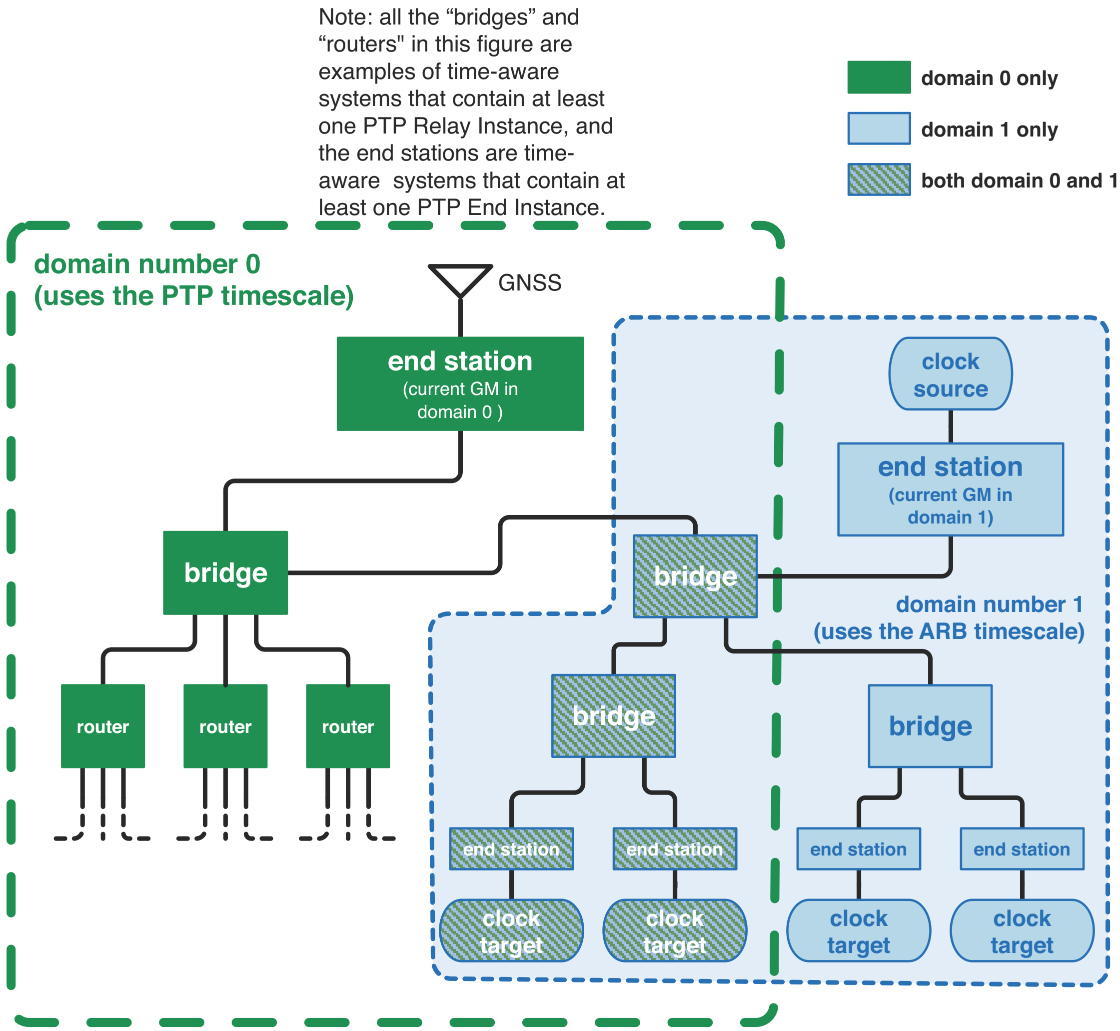


Figure 7-3—Time-aware network example for the case of multiple gPTP domains

- A detection component that triggers when the current grandmaster stops working (i.e., loss of Sync messages and Announce messages for a period of time) or if the link to the grandmaster goes down (i.e., immediate loss of Sync messages and Announce messages).
- A correction component that triggers the Best Master Clock Algorithm and the sending of Announce messages, so that a new grandmaster can be elected.
- An action component, where the winning grandmaster starts sending Announce and Sync messages and all the PTP Instances listen to this new grandmaster.

In addition to providing the basic level of redundancy, this standard provides the ability to support more sophisticated network configurations that provide additional levels of grandmaster and clock path redundancy. Figures 7-4 through 7-6 are examples of such networks that provide these additional levels of redundancy as a way to deal with these failures. The information necessary to implement and configure (i.e., the action component of) these network configurations is contained in this standard.

In order to take advantage of these failure correction configurations, new types of fault detection are required. The category of fault detection where a grandmaster completely fails and stops sending clock information is supported as mentioned above.

Other types of faults involve instability of the grandmaster clock, such as time glitches, excess jitter or wander, or various other impairments that could occur in the grandmaster clock. Techniques for identifying these types of failures, and the appropriate correction necessary, are outside the scope of this standard. However, if other techniques or standards are used for detection and correction of these types of failures, this standard provides the means to recover from these errors.

Figure 7-4 shows an example network realizing two redundant synchronization trees from a single GM, with each synchronization tree in a different gPTP domain (there are a total of two gPTP domains).

Figure 7-5 shows an example network with two redundant GMs, one as primary GM and the other as secondary GM, where each GM has one of the two redundant synchronization trees originating from it. This example supports hot-standby operating mode. In this mode, the secondary GM has to be synchronized to the primary GM, because it is part of the synchronization tree of the primary GM as shown in the figure.

Figure 7-6 shows another case of a ring topology, using the redundancy features of both Figure 7-4 and Figure 7-5.

7.3 Time synchronization

7.3.1 General

Time synchronization in gPTP is done the same way (in the abstract) as is done in IEEE Std 1588-2019: a grandmaster sends information including the current synchronized time to all directly attached PTP Instances. Each of these PTP Instances must correct the received synchronized time by adding the propagation time needed for the information to transit the gPTP communication path from the grandmaster. If the PTP Instance is a PTP Relay Instance, then it must forward the corrected time information (including additional corrections for delays in the forwarding process) to all the other attached PTP Instances.

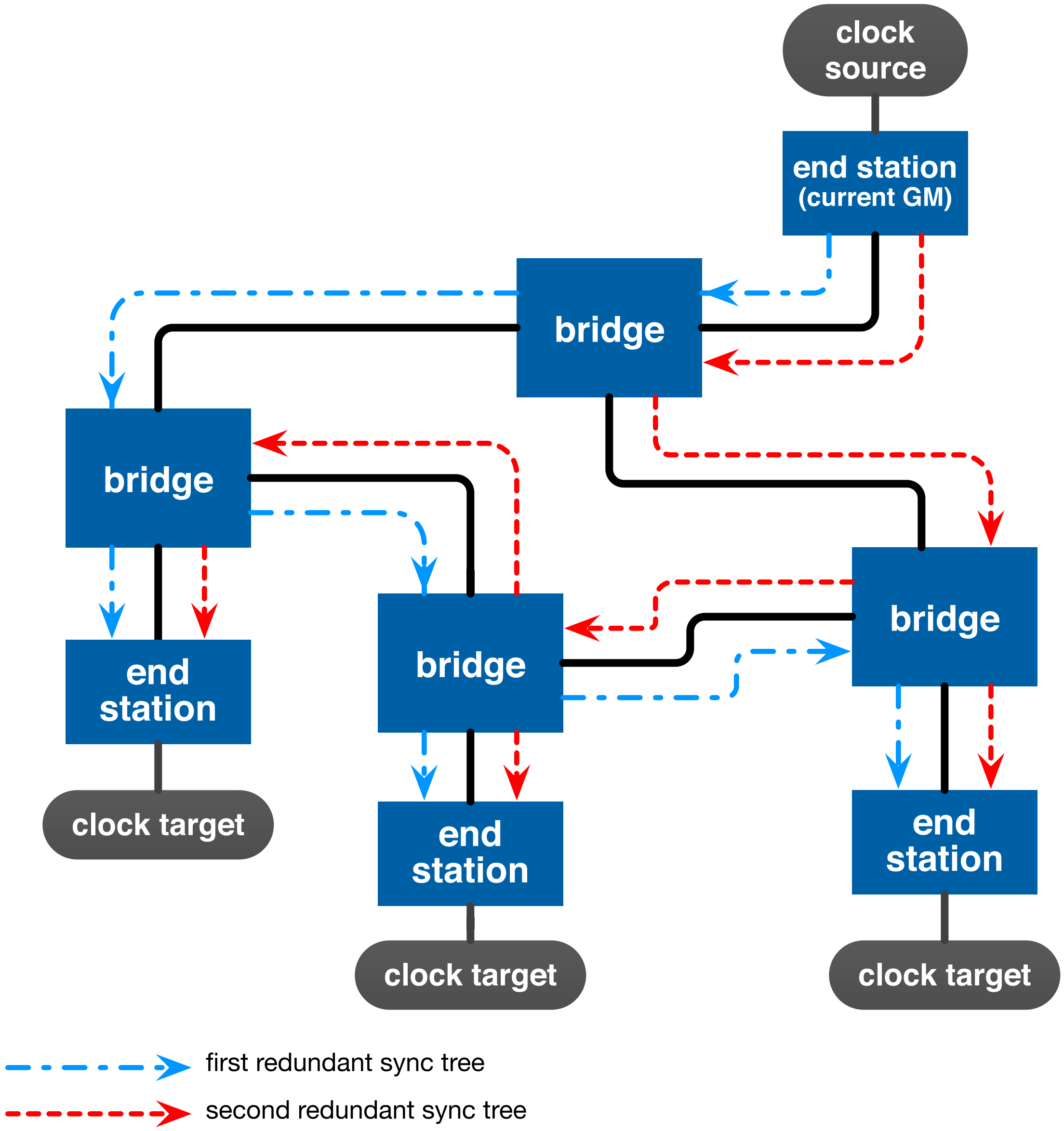
To make this all work, there are two time intervals that must be precisely known: the forwarding delay (called the *residence time*), and the time taken for the synchronized time information to transit the gPTP communication path between two PTP Instances. The residence time measurement is local to a PTP Relay Instance and easy to compute, while the gPTP communication path delay is dependent on many things including media- dependent properties and the length of the path.

7.3.2 Delay measurement

Each type of LAN or gPTP communication path has different methods for measuring propagation time, but they are all based on the same principal: measuring the time that a well-known part of a message is transmitted from one device and the time that the same part of the same message is received by the other device, then sending another message in the opposite direction and doing the same measurement as shown in Figure 7-7.

This basic mechanism is used in the various LANs in the following ways:

a) Full-duplex Ethernet LANs use the two-step peer-to-peer (P2P) path delay algorithm as defined in IEEE Std 1588-2019, where the messages are called Pdelay_Req, Pdelay_Resp, and Pdelay_Resp_Follow_Up (see Figure 11-1).
b) IEEE 802.11 wireless LANs use the Timing measurement procedure or the Fine timing measurement procedure defined in IEEE Std 802.11. The Timing measurement messages are the



Note 1: The methods used for merging the redundant sync msgs received at each end station are not specified in this standard.

Note 2: All the “bridges” in this figure are examples of time-aware systems that contain PTP Relay Instances, and the end stations are examples of time-aware systems that contain PTP End Instances.

Figure 7-4—Time-aware network example for synchronization path redundancy, with one clock source providing time to two domains

“timing measurement frame” and its corresponding “Ack” (see Figure 12-1). The Fine timing measurement messages are the “initial FTM request frame” and the “fine timing measurement frame” and its “Ack” (see Figure 12-2).

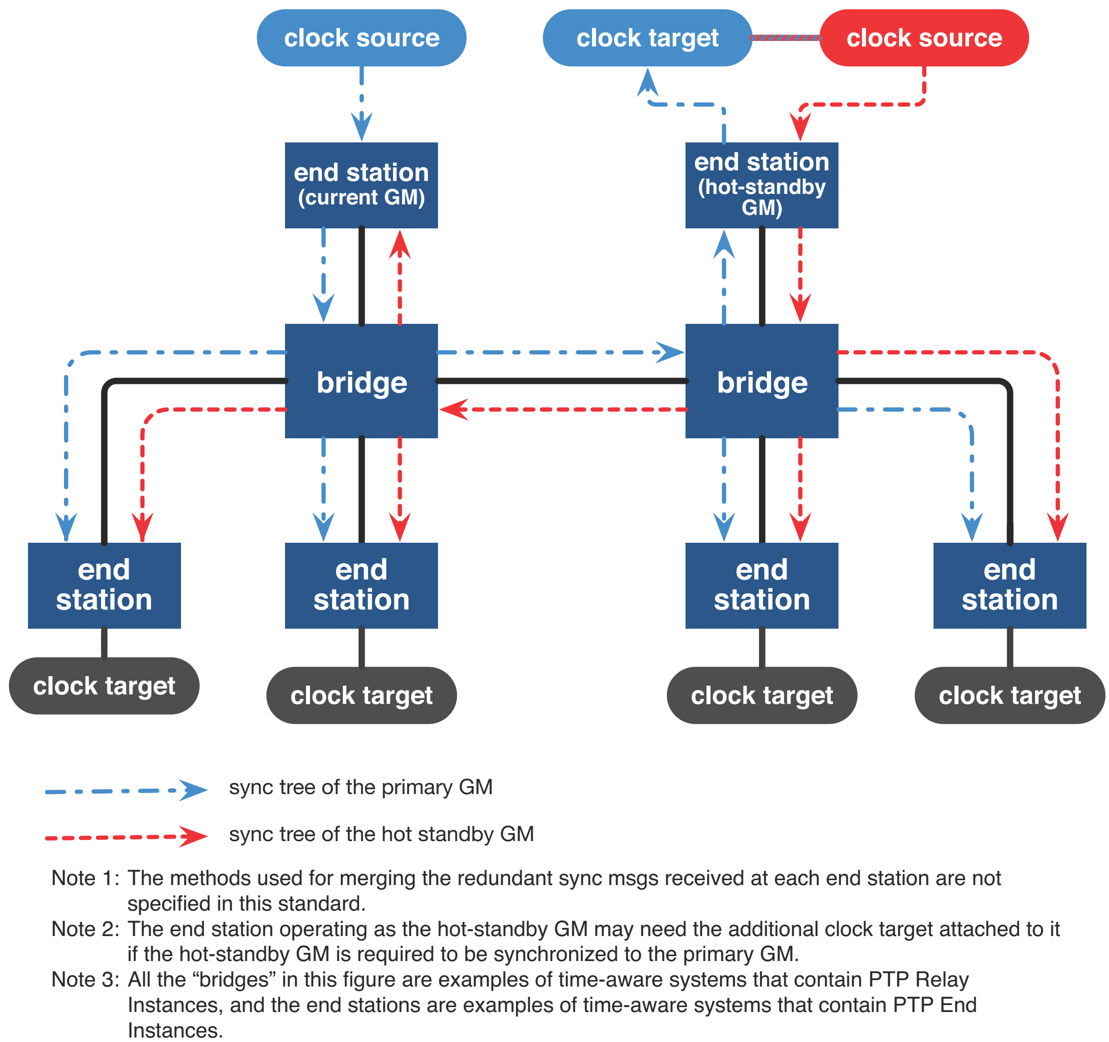


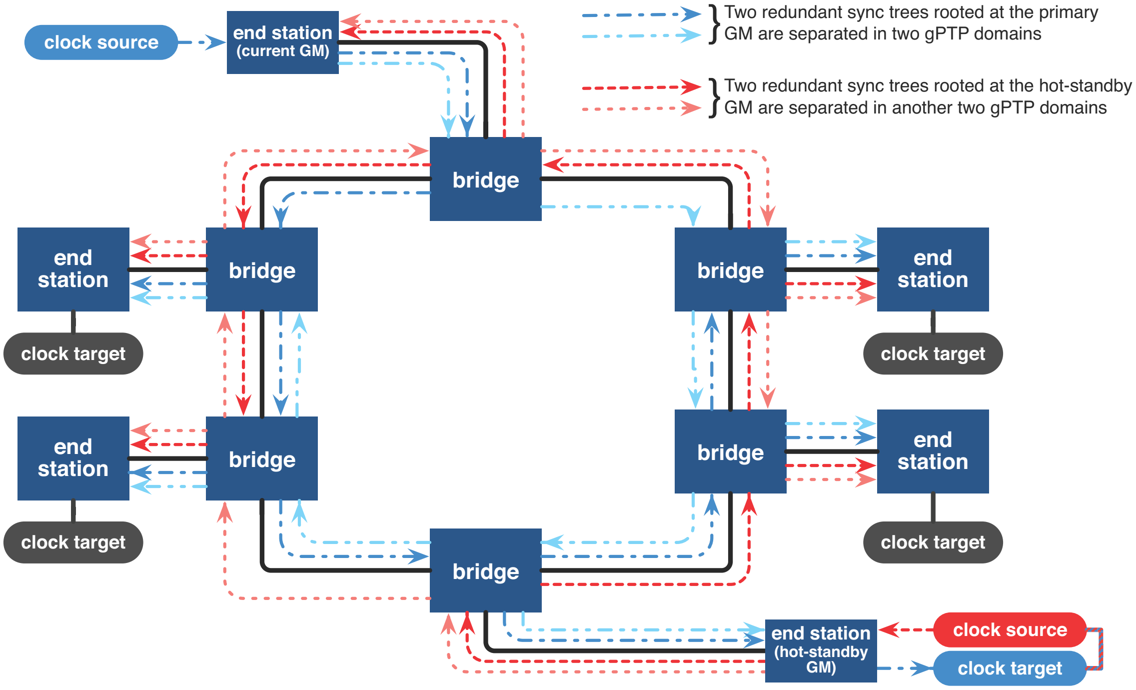
Figure 7-5—Time-aware network example for GM redundancy with one primary GM and one hot-standby GM, which are separated in two gPTP domains

c) EPON LANs use the discovery process, where the messages are "GATE" and "REGISTER_REQ" (see Clauses 64 and 77 of IEEE Std 802.3-2018).

d) CSNs either use the same mechanism as full-duplex Ethernet, or use a method native to the particular CSN (similar to the way native methods are used by IEEE 802.11 and EPON) (see Figure 16-5).

7.3.3 Logical syntonization

The time-synchronization correction previously described is dependent on the accuracy of the delay and residence time measurements. If the clock used for this purpose is frequency locked (syntonized) to the grandmaster, then all the time interval measurements use the same time base. Since actually adjusting the frequency of an oscillator (e.g., using a PLL) is slow and prone to gain peaking effects, PTP Relay Instances can correct time interval measurements using the grandmaster frequency ratio.



Note 1: The methods used for merging the redundant sync msgs received at each end station are not specified in this standard.
Note 2: The endstation operating as the hot-standby GM may need the additional clock target attached to it if the hot-standby GM is required to be synchronized to the primary GM.
Note 3: All the “bridges” in this figure are examples of time-aware systems that contain PTP Relay Instances, and the end stations are examples of time-aware systems that contain PTP End Instances.

Figure 7-6—Time-aware network example for GM+synchronziation Path redundancy, with one primary and one hot-standby GM. Each GM establishes two sync trees, resulting in a total of four Sync trees that are separated in four gPTP domains

Each PTP Instance measures, at each port, the ratio of the frequency of the PTP Instance at the other end of the link attached to that port to the frequency of its own clock. The cumulative ratio of the grandmaster frequency to the local clock frequency is accumulated in a standard organizational type, length, value (TLV) attached to the Follow_Up message (or the Sync message if the optional one-step processing is enabled). The frequency ratio of the grandmaster relative to the local clock is used in computing synchronized time, and the frequency ratio of the neighbor relative to the local clock is used in correcting the propagation time measurement.

The grandmaster frequency ratio is measured by accumulating neighbor frequency ratios for two main reasons. First, if there is a network reconfiguration and a new grandmaster is elected, the nearest neighbor frequency ratios do not have to be newly measured as they are constantly measured using the Pdelay messages. This results in the frequency offset relative to the new grandmaster being known when the first Follow_Up message (or first Sync message if the optional one-step processing is enabled) is received, which reduces the duration of any transient error in synchronized time during the reconfiguration. This is beneficial to many high-end audio applications. Second, there are no gain peaking effects because an error in frequency offset at one PTP Relay Instance, and resulting residence time error, does not directly affect the frequency offset at a downstream PTP Relay Instance.

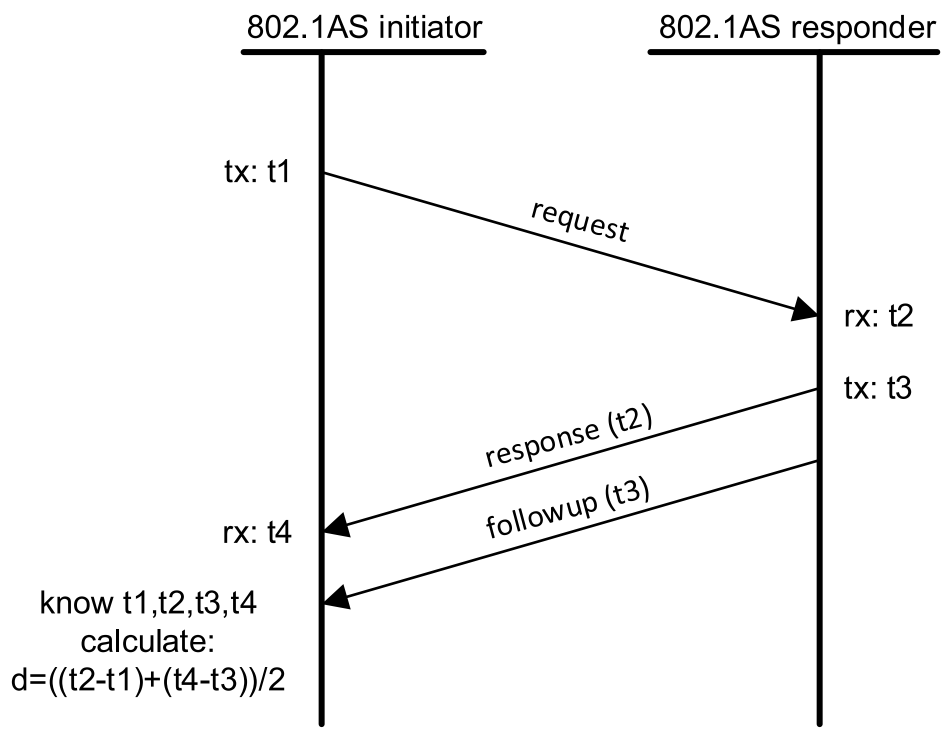


Figure 7-7— Conceptual medium delay measurement

7.3.4 Grandmaster (best master) selection and network establishment

All PTP Instances participate in best master selection so that the IEEE 802.1AS protocol can determine the synchronization spanning tree. This synchronization spanning tree can be different from the forwarding spanning tree determined by IEEE 802.1Q Rapid Spanning Tree Protocol (RSTP) since the spanning tree determined by RSTP can be suboptimal or even inadequate for synchronization, or for a different topology of nodes than the synchronization spanning tree.

gPTP requires that all systems in the gPTP domain be time-aware systems, i.e., the protocol does not transfer timing over systems that are not time-aware (e.g., those that meet the requirements of IEEE Std 802.1Q, but do NOT meet the requirements of this standard). A time-aware system uses the peer-to-peer delay mechanism on each port to determine if a non-time-aware system is at the other end of the link or between itself and the Pdelay responder. If, on sending Pdelay_Req

a) no response is received,
b) multiple responses are received, or
c) the measured propagation delay exceeds a specified threshold, then

the protocol concludes that a non-time-aware system or end-to-end TC (see IEEE Std 1588-2019) is present. In this case, the link attached to the port is deemed not capable of running gPTP, and BMCA ignores it. However, the port continues to attempt the measurement of propagation delay using the peer-to-peer delay mechanism (for full-duplex IEEE 802.3 links), MPCP messages (for EPON), or IEEE802.11 messages (for IEEE 802.11 links), and periodically checks whether the link is or is not capable of running IEEE 802.1AS.

7.3.5 Energy efficiency

Sending PTP messages at relatively high rates when there is otherwise little or no traffic conflicts with the goal of reducing energy consumption. This standard specifies a way to request that a neighbor port reduce the rate of sending Sync (and Follow_Up if optional one-step processing is not enabled), peer delay, and Announce messages, and also to inform the neighbor not to compute neighbor rate ratio and/or propagation delay on this link. A time-aware system could do this when it enters low-power mode, but this standard does not specify the conditions under which this is done; it specifies only the actions a time-aware system takes.

7.4 PTP Instance architecture

The model of a PTP Instance is shown in Figure 7-8.

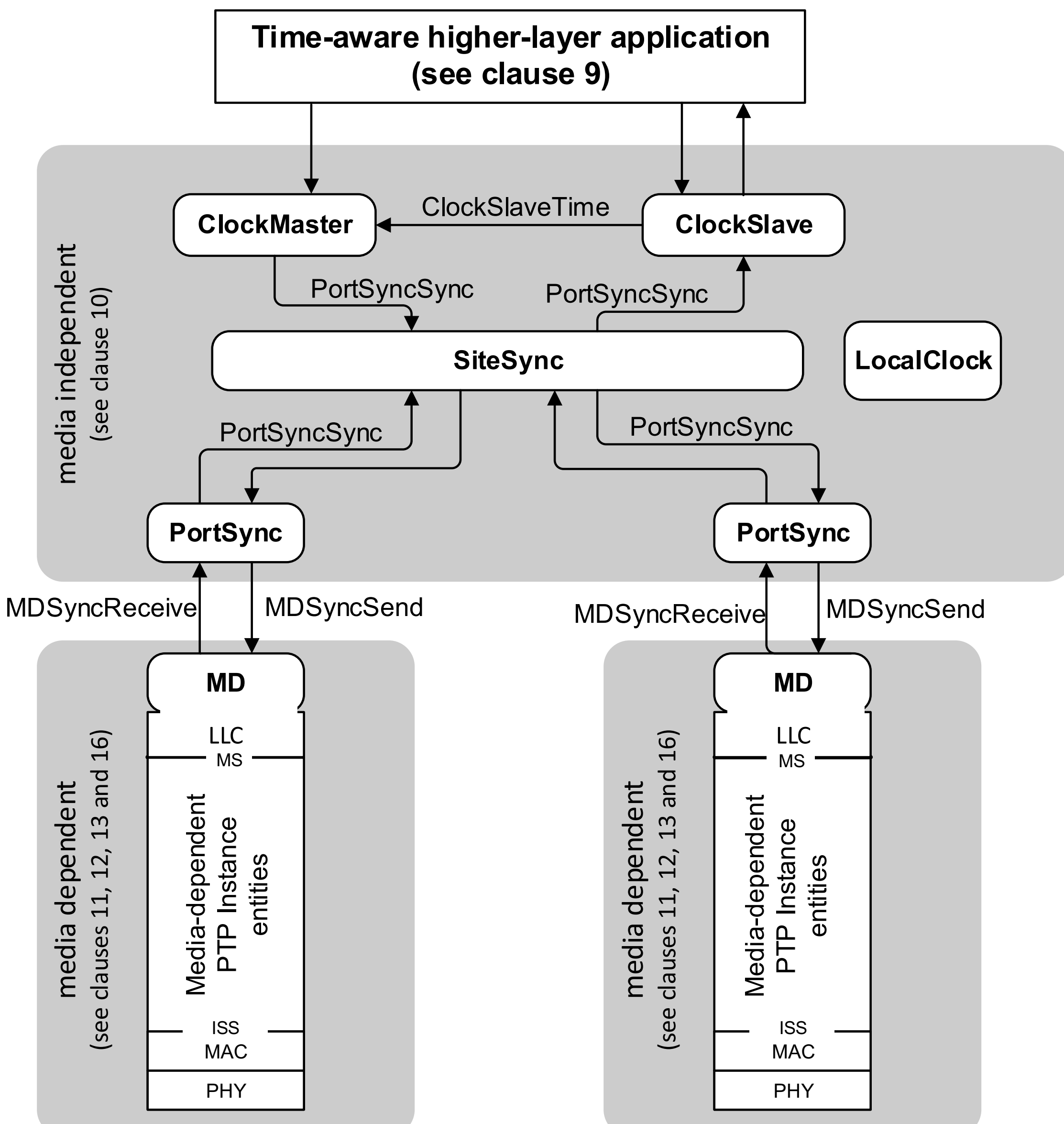


Figure 7-8—PTP Instance model

A PTP Instance consists of the following major parts:

a) If the PTP Instance includes application(s) that either use or source time information, then they interface with the gPTP information using the application interfaces specified in Clause 9.
b) A single media-independent part that consists of ClockMaster, ClockSlave, and SiteSync logical entities, one or more PortSync entities, and a LocalClock entity. The BMCA and forwarding of time information between logical ports and the ClockSlave and ClockMaster is done by the SiteSync entity, while the computation of port-specific delays needed for time-synchronization correction is done by the PortSync entities.
c) Media-dependent ports, which translate the abstract "MDSyncSend" and "MDSyncReceive" structures received from or sent to the media-independent layer and corresponding methods used for the particular LAN attached to the port.

In the case of full-duplex Ethernet ports, IEEE 1588 Sync and Follow_Up (or just Sync if the optional one-step processing is enabled) messages are used, with an additional TLV in the Follow_Up (or the Sync if the optional one-step processing is enabled) used for communication of rate ratio and information on phase and frequency change when there is a change in grandmaster. The path delay is measured using the two-step IEEE 1588 peer-to-peer delay mechanism. This is defined in Clause 11.

For IEEE 802.11 ports, timing information is communicated using the MAC Layer Management Entity to request a "timing measurement" or "fine timing measurement" [as defined in IEEE Std 802.11], which also sends everything that would be included in the Follow_up message for full-duplex Ethernet. The timing measurement or fine timing measurement result includes all the information to determine the path delay. This is defined in Clause 12.

For EPON, timing information is communicated using a "slow protocol" as defined in Clause 13. CSNs use the same communication system used by Ethernet full-duplex, as is defined in Clause 16.

7.5 Differences between gPTP (IEEE Std 802.1AS) and PTP (IEEE Std 1588)

a) gPTP assumes all communication between PTP Instances is done only using IEEE 802 MAC PDUs and addressing, while IEEE Std 1588 supports various layer 2 and layer 3-4 communication methods.
b) gPTP specifies a media-independent sublayer that simplifies the integration within a single timing domain of multiple different networking technologies with radically different media access protocols. gPTP specifies a media-dependent sublayer for each medium. The information exchanged between PTP Instances has been generalized to support different packet formats and management schemes appropriate to the particular networking technology. IEEE Std 1588-2019, on the other hand, has introduced a new architecture based on media-independent and media-dependent sublayers; however, this architecture is optional. The architecture of IEEE Std 1588-2008, which is not based on media-independent and media-dependent layers, has been retained for IP version 4, IP version 6, Ethernet LANs, and several industrial automation control protocols. The intent in IEEE Std 1588-2019 is that the new architecture, based on media-independent and media-dependent layers, will be used for IEEE 802.11, IEEE 802.3 EPON, CSN using the specifications of gPTP, and that the architecture must be used for transports that define native timing mechanisms if those native timing mechanisms are used. The new architecture in IEEE Std 1588-2019 can optionally be used for the transports described in IEEE Std 1588-2008.
c) In gPTP there are only two types of PTP Instances: PTP End Instances and PTP Relay Instances, while IEEE 1588 has ordinary clocks, boundary clocks, end-to-end transparent clocks, and P2P transparent clocks. A PTP End Instance corresponds to an IEEE 1588 ordinary clock, and a PTP Relay Instance is a type of IEEE 1588 boundary clock where its operation is very tightly defined, so much so that a PTP Relay Instance with Ethernet ports can be shown to be mathematically equivalent to a P2P transparent clock in terms of how synchronization is performed, as shown in

11.1.3. In addition, a PTP Relay Instance can operate in a mode (i.e., the mode where the variable syncLocked is TRUE, see 10.2.5.15) where the PTP Relay Instance is equivalent to a P2P transparent clock in terms of when time-synchronization messages are sent. A time-aware system measures link delay and residence time, and communicates these in a correction field. In summary, a PTP Relay Instance conforms to the specifications for a boundary clock in IEEE Std 1588, but a PTP Relay Instance does not conform to the complete specifications for a P2P transparent clock in IEEE Std 1588 because (i) when syncLocked is FALSE, the PTP Relay Instance sends Sync according to the specifications for a boundary clock, and (ii) the PTP Relay Instance invokes BMCA and has port states.

d) PTP Instances only communicate gPTP information directly with other PTP Instances. That is, a gPTP domain consists ONLY of PTP Instances. Non-PTP Relay Instances cannot be used to relay gPTP information. In IEEE 1588 it is possible to use non-IEEE-1588-aware relays in an IEEE 1588 domain, although this slows timing convergence and introduce extra jitter and wander that must be filtered by any IEEE 1588 clock.
e) For Ethernet full-duplex links, gPTP requires the use of the peer-to-peer delay mechanism, while IEEE 1588 also allows the use of end-to-end delay measurement.
f) For Ethernet full-duplex links, gPTP requires the use of two-step processing (use of Follow_Up and Pdelay_Resp_Follow_Up messages to communicate timestamps), with an optional one-step processing mode that embeds timestamps in the Sync "on the fly" as they are being transmitted (gPTP does not specify one-step processing for peer delay messages). IEEE 1588 allows either two-step or one-step processing to be required (for both Sync and peer delay messages) depending on a specific profile.
g) All PTP Instances in a gPTP domain are logically syntonized, meaning that they all measure time intervals using the same frequency. This is done by the process described in 7.3.3, and is mandatory. Syntonization in IEEE 1588 is optional. The syntonization method used by gPTP is supported as an option in IEEE 1588-2019, but uses a TLV standardized as part of IEEE Std 1588-2019 (this is a new feature for IEEE 1588-2019), while gPTP uses the ORGANIZATION_EXTENSION TLV specified in 11.4.4.3.
h) The BMCA of this standard is the default BMCA according to the specifications of IEEE Std 1588-2019.

NOTE—The BMCA used in gPTP was not the same as the default BMCA of IEEE 1588-2008.

i) Finally, this standard includes formal interface definitions, including primitives, for the time-aware applications (see Clause 9). IEEE Std 1588-2019 describes external interfaces less formally, i.e., without describing specific interface primitives.

8. IEEE 802.1AS concepts and terminology

8.1 gPTP domain

A gPTP domain, hereafter referred to as simply a *domain,* consists of one or more PTP Instances and links that meet the requirements of this standard and communicate with each other as defined by the IEEE 802.1AS protocol. A gPTP domain defines the scope of gPTP message communication, state, operations, data sets, and timescale.

A domain is identified by two attributes: domain number and sdoId. The sdoId of a domain is a 12-bit unsigned integer. The sdoId is structured as a two-part attribute as follows:

— the most significant 4 bits are named the majorSdoId, and
— the least significant 8 bits are named the minorSdoId.

The domain number of a gPTP domain shall be in the range 0-127. A time-aware system shall support a domain whose domain number is 0. A time-aware system may support one or more additional domains each with a distinct domain number in the range 1-127. Unless otherwise specified in this standard, the operation of the gPTP protocol and the timescale in any given domain is independent of operation in any other domain.

The value of majorSdoId for a gPTP domain shall be 0x1. The value of minorSdoId for a gPTP domain shall be 0x00.

NOTE 1—The above requirements for majorSdoId and minorSdoId are for gPTP domains. The requirements for the Common Mean Link Delay Service are given in 11.2.17.

Both the domainNumber and the sdoId are carried in the common header of all PTP messages (see 10.6.2.2).

NOTE 2—In the 2011 edition of this standard, the attribute majorSdoId was named transportSpecific and its value was specified as 0x1 in 10.5.2.2.1 of Corrigendum 1. The attribute minorSdoId did not exist in the 2011 edition, but its location in the common header was a reserved field, which was specified to be transmitted as 0 and ignored on receipt.

Unless otherwise stated, information in the remainder of this document is per domain.

NOTE 3—In steady state, all PTP Instances in a gPTP domain are traceable to a single grandmaster.

8.2 Timescale

8.2.1 Introduction

The timescale for a gPTP domain is established by the grandmaster. There are two types of timescales supported by gPTP:

— The timescale PTP: The epoch is the PTP epoch (see 8.2.2), and the timescale is continuous. The unit of measure of time is the second defined by International Atomic Time (TAI) (see [B15], with the further amplification of [B24], for the definition of TAI, and [B12], [B16], and [B25] for more information on TAI). The timescale of domain 0 shall be PTP. See IEEE Std 1588 for more details.

NOTE—The 2011 of this standard specified only domain number 0 and only the timescale PTP. In addition, the 2011 edition of this standard specified that the ptpTimescale flag bit of the PTP message common header (see 10.6.2.2.8) be set to TRUE on transmit and ignored on receipt. If the single PTP Instance of a time-aware system compliant with the 2011 edition of this standard and a PTP Instance compliant with the 2019 edition of this standard that supports the ARB timescale on domain 0 are in the same network, the former will transmit TRUE for the ptpTimescale flag bit, while the latter will transmit FALSE. However, this will not impact the operation of the protocol because this flag is only for information.

— The timescale ARB (arbitrary): The epoch is the domain startup time, and can be set by an administrative procedure. Between invocations of the administrative procedure, the timescale is continuous. Additional invocations of the administrative procedure can introduce discontinuities in the overall timescale. The unit of measure of time is determined by the grandmaster. The second used in the operation of the protocol may differ from the SI second.

8.2.2 Epoch

The epoch is the origin of the timescale of a gPTP domain.

The PTP epoch is 1 January 1970 00:00:00 TAI, which is 31 December 1969 23:59:51.999918 UTC.

NOTE—The common Portable Operating System Interface (POSIX) algorithms can be used for converting elapsed seconds since the PTP epoch to the ISO 8601:2004 printed representation of time of day on the TAI timescale (see ISO/IEC 9945:2003 [B8] and ISO 8601:2004 [B6]).

See Annex C for information on converting between common timescales.

8.2.3 UTC Offset

When the timescale is PTP, it is possible to calculate UTC time using the value of currentUtcOffset. The value of currentUtcOffset is given by

currentUtcOffset = TAI – UTC,

where the difference TAI – UTC is the IERS defined offset between TAI and UTC (see IERS Bulletin C).

The value of currentUtcOffset for the current grandmaster is maintained in the currentUtcOffset member of the time properties data set (see 14.5.1).

NOTE—As of 0 hours 1 January 2017 UTC, UTC was behind TAI by 37 s, i.e., TAI - UTC = +37 s. At that moment, the IEEE 802.1AS defined value of currentUtcOffset was +37 s, as designated in the applicable IERS Bulletin C (see Clause 2, and see also Service de la Rotation Terrestre [B15] and U.S. Naval Observatory [B17]).

When the timescale is PTP, it is possible to calculate local time from the time provided by a gPTP domain using the currentUtcOffset field value of the time properties data set and knowledge of the local time zone and whether and when daylight savings time is observed.

When the timescale is ARB, the value of currentUtcOffset cannot be used to compute UTC.

The mechanism for computing UTC does not change the synchronized time (see 3.24) of a PTP Instance.

8.2.4 Measurement of time within a gPTP domain

Time in a gPTP domain shall be measured as elapsed time since the epoch of the timescale of that domain.

8.3 Link asymmetry

This standard requires the measurement of the mean propagation time (also referred to as the *mean propagation delay*) between the time-aware systems that comprise the two endpoints of a link. The measurement is performed by one of the time-aware systems (the initiator time-aware system), sending a message to the other time-aware system (the responder time-aware system). The responder then sends a message back to the initiator at a later time. The departure of the message sent by the initiator time-aware system is timestamped, and the timestamp value is retained by that system. The arrival of this message at the responder time-aware system is timestamped; the timestamp value is conveyed to the initiator time-aware system in a subsequent message. The departure of the response message sent by the responder time-aware system (in response to the message it receives from the initiator time-aware system) is timestamped, and the timestamp value is conveyed to the initiator time-aware system in a subsequent message. The arrival of this response message at the initiator time-aware system is timestamped, and the timestamp value is retained by that system. The mean propagation time is computed by the initiator time-aware system after receiving the response message, from the four timestamp values it has at this point.

Typically, the propagation time is not exactly the same in both directions, and the degree to which it differs in the two directions is characterized by the delay asymmetry. The relation between the individual propagation times in the two directions, the mean propagation time, and the delay asymmetry is as follows. Let t_{ir} be the propagation time from the initiator to the responder, t_{ri} be the propagation time from the responder to the initiator, meanLinkDelay be the mean propagation time (see 10.2.5.8), and delayAsymmetry be the delay asymmetry. The propagation times in the two directions are illustrated in Figure 8-1.

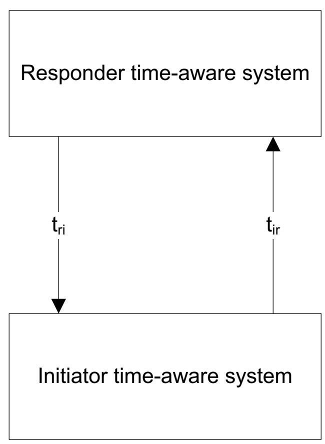


Figure 8-1—Propagation asymmetry

The meanLinkDelay is the mean value of t_{ir} and t_{ri}, i.e., meanLinkDelay = (t_{ir} + t_{ri}) / 2. The delayAsymmetry is defined as

t_{ir} = meanLinkDelay – delayAsymmetry, and

t_{ri} = meanLinkDelay + delayAsymmetry.

In other words, delayAsymmetry is defined to be positive when the responder to initiator propagation time is longer than the initiator to responder propagation time.

This standard does not explicitly require the measurement of delayAsymmetry; however, if delayAsymmetry is modeled, it shall be modeled as specified in this clause.

NOTE 1—A time-aware system can change the value of delayAsymmetry during operation (see 14.8.10, 14.16.8, and Annex G).

NOTE 2—A time-aware system port cannot measure the value of delayAsymmetry during live operation of the system (i.e., asymmetryMeasurementMode, see 10.2.5.2, is FALSE), so the value of delayAsymmetry must be defined separately using information from the supplier or additional testing before running the live system. The methods for measuring asymmetry are not specified in this standard. These values can be added to the system configuration to improve the accuracy of time synchronization. The inaccuracy caused by asymmetry is half the value of the difference between t_{ri} and t_{ir}, and these inaccuracies can either accumulate over successive hops or, if the successive asymmetries have different signs, can cancel each other over the successive hops.

8.4 Messages

8.4.1 General

All communications occur via PTP messages and/or media-specific messages.

8.4.2 Message attributes

8.4.2.1 General

All messages used in this standard have the following attributes:

a) Message class
b) Message type

The message class attribute is defined in this clause. The message type attribute is defined in 3.13**.** Some messages have additional attributes; these are defined in the subclauses where the respective messages are defined.

8.4.2.2 Message class

There are two message classes, the event message class and the general message class. Event messages are timestamped on egress from a PTP Instance and ingress to a PTP Instance. General messages are not timestamped. Every message is either an event message or a general message.

8.4.3 Generation of event message timestamps

All event messages are timestamped on egress and ingress. The timestamp shall be the time, relative to the LocalClock entity (see 10.1) at which the message timestamp point passes the reference plane marking the boundary between the PTP Instance and the network media.

The definition of the timestamp measurement plane (see 3.28), along with the corrections defined as follows, allows transmission delays to be measured in such a way (at such a low layer) that they appear fixed and symmetrical to gPTP even though the MAC client might otherwise observe substantial asymmetry and transmission variation. For example, the timestamp measurement plane is located below any retransmission and queuing performed by the MAC.

NOTE 1—If an implementation generates event message timestamps using a point other than the message timestamp point, then the generated timestamps should be appropriately corrected by the time interval (fixed or otherwise) between the actual time of detection and the time the message timestamp point passed the reference plane. Failure to make these corrections results in a time offset between PTP Instances.

NOTE 2—In general, the timestamps can be generated at a timestamp measurement plane that is removed from the reference plane. Furthermore, the timestamp measurement plane, and therefore the time offset of this plane from the reference plane, is likely to be different for inbound and outbound event messages. To meet the requirement of this clause, the generated timestamps should be corrected for these offsets. Figure 8-2 illustrates these offsets. Based on this model the appropriate corrections are as follows:

egressTimestamp = egressMeasuredTimestamp + egressLatency

ingressTimestamp = ingressMeasuredTimestamp − ingressLatency

where the timestamps relative to the reference plane, egressTimestamp and ingressTimestamp, are computed from the timestamps relative to the timestamp measurement plane, egressMeasuredTimestamp and ingressMeasuredTimestamp, respectively, using their respective latencies, egressLatency and ingressLatency. Failure to make these corrections results in a time offset between the slave and master clocks.

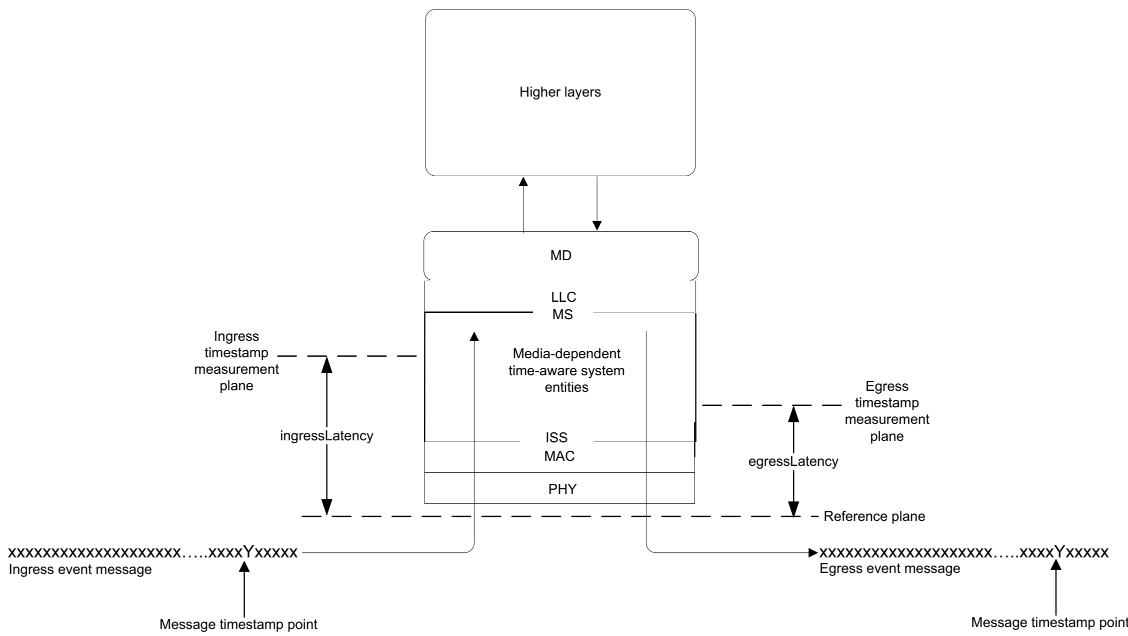


Figure 8-2—Definition of message timestamp point, reference plane, timestamp measurement plane, and latency constants

8.4.4 Priorities

The ISS priority for frames carrying IEEE 802.1AS messages shall be 6.

NOTE—Frames carrying IEEE 802.1AS messages are neither VLAN-tagged nor priority-tagged, i.e., they are untagged (see 11.3.3).

8.5 Ports

8.5.1 General

The PTP Instances in a gPTP domain interface with the network media via physical ports. gPTP defines a logical port in such a way that communication between PTP Instances is point-to-point even over physical ports that are attached to shared media. One logical port, consisting of one PortSync entity and one MD entity, is instantiated for each PTP Instance with which the PTP Instance communicates. In the case of shared media, multiple logical ports can be associated with a single physical port.

Unless otherwise qualified, each instance of the term *port* refers to a *logical port.*

8.5.2 Port identity

8.5.2.1 General

A port is identified by a port identity of type PortIdentity (see 6.4.3.7). The value is maintained in portDS.portIdentity (see 14.8.2). A port identity consists of the following two attributes:

a) portIdentity.clockIdentity
b) portIdentity.portNumber

8.5.2.2 clockIdentity

The clockIdentity attribute shall be as specified in 7.5.2.2, and its subclauses, of IEEE Std 1588-2019.

8.5.2.3 Port number

The portNumber values for the ports on a time-aware system shall be distinct in the range 1, 2, …, 0xFFFE.

The portNumber value 0 is assigned to the interface between the ClockMaster and ClockSource entities (see 10.1 and Figure 10-1). The value 0xFFFF is reserved.

8.5.2.4 Ordering of clockIdentity and portIdentity values

Two clockIdentity values X and Y are compared as follows. Let x be the unsigned integer formed by concatenating octets 0 through 7 of X such that octet j+1 follows octet j (i.e., is less significant than octet j) in x (j = 0, 1, ..., 6). Let y be the unsigned integer formed by concatenating octets 0 through 7 of Y such that octet j+1 follows octet j (i.e., is less significant than octet j) in y (j = 0, 1, ..., 6). Then

a) X = Y if and only if $x = y$,
b) X > Y if and only if $x > y$, and
c) X < Y if and only if $x < y$.

Two portIdentity values A and B with members clockIdentity and portNumber are compared as follows. Let a be the unsigned integer formed by concatenating octets 0 through 7 of A.clockIdentity, such that octet j+1 follows octet j (i.e., is less significant than octet j) in a (j = 0, 1, ..., 6), followed by octet 0 of A.portNumber, followed by octet 1 of A.portNumber. Let b be the unsigned integer formed by concatenating octets 0 through 7 of B.clockIdentity, such that octet j+1 follows octet j (i.e., is less significant than octet j) in b (j = 0, 1, ..., 6), followed by octet 0 of B.portNumber, followed by octet 1 of B.portNumber. Then

d) A = B if and only if $a = b$,
e) A > B if and only if $a > b$, and
f) A < B if and only if $a < b$.

A portIdentity A with members clockIdentity and portNumber and a clockIdentity B are compared as follows. The unsigned integer a is formed from portIdentity A as described above. The unsigned integer b is formed by first forming a portIdentity B' whose clockIdentity is B and portNumber is 0. b is then formed from B' as described above. A and B are then compared as described in items d) through f) above.

8.6 PTP Instance characterization

8.6.1 PTP Instance type

There are two types of PTP Instances used in a gPTP domain, as follows:

a) PTP End Instance
b) PTP Relay Instance.

All PTP Instances are identified by clockIdentity.

In addition, PTP Instances are characterized by the following attributes:

c) priority1
d) clockClass
e) clockAccuracy
f) offsetScaledLogVariance
g) priority2
h) clockIdentity
i) timeSource
j) numberPorts

NOTE—Attributes c) through i) can be considered to be associated with the ClockMaster entity of the PTP Instance.

8.6.2 PTP Instance attributes

8.6.2.1 priority1

priority1 is used in the execution of the BMCA (see 10.3). The value of priority1 is an integer selected from the range 0 through 255. The ordering of priority1 in the operation of the BMCA (see 10.3.4 and 10.3.5) is specified as follows. A ClockMaster A shall be deemed better than a ClockMaster B if the value of priority1 of A is numerically less than that of B.

The value of priority1 shall be 255 for a PTP Instance that is not grandmaster-capable. The value of priority1 shall be less than 255 for a PTP Instance that is grandmaster-capable. The value 0 shall be reserved for management use, i.e., the value of priority1 shall be set to 0 only via management action. The default value shall be set to one of the values listed in Table 8-1, and the choice of value from Table 8-1 is left to the implementer of the PTP Instance.

Table 8-1—Default values for priority1, for the respective media

System type	Default value for priority1
Devices that are a central/critical part of the network and are not expected to ever be turned off during normal network operation. For example, these include Bridges, Wireless Access Points, and grandmaster-specific end nodes. (End node grandmasters are not a 'central' part of the network, but they are 'critical' to this gPTP operation)	246
Devices that can be turned off at any time, so they cannot be considered a central/critical part of the network. When these devices are turned off, they only affect the function(s) they support and do not affect any other functions of the network. For example, these include desktop computers, fixed (heavy or otherwise) end nodes, speakers, receivers, amplifiers, and televisions.	248
Devices that can go away (physically or otherwise) at any time. This includes devices that are designed to be transient to the network. For example, these include laptop computers, cell phones, and battery powered speakers.	250
PTP Instance that is not grandmaster-capable	255

NOTE 1—Care must be applied to multi-function device design, specifically for an end station that also contains a Bridge. The Bridge function inside multi-function devices must not be powered down when the end station function is

powered down, or else the data and controls (like gPTP) passing through the Bridge function to other network devices will cease. Example devices of this are a television and/or amplifier/receiver, each of which contains a Bridge.

NOTE 2—The BMCA (see 10.3) considers priority1 before other attributes; the priority1 attribute can therefore be used to force a desired ordering of PTP Instances for best master selection.

NOTE 3—The settings for priority1 in Table 8-1 guarantee that a PTP Instance that is grandmaster-capable is always preferred by the BMCA over a PTP Instance that is not grandmaster-capable.

NOTE 4—These values are assigned so that devices with priority1 value of 246 are selected as grandmaster over devices with priority1 values of 248 or 250 (devices with priority1 value of 255 are never selected as grandmaster).

NOTE 5—These default values are suitable for applications in which the availability of the grandmaster is the most important criterion for grandmaster selection. A device built for a specific application for which this is not the case can be capable of having priority1 changed via management.

8.6.2.2 clockClass

The clockClass attribute denotes the traceability of the synchronized time distributed by a ClockMaster when it is the grandmaster.

The value shall be selected as follows:

a) If defaultDS.gmCapable (see 14.2.6) is TRUE, then
   1) clockClass is set to the value that reflects the combination of the LocalClock and ClockSource entities; else
   2) if the value that reflects the LocalClock and ClockSource entities is not specified or not known, clockClass is set to 248;
b) If the defaultDS.gmCapable is FALSE, clockClass is set to 255 (see 8.6.2.1).

The ordering of clockClass in the operation of the best master clock algorithm (see 10.3.4 and 10.3.5) is specified as follows. When comparing clockClass values, PTP Instance A shall be deemed better than PTP Instance B if the value of the clockClass of A is lower than that of B.

See 7.6.2.5 of IEEE Std 1588-2019 for a more detailed description of clockClass.

NOTE—The PTP Instance has a LocalClock entity, which can be the free-running quartz crystal that just meets the IEEE 802.3 requirements, but could also be better. There can be a ClockSource entity, e.g., timing taken from a GNSS, available in the local system that provides timing to the ClockSource entity. The time provided by the PTP Instance, if it is the grandmaster, is reflected by the combination of these two entities, and the clockClass reflects this combination as specified in 7.6.2.5 of IEEE Std 1588-2019. For example, when the LocalClock entity uses a quartz oscillator that meets the requirements of IEEE Std 802.3 and B.1 of this standard, clockClass is set to 248. But, if a GNSS receiver is present and synchronizes the PTP Instance, then the clockClass is set to the value 6, indicating traceability to a primary reference time source (see 7.6.2.5 of IEEE Std 1588-2019).

8.6.2.3 clockAccuracy

The clockAccuracy attribute indicates the expected time accuracy of a ClockMaster.

The value shall be selected as follows:

a) clockAccuracy is set to the value that reflects the combination of the LocalClock and ClockSource entities; else
b) if the value that reflects the LocalClock and ClockSource entities is not specified or unknown, clockAccuracy is set to 254 (FE_{16}).

The ordering of clockAccuracy in the operation of the best master clock algorithm (see 10.3.4 and 10.3.5) is specified as follows. When comparing clockAccuracy values, PTP Instance A shall be deemed better than PTP Instance B if the value of the clockAccuracy of A is lower than that of B.

See 7.6.2.6 of IEEE Std 1588-2019 for more detailed description of clockAccuracy.

8.6.2.4 offsetScaledLogVariance

The offsetScaledLogVariance is a scaled, offset representation of an estimate of the PTP variance. The PTP variance characterizes the precision and frequency stability of the ClockMaster. The PTP variance is the square of PTPDEV (see B.1.3.2).

The value shall be selected as follows:

a) offsetScaledLogVariance is set to the value that reflects the combination of the LocalClock and ClockSource entities; else
b) if the value that reflects these entities is not specified or not known, offsetScaledLogVariance is set to 17258 ($436A_{16}$). This value corresponds to the value of PTPDEV for observation interval equal to the default Sync message transmission interval (i.e., observation interval of 0.125 s, see 11.5.2.3 and B.1.3.2).

The ordering of offsetScaledLogVariance in the operation of the best master clock algorithm (see 10.3.4 and 10.3.5) is specified as follows. When comparing offsetScaledLogVariance values, PTP Instance A shall be deemed better than PTP Instance B if the value of the offsetScaledLogVariance of A is lower than that of B.

See 7.6.3 of IEEE Std 1588-2019 for more detailed description of PTP variance and offsetScaledLogVariance (7.6.3.3 of IEEE Std 1588-2019 provides a detailed description of the computation of offsetScaledLogVariance from PTP variance, along with an example).

8.6.2.5 priority2

priority2 is used in the execution of the BMCA (see 10.3). The value of priority2 shall be an integer selected from the range 0 through 255. The ordering of priority2 in the operation of the BMCA is the same as the ordering of priority1 (see 8.6.2.1).

The default value of priority2 shall be 248. See 7.6.2.4 of IEEE Std 1588-2019 for a more detailed description of priority2.

8.6.2.6 clockIdentity

The clockIdentity value for a PTP Instance shall be as specified in 8.5.2.2.

8.6.2.7 timeSource

The timeSource is an information only attribute indicating the type of source of time used by a ClockMaster. The value is not used in the selection of the grandmaster. The data type of timeSource shall be TimeSource, which is an Enumeration8. The values of TimeSource are specified in Table 8-2. These represent categories. For example, the GPS entry includes not only the GPS system of the U.S. Department of Defense but the European Galileo system and other present and future GNSSs.

All unused values are reserved.

See 7.6.2.8 of IEEE Std 1588-2019 for a more detailed description of timeSource.

Table 8-2—TimeSource enumeration

Value	Time source	Description
0x10	ATOMIC_CLOCK	Any device, or device directly connected to such a device, that is based on atomic resonance for frequency and that has been calibrated against international standards for frequency and time
0x20	GPS NOTE—In this standard, this value refers to any GNSS or Regional Navigation Satellite System (i.e., not only GPS).	Any device synchronized to any of the satellite systems that distribute time and frequency tied to international standards
0x30	TERRESTRIAL_RADIO	Any device synchronized via any of the radio distribution systems that distribute time and frequency tied to international standards
0x40	PTP	Any device synchronized to an IEEE 1588 PTP-based source of time external to the gPTP domain (Sse NOTE).
0x50	NTP	Any device synchronized via NTP to servers that distribute time and frequency tied to international standards
0x60	HAND_SET	Used in all cases for any device whose time has been set by means of a human interface based on observation of an international standards source of time to within the claimed clock accuracy
0x90	OTHER	Any source of time and/or frequency not covered by other values, or for which the source is not known
0xA0	INTERNAL_OSCILLATOR	Any device whose frequency is not based on atomic resonance nor calibrated against international standards for frequency, and whose time is based on a free-running oscillator with epoch determined in an arbitrary or unknown manner
NOTE—For example, a clock that implements both a gPTP domain and a separate IEEE 1588 (i.e., PTP) domain, and is synchronized by the separate IEEE 1588 domain, would have time source of PTP in the gPTP domain.		

The initialization value is selected as follows:

a) If the timeSource (8.6.2.7 and Table 8-2) is known at the time of initialization, the value is derived from the table, else
b) The value is set to $A0_{16}$ (INTERNAL_OSCILLATOR).

8.6.2.8 numberPorts

The numberPorts indicates the number of ports on the PTP Instance.

9. Application interfaces

9.1 Overview of the interfaces

Unless otherwise stated, information in this clause is per domain.

The following subclauses define one application interface between the ClockSource entity and ClockMaster entity (see 10.1.1) and four application interfaces between the ClockTarget entity and ClockSlave entity (see 10.1.1). The ClockSource is an entity that can be used as an external timing source for the gPTP domain. The ClockSource entity either contains or has access to a clock (see 3.3). The ClockTarget entity represents any application that uses information provided by the ClockSlave entity via any of the application interfaces.

NOTE—The manner in which the ClockSource entity obtains time from a clock is outside the scope of this standard. The manner in which the ClockTarget uses the information provided by application interfaces is outside the scope of this standard.

The five interfaces are illustrated in Figure 9-1. They include the following:

a) the ClockSourceTime interface, which provides external timing to a PTP Instance,
b) the ClockTargetEventCapture interface, which returns the synchronized time of an event signaled by a ClockTarget entity,
c) the ClockTargetTriggerGenerate interface, which causes an event to be signaled at a synchronized time specified by a ClockTarget entity,
d) the ClockTargetClockGenerator interface, which causes a periodic sequence of results to be generated, with a phase and rate specified by a ClockTarget entity, and
e) the ClockTargetPhaseDiscontinuity interface, which supplies information that an application can use to determine if a discontinuity in grandmaster phase or frequency has occurred.

NOTE—The application interfaces described in this clause are models for behavior and not application program interfaces. Other application interfaces besides items a) through e) above are possible, but are not described here. In addition, there can be multiple instances of a particular interface.

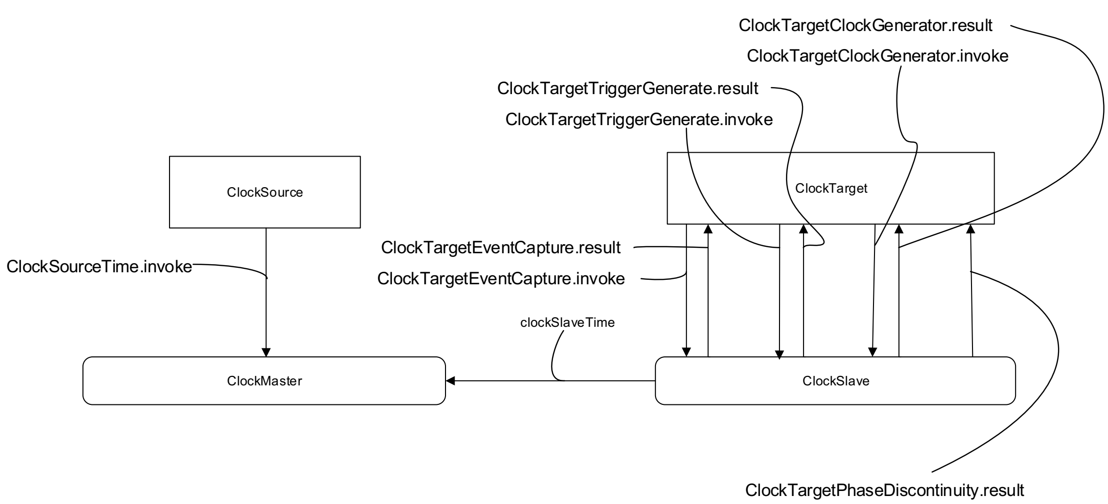


Figure 9-1—Application interfaces

9.2 ClockSourceTime interface

9.2.1 General

This interface is used by the ClockSource entity to provide time to the ClockMaster entity of a PTP Instance. The ClockSource entity invokes the ClockSourceTime.invoke function to provide the time, relative to the ClockSource, that this function is invoked.

9.2.2 ClockSourceTime.invoke function parameters

```
ClockSourceTime.invoke {
        domainNumber,
        sourceTime,
        timeBaseIndicator,
        lastGmPhaseChange,
        lastGmFreqChange
}
```

The parameter definitions are as follows:

9.2.2.1 domainNumber (UInteger8)

The domain number of the gPTP domain that this ClockSource entity is providing time to.

9.2.2.2 sourceTime (ExtendedTimestamp)

The value of sourceTime is the time this function is invoked by the ClockSource entity.

9.2.2.3 timeBaseIndicator (UInteger16)

The timeBaseIndicator is a binary value that is set by the ClockSource entity. The ClockSource entity changes the value whenever its time base changes. The ClockSource entity shall change the value of timeBaseIndicator if and only if there is a phase or frequency change.

NOTE—While the clock that supplies time to the ClockSource entity can be lost, i.e., the PTP Instance can enter holdover, the ClockSource entity itself is not lost. The ClockSource entity ensures that timeBaseIndicator changes if the source of time is lost.

9.2.2.4 lastGmPhaseChange (ScaledNs)

The value of lastGmPhaseChange is the phase change (i.e., change in sourceTime) that occurred on the most recent change in timeBaseIndicator. The value is initialized to 0.

9.2.2.5 lastGmFreqChange (Float64)

The value of lastGmFreqChange is the fractional frequency change (i.e., frequency change expressed as a pure fraction) that occurred on the most recent change in timeBaseIndicator. The value is initialized to 0.

9.3 ClockTargetEventCapture interface

9.3.1 General

This interface is used by the ClockTarget entity to request the synchronized time of an event that it signals to the ClockSlave entity of a PTP Instance. The ClockTarget entity invokes the

ClockTargetEventCapture.invoke function to signal an event to the ClockSlave entity. The ClockSlave entity invokes the ClockTargetEventCapture.result function to return the time of the event relative to the current grandmaster or, if no PTP Instance is grandmaster-capable, the LocalClock. The ClockTargetEventCapture.result function also returns gmPresent, to indicate to the ClockTarget whether or not a grandmaster is present.

9.3.2 ClockTargetEventCapture.invoke parameters

```
ClockTargetEventCapture.invoke {
        domainNumber
}
```

The parameter definitions are:

9.3.2.1 domainNumber (UInteger8)

The domain number of the ClockSlave entity that is requested to provide the synchronized time of the signaled event.

9.3.3 ClockTargetEventCapture.result parameters

```
ClockTargetEventCapture.result {
        domainNumber,
        slaveTimeCallback,
        gmPresent
}
```

The parameter definitions are:

9.3.3.1 domainNumber (UInteger8)

The domain number of the ClockSlave entity that is providing the synchronized time of the signaled event.

9.3.3.2 slaveTimeCallback (ExtendedTimestamp)

The value of slaveTimeCallback is the time, relative to the grandmaster, that the corresponding ClockTargetEventCapture.invoke function is invoked.

NOTE—The invocation of the ClockTargetEventCapture.invoke function and the detection of this invocation by the ClockSlave entity are simultaneous in this abstract interface.

9.3.3.3 gmPresent (Boolean)

The value of gmPresent is set equal to the value of the global variable gmPresent (see 10.2.4.13). This parameter indicates to the ClockTarget whether or not a grandmaster is present.

9.4 ClockTargetTriggerGenerate interface

9.4.1 General

This interface is used by the ClockTarget entity to request that the ClockSlave entity send a result at a specified time relative to the grandmaster. The ClockTarget entity invokes the ClockTargetTriggerGenerate.invoke function to indicate the synchronized time of the event. The ClockSlave entity invokes the ClockTargetTriggerGenerate.result function to either signal the event at the requested synchronized time or indicate an error condition.

9.4.2 ClockTargetTriggerGenerate.invoke parameters

```
ClockTargetTriggerGenerate.invoke {
        domainNumber,
        slaveTimeCallback
}
```

The parameter definition is:

9.4.2.1 domainNumber (UInteger8)

The domain number of the ClockSlave entity that is requested to signal an event at the specified time.

9.4.2.2 slaveTimeCallback (ExtendedTimestamp)

If slaveTimeCallback is nonzero, its value is the synchronized time the corresponding ClockTargetTriggerGenerate.result function, i.e., the trigger, is to be invoked. If slaveTimeCallback is zero, any previous ClockTargetTriggerGenerate.invoke function for which a ClockTargetTriggerGenerate.result function has not yet been issued is canceled.

9.4.3 ClockTargetTriggerGenerate.result parameters

```
ClockTargetTriggerGenerate.result {
        domainNumber,
        errorCondition,
        gmPresent
}
```

The parameter definitions are:

9.4.3.1 domainNumber (UInteger8)

The domain number of the ClockSlave entity that is signaling an event at the specified time.

9.4.3.2 errorCondition (Boolean)

A value of FALSE indicates that the ClockTargetTriggerGenerate.result function was invoked at the time, relative to the grandmaster, contained in the corresponding ClockTargetTriggerGenerate.invoke function. A value of TRUE indicates that the ClockTargetTriggerGenerate.result function could not be invoked at the synchronized time contained in the corresponding ClockTargetTriggerGenerate.invoke function.

NOTE—For example, the ClockTargetTriggerGenerate.result function is invoked with errorCondition = TRUE if the requested slaveTimeCallback is a time prior to the synchronized time when the corresponding ClockTargetTriggerGenerate.invoke function is invoked. As another example, the ClockTargetTriggerGenerate.result function is invoked with errorCondition = TRUE if a discontinuity in the synchronized time causes the requested slaveTimeCallback to be skipped over.

9.4.3.3 gmPresent (Boolean)

The value of gmPresent is set equal to the value of the global variable gmPresent (see 10.2.4.13). This parameter indicates to the ClockTarget whether or not a grandmaster is present.

9.4.4 ClockTargetTriggerGenerate interface definition

The invocation of the ClockTargetTriggerGenerate.invoke function causes the ClockSlave entity to store the value of the slaveTimeCallback parameter in an internal variable (replacing any previous value of that variable) until the synchronized time, or LocalClock time if gmPresent is FALSE, equals the value of that variable, at which time the ClockTargetTriggerGenerate.result function is invoked with errorCondition = FALSE. If it is not possible to invoke the ClockTargetTriggerGenerate.result function at slaveTimeCallback, e.g., if slaveTimeCallback is earlier than the synchronized time (or LocalClock time if gmPresent is FALSE) when the ClockTargetTriggerGenerate.invoke function is invoked, the ClockTargetTriggerGenerate.result function is invoked with errorCondition = TRUE. Invocation of the ClockTargetTriggerGenerate.invoke function with slaveTimeCallback = 0 (which is earlier than any synchronized time) is used to cancel a pending request.

9.5 ClockTargetClockGenerator interface

9.5.1 General

This interface is used by the ClockTarget entity to request that the ClockSlave entity deliver a periodic clock signal of specified period and phase. The ClockTarget entity invokes the ClockTargetClockGenerator.invoke function to request that the ClockSlave entity generate the periodic clock signal. The ClockSlave entity invokes the ClockTargetClockGenerator.result function at significant instants of the desired clock signal.

9.5.2 ClockTargetClockGenerator.invoke parameters

```
ClockTargetClockGenerator.invoke {
        domainNumber,
        clockPeriod,
        slaveTimeCallbackPhase
}
```

The parameter definitions are:

9.5.2.1 domainNumber (UInteger8)

The domain number of the ClockSlave entity that is requested to deliver a periodic clock signal.

9.5.2.2 clockPeriod (TimeInterval)

The value of clockPeriod is the period between successive invocations of the ClockTargetClockGenerator.result function. A value that is zero or negative causes any existing periodic clock signal generated via this application interface to be terminated.

9.5.2.3 slaveTimeCallbackPhase (ExtendedTimestamp)

The value of slaveTimeCallbackPhase describes phase of the generated clock signal by specifying a point on the timescale in use such that ClockTargetClockGenerator.result invocations occur at synchronized times that differ from slaveTimeCallbackPhase by $n \times$ clockPeriod, where n is an integer.

NOTE—The value of slaveTimeCallbackPhase can be earlier or later than the synchronized time the ClockTargetClockGenerator.invoke function is invoked; use of a slaveTimeCallbackPhase value in the future does not imply that the initiation of the periodic clock signal is suppressed until that synchronized time.

9.5.3 ClockTargetClockGenerator.result parameters

The semantics of the function are:

```
ClockTargetClockGenerator.result {
        domainNumber,
        slaveTimeCallback,
}
```

The parameter definition is:

9.5.3.1 domainNumber (UInteger8)

The domain number of the ClockSlave entity that is delivering a periodic clock signal.

9.5.3.2 slaveTimeCallback (ExtendedTimestamp)

The value of slaveTimeCallback is the synchronized time of this event.

9.6 ClockTargetPhaseDiscontinuity interface

9.6.1 General

This interface provides discontinuity information, sent from the grandmaster, to an application within a station. It is used by the ClockSlave entity to supply sufficient information to the ClockTarget entity to enable the ClockTarget entity to determine whether a phase or frequency discontinuity has occurred. The ClockSlave invokes the ClockTargetPhaseDiscontinuity.result function in the SEND_SYNC_INDICATION block of the ClockSlaveSync state machine (see 10.2.13 and Figure 10-9). The invocation occurs when a PortSyncSync structure is received, after the needed information has been computed by the ClockSlaveSync state machine.

9.6.2 ClockTargetPhaseDiscontinuity.result parameters

```
ClockTargetPhaseDiscontinuity.result {
        domainNumber,
        gmIdentity,
        gmTimeBaseIndicator,
        lastGmPhaseChange,
        lastGmFreqChange
}
```

The parameter definitions are:

9.6.2.1 domainNumber (UInteger8)

The domain number of the ClockSlave entity that is providing discontinuity information.

9.6.2.2 gmIdentity (ClockIdentity)

If gmPresent (see 10.2.4.13) is TRUE, the value of gmIdentity is the ClockIdentity of the current grandmaster. If gmPresent is FALSE, the value of gmIdentity is 0x0.

9.6.2.3 gmTimeBaseIndicator (UInteger16)

The value of gmTimeBaseIndicator is the timeBaseIndicator of the current grandmaster.

9.6.2.4 lastGmPhaseChange (ScaledNs)

The value of the global lastGmPhaseChange parameter (see 10.2.4.16) received from the grandmaster.

9.6.2.5 lastGmFreqChange (Float64)

The value of lastGmFreqChange parameter (see 10.2.4.17) received from the grandmaster.

10. Media-independent layer specification

10.1 Overview

Unless otherwise stated, information in this clause is per domain.

10.1.1 Model of operation

A PTP Instance contains a best master selection function and a synchronization function. These functions include port-specific aspects and aspects associated with the PTP Instance as a whole. The functions are distributed among a number of entities, which together describe the behavior of a compliant implementation. The functions are specified by a number of state machines.

The model for the media-independent layer of a PTP Instance is shown in Figure 10-1. It includes a single SiteSync entity, ClockMaster entity, and ClockSlave entity for the PTP Instance as a whole, plus one PortSync for each port. The PTP Instance also includes one MD entity for each port, which is part of the media-dependent layer. The media-dependent functions performed by the MD entity are described in the clauses for the respective media. In addition to the entities, Figure 10-1 shows the information that flows between the entities via the PortSyncSync, MDSyncSend, and MDSyncReceive structures (see 10.2.2.3, 10.2.2.1, and 10.2.2.2, respectively).

The SiteSync, ClockMaster, ClockSlave, and PortSync entities each contain a number of cooperating state machines, which are described later in this clause (the MD entity state machines are described in the respective media-dependent clauses). The ClockMaster entity receives information from an external time source, referred to as a ClockSource entity (see 9.2), via an application interface, and provides the information to the SiteSync entity. The ClockSlave entity receives grandmaster time-synchronization and current grandmaster information from the SiteSync entity, and makes the information available to an external application, referred to as a clockTarget entity (see 9.3 through 9.6), via one or more application service interfaces. The SiteSync entity executes the portion of best master clock selection associated with the PTP Instance as a whole, i.e., it uses the best master information received on each port to determine which port has received the best information, and updates the states of all the ports (see 10.3.1 for a discussion of port states). It also distributes synchronization information received on the SlavePort to all the ports whose state is MasterPort (see 10.3.1). The PortSync entity for a SlavePort receives best master selection information from the PTP Instance at the other end of the associated link, compares this to the current best master information that it has, and forwards the result of the comparison to the Site Sync entity. The PortSync entity for a SlavePort also receives time-synchronization information from the MD entity associated with the port, and forwards it to the SiteSync entity. The PortSync entity for a MasterPort sends best master selection and time-synchronization information to the MD entity for the port, which in turn sends the respective messages.

NOTE—This clause does not require a one-to-one correspondence between the PortSync entities of PTP Instances attached to the same gPTP communication path (see 3.9), i.e., more than two PTP Instances can be attached to a gPTP communication path that uses a shared medium and meet the requirements of this clause. However, it is possible for a media-dependent clause to have additional requirements that limit the gPTP communication paths to point-to-point links for that medium; in this case, each link has exactly two PortSync entities, which can be considered to be in one-to-one correspondence. One example of this is the full-duplex, point-to-point media-dependent layer specified in Clause 11. In addition, one or more gPTP communication paths can be logically point-to-point but traverse the same shared medium.

The LocalClock entity is a free-running clock (see 3.3) that provides a common time to the PTP Instance, relative to an arbitrary epoch. A PTP Instance contains a LocalClock entity. The requirements for the LocalClock entity are specified in B.1. All timestamps are taken relative to the LocalClock entity (see 8.4.3). The LocalClock entity also provides the value of currentTime (see 10.2.4.12), which is used in the state machines to specify the various timers.

NOTE—The epoch for the LocalClock entity can be the time that the PTP Instance is powered on.

The time-synchronization state machines are described in 10.2. The best master clock selection state machines are described in 10.3. The attributes and format of the Announce message are described in 10.5 and 10.6. The timing characterization of the protocol is described in 10.7.

10.1.2 Grandmaster-capable PTP Instance

A PTP Instance may be grandmaster-capable. An implementation may provide the ability to configure a PTP Instance as grandmaster-capable via a management interface.

NOTE—The managed object gmCapable is read only (see Table 14-1). gmCapable is configured by setting the value of the managed object priority1, which is read/write (see Table 14-1). If the value of priority1 is 255, then:

a) gmCapable is set to FALSE (see 8.6.2.1), and
b) the value of the managed object clockClass, which is read only, is set to 255 (see 8.6.2.2).

NOTE—While a PTP Instance that is not grandmaster-capable can never be the grandmaster of the gPTP domain, such a PTP Instance contains a best master selection function, invokes the best master selection algorithm, and conveys synchronization information received from the current grandmaster.

10.2 Time-synchronization state machines

10.2.1 Overview

The time-synchronization function in a PTP Instance is specified by a number of cooperating state machines. Figure 10-2 illustrates these state machines, their local variables, their interrelationships, and the global variables and structures used to communicate between them. The figure indicates the interaction between the state machines and the media-dependent layer and LocalClock entity.

The ClockMasterSyncReceive, ClockMasterSyncOffset, and ClockMasterSyncSend state machines are optional for PTP Instances that are not grandmaster-capable (see 8.6.2.1 and 10.1.2). These state machines may be present in a PTP Instance that is not grandmaster-capable; however, any information supplied by them, via the ClockMasterSyncSend state machine, to the SiteSyncSync state machine is not used by the SiteSyncSync state machine if the PTP Instance is not grandmaster-capable.

The media-independent layer state machines in Figure 10-2 are as follows:

a) ClockMasterSyncReceive (one instance per PTP Instance): receives ClockSourceTime.invoke functions from the ClockSource entity and notifications of LocalClock entity ticks (see 10.2.4.18), updates masterTime, and provides masterTime to ClockMasterSyncOffset and ClockMasterSyncSend state machines.
b) ClockMasterSyncOffset (one instance per PTP Instance): receives syncReceiptTime from the ClockSlave entity and masterTime from the ClockMasterSyncReceive state machine, computes phase offset and frequency offset between masterTime and syncReceiptTime if the PTP Instance is not the grandmaster, and provides the frequency and phase offsets to the ClockMasterSyncSend state machine.
c) ClockMasterSyncSend (one instance per PTP Instance): receives masterTime from the ClockMasterSyncReceive state machine, receives phase and frequency offset between masterTime and syncReceiptTime from the ClockMasterSyncOffset state machine, and provides masterTime (i.e., synchronized time) and the phase and frequency offset to the SiteSync entity using a PortSyncSync structure.

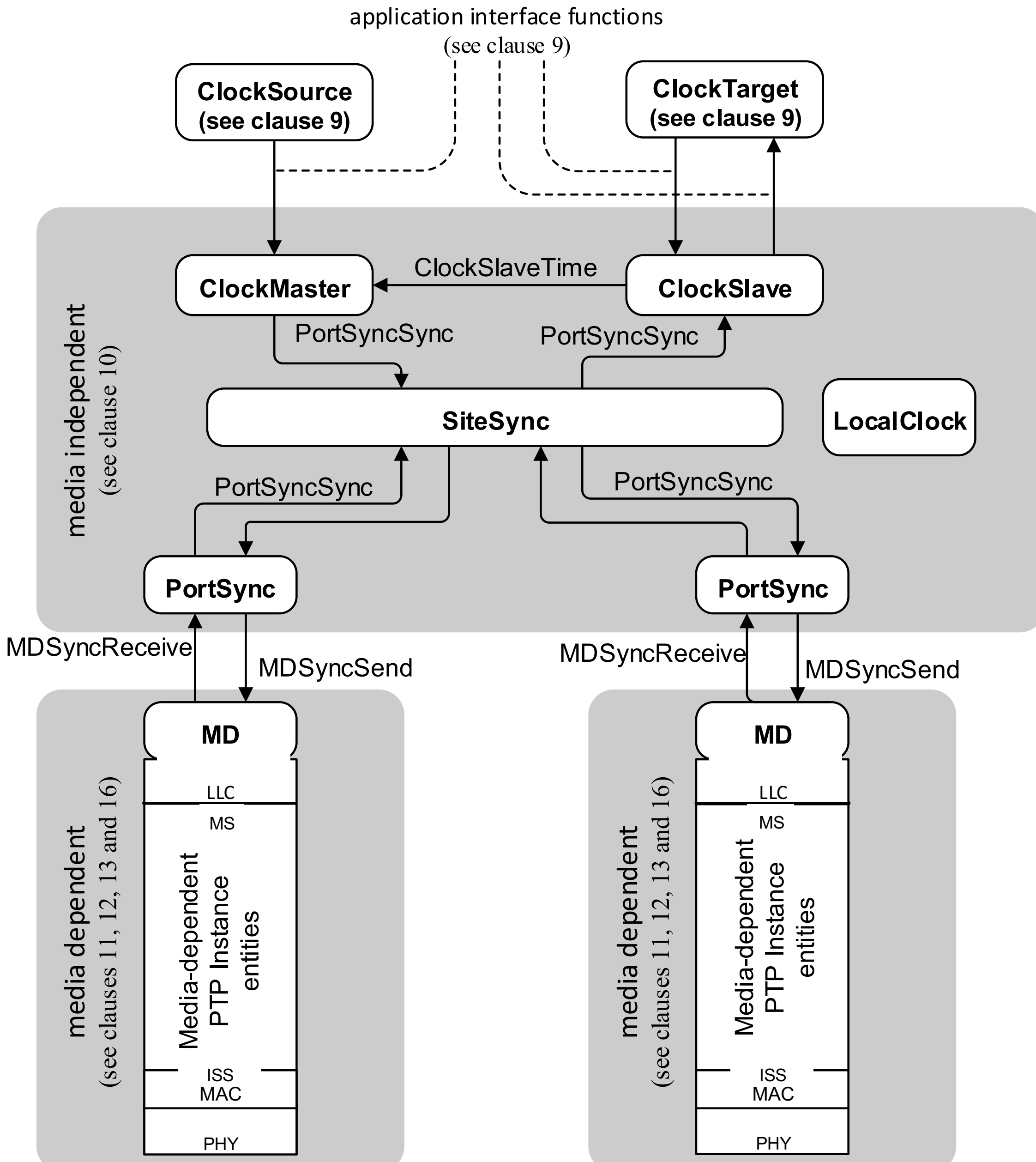


Figure 10-1—Model for media-independent layer of PTP Instance

d) PortSyncSyncReceive (one instance per PTP Instance, per port): receives time-synchronization information from the MD entity of the corresponding port, computes accumulated rateRatio, computes syncReceiptTimeoutTime, and sends the information to the SiteSync entity.

e) SiteSyncSync (one instance per PTP Instance): receives time-synchronization information, accumulated rateRatio, and syncReceiptTimeoutTime from the PortSync entity of the current slave port or from the ClockMaster entity, and sends the information to the PortSync entities of all the ports and to the ClockSlave entity.

f) PortSyncSyncSend (one instance per PTP Instance, per port): receives time-synchronization information from the SiteSync entity, requests that the MD entity of the corresponding port send a time-synchronization event message, receives the syncEventEgressTimestamp for this event message from the MD entity, uses the most recent time-synchronization information received from the SiteSync entity and the timestamp to compute time-synchronization information that will be sent by the MD entity in a general message (e.g., for full-duplex IEEE 802.3 media) or a subsequent event message (e.g., for IEEE 802.11 media), and sends this latter information to the MD entity.

g) ClockSlaveSync (one instance per PTP Instance): receives time-synchronization information from the SiteSync entity, computes clockSlaveTime and syncReceiptTime, sets syncReceiptLocalTime, GmTimeBaseIndicator, lastGmPhaseChange, lastGmFreqChange, sends clockSlaveTime to the ClockMaster entity, and provides information to the ClockTarget entity (via the ClockTargetPhaseDiscontinuity interface, see 9.6) to enable that entity to determine if a phase or frequency discontinuity has occurred.

10.2.2 Data structures communicated between state machines

The following subclauses describe the data structures communicated between the time-synchronization state machines.

10.2.2.1 MDSyncSend

10.2.2.1.1 General

This structure contains information that is sent by the PortSync entity of a port to the MD entity of that port when requesting that the MD entity cause time-synchronization information to be sent. The structure contains information that reflects the most recent time-synchronization information received by this PTP Instance, and is used to determine the contents of the time-synchronization event message and possibly separate general message that will be sent by this port.

The structure is:

```
MDSyncSend {
        domainNumber,
        followUpCorrectionField,
        sourcePortIdentity,
        logMessageInterval,
        preciseOriginTimestamp,
        upstreamTxTime,
        rateRatio,
        gmTimeBaseIndicator,
        lastGmPhaseChange,
        lastGmFreqChange
}
```

The members of the structure are defined in the following subclauses.

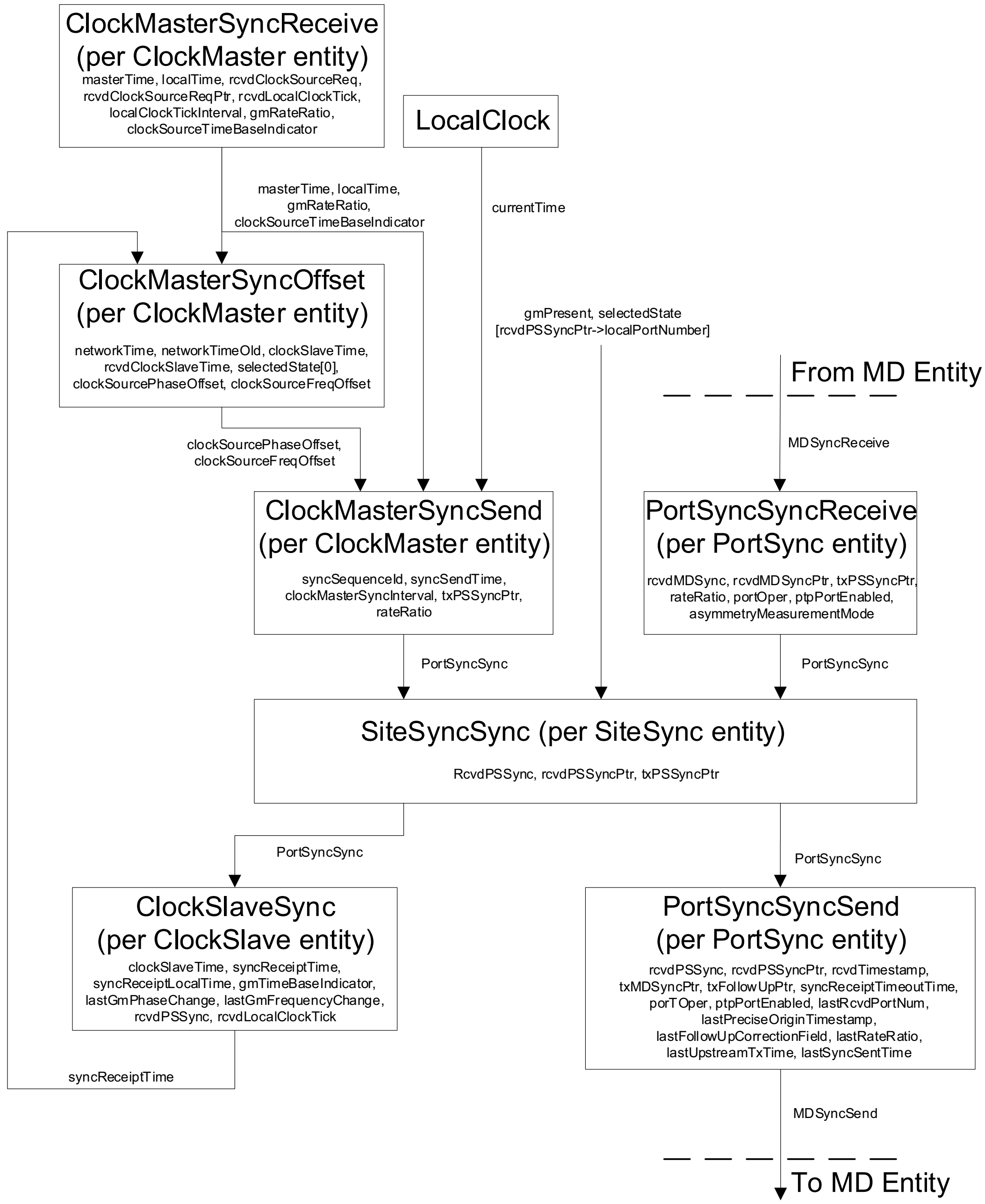
10.2.2.1.2 domainNumber (UInteger8)

The domainNumber of the gPTP domain in which this structure is sent.

NOTE—The domain number member is not essential because the state machines that send and receive this structure are per domain, and each state machine implicitly knows the number of the domain in which it operates.

10.2.2.1.3 followUpCorrectionField (ScaledNs)

The followUpCorrectionField contains the accumulated time since the preciseOriginTimestamp was captured by the grandmaster. This is equal to the elapsed time, relative to the grandmaster, between the time



Notes:
a) selectedState for each port and gmPresent are set by Port State Selection state machine (see 10.3.12)
b) currentTime is a global variable that is always equal to the current time relative to the local oscillator
c) application interfaces to higher layers are not shown
d) the ClockMasterSyncReceive, ClockMasterSyncSend, and ClockMasterSyncOffset state machines are optional for time-aware systems that are not grandmaster-capable.

Figure 10-2—Time-synchronization state machines—overview and interrelationships

the grandmaster sent the received time-synchronization event message, truncated to the nearest nanosecond, and the time at which that event message was sent by the upstream PTP Instance. The followUpCorrectionField is equal to the value of the followUpCorrectionField member of the most recently received PortSyncSync structure from the PortSync entity of this port (see 10.2.2.3.5).

10.2.2.1.4 sourcePortIdentity (PortIdentity)

The sourcePortIdentity is the portIdentity of this port (see 8.5.2).

10.2.2.1.5 logMessageInterval (Integer8)

The logMessageInterval is the value of currentLogSyncInterval for this port (see 10.7.2.3).

10.2.2.1.6 preciseOriginTimestamp (Timestamp)

The preciseOriginTimestamp is the sourceTime of the ClockMaster entity of the grandmaster, with any fractional nanoseconds truncated, when the received time-synchronization information was sent by the grandmaster. The preciseOriginTimestamp is the value of the preciseOriginTimestamp member of the most recently received PortSyncSync structure from the PortSync entity of this port (see 10.2.2.3.8).

10.2.2.1.7 upstreamTxTime (UScaledNs)

The upstreamTxTime is given by:

$$\text{upstreamTxTime} = \text{syncEventIngressTimestamp} - \frac{\text{meanLinkDelay}}{\text{neighborRateRatio}}$$

where the value of syncEventIngressTimestamp corresponds to the receipt of the time-synchronization information at the slave port of this PTP Instance, meanLinkDelay is defined in 10.2.5.8), and neighborRateRatio is defined in 10.2.5.7). The upstreamTxTime is the value of the upstreamTxTime member of the most recently received PortSyncSync structure from the PortSync entity of this port (see 10.2.2.3.9).

10.2.2.1.8 rateRatio (Float64)

The rateRatio is the value of the rateRatio member of the most recently received PortSyncSync structure from the PortSync entity of this port (see 10.2.2.3.10). It is equal to the ratio of the frequency of the grandmaster to the frequency of the LocalClock entity of this PTP Instance (see 10.2.8.1.4).

10.2.2.1.9 gmTimeBaseIndicator (UInteger16)

The gmTimeBaseIndicator is the timeBaseIndicator of the ClockSource entity of the current grandmaster. It is set equal to the gmTimeBaseIndicator of the received time-synchronization information. The gmTimeBaseIndicator is the value of the gmTimeBaseIndicator member of the most recently received PortSyncSync structure from the PortSync entity of this port (see 10.2.2.3.11).

10.2.2.1.10 lastGmPhaseChange (ScaledNs)

The lastGmPhaseChange is the time of the current grandmaster minus the time of the previous grandmaster, at the time that the current grandmaster became grandmaster, or the step change in the time of the current grandmaster at the time of the most recent gmTimeBaseIndicator change. It is set equal to the lastGmPhaseChange of the received time-synchronization information. The lastGmPhaseChange is the value of the lastGmPhaseChange member of the most recently received PortSyncSync structure from the PortSync entity of this port (see 10.2.2.3.12).

10.2.2.1.11 lastGmFreqChange (Float64)

The lastGmFreqChange is the fractional frequency offset of the current grandmaster relative to the previous grandmaster, at the time that the current grandmaster became grandmaster, or relative to itself prior to the last change in gmTimeBaseIndicator. It is set equal to the lastGmFreqChange of the received time-synchronization information. The lastGmFreqChange is the value of the lastGmFreqChange member of the most recently received PortSyncSync structure from the PortSync entity of this port (see 10.2.2.3.13).

10.2.2.2 MDSyncReceive

10.2.2.2.1 General

This structure contains information that is sent by the MD entity of a port to the PortSync entity of that port. It provides the PortSync entity with master clock timing information and timestamp of receipt of a time-synchronization event message compensated for propagation time on the upstream link. The information is sent to the PortSync entity upon receipt of time-synchronization information by the MD entity of the port. The information is in turn provided by the PortSync entity to the SiteSync entity. The information is used by the PortSyncSyncReceive state machine of the PortSync entity to compute the rate ratio of the grandmaster relative to the local clock and is communicated to the SiteSync entity, and then by the SiteSync entity to the other PortSync entities for use in computing master clock timing information.

The structure is:

```
MDSyncReceive {
        domainNumber,
        followUpCorrectionField,
        sourcePortIdentity,
        logMessageInterval,
        preciseOriginTimestamp,
        upstreamTxTime,
        rateRatio,
        gmTimeBaseIndicator,
        lastGmPhaseChange,
        lastGmFreqChange
}
```

The members of the structure are defined in the following subclauses.

10.2.2.2.2 domainNumber (UInteger8)

The domainNumber is the domain number of the gPTP domain in which this structure is sent.

NOTE—The domain number member is not essential because the state machines that send and receive this structure are per domain, and each state machine implicitly knows the number of the domain in which it operates.

10.2.2.2.3 followUpCorrectionField (ScaledNs)

The followUpCorrectionField contains the elapsed time, relative to the grandmaster, between the time the grandmaster sent the received time-synchronization information, truncated to the nearest nanosecond, and the time at which this information was sent by the upstream PTP Instance.

NOTE 1—The sum of followUpCorrectionField and preciseOriginTimestamp is the synchronized time that corresponds to the time the most recently received time-synchronization event message was sent by the upstream PTP Instance.

NOTE 2—For a medium that uses separate event and general messages (for example, full-duplex, point-to-point media described in Clause 11), the event message corresponding to the most recently received network synchronization information is the event message that corresponds to the most recently received general message. For a medium that places synchronization information based on the event message timestamp in the next event message (for example, IEEE 802.11 media described in Clause 12), the event message corresponding to the most recently received network synchronization information is the previous event message; in this case, the time-synchronization information in the current event message refers to the previous event message.

10.2.2.2.4 sourcePortIdentity (PortIdentity)

The sourcePortIdentity is the value of the sourcePortIdentity of the time-synchronization event message received by this port. It is the portIdentity of the upstream MasterPort that sent the event message.

10.2.2.2.5 logMessageInterval (Integer8)

The logMessageInterval is the value of the logMessageInterval of the time-synchronization event message received by this port. It is the currentLogSyncInterval (see 10.7.2.3) of the upstream MasterPort that sent the event message.

10.2.2.2.6 preciseOriginTimestamp (Timestamp)

The preciseOriginTimestamp is the sourceTime of the ClockMaster entity of the grandmaster, with any fractional nanoseconds truncated, when the time-synchronization event message was sent by the grandmaster.

10.2.2.2.7 upstreamTxTime (UScaledNs)

The upstreamTxTime is given by:

$$\text{upstreamTxTime} = \text{syncEventIngressTimestamp} - \frac{\text{meanLinkDelay}}{\text{neighborRateRatio}}$$

where the value of syncEventIngressTimestamp corresponds to the receipt of the time-synchronization information at the slave port of this PTP Instance (i.e., at this port), meanLinkDelay is defined in 10.2.5.8), and neighborRateRatio is defined in 10.2.5.7).

10.2.2.2.8 rateRatio (Float64)

The rateRatio is the value of rateRatio of the received time-synchronization information. It is equal to the ratio of the frequency of the grandmaster to the frequency of the LocalClock entity of the PTP Instance at the other end of the link attached to this port, i.e., the PTP Instance that sent the most recently received time-synchronization event message (see 10.2.8.1.4).

10.2.2.2.9 gmTimeBaseIndicator (UInteger16)

The gmTimeBaseIndicator is the timeBaseIndicator of the ClockSource entity of the current grandmaster. It is set equal to the gmTimeBaseIndicator of the received time-synchronization information.

10.2.2.2.10 lastGmPhaseChange (ScaledNs)

The lastGmPhaseChange is the time of the current grandmaster minus the time of the previous grandmaster, at the time that the current grandmaster became grandmaster, or the step change in the time of the current grandmaster at the time of the most recent gmTimeBaseIndicator change. It is set equal to the lastGmPhaseChange of the received time-synchronization information.

10.2.2.2.11 lastGmFreqChange (Float64)

The lastGmFreqChange is the fractional frequency offset of the current grandmaster relative to the previous grandmaster, at the time that the current grandmaster became grandmaster, or relative to itself prior to the last change in gmTimeBaseIndicator. It is set equal to the lastGmFreqChange of the received time-synchronization information.

10.2.2.3 PortSyncSync

10.2.2.3.1 General

This structure is sent by the PortSync and ClockMaster entities to the SiteSync entity, and also from the SiteSync entity to the PortSync and ClockSlave entities.

When sent from the PortSync or ClockMaster entity, it provides the SiteSync entity with master clock timing information, timestamp of receipt of a time-synchronization event message compensated for propagation time on the upstream link, and the time at which sync receipt timeout occurs if a subsequent Sync message is not received by then. The information is used by the SiteSync entity to compute the rate ratio of the grandmaster relative to the local clock and is communicated to the other PortSync entities for use in computing master clock timing information.

When sent from the SiteSync entity to the PortSync or ClockSlave entity, the structure contains information needed to compute the synchronization information that will be included in respective fields of the time-synchronization event and general messages that will be sent, and also to compute the synchronized time that the ClockSlave entity will supply to the ClockTarget entity.

The structure is:

```
PortSyncSync {
        domainNumber,
        localPortNumber,
        syncReceiptTimeoutTime,
        followUpCorrectionField,
        sourcePortIdentity,
        logMessageInterval,
        preciseOriginTimestamp,
        upstreamTxTime,
        rateRatio,
        gmTimeBaseIndicator,
        lastGmPhaseChange,
        lastGmFreqChange
}
```

The parameters of the PortSyncSync structure are defined in the following subclauses for the case where the structure is sent from the PortSync or ClockMaster entity to the SiteSync entity. If the structure is sent from the SiteSync entity to the PortSync or ClockSlave entity, the member values are copied from the most recently received PortSyncSync structure where the port that received this structure has port state of SlavePort.

10.2.2.3.2 domainNumber (UInteger8)

The domainNumber is the domain number of the gPTP domain in which this structure is sent.

NOTE—The domain number member is not essential because the state machines that send and receive this structure are per domain, and each state machine implicitly knows the number of the domain in which it operates.

10.2.2.3.3 localPortNumber (UInteger16)

If the structure is sent by a PortSync entity, the localPortNumber is the port number of the port whose PortSync entity sent this structure. If the structure is sent by a ClockMaster entity, the localPortNumber is zero.

10.2.2.3.4 syncReceiptTimeoutTime (UScaledNs)

If the structure is sent by a PortSync entity, the syncReceiptTimeoutTime is the value of the local time (i.e., the free-running, local clock time) at which sync receipt timeout occurs if a subsequent time-synchronization event message is not received by that time. If the structure is sent by a ClockMaster entity, the syncReceiptTimeoutTime is $FFFFFFFFFFFFFFFF_{16}$ [see 10.2.9.2.1, item h)].

10.2.2.3.5 followUpCorrectionField (ScaledNs)

If the structure is sent by a PortSync entity, the followUpCorrectionField is the value of the followUpcorrectionField member of the MDSyncReceive structure whose receipt caused the sending of this structure (see 10.2.2.2.2). If the structure is sent by a ClockMaster entity, the followUpCorrectionField is the sub-nanosecond portion of the ClockMaster time.

10.2.2.3.6 sourcePortIdentity (PortIdentity)

If the structure is sent by a PortSync entity, the sourcePortIdentity is the value of the sourcePortIdentity member of the MDSyncReceive structure whose receipt caused the sending of this structure (see 10.2.2.2.4). If the structure is sent by a ClockMaster entity, the clockIdentity member of the sourcePortIdentity is the clockIdentity of this PTP Instance, and the portNumber member of the sourcePortIdentity is 0.

10.2.2.3.7 logMessageInterval (Integer8)

If the structure is sent by a PortSync entity, the logMessageInterval is the value of the logMessageInterval member of the MDSyncReceive structure whose receipt caused the sending of this structure (see 10.2.2.2.5). If the structure is sent by a ClockMaster entity, the logMessageInterval is the value of clockMasterLogSyncInterval (see 10.7.2.4).

10.2.2.3.8 preciseOriginTimestamp (Timestamp)

If the structure is sent by a PortSync entity, the preciseOriginTimestamp is the value of the preciseOriginTimestamp member of the MDSyncReceive structure whose receipt caused the sending of this structure (see 10.2.2.2.6). If the structure is sent by a ClockMaster entity, the preciseOriginTimestamp is the ClockMaster time truncated to the next lower nanosecond.

10.2.2.3.9 upstreamTxTime (UScaledNs)

If the structure is sent by a PortSync entity, the upstreamTxTime is the value of the upstreamTxTime member of the MDSyncReceive structure whose receipt caused the sending of this structure (see 10.2.2.2.7). If the structure is sent by a ClockMaster entity, the upstreamTxTime is the local clock time corresponding to the ClockMaster time.

10.2.2.3.10 rateRatio (Float64)

If the structure is sent by a PortSync entity, the rateRatio is the value of the rateRatio member of the MDSyncReceive structure whose receipt caused the sending of this structure (see 10.2.2.2.8). It is equal to the ratio of the frequency of the grandmaster to the frequency of the LocalClock entity of the PTP Instance at the other end of the link attached to this port, i.e., the PTP Instance that sent the most recently-received time-synchronization event message (see 10.2.8.1.4). If the structure is sent by a ClockMaster entity, the rateRatio is equal to gmRateRatio (see 10.2.4.14).

10.2.2.3.11 gmTimeBaseIndicator (UInteger16)

If the structure is sent by a PortSync entity, the gmTimeBaseIndicator is the value of the gmTimeBaseIndicator member of the MDSyncReceive structure whose receipt caused the sending of this structure (see 10.2.2.2.9). If the structure is sent by a ClockMaster entity, the gmTimeBaseIndicator is equal to clockSourceTimeBaseIndicator (see 10.2.4.8).

10.2.2.3.12 lastGmPhaseChange (ScaledNs)

If the structure is sent by a PortSync entity, the lastGmPhaseChange is the value of the lastGmPhaseChange member of the MDSyncReceive structure whose receipt caused the sending of this structure (see 10.2.2.2.9). If the structure is sent by a ClockMaster entity, the lastGmPhaseChange is equal to clockSourcePhaseOffset (see 10.2.4.7).

10.2.2.3.13 lastGmFreqChange (Float64)

If the structure is sent by a PortSync entity, the lastGmFreqChange is the value of the lastGmFreqChange member of the MDSyncReceive structure whose receipt caused the sending of this structure (see 10.2.2.2.9). If the structure is sent by a ClockMaster entity, the lastGmFreqChange is equal to clockSourceFreqOffset (see 10.2.4.6).

10.2.3 Overview of global variables used by time synchronization state machines

The two subclauses that follow this subclause, i.e., 10.2.4 and 10.2.5, define global variables used by time synchronization state machines whose scopes are per PTP Instance, common across all PTP Instances (e.g., used by CMLDS, see 11.2.17), per port per PTP Instance, and per port but common across all PTP Instances. Table 10-1 summarizes the scope of each global variable of 10.2.4 and 10.2.5.

Table 10-1 — Summary of scope of global variables used by time synchronization state machines (see 10.2.4 and 10.2.5)

Variable name	Subclause of definition	Per PTP Instance (i.e., per domain)	Per PTP Instance, per PTP Port	Instance used by CMLDS (see 11.2.17), i.e., that is common across all LinkPorts	Instance used by CMLDS, per LinkPort
BEGIN	10.2.4.1	Yes	No	Yes	No
clockMasterSyncInterval	10.2.4.2	Yes	No	No	No
clockSlaveTime	10.2.4.3	Yes	No	No	No
syncReceiptTime	10.2.4.4	Yes	No	No	No
syncReceiptLocalTime	10.2.4.5	Yes	No	No	No
clockSourceFreqOffset	10.2.4.6	Yes	No	No	No
clockSourcePhaseOffset	10.2.4.7	Yes	No	No	No
clockSourceTimeBaseIndicator	10.2.4.8	Yes	No	No	No
clockSourceTimeBaseIndicatorOld	10.2.4.9	Yes	No	No	No
clockSourceLastGmPhaseChange	10.2.4.10	Yes	No	No	No
clockSourceLastGmFreqChange	10.2.4.11	Yes	No	No	No
currentTime	10.2.4.12	Yes	No	No	No
gmPresent	10.2.4.13	Yes	No	No	No
gmRateRatio	10.2.4.14	Yes	No	No	No
gmTimeBaseIndicator	10.2.4.15	Yes	No	No	No
lastGmPhaseChange	10.2.4.16	Yes	No	No	No
lastGmFreqChange	10.2.4.17	Yes	No	No	No
localClockTickInterval	10.2.4.18	Yes	No	No	No
localTime	10.2.4.19	Yes	No	No	No
selectedState	10.2.4.20	Yes	No	No	No
masterTime	10.2.4.21	Yes	No	No	No
thisClock	10.2.4.22	Yes	No	Yes	No
parentLogSyncInterval	10.2.4.23	Yes	No	No	No
instanceEnable	10.2.4.24	Yes	No	No	No
asCapable	10.2.5.1	No	Yes	No	No
asymmetryMeasurementMode	10.2.5.2	No	Yes[a]	No	Yes
syncReceiptTimeoutTimeInterval	10.2.5.3	No	Yes	No	No
currentLogSyncInterval	10.2.5.4	No	Yes	No	No
initialLogSyncInterval	10.2.5.5	No	Yes	No	No
syncInterval	10.2.5.6	No	Yes	No	No

Table 10-1 — Summary of scope of global variables used by time synchronization state machines (see 10.2.4 and 10.2.5)

Variable name	Subclause of definition	Per PTP Instance (i.e., per domain)	Per PTP Instance, per PTP Port	Instance used by CMLDS (see 11.2.17), i.e., that is common across all LinkPorts	Instance used by CMLDS, per LinkPort
neighborRateRatio	10.2.5.7	No	Yes[a]	No	Yes
meanLinkDelay	10.2.5.8	No	Yes[a]	No	Yes
delayAsymmetry	10.2.5.9	No	Yes[a]	No	Yes
computeNeighborRateRatio	10.2.5.10	No	Yes[a]	No	Yes
computeMeanLinkDelay	10.2.5.11	No	Yes[a]	No	Yes
portOper[b]	10.2.5.12	No	Yes	No	Yes
ptpPortEnabled	10.2.5.13	No	Yes	No	No
thisPort	10.2.5.14	No	Yes	No	Yes
syncLocked	10.2.5.15	No	Yes	No	No
neighborGptpCapable	10.2.5.16	No	Yes	No	No
syncSlowdown	10.2.5.17	No	Yes	No	No
oldSyncInterval	10.2.5.18	No	Yes	No	No
gPtpCapableMessageSlowdown	10.2.5.19	No	Yes	No	No
gPtpCapableMessageInterval	10.2.5.20	No	Yes	No	No
oldGptpCapableMessageInterval	10.2.5.21	No	Yes	No	No
currentLogGptpCapableMessageInterval	10.2.5.22	No	Yes	No	No
initialLogGptpCapableMessageInterval	10.2.5.23	No	Yes	No	No

[a]The instance of this variable that is per PTP Instance, per PTP Port exists only for domain 0.
[b]There is one instance of this variable per physical port, which is accessible by all PTP ports and LinkPorts associated with the physical port

10.2.4 Per PTP Instance global variables

10.2.4.1 BEGIN: a Boolean controlled by the system initialization. If BEGIN is true, all state machines, including per-port state machines, continuously execute their initial state. See Annex E of IEEE Std 802.1Q-2018.

10.2.4.2 clockMasterSyncInterval: a variable containing the mean time interval between successive messages providing time-synchronization information by the ClockMaster entity to the SiteSync entity. This value is given by $1000000000 \times 2^{\text{clockMasterLogSyncInterval}}$ ns, where clockMasterLogSyncInterval is the logarithm to base 2 of the mean time between the successive providing of time-synchronization information by the ClockMaster entity (see 10.7.2.4). The data type for clockMasterSyncInterval is UScaledNs.

10.2.4.3 clockSlaveTime: the synchronized time maintained, at the slave, at the granularity of the LocalClock entity [i.e., a new value is computed every localClockTickInterval (see 10.2.4.18) by the ClockSlave entity]. The data type for clockSlaveTime is ExtendedTimestamp.

10.2.4.4 syncReceiptTime: the synchronized time computed by the ClockSlave entity at the instant time-synchronization information, contained in a PortSyncSync structure, is received. The data type for syncReceiptTime is ExtendedTimestamp.

10.2.4.5 syncReceiptLocalTime: the value of currentTime (i.e., the time relative to the LocalClock entity) corresponding to syncReceiptTime. The data type for syncReceiptLocalTime is UScaledNs.

10.2.4.6 clockSourceFreqOffset: the fractional frequency offset of the ClockSource entity frequency relative to the current grandmaster frequency. The data type for clockSourceFreqOffset is Float64.

10.2.4.7 clockSourcePhaseOffset: the time provided by the ClockSource entity, minus the synchronized time. The data type for clockSourcePhaseOffset is ScaledNs.

10.2.4.8 clockSourceTimeBaseIndicator: a global variable that is set equal to the timeBaseIndicator parameter of the ClockSourceTime.invoke application interface function (see 9.2.2.3), by the ClockMaster entity. The parameter timeBaseIndicator of ClockSourceTime.invoke is set by the ClockSource entity and is changed by that entity whenever the time base changes. The data type for clockSourceTimeBaseIndicator is UInteger16.

10.2.4.9 clockSourceTimeBaseIndicatorOld: a global variable that is set equal to the previous value of clockSourceTimeBaseIndicator. The data type for clockSourceTimeBaseIndicatorOld is UInteger16.

10.2.4.10 clockSourceLastGmPhaseChange: a global variable that is set equal to the lastGmPhaseChange parameter of the ClockSourceTime.invoke application interface function (see 9.2.2.4). That parameter is set by the ClockSource entity and is changed by that entity whenever the time base changes. The data type for clockSourceLastGmPhaseChange is ScaledNs.

10.2.4.11 clockSourceLastGmFreqChange: a global variable that is set equal to the lastGmFreqChange parameter of the ClockSourceTime.invoke application interface function (see 9.2.2.5). That parameter is set by the ClockSource entity and is changed by that entity whenever the time base changes. The data type for clockSourceLastGmFreqChange is Float64.

10.2.4.12 currentTime: the current value of time relative to the LocalClock entity clock. The data type for currentTime is UScaledNs.

10.2.4.13 gmPresent: a Boolean that indicates whether a grandmaster-capable PTP Instance is present in the domain. If TRUE, a grandmaster-capable PTP Instance is present; if FALSE, a grandmaster-capable PTP Instance is not present.

10.2.4.14 gmRateRatio: the measured ratio of the frequency of the ClockSource entity to the frequency of the LocalClock entity. The data type for gmRateRatio is Float64.

10.2.4.15 gmTimeBaseIndicator: the most recent value of gmTimeBaseIndicator provided to the ClockSlaveSync state machine via a PortSyncSync structure. The data type for gmTimeBaseIndicator is UInteger16.

10.2.4.16 lastGmPhaseChange: the most recent value of lastGmPhaseChange provided to the ClockSlaveSync state machine via a PortSyncSync structure. The data type for lastGmPhaseChange is ScaledNs.

10.2.4.17 lastGmFreqChange: the most recent value of lastGmFreqChange provided to the ClockSlaveSync state machine via a PortSyncSync structure. The data type for lastGmFreqChange is Float64.

10.2.4.18 localClockTickInterval: the time interval between two successive significant instants (i.e., "ticks") of the LocalClock entity. The data type for localClockTickInterval is TimeInterval.

10.2.4.19 localTime: the value of currentTime when the most recent ClockSourceTime.invoke function (see 9.2) was received from the ClockSource entity, or when the LocalClock entity most recently updated its time. The data type for localTime is UScaledNs.

10.2.4.20 selectedState: an Enumeration2 array of length numberPorts+1 (see 8.6.2.8). selectedState[j] is set equal to the port state (see Table 10-2) of port whose portNumber is j.

10.2.4.21 masterTime: the time maintained by the ClockMaster entity, based on information received from the ClockSource and LocalClock entities. The data type for masterTime is ExtendedTimestamp.

10.2.4.22 thisClock: the clockIdentity of the current PTP Instance. The data type for thisClock is ClockIdentity.

10.2.4.23 parentLogSyncInterval: the most recent logMessageInterval value received on the slave port. If this PTP Instance is the grandmaster, then this is the clockMasterLogSyncInterval (see 10.7.2.4). The data type for parentLogSyncInterval is Integer8.

10.2.4.24 instanceEnable: a per-domain Boolean used to enable gPTP on all ports that are enabled for that domain (i.e., ports for which portOper and ptpPortEnabled are both TRUE). Setting instanceEnable to FALSE causes all per-domain state machines to go to the initial state.

NOTE—instanceEnable has no effect on the operation of the MDPdelayReq (see 11.2.19) and MDPdelayResp (see 11.2.20) state machines, because those state machines are not per domain (i.e., there is a single instance of each of those state machines, per link, for all domains).

10.2.5 Per-port global variables

10.2.5.1 asCapable: a Boolean that is TRUE if and only if it is determined that this PTP Instance and the PTP Instance at the other end of the link attached to this port can interoperate with each other via the IEEE 802.1AS protocol. This means that

a) this PTP Instance is capable of executing the IEEE 802.1AS protocol,
b) the PTP Instance at the other end of the link is capable of executing the IEEE 802.1AS protocol, and
c) there are no non-IEEE-802.1AS systems in between this PTP Instance and the PTP Instance at the other end of the link that introduce sufficient impairments that the end-to-end time-synchronization performance of B.3 cannot be met.

The determination of asCapable is different for each medium, and is described in the respective media-dependent clauses.

There is one instance of this variable per PTP Instance (i.e., per domain), per port.

NOTE—The per-port global variable asCapableAcrossDomains (see 11.2.13.12) is common across and accessible by all the domains. It is computed by the MDPdelayReq state machine (see 11.2.19). In the case of full-duplex, point-to-point links (see Clause 11), asCapableAcrossDomains is used when setting the instance of asCapable for each domain (for the link in question).

10.2.5.2 asymmetryMeasurementMode: a Boolean that contains the value of the managed object asymmetryMeasurementMode (see 14.8.46). For full-duplex IEEE 802.3 media, the value is TRUE if an asymmetry measurement is being performed for the link attached to this port, and FALSE otherwise. For all other media, the value is FALSE. There is one instance of this variable for all the domains, i.e. all the PTP Instances (per port). The variable is accessible by all the domains.

10.2.5.3 syncReceiptTimeoutTimeInterval: the time interval after which sync receipt timeout occurs if time-synchronization information has not been received during the interval. The value of syncReceiptTimeoutTimeInterval is equal to syncReceiptTimeout (see 10.7.3.1) multiplied by the syncInterval (see 10.2.5.6) for the port at the other end of the link to which this port is attached. The value of syncInterval for the port at the other end of the link is computed from logMessageInterval of the received Sync message (see 10.6.2.2.14). The data type for syncReceiptTimeoutTimeInterval is UScaledNs.

10.2.5.4 currentLogSyncInterval: the current value of the logarithm to base 2 of the mean time interval, in seconds, between the sending of successive time-synchronization event messages (see 10.7.2.3). This value is set in the SyncIntervalSetting state machine (see 10.3.18). The data type for currentLogSyncInterval is Integer8.

10.2.5.5 initialLogSyncInterval: the initial value of the logarithm to base 2 of the mean time interval, in seconds, between the sending of successive time-synchronization event messages (see 10.7.2.3). The data type for initialLogSyncInterval is Integer8.

10.2.5.6 syncInterval: a variable containing the mean time-synchronization event message transmission interval for the port. This value is set in the SyncIntervalSetting state machine (see 10.3.18). The data type for syncInterval is UScaledNs.

10.2.5.7 neighborRateRatio: the measured ratio of the frequency of the LocalClock entity of the time-aware system at the other end of the link attached to this port, to the frequency of the LocalClock entity of this time-aware system. The data type for neighborRateRatio is Float64. There is one instance of this variable for all the domains, i.e., all the PTP Instances (per port). The variable is accessible by all the domains.

10.2.5.8 meanLinkDelay: the measured mean propagation delay (see 8.3) on the link attached to this port, relative to the LocalClock entity of the time-aware system at the other end of the link (i.e., expressed in the time base of the time-aware system at the other end of the link). The data type for meanLinkDelay is UScaledNs. There is one instance of this variable for all the domains, i.e., all the PTP Instances (per port). The variable is accessible by all the domains.

NOTE—The variable meanLinkDelay was named neighborPropDelay in the 2011 edition of this standard.

10.2.5.9 delayAsymmetry: the asymmetry in the propagation delay on the link attached to this port.If propagation delay asymmetry is not modeled, then delayAsymmetry is zero. The data type for delayAsymmetry is ScaledNs. There is one instance of this variable for CMLDS (see 11.2.17), and also one instance of this variable for each domain that uses the instance-specific peer-to-peer delay mechanism. The instance of this variable for CMLDS is relative to the local clock. The instance of this variable for a domain that uses the instance-specific peer-to-peer delay mechanism is relative to the grandmaser time base.

10.2.5.10 computeNeighborRateRatio: a Boolean, set by the LinkDelayIntervalSetting state machine (see 11.2.21), that indicates whether neighborRateRatio is to be computed by this port. There is one instance of this variable for all the domains, i.e., all the PTP Instances (per port). The variable is accessible by all the domains.

10.2.5.11 computeMeanLinkDelay: a Boolean, set by the LinkDelayIntervalSetting state machine (see 11.2.21), that indicates whether meanLinkDelay is to be computed by this port. There is one instance of this variable for all the domains, i.e., all the PTP Instances (per port). The variable is accessible by all the domains.

10.2.5.12 portOper: a Boolean that is TRUE if and only if the port is up and able to send and receive messages. There is one instance of this variable for all the domains, i.e., all the PTP Instances (per port). The variable is accessible by all the domains. The term *port* in this definition is a physical port.

NOTE 1—portOper is an indicator, and not a control, and reflects the operational status of the underlying medium. It is not administratively set by gPTP.

NOTE 2—The variable portOper corresponds to the variable portEnabled in the previous edition of this standard. The change is reflected in many state machines.

NOTE 3—portOper is the same as MAC_Operational (see IEEE Std 802.1AC).

10.2.5.13 ptpPortEnabled: a Boolean that is administratively set to TRUE if time-synchronization is to be enabled on this port.

NOTE 1—It is expected that the value of ptpPortEnabled will be set via the management interface (see 14.8.4). A port can be enabled for data transport but not for synchronization transport.

NOTE 2—The variable ptpPortEnabled was named pttPortEnabled in the previous edition of this standard. This is only a name change; the definition and function of this variable is the same as in the previous edition of this standard. The name change is reflected in many state machines.

10.2.5.14 thisPort: the portNumber of the current port. The data type for thisPort is UInteger16.

10.2.5.15 syncLocked: a Boolean, set by the PortSyncSyncSend state machine (see 10.2.12.3), that indicates that this port, when operating as a Master port, shall transmit a Sync as soon as possible after the Slave port received a Sync (ignoring syncInterval). If FALSE, the port shall use the timing set by syncInterval.

10.2.5.16 neighborGptpCapable: a Boolean, set by the GptpCapableReceive state machine (see 10.4.2), that indicates that the neighbor of this port (i.e, the port at the other end of the link attached to this port) is capable of invoking the gPTP protocol.

10.2.5.17 syncSlowdown: a Boolean that is set to TRUE if the SyncIntervalSetting state machine (see Figure 10-20) receives a TLV that requests a larger sync interval (see 10.7.2.3), and FALSE otherwise. When syncSlowdown is set to TRUE, the PortSyncSyncSend state machine (see Figure 10-8) continues to send time synchronization event messages (see 11.4.3, 12.1 and 12.2, and 13.3.1) at the old (i.e., faster) rate until the number of time synchronization event messages equal to syncReceiptTimeout (see 10.7.3.1) have been sent, but with the respective time synchronization event message transmission interval field (see 11.4.2.8, 12.7 and Figure 12-8, and 13.3.1.2.10) of the time synchronization event message set equal to the new sync interval (i.e., corresponding to the slower rate). After syncReceiptTimeout Sync messages have been sent, subsequent time synchronization event messages are sent at the new (i.e., slower) rate, and with the respective time synchronization event message transmission interval field of the time synchronization event message set to the new sync interval. When syncSlowdown is set to FALSE, the PortSyncSyncSend state machine immediately sends time synchronization event messages at the new (i.e., slower) rate.

NOTE—If a receiver of time synchronization event messages (see 11.4.3, 12.1 and 12.2, and 13.3.1) requests a slower rate, the receiver will continue to use the upstream sync interval value, which it obtains from the respective time synchronization event message transmission interval field (see 11.4.2.8, 12.7 and Figure 12-8, and 13.3.1.2.10) of the received time synchronization event message, until it receives a time synchronzation event message where that value has changed. If, immediately after requesting a slower time synchronization event message rate, up to syncReceiptTimeout consecutive time synchronization event messages sent to the receiver are lost, sync receipt timeout could occur if the sender had changed to the slower rate immediately. Delaying the slowing down of the sending rate of time synchronization event messages for syncReceiptTimeout messages prevents this from happening.

10.2.5.18 oldSyncInterval: the saved value of the previous sync interval, when a new time-synchronization event message transmission interval is requested via a Signaling message that contains a message interval request TLV. The data type for oldSyncInterval is UScaledNs.

10.2.5.19 gPtpCapableMessageSlowdown: a Boolean that is set to TRUE if the GptpCapableIntervalSetting state machine (see Figure 10-19) receives a TLV that requests a larger gPTP capable message interval (see 10.7.2.5), and FALSE otherwise. When gPtpCapableMessageSlowdown is set to TRUE, the GptpCapableTransmit state machine (see Figure 10-21) continues to send Signaling messages containing the gPTP capable TLV at the old (i.e., faster) rate until a number of Signaling messages containing the gPTP capable TLV, equal to gPtpCapableReceiptTimeout (see 10.7.3.3), have been sent, but with the logGptpCapableMessageIntervalField of the gPTP capable TLV (see 10.6.4.5.6) set equal to the new gPTP capable message interval (i.e., corresponding to the slower rate). After gPtpCapableReceiptTimeout Signaling messages containing the gPTP capable TLV have been sent, subsequent such Signaling messages are sent at the new (i.e., slower) rate, and with the logGptpCapableMessageInterval field of the gPTP capable TLV set to the new gPTP capable message interval. When gPtpCapableSlowdown is set to FALSE, the GptpCapableTransmit state machine immediately sends Signaling messages containing the gPTP capable TLV at the new (i.e., slower) rate.

NOTE—If a receiver of Signaling messages containing the gPTP capable TLV requests a slower rate, the receiver will continue to use the old gPTP capable message interval value in determining, via the GptpCapableReceive state machine (see 10.4.2), if its neighbor is no longer capable of invoking gPTP, until it has received gPtpCapableReceiptTimeout such Signaling messages. If, immediately after requesting a slower rate, up to gPtpCapableReceiptTimeout consecutive Signaling messages, containing the gPTP capable TLV, sent to the receiver are lost, a declaration that the sender is no longer capable of invoking gPTP could occur if the sender had changed to the slower rate immediately. Delaying the slowing down of the sending rate of Signaling messages containing the gPTP capable TLV for gPtpCapableReceiptTimeout messages prevents this from happening.

10.2.5.20 gPtpCapableMessageInterval: a variable whose value is the mean time, in seconds, between successive Signaling messages that carry the gPTP capable TLV (see 10.7.2.5 and 10.6.4.4). The data type for gPtpCapableMessageInterval is UScaledNs.

10.2.5.21 oldGptpCapableMessageInterval: the saved value of the previous interval between successive Signaling messages that carry the gPTP capable TLV(see 10.2.5.20), when a new such interval is requested via a Signaling message that contains a gPTP capable message interval request TLV. The data type for oldGptpCapableMessageInterval is UScaledNs.

10.2.5.22 currentLogGptpCapableMessageInterval: the current value of the logarithm to base 2 of the mean time interval, in seconds, between the sending of successive Signaling messages that carry the gPTP capable TLV(see 10.6.4.4). This value is set in the GptpCapableIntervalSetting state machine (see 10.4.3). The data type for currentGptpCapableMessageInterval is Integer8.

10.2.5.23 initialLogGptpCapableMessageInterval: the initial value of the logarithm to base 2 of the mean time interval, in seconds, between the sending of successive Signaling messages that carry the gPTP capable TLV (see 10.6.4.4). The data type for initialLogGptpCapableMessageInterval is Integer8.

10.2.6 Functions used by multiple state machines

10.2.6.1 random(): returns a uniformly-distributed pseudo-random number whose data type is UInteger16 (i.e., the function returns a uniformly distributed, pseudo-random integer in the range $[0, 2^{16} – 1]$).

10.2.7 SiteSyncSync state machine

10.2.7.1 State machine variables

The following variables are used in the state diagram of 10.2.7.3:

10.2.7.1.1 rcvdPSSyncSSS: a Boolean variable that notifies the current state machine when a PortSyncSync structure (see 10.2.2.3) is received from the PortSyncSyncReceive state machine of a PortSync entity or from the ClockMasterSyncSend state machine of the ClockMaster entity. This variable is reset by this state machine.

10.2.7.1.2 rcvdPSSyncPtrSSS: a pointer to the received PortSyncSync structure indicated by rcvdPSSyncSSS.

10.2.7.1.3 txPSSyncPtrSSS: a pointer to the PortSyncSync structure transmitted by the state machine.

10.2.7.2 State machine functions

10.2.7.2.1 setPSSyncSend (rcvdPSSyncPtrSSS): creates a PortSyncSync structure to be transmitted, and returns a pointer to this structure. The members are copied from the received PortSyncSync structure pointed to by rcvdPSSyncPtrSSS.

10.2.7.2.2 txPSSync (txPSSyncPtrSSS): transmits a copy of the PortSyncSync structure pointed to by txPSSyncPtrSSS to the PortSyncSyncSend state machine of each PortSync entity and the ClockSlaveSync state machine of the ClockSlave entity of this PTP Instance.

10.2.7.3 State diagram

The SiteSyncSync state machine shall implement the function specified by the state diagram in Figure 10-3, the local variables specified in 10.2.7.1, the functions specified in 10.2.7.2, the structure specified in 10.2.2.3, and the relevant global variables and functions specified in 10.2.4 through 10.2.6. The state machine receives time-synchronization information, accumulated rateRatio, and syncReceiptTimeoutTime from the PortSync entity (PortSyncSyncReceive state machine) of the current slave port or from the ClockMaster entity (ClockMasterSyncSend state machine). If the information was sent by a PortSync entity, the state machine also receives the portIdentity of the port on the upstream PTP Instance that sent the information to this PTP Instance (if the information was sent by the ClockMaster entity, the portIdentity is that of the ClockMaster entity, i.e., it has clockIdentity equal to the clockIdentity of this PTP Instance and portNumber 0). The state machine sends a PortSyncSync structure to the PortSync entities of all the ports and to the ClockSlave entity.

10.2.8 PortSyncSyncReceive state machine

10.2.8.1 State machine variables

The following variables are used in the state diagram of 10.2.8.3:

10.2.8.1.1 rcvdMDSyncPSSR: a Boolean variable that notifies the current state machine when an MDSyncReceive structure is received from an MD entity of the same port (see 10.2.2.1). This variable is reset by this state machine.

10.2.8.1.2 rcvdMDSyncPtrPSSR: a pointer to the received MDSyncReceive structure indicated by rcvdMDSyncPSSR.

10.2.8.1.3 txPSSyncPtrPSSR: a pointer to the PortSyncSync structure transmitted by the state machine.

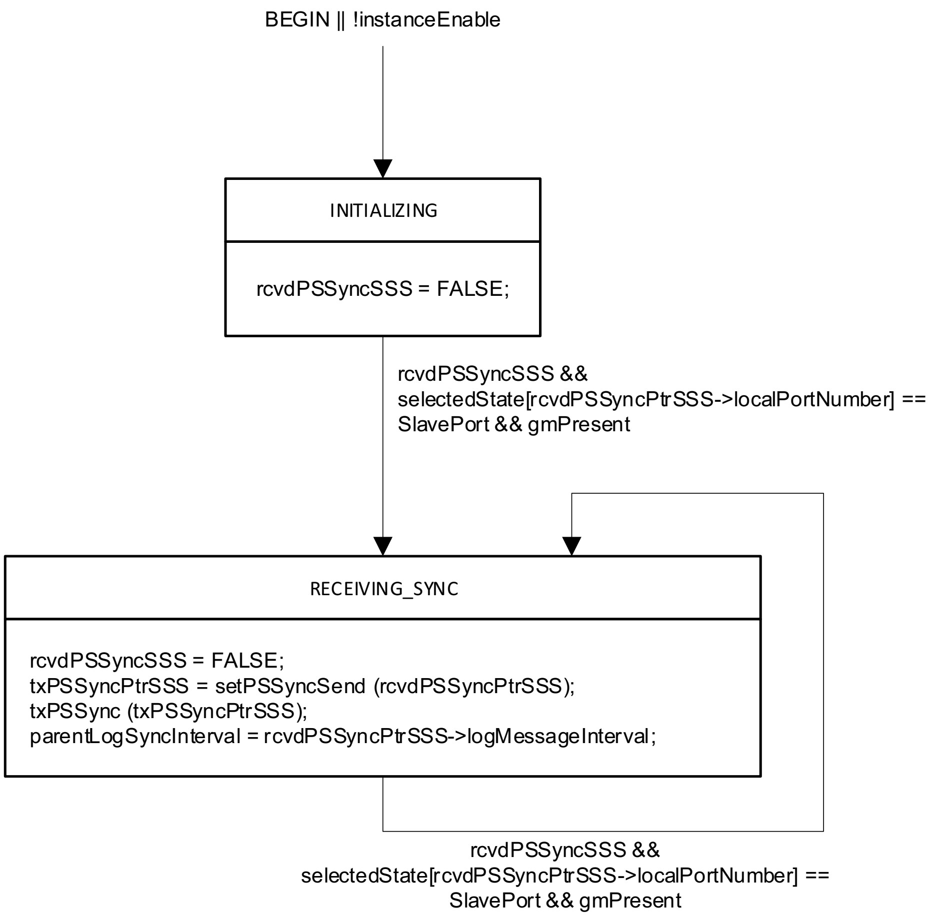


Figure 10-3—SiteSyncSync state machine

10.2.8.1.4 rateRatio: a Float64 variable that holds the ratio of the frequency of the grandmaster to the frequency of the LocalClock entity. This frequency ratio is computed by (a) measuring the ratio of the grandmaster frequency to the LocalClock frequency at the grandmaster PTP Instance and initializing rateRatio to this value in the ClockMasterSend state machine of the grandmaster PTP Instance, and (b) accumulating, in the PortSyncSyncReceive state machine of each PTP Instance, the frequency offset of the LocalClock entity of the PTP Instance at the remote end of the link attached to that port to the frequency of the LocalClock entity of this PTP Instance.

10.2.8.2 State machine functions

10.2.8.2.1 setPSSyncPSSR (rcvdMDSyncPtrPSSR syncReceiptTimeoutTimeInterval, rateRatio): creates a PortSyncSync structure to be transmitted, and returns a pointer to this structure. The members are set as follows:

a) localPortNumber is set equal to thisPort,

b) domainNumber, followUpCorrectionField, sourcePortIdentity, logMessageInterval,preciseOriginTimestamp, and upstreamTxTime are copied from the received MDSyncReceive structure pointed to by rcvdMDSyncPtrPSSR,
c) syncReceiptTimeoutTime is set equal to currentTime plus syncReceiptTimeoutTimeInterval (see 10.2.5.3), and
d) the function argument rateRatio is set equal to the local variable rateRatio (computed just prior to invoking setPSSyncPSSR (see Figure 10-4)). The rateRatio member of the PortSyncSync structure is then set equal to the function argument rateRatio.

10.2.8.2.2 txPSSyncPSSR (txPSSyncPtrPSSR): transmits a copy of the PortSyncSync structure pointed to by txPSSyncPtrPSSR to the SiteSyncSync state machine of this PTP Instance.

10.2.8.3 State diagram

The PortSyncSyncReceive state machine shall implement the function specified by the state diagram in Figure 10-4, the local variables specified in 10.2.8.1, the functions specified in 10.2.8.2, the structures specified in 10.2.2.1 and 10.2.2.3, and the relevant global variables and functions specified in 10.2.4 through 10.2.6. The state machine receives time-synchronization information, accumulated rateRatio, and syncReceiptTimeoutTime from the MD entity of the same port. The state machine adds, to rateRatio, the fractional frequency offset of the LocalClock entity relative to the LocalClock entity of the upstream PTP Instance at the remote end of the link attached to this port. The state machine computes syncReceiptTimeoutTime. The state machine sends this information to the SiteSync entity (SiteSyncSync state machine).

10.2.9 ClockMasterSyncSend state machine

10.2.9.1 State machine variables

The following variables are used in the state diagram of 10.2.9.3:

10.2.9.1.1 syncSendTime: the time in seconds, relative to the LocalClock entity, when synchronization information will next be sent to the SiteSync entity, via a PortSyncSync structure. The data type for syncSendTime is UScaledNs.

10.2.9.1.2 txPSSyncPtrCMSS: a pointer to the PortSyncSync structure transmitted by the state machine.

10.2.9.2 State machine functions

10.2.9.2.1 setPSSyncCMSS (gmRateRatio): creates a PortSyncSync structure to be transmitted, and returns a pointer to this structure. The members are set as follows:
a) localPortNumber is set to 0
b) preciseOriginTimestamp is set equal to the masterTime, with any fractional nanoseconds truncated
c) followUpCorrectionField is set equal to the sum of
   1) the fractional nanoseconds portion of masterTime.fractionalNanoseconds,
   2) the quantity gmRateRatio × (currentTime – localTime)
d) the clockIdentity member of sourcePortIdentity is set equal to the clockIdentity of this PTP Instance
e) the portNumber member of the sourcePortIdentity is set to 0

NOTE—This quantity and localPortNumber are redundant; both are retained so that the SiteSync entity can process PortSyncSync structures received from a PortSync entity or the ClockMaster entity in the same manner.

f) logMessageInterval is set to clockMasterLogSyncInterval
g) upstreamTxTime is set equal to localTime

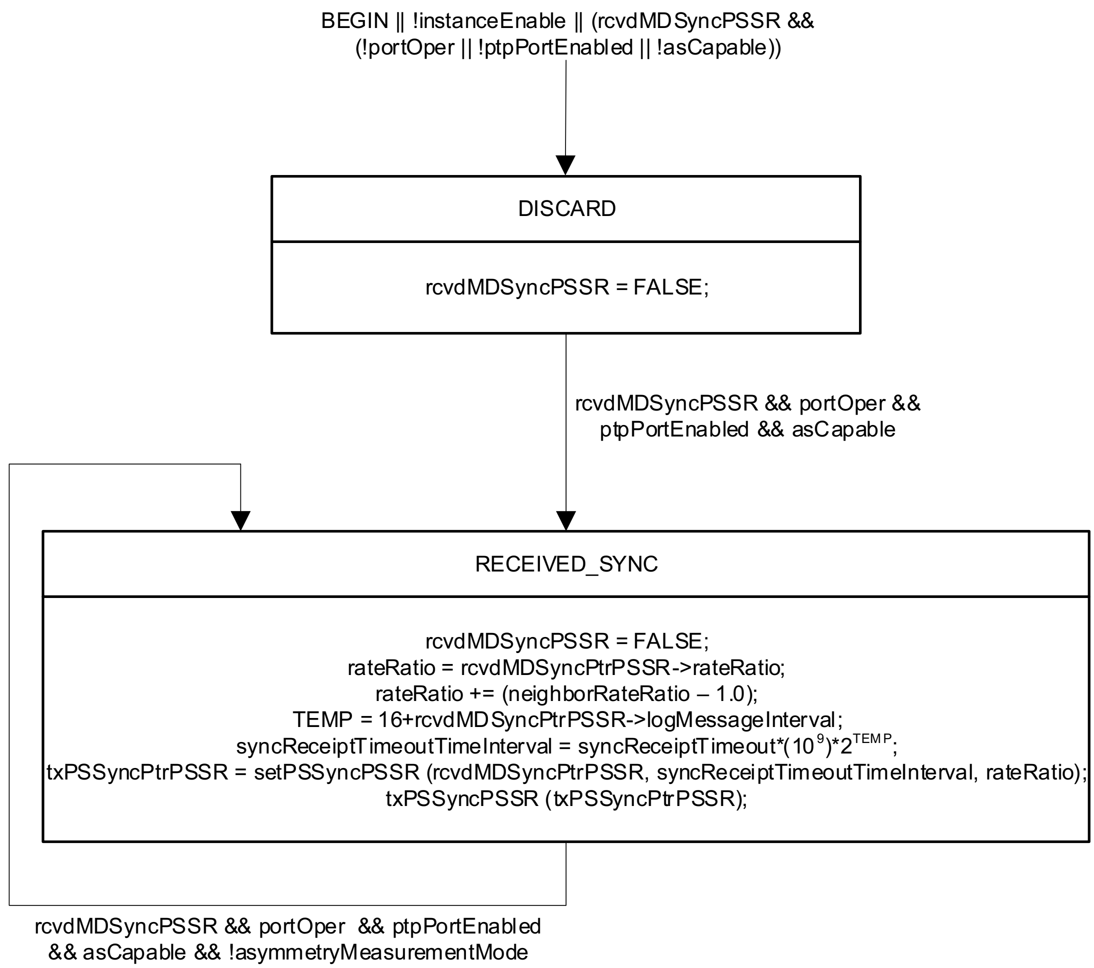


Figure 10-4—PortSyncSyncReceive state machine

h) syncReceiptTimeoutTime is set equal to $FFFFFFFFFFFFFFFF_{16}$, which indicates that there is no sync receipt timeout.

NOTE—A ClockMaster entity does not receive Sync messages, and there is no notion of sync receipt timeout.

i) rateRatio is set equal to gmRateRatio
j) gmTimeBaseIndicator is set equal to clockSourceTimeBaseIndicator
k) lastGmPhaseChange is set equal to clockSourcePhaseOffset
l) lastGmFreqChange is set equal to clockSourceFreqOffset, and
m) domainNumber is set equal to the domain number of this gPTP domain.

10.2.9.2.2 txPSSyncCMSS (txPSSyncPtrCMSS): transmits a copy of the PortSyncSync structure pointed to by txPSSyncPtrCMSS to the SiteSync state machine.

10.2.9.2.3 computeClockMasterSyncInterval(): computes the value of clockMasterSyncInterval (see 10.2.4.2) as $1000000000 \times 2^{\mathrm{clockMasterLogSyncInterval}}$ ns, where clockMasterLogSyncInterval is the minimum currentLogSyncInterval value, taken over all the gPTP Ports of the PTP Instance (see 10.7.2.4).

10.2.9.3 State diagram

The ClockMasterSyncSend state machine shall implement the function specified by the state diagram in Figure 10-5, the local variables specified in 10.2.9.1, the functions specified in 10.2.9.2, the structure specified in 10.2.2.3, and the relevant global variables and functions specified in 10.2.4 through 10.2.6. The state machine receives masterTime and clockSourceTimeBaseIndicator from the ClockMasterSyncReceive state machine, and phase and frequency offset between masterTime and syncReceiptTime from the ClockMasterSyncOffset state machine. It provides masterTime (i.e., synchronized time) and the phase and frequency offset to the SiteSync entity via a PortSyncSync structure.

The ClockMasterSyncSend state machine is optional for PTP Instances that are not grandmaster-capable (see 8.6.2.1, 10.1.2, and 10.2.1). This state machine may be present in a PTP Instance that is not grandmaster-capable; however, any information supplied by it to the SiteSyncSync state machine is not used by the SiteSyncSync state machine if the PTP Instance is not grandmaster-capable.

10.2.10 ClockMasterSyncOffset state machine

10.2.10.1 State machine variables

The following variables are used in the state diagram of 10.2.10.3:

10.2.10.1.1 rcvdSyncReceiptTime: a Boolean variable that notifies the current state machine that syncReceiptTime has been updated by the ClockSlave entity. This variable is reset by this state machine.

10.2.10.2 State machine functions

10.2.10.2.1 computeClockSourceFreqOffset(): computes and returns clockSourceFreqOffset (see 10.2.4.6), using successive values of masterTime computed by the ClockMasterSyncReceive state machine (see 10.2.11) and successive values of syncReceiptTime computed by the ClockSlaveSync state machine (see 10.2.13). The data type for the returned value is Float64. Any scheme that uses this information to compute clockSourceFreqOffset is acceptable as long as the performance requirements specified in B.2.4 are met.

NOTE—As one example, clockSourceFreqOffset can be estimated as the ratio of the duration of a time interval measured by the ClockSource entity to the duration of the same time interval computed from ClockSlaveTime values, minus 1.

10.2.10.3 State diagram

The ClockMasterSyncOffset state machine shall implement the function specified by the state diagram in Figure 10-6, the local variables specified in 10.2.10.1, and the relevant global variables and functions specified in 10.2.4 through 10.2.6. The state machine receives syncReceiptTime from the ClockSlaveSync state machine and masterTime from the ClockMasterSyncReceive state machine. It computes clockSourcePhaseOffset and clockSourceFrequency offset if this PTP Instance is not currently the grandmaster, i.e., if selectedState[0] is equal to PassivePort.

The ClockMasterSyncOffset state machine is optional for PTP Instances that are not grandmaster-capable (see 8.6.2.1, 10.1.2, and 10.2.1). This state machine may be present in a PTP Instance that is not

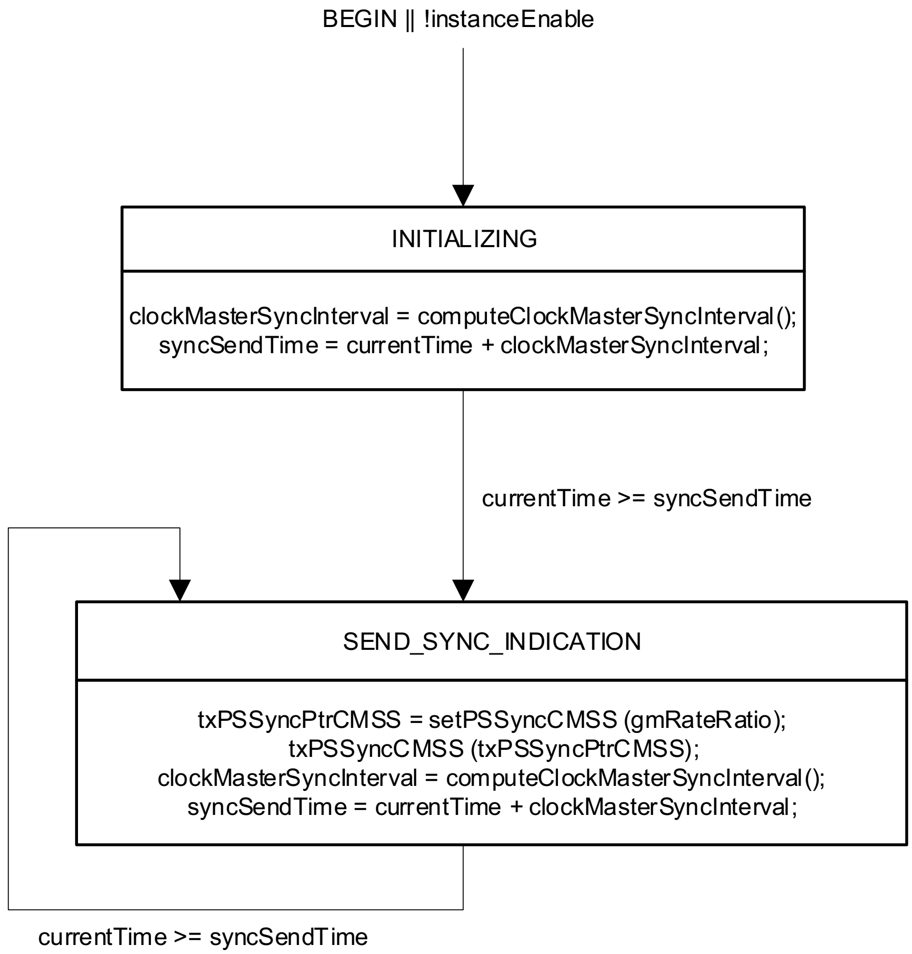


Figure 10-5—ClockMasterSyncSend state machine

grandmaster-capable; however, any information supplied by it, via the ClockMasterSyncSend state machine, to the SiteSyncSync state machine is not used by the SiteSyncSync state machine if the PTP Instance is not grandmaster-capable.

10.2.11 ClockMasterSyncReceive state machine

10.2.11.1 State machine variables

The following variables are used in the state diagram of 10.2.11.3:

10.2.11.1.1 rcvdClockSourceReq: a Boolean variable that notifies the current state machine when sourceTime is received from the ClockSource entity, due to the ClockSourceTime.invoke primitive having been invoked at that entity. This variable is reset by this state machine.

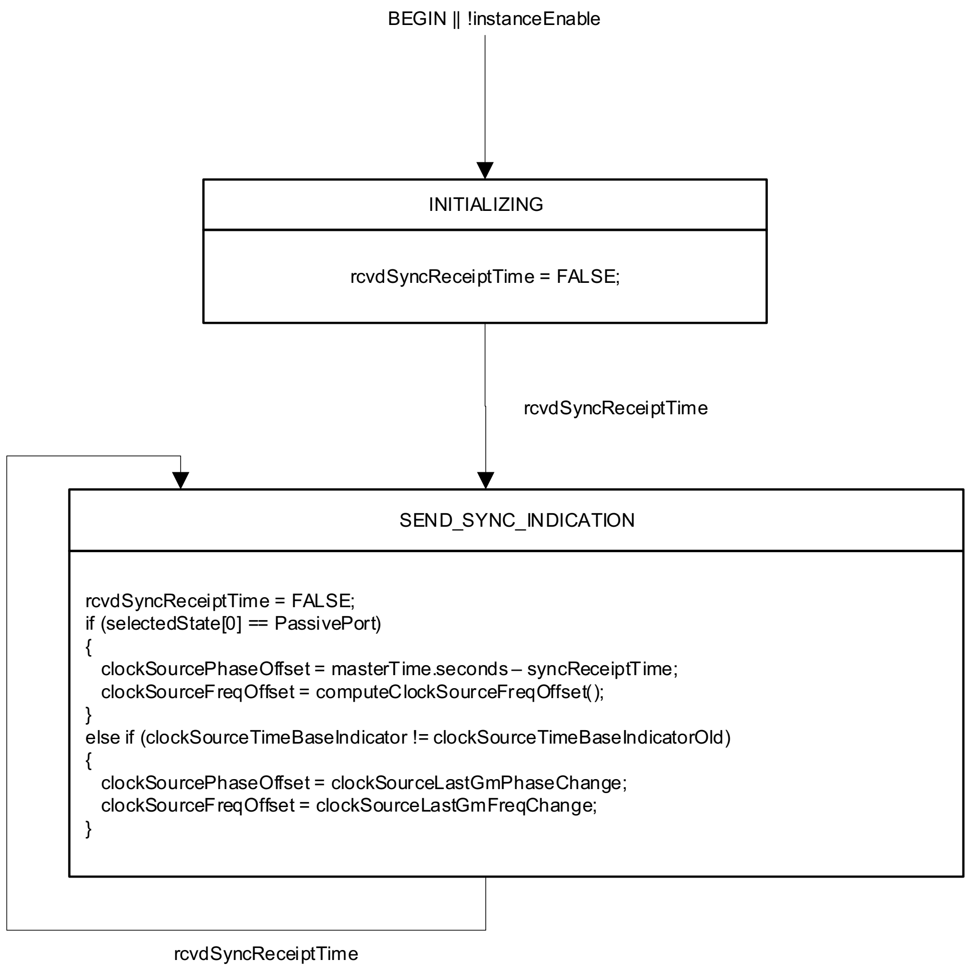


Figure 10-6—ClockMasterSyncOffset state machine

10.2.11.1.2 rcvdClockSourceReqPtr: a pointer to the received ClockSourceTime.invoke function parameters.

10.2.11.1.3 rcvdLocalClockTickCMSR: a Boolean variable that notifies the current state machine when the LocalClock entity updates its time. This variable is reset by this state machine.

10.2.11.2 State machine functions

10.2.11.2.1 computeGmRateRatio(): computes gmRateRatio(see 10.2.4.14), using values of sourceTime conveyed by successive ClockSourceTime.invoke functions (see 9.2.2.1), and corresponding values of localTime (see 10.2.4.19). Any scheme that uses this information, along with any other information conveyed by the successive ClockSourceTime.invoke functions and corresponding values of localTime, to compute gmRateRatio is acceptable as long as the performance requirements specified in B.2.4 are met.

NOTE—As one example, gmRateRatio can be estimated as the ratio of the elapsed time of the ClockSource entity that supplies time to this PTP Instance, to the elapsed time of the LocalClock entity of this PTP Instance. This ratio can be computed for the time interval between a received ClockSourceTime.invoke function and a second received ClockSourceTime.invoke function some number of ClockSourceTime.invoke functions later, i.e.,

$$\frac{\text{ClockSource.invoke.sourceTime}_N - \text{ClockSource.invoke.sourceTime}_0}{\text{localTime}_N - \text{localTime}_0}$$

where the successive received ClockSourceTime.invoke functions are indexed from 0 to *N,* with the first such function indexed as 0, and localTime_j is the value of localTime when the ClockSourceTime.invoke function whose index is *j* is received.

10.2.11.2.2 updateMasterTime(): updates the global variable masterTime (see 10.2.4.21), based on information received from the ClockSource and LocalClock entities. It is the responsibility of the application to filter master times appropriately. As one example, masterTime can be set equal to the sourceTime member of the ClockSourceTime.invoke function when this function is invoked at the ClockSource entity, and can be incremented by localClockTickInterval (see 10.2.4.18) multiplied by gmRateRatio (see 10.2.4.14) when rcvdLocalClockTickCMSR is TRUE.

10.2.11.3 State diagram

The ClockMasterSyncReceive state machine shall implement the function specified by the state diagram in Figure 10-7, the local variables specified in 10.2.11.1, and the relevant global variables and functions specified in 10.2.4 through 10.2.6. The state machine updates the global variable masterTime with information received from the ClockSource entity via the ClockSourceTime.invoke function and information received from the LocalClock entity. It also computes gmRateRatio, i.e., the ratio of the ClockSource entity frequency and the LocalClock entity frequency.

The ClockMasterSyncReceive state machine is optional for PTP Instances that are not grandmaster-capable (see 8.6.2.1, 10.1.2, and 10.2.1). This state machine may be present in a PTP Instance that is not grandmaster-capable; however, any information supplied by it, via the ClockMasterSyncSend state machine, to the SiteSyncSync state machine is not used by the SiteSyncSync state machine if the PTP Instance is not grandmaster-capable.

10.2.12 PortSyncSyncSend state machine

10.2.12.1 State machine variables

The following variables are used in the state diagram of 10.2.12.3:

10.2.12.1.1 rcvdPSSyncPSSS: a Boolean variable that notifies the current state machine when a PortSyncSync structure is received from the SiteSyncSync state machine of the SiteSync entity of the PTP Instance (see 10.2.2.3). This variable is reset by this state machine.

10.2.12.1.2 rcvdPSSyncPtrPSSS: a pointer to the received PortSyncSync structure indicated by rcvdPSSyncPSSS.

10.2.12.1.3 lastPreciseOriginTimestamp: the preciseOriginTimestamp member of the most recently received PortSyncSync structure. The data type for lastPreciseOriginTimestamp is Timestamp.

10.2.12.1.4 lastFollowUpCorrectionField: the followUpCorrectionField member of the most recently received PortSyncSync structure. The data type for lastFollowUpCorrectionField is ScaledNs.

10.2.12.1.5 lastRateRatio: the rateRatio member of the most recently received PortSyncSync structure. The data type for lastRateRatio is Float64.

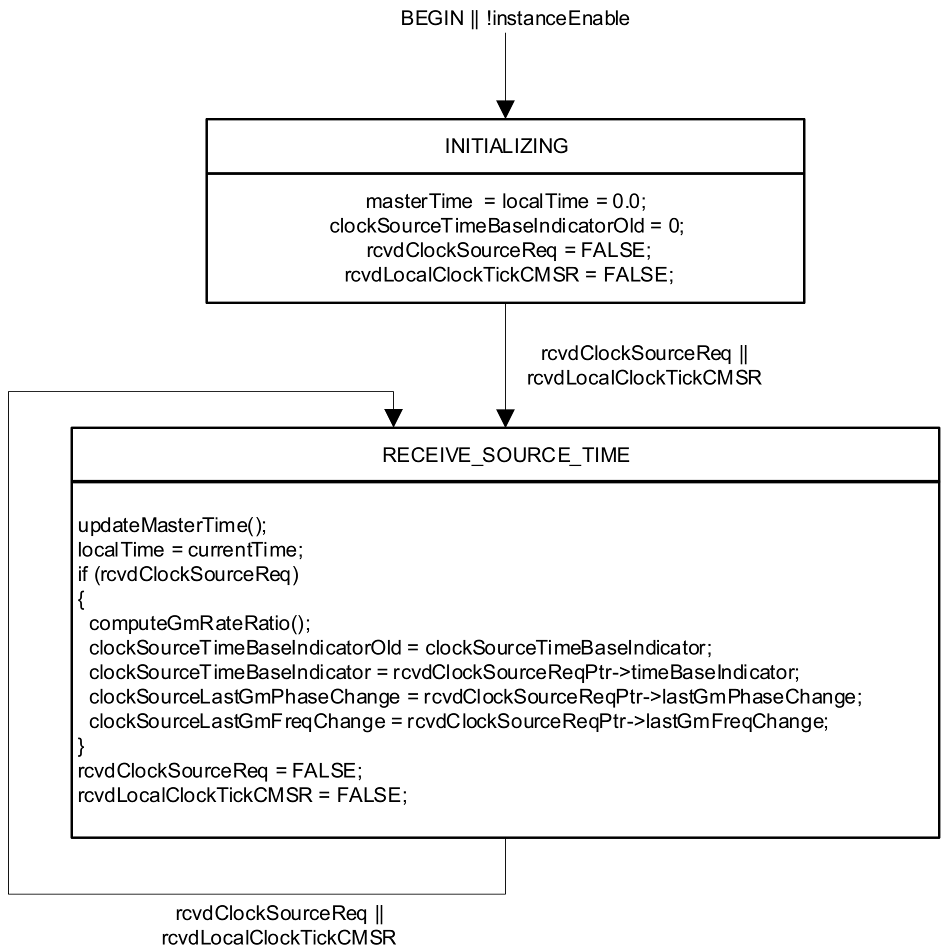


Figure 10-7—ClockMasterSyncReceive state machine

10.2.12.1.6 lastUpstreamTxTime: the upstreamTxTime member of the most recently received PortSyncSync structure. The data type for lastUpstreamTxTime is UScaledNs.

10.2.12.1.7 lastSyncSentTime: the value of currentTime (i.e., the time relative to the LocalClock entity) when the most recent MDSyncSend structure was sent. The data type for lastSyncSentTime is UScaledNs.

NOTE—lastSyncSentTime is the time the abstract MDSyncSend structure was sent, NOT the time the corresponding Sync message (or equivalent) was sent on a physical link.

10.2.12.1.8 lastRcvdPortNum: the portNumber of the port on which time-synchronization information was most recently received. The data type for lastRcvdPortNum is UInteger16.

10.2.12.1.9 lastGmTimeBaseIndicator: the gmTimeBaseIndicator of the most recently received PortSyncSync structure. The data type for lastGmTimeBaseIndicator is UInteger16.

10.2.12.1.10 lastGmPhaseChangePSSS: the lastGmPhaseChange of the most recently received PortSyncSync structure. The data type for lastGmPhaseChange is ScaledNs.

10.2.12.1.11 lastGmFreqChangePSSS: the lastGmFreqChange of the most recently received PortSyncSync structure. The data type for lastGmPhaseChange is Float64.

10.2.12.1.12 txMDSyncPtr: a pointer to the MDSyncSend structure sent to the MD entity of this port.

10.2.12.1.13 syncReceiptTimeoutTime: the value of the syncReceiptTimeoutTime member of the most recently received PortSyncSync structure. The data type for syncReceiptTimeoutTime is UScaledNs.

10.2.12.1.14 numberSyncTransmissions: a count of the number of consecutive Sync message transmissions after the SyncIntervalSetting state machine (see Figure 10-20) has set syncSlowdown (see 10.2.5.17) to TRUE. The data type for numberSyncTransmissions is UInteger8.

10.2.12.1.15 interval1: a local variable that holds either syncInterval or oldSyncInterval. The data type for interval1 is UScaledNs.

10.2.12.2 State machine functions

10.2.12.2.1 setMDSync(): creates an MDSyncSend structure, and returns a pointer to this structure. The members are set as follows:

a) sourcePortIdentity is set to the portIdentity of this port(see 8.5.2)
b) logMessageInterval is set equal to the value of currentLogSyncInterval for this port (see 10.7.2.3),
c) preciseOriginTimestamp is set equal to lastPreciseOriginTimestamp (see 10.2.12.1.3),
d) rateRatio is set equal to lastRateRatio (see 10.2.12.1.5),
e) followUpCorrectionField is set equal to lastFollowUpCorrectionField (see 10.2.12.1.4),
f) upstreamTxTime is set equal to lastUpstreamTxTime (see 10.2.12.1.6),
g) gmTimeBaseIndicator is set to lastGmTimeBaseIndicator (see 10.2.12.1.9),
h) lastGmPhaseChange (structure member) is set to lastGmPhaseChangePSSS (see 10.2.12.1.10),
i) lastGmFreqChange (structure member) is set to lastGmFreqChangePSSS (see 10.2.12.1.11), and
j) domainNumber is set equal to the domain number of this gPTP domain (see 8.1).

10.2.12.2.2 txMDSync(txMDSyncPtr): transmits the MDSyncSend structure pointed to by txMDSyncPtr, to the MD entity of this port.

10.2.12.3 State diagram

The PortSyncSyncSend state machine shall implement the function specified by the state diagram in Figure 10-8, the local variables specified in 10.2.12.1, the functions specified in 10.2.12.2, the structures specified in 10.2.2.1 through 10.2.2.3, and the relevant global variables and functions specified in 10.2.4 through 10.2.6. The state machine receives time-synchronization information from the SiteSyncSync state machine, corresponding to the receipt of the most recent synchronization information on either the slave port, if this PTP Instance is not the grandmaster, or from the ClockMasterSyncSend state machine, if this PTP Instance is the grandmaster. The state machine causes time-synchronization information to be sent to the MD entity if this port is a MasterPort.

10.2.13 ClockSlaveSync state machine

10.2.13.1 State machine variables

The following variables are used in the state diagram of 10.2.13.3:

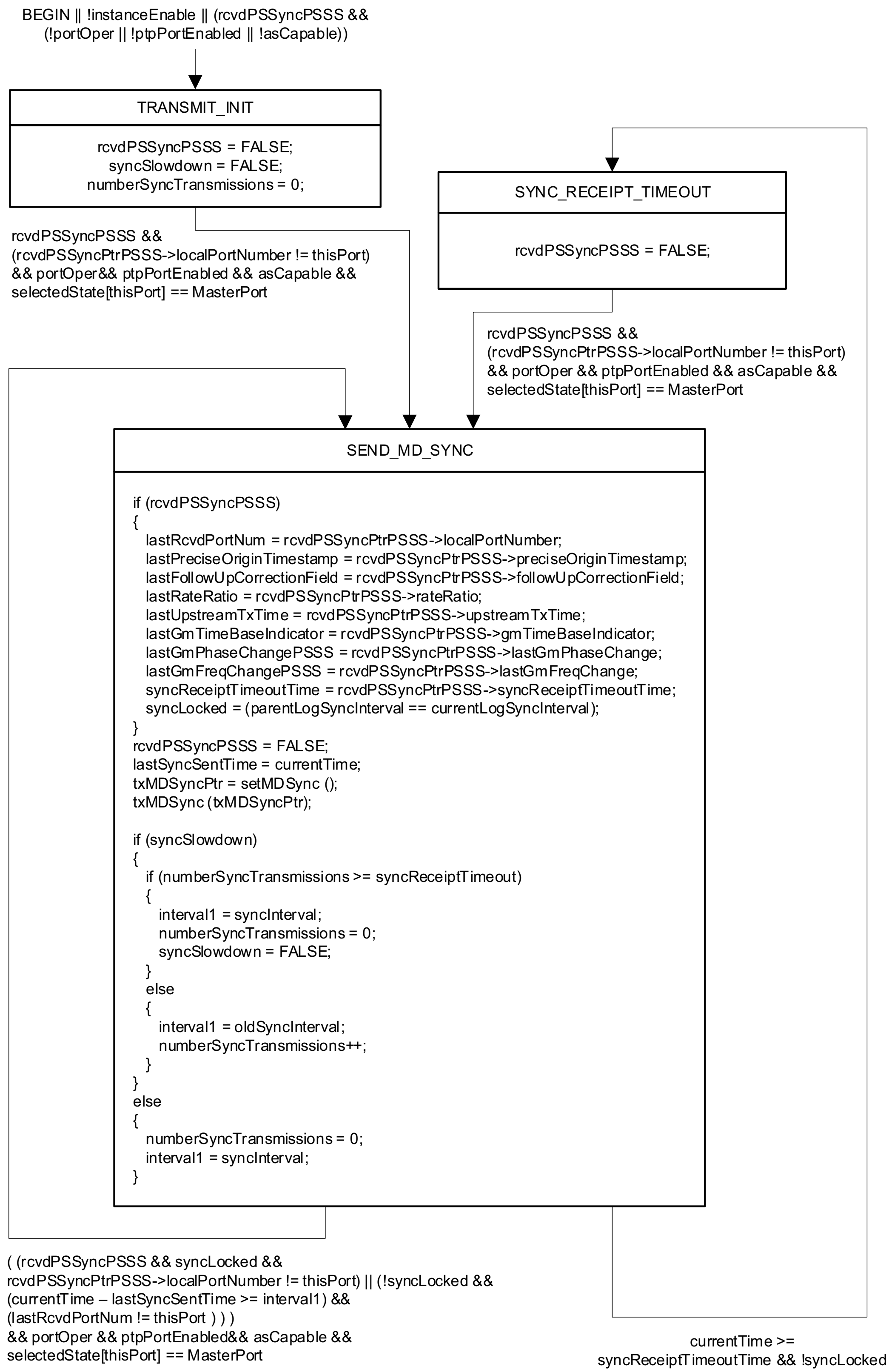


Figure 10-8—PortSyncSyncSend state machine

10.2.13.1.1 rcvdPSSyncCSS: a Boolean variable that notifies the current state machine when a PortSyncSync structure is received from the SiteSyncSync state machine of the SiteSync entity. This variable is reset by this state machine.

10.2.13.1.2 rcvdLocalClockTickCSS: a Boolean variable that notifies the current state machine when the LocalClock entity updates its time. This variable is reset by this state machine.

10.2.13.1.3 rcvdPSSyncPtrCSS: a pointer to the received PortSyncSync structure.

10.2.13.2 State machine functions

10.2.13.2.1 updateSlaveTime(): updates the global variable clockSlaveTime (see 10.2.4.3), based on information received from the SiteSync and LocalClock entities. It is the responsibility of the application to filter slave times appropriately (see B.3 and B.4 for examples). As one example, clockSlaveTime can be

a) set to syncReceiptTime at every LocalClock update immediately after a PortSyncSync structure is received, and
b) incremented by localClockTickInterval (see 10.2.4.18) multiplied by the rateRatio member of the previously received PortSyncSync structure during all other LocalClock updates.

If no PTP Instance is grandmaster-capable, i.e., gmPresent is FALSE, then clockSlaveTime is set to the time provided by the LocalClock. This function is invoked when rcvdLocalClockTickCSS is TRUE.

10.2.13.2.2 invokeApplicationInterfaceFunction (functionName): invokes the application interface function whose name is functionName. In the case here, functionName is clockTargetPhaseDiscontinuity.result (see 9.6.2).

10.2.13.3 State diagram

The ClockSlaveSync state machine shall implement the function specified by the state diagram in Figure 10-9, the local variables specified in 10.2.13.1, and the relevant global variables and functions specified in 10.2.4 through 10.2.6. The state machine receives a PortSyncSync structure from the SiteSyncSync state machine. It computes syncReceiptTime and clockSlaveTime, and sets syncReceiptLocalTime (i.e., the time relative to the LocalClock entity corresponding to syncReceiptTime), GmTimeBaseIndicator, lastGmPhaseChange, and lastGmFreqChange. It provides clockSlaveTime to the ClockMasterSyncOffset state machine, and provides information to the ClockTarget entity (via the ClockTargetPhaseDiscontinuity interface, see 9.6) to enable that entity to determine if a phase or frequency discontinuity has occurred.

The per-Port global variables used in the ClockSlaveSync state machine are determined based on rcvdPSSyncPtrCSS->localPortNumber, as follows:

a) If rcvdPSSyncPtrCSS->localPortNumber > 0, the per-Port global variables of port number rcvdPSSyncPtrCSS->localPortNumber are used.
b) If rcvdPSSyncPtrCSS->localPortNumber == 0, the values of the used per-Port global variables are fixed as:
    1) meanLinkDelay = 0
    2) delayAsymmetry = 0
    3) neighborRateRatio = 1.0.

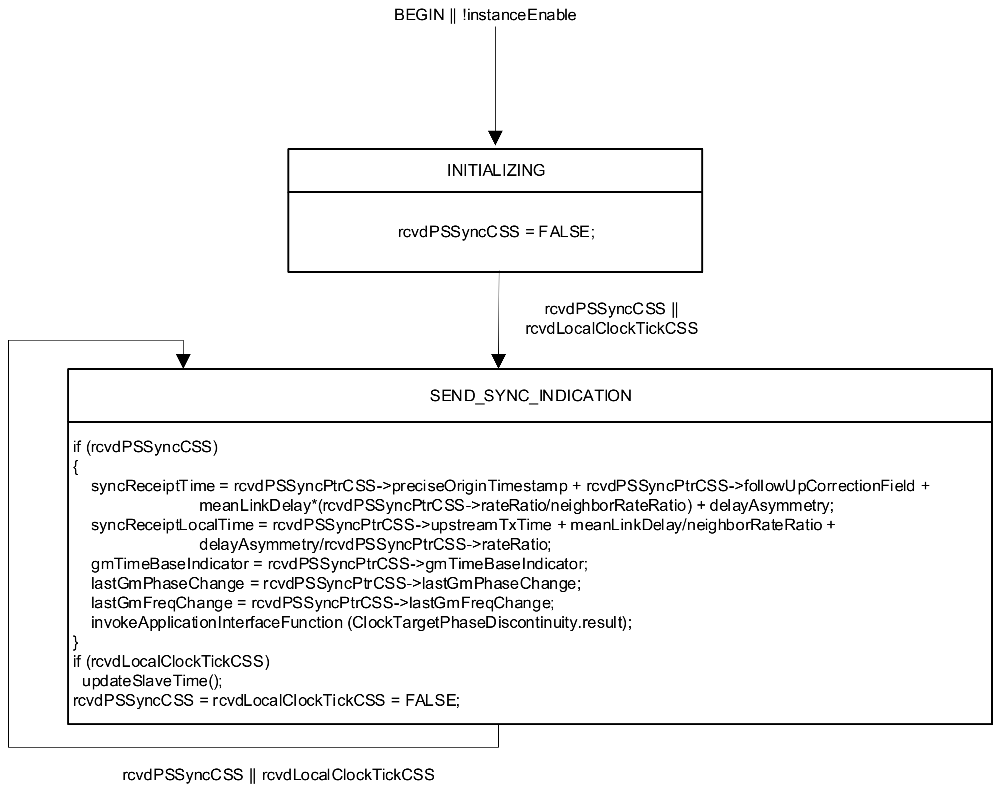


Figure 10-9—ClockSlaveSync state machine

10.3 Best master clock selection, external port configuration, and announce interval setting state machines

10.3.1 Best master clock selection and external port configuration overview

There are two methods for setting the grandmaster and time-synchronization spanning tree for a gPTP domain:

a) The BMCA is used to determine the grandmaster for a gPTP domain and construct the time-synchronization spanning tree with that grandmaster as the root. In this case, the network is configured automatically, i.e., the port states are set, using the results of the BMCA.
b) The port states are configured to force a desired grandmaster, and to construct a desired time-synchronization spanning tree with the grandmaster as the root.

The port state definitions are given in Table 10-2.

The per PTP Instance global variable externalPortConfigurationEnabled indicates whether (a) or (b) is used; a value of TRUE indicates (b) and a value of FALSE indicates (a) (see 10.3.9.24). The data type of externalPortConfigurationEnabled is Boolean. Method (a) is implemented, and is the default mode of operation (i.e., externalPortConfigurationEnabled is FALSE) on domain 0 to maintain backward compatibility. For domains other than domain 0:

c) at least one of the possibilities (a) or (b) is implemented,
d) both possibilities can be implemented, and
e) if both possibilities are implemented, the default value of externalPortConfigurationEnabled is FALSE.

Once an Announce message is transmitted by a port, subsequent timing information (see 7.4) transmitted by that port is derived from the grandmaster indicated in that Announce message.

Table 10-2—Port state definitions

Port state	Description
MasterPort	Any port, P, of the PTP Instance that is closer to the root than any other port of the gPTP communication path connected to P
SlavePort	The one port of the PTP Instance that is closest to the root PTP Instance. If the root is grandmaster-capable, the SlavePort is also closest to the grandmaster. The PTP Instance does not transmit Sync or Announce messages on the SlavePort.
PassivePort	Any port of the PTP Instance whose port state is not MasterPort, SlavePort, or DisabledPort.
DisabledPort	Any port of the PTP Instance for which portOper, ptpPortEnabled, and asCapable are not all TRUE.
NOTE—Port states are per port and per domain (i.e., per PTP Instance, see 8.1).	

NOTE 1—Information contained in Sync and associated Follow_Up messages received on ports whose port state is PassivePort is discarded; the SiteSyncSync state machine (see 10.2.7) uses only information received from a port whose port state is SlavePort.

An example master/slave hierarchy of PTP Instances is shown in Figure 10-10. The grandmaster ports all have port state of MasterPort. All the other PTP Instances have exactly one slave port. The time-synchronization spanning tree is composed of the PTP Instances and the links that do not have an endpoint port whose port state is PassivePort.

10.3.1.1 Best master clock algorithm overview

In the BMCA (i.e., method (a) of 10.3.1), best master selection information is exchanged between PTP Instances of time-aware systems via Announce messages (see 10.5 and 10.6). Each Announce message contains time-synchronization spanning tree vector information that identifies one PTP Instance as the root of the time-synchronization spanning tree and, if the PTP Instance is grandmaster-capable, the grandmaster. Each PTP Instance in turn uses the information contained in the Announce messages it receives, along with its knowledge of itself, to compute which of the PTP Instances that it has knowledge of ought to be the root of the spanning tree and, if grandmaster-capable, the grandmaster. As part of constructing the time-synchronization spanning tree, each port of each PTP Instance is assigned a port state from Table 10-2 by state machines associated with the ports and with the PTP Instance as a whole.

NOTE—The BMCA described in this standard is equivalent to a subset of the BMCA described in IEEE Std 1588-2019. It is also equivalent to a subset of the Rapid Spanning Tree Protocol (RSTP) described in IEEE Std 802.1Q (though the full RSTP described inEEE Std 802.1Q is not equivalent to the full BMCA described in IEEE Std 1588-2019). The BMCA description here uses the formalism of the RSTP description in the latter standard.

10.3.1.2 external port configuration overview

In external port configuration (i.e., method (b), of 10.3.1), an external entity determines the synchronization spanning tree and sets the port states accordingly. The method used by the external entity to determine the synchronization spanning tree is outside the scope of this standard. However, as with the BMCA, Announce messages are used to transport information on the time-synchronization spanning tree and grandmaster time properties information from one PTP Instance to the next in the tree. The external entity sets the state of a gPTP Port by setting the value of externalPortConfigurationPortDS.desiredState to the desired state.

10.3.2 systemIdentity

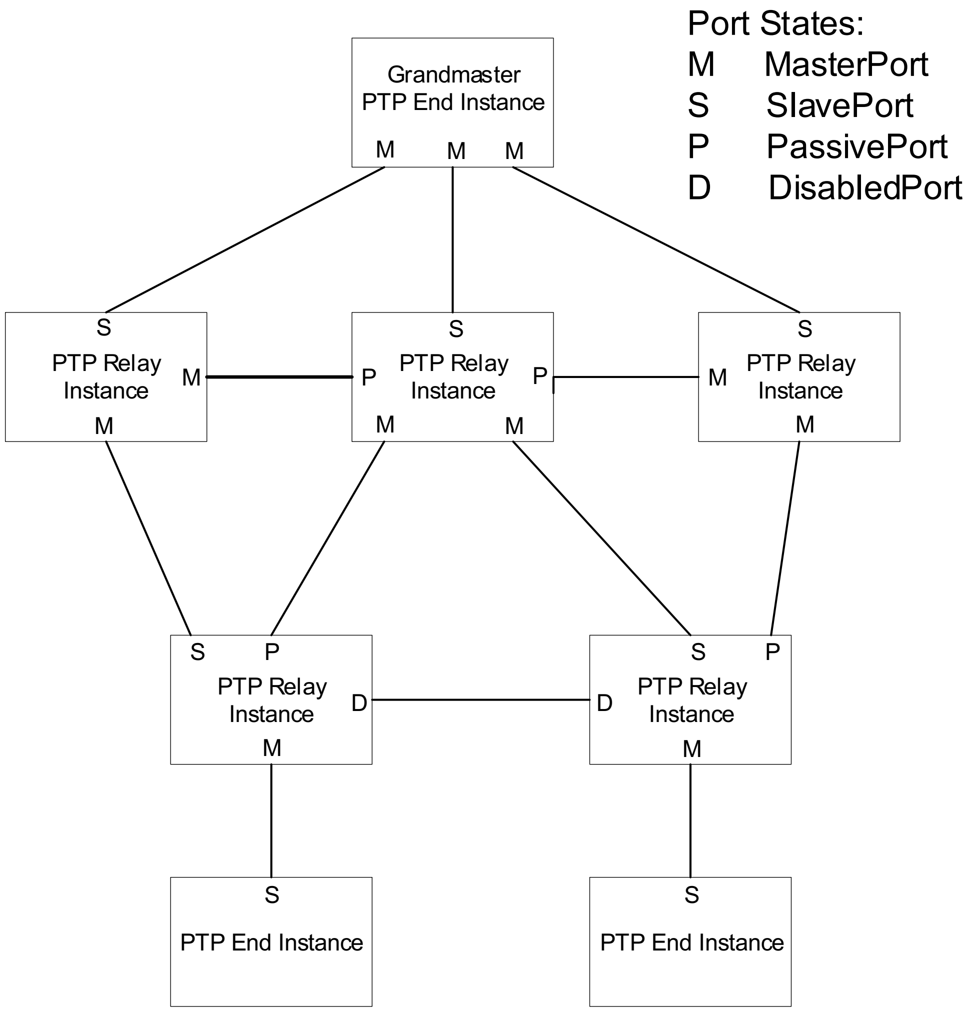


Figure 10-10—Example master/slave hierarchy of PTP Instances

The systemIdentity attribute of a PTP Instance is a UInteger112 (i.e., a 14-byte, unsigned integer) formed by concatenating the following attributes, in the following order, from most significant to least significant octet:

a) priority1 (1 octet, see 8.6.2.1)
b) clockClass (1 octet, see 8.6.2.2 and 6.4.3.8)
c) clockAccuracy (1 octet, see 8.6.2.3 and 6.4.3.8)
d) offsetScaledLogVariance (2 octets, see 8.6.2.4 and 6.4.3.8)
e) priority2 (1 octet, see 8.6.2.5)
f) clockIdentity (8 octets, see 8.5.2.2 and 6.4.3.6)

The systemIdentity attribute is defined for convenience when comparing two PTP Instances to determine, when using the BMCA (i.e., (a) of 10.3.1) which is a better candidate for root and if the PTP Instance is grandmaster-capable (i.e., the value of priority1 is less than 255, see 8.6.2.1). Two PTP Instances are compared as follows. Let the systemIdentity of PTP Instance A be S_A and the systemIdentity of PTP Instance B be S_B. Let the clockIdentity of A be C_A and the clockIdentity of B be C_B. Then, if $C_A \neq C_B$, i.e., A and B represent different PTP Instances:

g) A is better than B if and only if $S_A < S_B$, and
h) B is better than A if and only if $S_B < S_A$.

If $C_A = C_B$, i.e., A and B represent the same PTP Instance:

i) $S_A < S_B$ means that A represents an upgrading of the PTP Instance compared to B or, equivalently, B represents a downgrading of the PTP Instance compared to A,
j) $S_B < S_A$ means that B represents an upgrading of the PTP Instance compared to A or, equivalently, A represents a downgrading of the PTP Instance compared to B, and
k) $S_A = S_B$ means that A and B represent the same PTP Instance that has not changed.

Comparisons g) and h) above imply that, with the ordering of attributes in the systemIdentity, the clockIdentity is a tie-breaker for the case where two different PTP Instances that have identical attributes a) through e) are compared.

Comparisons g) and h) also imply that a PTP Instance that is grandmaster-capable is always better than another PTP Instance that is not grandmaster-capable, because the priority1 is less than 255 if the PTP Instance is grandmaster-capable and 255 if it is not grandmaster-capable (see 8.6.2.1).

The cases where A and B represent different PTP Instances and represent the same PTP Instance are handled separately in the BMCA. When comparing two different PTP Instances, the better PTP Instance is selected as the grandmaster candidate. However, if A and B represent the same PTP Instance with attributes that have changed, the PTP Instance is considered as having the most recent attributes when doing subsequent comparisons with other PTP Instances.

10.3.3 stepsRemoved

Every PTP Instance has a stepsRemoved associated with it. For the root PTP Instance, and therefore the grandmaster in the case where the root is grandmaster-capable, it is zero. For all other PTP Instances, it is the number of links in the path from the root to the respective PTP Instance.

The stepsRemoved attributes of different ports of a PTP Instance are compared after comparisons of other attributes that take precedence (i.e., priority1, clockClass, clockAccuracy, offsetScaledLogVariance, priority2) do not result in one port being declared better than the other. Among those ports whose stepsRemoved attributes are compared, the port on the PTP Instance with the lowest stepsRemoved is assigned the state of SlavePort for that PTP Instance (the root PTP Instance does not have a SlavePort). This lowest stepsRemoved is also considered the stepsRemoved for the PTP Instance. If a PTP Instance has two or more ports with the same stepsRemoved, then the port with the smallest portNumber is selected as the SlavePort.

10.3.4 time-synchronization spanning tree priority vectors

PTP Instances send best master selection information to each other in Announce messages. The information is structured in a time-synchronization spanning tree priority vector. Time-synchronization spanning tree priority vectors provide the basis for a concise specification of the BMCA's determination of the time-synchronization spanning tree and grandmaster. A priority vector is formed by concatenating the following attributes, in the following order, from most significant to least significant octet:

a) rootSystemIdentity (14 octets, see 10.3.2)
b) stepsRemoved (2 octets, see 10.3.3)

c) sourcePortIdentity (i.e., portIdentity of the transmitting PTP Instance; 10 octets, see 8.5.2 and 10.6.2)
d) portNumber of the receiving port (2 octets, see 8.5.2.3)

The first two components of a priority vector are significant throughout the gPTP domain; they are propagated via Announce messages and updated through invocation of BMCA state machines. The next component is assigned hop-by-hop for each gPTP communication path or PTP Instance, and thus is of local significance only. It is used as a tie-breaker in decisions between time-synchronization spanning tree priority vectors that are otherwise equal. The fourth component is not conveyed in Announce messages, but is used as a tie-breaker within a PTP Instance.

The set of all time-synchronization spanning tree priority vectors is totally ordered. For all components, a lesser numerical value is better, and earlier components in the preceding list are more significant. In addition, as mentioned earlier, a priority vector that reflects a root PTP Instance that is grandmaster-capable is always better than a priority vector that reflects a root PTP Instance that is not grandmaster-capable. As each PTP Instance port receives a priority vector, via an Announce message, from ports closer to the root, additions are made to one or more components to yield a worse priority vector. This process of receiving information, adding to it, and passing it on, can be described in terms of the message priority vector received and a set of priority vectors used to facilitate the computation of a priority vector for each port, to be transmitted in further Announce Messages to PTP Instances further from the root.

10.3.5 Priority vector calculations

The portPriorityVector is the time-synchronization spanning tree priority vector held for the port when the reception of Announce messages and any pending update of information has been completed:

portPriorityVector = {rootSystemIdentity : stepsRemoved : sourcePortIdentity : portNumber}

A messagePriorityVector is the time-synchronization spanning tree priority vector conveyed in a received Announce Message. For a PTP Instance S receiving an Announce Message on Port P_S with portNumber PN_S, from a MasterPort with portIdentity P_M on PTP Instance M claiming a rootSystemIdentity of R_M and a stepsRemoved of SR_M:

messagePriorityVector = $\{R_M : SR_M : P_M : PN_S\}$

This messagePriorityVector is superior to the portPriorityVector and will replace it if, and only if, the messagePriorityVector is better than the portPriorityVector, or the Announce message has been transmitted from the same master PTP Instance and MasterPort as the portPriorityVector, i.e., if the following is true:

((R_M < rootSystemIdentity)) ||

((R_M == rootSystemIdentity) && (SR_M < stepsRemoved)) ||

((R_M == rootSystemIdentity) && (SR_M == stepsRemoved) && (P_M < sourcePortIdentity (of current master PTP Instance))) ||

((R_M == rootSystemIdentity) && (SR_M == stepsRemoved)
&& (P_M == sourcePortIdentity (of current master PTP Instance)) && (PN_S < portNumber)) ||

((P_M.clockIdentity == sourcePortIdentity.clockIdentity (of current master PTP Instance)) && (P_M.portNumber == sourcePortIdentity.PortNumber (of the current master PTP Instance)))

A gmPathPriorityVector can be calculated from a received portPriorityVector by adding one to the stepsRemoved component:

gmPathPriorityVector = $\{R_M : SR_M + 1 : P_M : PN_S\}$

The systemPriorityVector for a PTP Instance S with systemIdentity S_S and clockIdentity C_S is the priority vector that would, with the portIdentity of the SlavePort set equal to the portIdentity of the transmitting port, be used as the message priority vector in Announce Messages transmitted on S's ports whose state is MasterPort if S was selected as the root:

systemPriorityVector = $\{S_S : 0 : \{C_S : 0\} : 0\}$

The gmPriorityVector for S is the best of the set comprising the systemPriorityVector vector plus every gmPathPriorityVector for which the clockIdentity of the master PTP Instance portIdentity is not the clockIdentity of S. If the systemPriorityVector is best, S has been selected as the root. For the case where the best gmPathPriorityVector is that of port PN_S above, then:

gmPriorityVector = $\{S_S : 0 : \{C_S : 0\} : 0\}$if S is better than R_M, or

gmPriorityVector = $\{R_M : SR_M + 1 : P_M : PN_S\}$ if S is worse than R_M.

The masterPriorityVector for a port Q on PTP Instance S is the gmPriorityVector with S's clockIdentity C_S substituted for the clockIdentity of the master portIdentity, and Q's portNumber PN_Q substituted for the portNumber of the master portIdentity and for the portNumber of the receiving port:

masterPriorityVector = $\{S_S : 0 : \{C_S : PN_Q\} : PN_Q\}$if S is better than R_M, or
masterPriorityVector = $\{R_M : SR_M + 1 : \{C_S : PN_Q\} : PN_Q\}$ if S is worse than R_M.

If the masterPriorityVector is better than the portPriorityVector, the port will be the MasterPort for the attached gPTP communication path and the portPriorityVector will be updated. The messagePriorityVector information in Announce messages transmitted by a port always includes the first three components of the masterPriorityVector of the port.

NOTE—The consistent use of lower numerical values to indicate better information is deliberate as the MasterPort that is closest to the root, i.e., has a numerically lowest path cost component, is selected from amongst potential alternatives for any given gPTP communication path. Adopting the conventions that lower numerical values indicate better information, that where possible more significant priority components are encoded earlier in the octet sequence of an Announce message, and that earlier octets in the encoding of individual components are more significant, allow concatenated octets that compose a priority vector to be compared as if they were a multiple octet encoding of a single number, without regard to the boundaries between the encoded components. To reduce the confusion that naturally arises from having the lesser of two numerical values represent the better of the two, i.e., the one to be chosen all other factors being equal, this clause uses the following consistent terminology. Relative numeric values are described as "least," "lesser," "equal," and "greater," and their comparisons as "less than," "equal to," or "greater than," while relative time-synchronization spanning tree priorities are described as "best," "better," "the same," "different," and "worse" and their comparisons as "better than," "the same as," "different from," and "worse than." The operators "<" and "==" represent less than and equal to respectively. The terms "superior" and "inferior" are used for comparisons that are not simply based on priority but can include the fact that the priority vector of a MasterPort can replace an earlier vector transmitted in an Announce message by the same port.

10.3.6 Port state assignments

10.3.6.1 Port state assignments when the BMCA is used

The BMCA assigns one of the following port states to each gPTP Port: MasterPort, SlavePort, PassivePort, or DisabledPort.

The DisabledPort state is assigned if portOper is FALSE (see 10.2.5.12), ptpPortEnabled is FALSE (see 10.2.5.13), or asCapable is FALSE (see 10.2.5.1).

A gPTP Port for which portOper, ptpPortEnabled, and asCapable are all TRUE has its port state assigned according to the source and relative priority of the time-synchronization spanning tree portPriorityVector (see 10.3.4 and 10.3.5) as follows:

a) If the PTP Instance is not the root, the source of the gmPriorityVector is the SlavePort.
b) Each port whose portPriorityVector is its masterPriorityVector is a MasterPort.
c) Each Port, other than the SlavePort, whose portPriorityVector has been received from another PTP Instance or another port on this PTP Instance is a PassivePort.

10.3.6.2 Port state assignments when external port configuration is used

If external port configuration is used, one of the states MasterPort, SlavePort, PassivePort, or DisabledPort, is assigned to each gPTP Port by an external entity, as described below.

The DisabledPort state is assigned if portOper is FALSE (see 10.2.5.12), ptpPortEnabled is FALSE (see 10.2.5.13), or asCapable is FALSE (see 10.2.5.1).

The member externalPortConfigurationPortDS.desiredState (see 14.12.2) is used by an external entity to set the state of the respective gPTP port to MasterPort, SlavePort, or PassivePort. When this member is set, its value is copied to the per port local variable portStateInd (see 10.3.15.1.5). If portOper, ptpPortEnabled, and asCapable are all TRUE for this port, the port state is set equal to the value of externalPortConfigurationPortDS.desiredState by copying the value of this member to the element of the selectedState array (see 10.2.4.20) for this gPTP port.

10.3.7 Overview of best master clock selection, external port configuration, and announce interval setting state machines

10.3.7.1 Best master clock selection state machines overview

The best master clock selection function in a PTP Instance is specified by a number of cooperating state machines. Figure 10-11 is not itself a state machine, but illustrates the machines, their interrelationships, the principle variables and structures used to communicate between them, their local variables, and performance parameters.

NOTE—The BMCA state machines are all invoked by the media-independent layer, i.e., by the SiteSync and PortSync entities. The media-dependent layer, i.e., the MD entity, simply takes an Announce message received from the PortSync entity of the same port and gives it to the next lower layer (e.g., IEEE 802.3, IEEE 802.11). It is the PortSync entity that generates and consumes Announce messages.

The media-independent layer state machines in Figure 10-11 are:

a) PortAnnounceReceive (one instance per PTP Instance, per port): receives Announce information from the MD entity of the same port, determines if the Announce message is qualified and, if so, sets the rcvdMsg variable. This state machine is invoked by the PortSync entity of the port.
b) PortAnnounceInformation (one instance per PTP Instance, per port): receives new qualified Announce information from the PortAnnounceReceive state machine, determines if the Announce information is better than the current best master information it knows about, and updates the current best master information when it receives updated port state information from the PortStateSelection state machine and when announce receipt timeout or, in the case where gmPresent is TRUE, sync receipt timeout occurs. This state machine is invoked by the PortSync entity of the port.
c) PortStateSelection (one instance per PTP Instance): updates the gmPathPriority vector for each port of the PTP Instance, the gmPriorityVector for the PTP Instance, and the masterPriorityVector for each port of the PTP Instance; determines the port state for each port; and updates gmPresent. This state machine is invoked by the SiteSync entity of the PTP Instance.

d) PortAnnounceTransmit (one instance per PTP Instance, per port): if the port state is MasterPort, transmits Announce information to the MD entity when an announce interval has elapsed, port states have been updated, and portPriority and portStepsRemoved information has been updated with newly determined masterPriority and masterStepsRemoved information. This state machine is invoked by the PortSync entity of the port, and is also used when external port configuration is used.

10.3.7.2 External port configuration state machines overview

The external port configuration function in a PTP Instance is specified by a number of cooperating state machines. Figure 10-12 is not itself a state machine, but illustrates the machines, their interrelationships, the principle variables and structures used to communicate between them, their local variables, and performance parameters.

NOTE—The external port configuration state machines are all invoked by the media-independent layer and are per port, i.e., they are invoked by the PortSync entity for the respective gPTP port. The media-dependent layer, i.e., the MD entity, simply takes an Announce message received from the PortSync entity of the same port and gives it to the next lower layer (e.g., IEEE 802.3, IEEE 802.11). It is the PortSync entity that generates and consumes Announce messages.

The media-independent layer state machines in Figure 10-11 are:

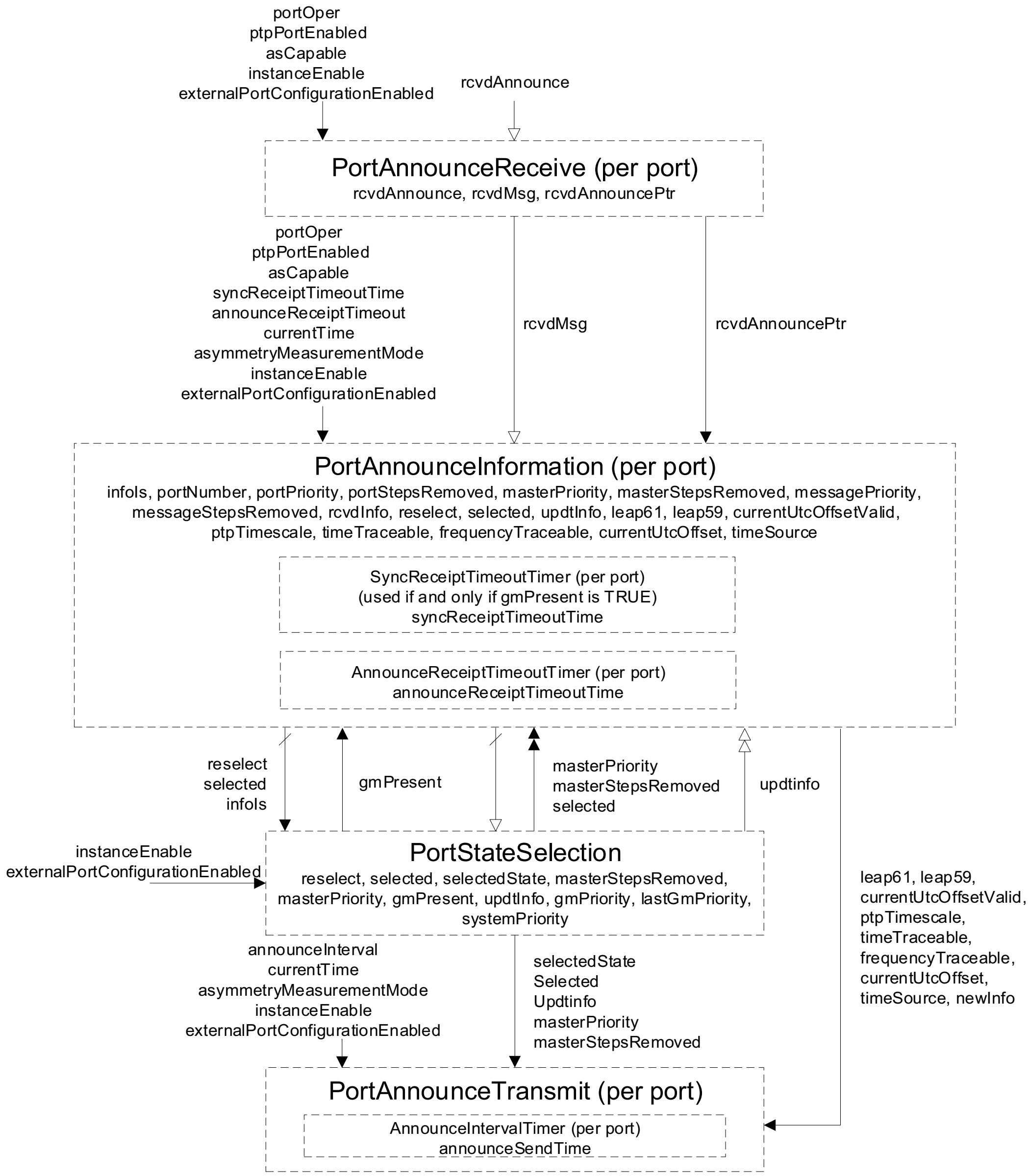
a) PortAnnounceInformationExt (one instance per PTP Instance, per port): receives and stores new Announce information received in Announce messages.
b) PortStateSettingExt (one instance per PTP Instance): copies the desired port state for the port to the respective selectedState array element, updates gmPresent, computes masterStepsRemoved, stores the time properties information in the respective global variables, and computes the gmPriorityVector and masterPriorityVector.
c) PortAnnounceTransmit (one instance per PTP Instance, per port): if the port state is MasterPort, transmits Announce information to the MD entity when an announce interval has elapsed, port states have been updated, and portPriority and portStepsRemoved information has been updated with newly determined masterPriority and masterStepsRemoved information. This state machine is invoked by the PortSync entity of the port, and is also used when the BMCA is used.

10.3.7.3 AnnounceIntervalSetting state machine overview

An additional state machine, the AnnounceIntervalSetting state machine, receives a Signaling message that contains a Message Interval Request TLV (see 10.6.4.3), and sets the global variables that give the duration of the mean interval between successive Announce messages..

10.3.8 Overview of global variables used by best master clock selection, external port configuration, and announce interval setting state machines

The subclause that follows this subclause, i.e., 10.3.9, defines global variables used by best master clock selection, external port configuration, and announce interval setting state machines, whose scopes are per PTP Instance, common across all PTP Instances (e.g., used by CMLDS, see 11.2.17), per port per PTP



Notation:

Variables are shown both within the machine where they are principally used and between machines where they are used to communicate information. In the latter case a variety of arrow styles, running from one machine to another, show how each is typically used:

Not changed by the target machine. Where the machines are both per port, this variable communicates between instances for the same port

Set (or cleared) by the originating machine, cleared (or set) by the target machine. Where the machines are both per port, this communicates between instances for the same port.

As above, except that the originating per-port machine instance communicates with multiple port machine instances (by setting or clearing variables owned by those ports).

As above, except that multiple per-port instances communicate with (an)other instance(s) (by setting or clearing variables owned by the originating ports).

Figure 10-11—Best master clock selection state machines—overview and interrelationships

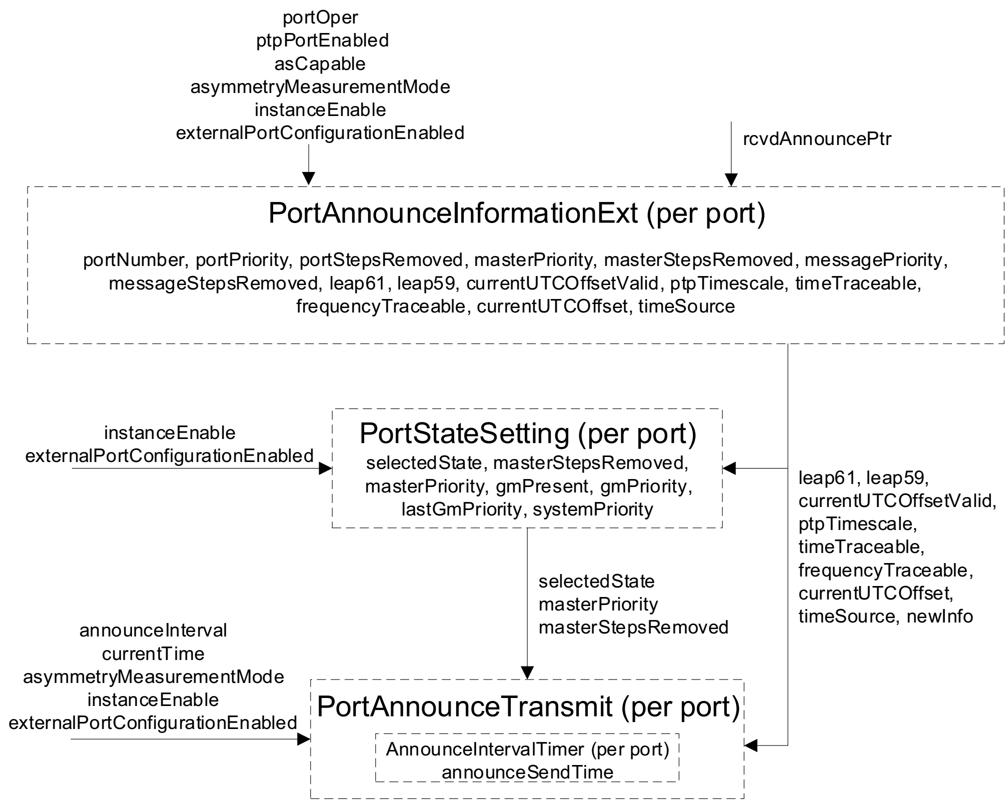


Figure 10-12—External port configuration state machines—overview and interrelationships

Instance, and per port but common across all PTP Instances. Table 10-3 summarizes the scope of each global variable of 10.3.9.

Table 10-3 — Summary of scope of global variables used by best master clock selection, external port configuration, and announce interval setting state machines 10.3.9)

Variable name	Subclause of definition	Per PTP Instance (i.e., per domain)	Per PTP Instance, per PTP Port	Instance used by CMLDS (see 11.2.17), i.e., that is common across all LinkPorts	Instance used by CMLDS, per LinkPort
reselect	10.3.9.1	Yes	No	No	No
selected	10.3.9.2	Yes	No	No	No
masterStepsRemoved	10.3.9.3	Yes	No	No	No
leap61	10.3.9.4	Yes	No	No	No
leap59	10.3.9.5	Yes	No	No	No
currentUtcOffsetValid	10.3.9.6	Yes	No	No	No

Table 10-3 — Summary of scope of global variables used by best master clock selection, external port configuration, and announce interval setting state machines 10.3.9)

Variable name	Subclause of definition	Per PTP Instance (i.e., per domain)	Per PTP Instance, per PTP Port	Instance used by CMLDS (see 11.2.17), i.e., that is common across all LinkPorts	Instance used by CMLDS, per LinkPort
ptpTimescale	10.3.9.7	Yes	No	No	No
timeTraceable	10.3.9.8	Yes	No	No	No
frequencyTraceable	10.3.9.9	Yes	No	No	No
currentUtcOffset	10.3.9.10	Yes	No	No	No
timeSource	10.3.9.11	Yes	No	No	No
sysLeap61	10.3.9.12	Yes	No	No	No
sysLeap59	10.3.9.13	Yes	No	No	No
sysCurrentUtcOffsetValid	10.3.9.14	Yes	No	No	No
sysPtpTimescale	10.3.9.15	Yes	No	No	No
sysTimeTraceable	10.3.9.16	Yes	No	No	No
sysFrequencyTraceable	10.3.9.17	Yes	No	No	No
sysCurrentUtcOffset	10.3.9.18	Yes	No	No	No
sysTimeSource	10.3.9.19	Yes	No	No	No
systemPriority	10.3.9.20	Yes	No	No	No
gmPriority	10.3.9.21	Yes	No	No	No
lastGmPriority	10.3.9.22	Yes	No	No	No
pathTrace	10.3.9.23	Yes	No	No	No
externalPortConfigurationEnabled	10.3.9.24	Yes	No	No	No
lastAnnouncePort	10.3.9.25	Yes	No	No	No
announceReceiptTimeoutTimeInterval	10.3.10.1	No	Yes[a]	No	No
announceSlowdown	10.3.10.2	No	Yes	No	No
oldAnnounceInterval	10.3.10.3	No	Yes	No	No
infoIs	10.3.10.4	No	Yes	No	No
masterPriority	10.3.10.5	No	Yes	No	No
currentLogAnnounceInterval	10.3.10.6	No	Yes	No	No
initialLogAnnounceInterval	10.3.10.7	No	Yes	No	No
announceInterval	10.3.10.8	No	Yes	No	No
messageStepsRemoved	10.3.10.9	No	Yes	No	No
newInfo	10.3.10.10	No	Yes	No	No
portPriority	10.3.10.11	No	Yes	No	No
portStepsRemoved	10.3.10.12	No	Yes	No	No
rcvdAnnouncePtr	10.3.10.13	No	Yes	No	No
rcvdMsg	10.3.10.14	No	Yes	No	No
updtInfo	10.3.10.15	No	Yes	No	No

Table 10-3 — Summary of scope of global variables used by best master clock selection, external port configuration, and announce interval setting state machines 10.3.9)

Variable name	Subclause of definition	Per PTP Instance (i.e., per domain)	Per PTP Instance, per PTP Port	Instance used by CMLDS (see 11.2.17), i.e., that is common across all LinkPorts	Instance used by CMLDS, per LinkPort
annLeap61	10.3.10.16	No	Yes	No	No
annLeap59	10.3.10.17	No	Yes	No	No
annCurrentUtcOffsetValid	10.3.10.18	No	Yes	No	No
annPtpTimescale	10.3.10.19	No	Yes	No	No
annTimeTraceable	10.3.10.20	No	Yes	No	No
annFrequencyTraceable	10.3.10.21	No	Yes	No	No
annCurrentUtcOffset	10.3.10.22	No	Yes	No	No
annTimeSource	10.3.10.23	No	Yes	No	No

10.3.9 Per PTP Instance global variables

10.3.9.1 reselect: a Boolean array of length numberPorts+1 (see 8.6.2.8). Setting reselect[j], where 0 ≤ j ≤ numberPorts, to TRUE causes the STATE_SELECTION block of the PortStateSelection state machine (see 10.3.13) to be re-entered, which in turn causes the port state of each port of the PTP Instance to be updated (via the function updtStatesTree(), see 10.3.13.2.4). This variable is used only by the BMCA, i.e., not by the explicit port state configuration option.

10.3.9.2 selected: a Boolean array of length numberPorts+1 (see 8.6.2.8). selected[j], where 0 ≤ j ≤ numberPorts, is set to TRUE immediately after the port states of all the ports are updated. This indicates to the PortAnnounceInformation state machine (see 10.3.12) that it can update the portPriorityVector and other variables for each port. This variable is used both by the BMCA and the explicit port state configuration option; however, its value does not impact the explicit port state configuration option (see the NOTE in 10.3.16.3, just after Figure 10-18).

NOTE—Array elements 0 of the reselect and selected arrays are not used, except that the function clearReselectTree() sets reselect[0] to FALSE when it sets the entire array to zero and the function setSelectedTree() sets selected[0] to TRUE when it sets the entire array to TRUE. This is done only for convenience, so that array element j can correspond to port j. Note also that, in contrast, selectedState[0] is not used (see 10.2.4.20).

10.3.9.3 masterStepsRemoved: the value of stepsRemoved for the PTP Instance, after the port states of all the ports have been updated (see 10.3.13.2.4 for details on the computation of masterStepsRemoved). The data type for masterStepsRemoved is UInteger16. This variable is used both by the BMCA and the explicit port state configuration option.

10.3.9.4 leap61: a Boolean variable whose value is TRUE if the last minute of the current UTC day, relative to the current grandmaster, contains 61 s, and FALSE if the last minute of the current UTC day does not contain 61 s. This variable is used both by the BMCA and the explicit port state configuration option.

10.3.9.5 leap59: a Boolean variable whose value is TRUE if the last minute of the current UTC day, relative to the current grandmaster, contains 59 s, and FALSE if the last minute of the current UTC day does not contain 59 s. This variable is used both by the BMCA and the explicit port state configuration option.

10.3.9.6 currentUtcOffsetValid: a Boolean variable whose value is TRUE if currentUtcOffset (see 10.3.9.10), relative to the current grandmaster, is known to be correct, and FALSE if currentUtcOffset is not known to be correct. This variable is used both by the BMCA and the explicit port state configuration option.

10.3.9.7 ptpTimescale: a Boolean variable whose value is TRUE if the timescale of the current grandmaster is PTP (see 8.2.1) and FALSE if the timescale is ARB. This variable is used both by the BMCA and the explicit port state configuration option.

10.3.9.8 timeTraceable: a Boolean variable whose value is TRUE if both clockSlaveTime (i.e., the synchronized time maintained at the slave (see 10.2.4.3)) and currentUtcOffset (see 10.3.9.10) relative to the current grandmaster are traceable to a primary reference, and FALSE if one or both are not traceable to a primary reference. This variable is used both by the BMCA and the explicit port state configuration option.

10.3.9.9 frequencyTraceable: a Boolean variable whose value is TRUE if the frequency that determines clockSlaveTime, i.e., the frequency of the LocalClockEntity multiplied by the most recently computed rateRatio by the PortSyncSyncReceive state machine (see 10.2.8.1.4), is traceable to a primary reference, and FALSE if this frequency is not traceable to a primary reference. This variable is used both by the BMCA and the explicit port state configuration option.

10.3.9.10 currentUtcOffset: the difference between TAI time and UTC time, i.e., TAI time minus UTC time, expressed in seconds, and relative to the current grandmaster, when known; otherwise the value has no meaning (see 10.3.9.6). The data type for currentUtcOffset is Integer16. This variable is used both by the BMCA and the explicit port state configuration option.

NOTE—For example, 2006-01-01 00:00:00 UTC and 2006-01-01 00:00:33 TAI represent the same instant of time. At this time, currentUtcOffset was equal to 33 s.[13]

10.3.9.11 timeSource: the value of the timeSource attribute of the current grandmaster. The data type for timeSource is TimeSource (see 8.6.2.7). This variable is used both by the BMCA and the explicit port state configuration option.

10.3.9.12 sysLeap61: a Boolean variable whose value is TRUE if the last minute of the current UTC day, relative to the ClockMaster entity of this PTP Instance, contains 61 s, and FALSE if the last minute of the current UTC day does not contain 61 s. This variable is used both by the BMCA and the explicit port state configuration option.

10.3.9.13 sysLeap59: a Boolean variable whose value is TRUE if the last minute of the current UTC day, relative to the ClockMaster entity of this PTP Instance, contains 59 s, and FALSE if the last minute of the current UTC day does not contain 59 s. This variable is used both by the BMCA and the explicit port state configuration option.

10.3.9.14 sysCurrentUtcOffsetValid: a Boolean variable whose value is TRUE if currentUtcOffset (see 10.3.9.10), relative to the ClockMaster entity of this PTP Instance, is known to be correct, and FALSE if currentUtcOffset is not known to be correct. This variable is used both by the BMCA and the explicit port state configuration option.

10.3.9.15 sysPtpTimescale: a Boolean variable whose value is TRUE if the timescale of the ClockMaster entity of this PTP Instance is PTP (see 8.2.1), and FALSE if the timescale of the ClockMaster entity of this PTP Instance is ARB. This variable is used both by the BMCA and the explicit port state configuration option.

10.3.9.16 sysTimeTraceable: a Boolean variable whose value is TRUE if both masterTime (i.e., the time maintained by the ClockMaster entity of this PTP Instance (see 10.2.4.21)) and currentUtcOffset (see

[13]Note also that a leap second was not added at the end of the last UTC minute of 2005-12-31.

10.3.9.10) relative to the ClockMaster entity of this PTP Instance, are traceable to a primary reference, and FALSE if one or both are not traceable to a primary reference. This variable is used both by the BMCA and the explicit port state configuration option.

10.3.9.17 sysFrequencyTraceable: a Boolean variable whose value is TRUE if the frequency that determines masterTime of the ClockMaster entity of this PTP Instance, i.e., the frequency of the LocalClockEntity multiplied by the most recently computed gmRateRatio by the ClockMasterSyncReceive state machine (see 10.2.4.14 and 10.2.11), is traceable to a primary reference, and FALSE if this frequency is not traceable to a primary reference. This variable is used both by the BMCA and the explicit port state configuration option.

10.3.9.18 sysCurrentUtcOffset: the difference between TAI time and UTC time, i.e., TAI time minus UTC time, expressed in seconds, and relative to the ClockMaster entity of this PTP Instance, when known; otherwise the value has no meaning (see 10.3.9.14). The data type for sysCurrentUtcOffset is Integer16. This variable is used both by the BMCA and the explicit port state configuration option.

NOTE—See the NOTE in 10.3.9.10 for more detail on the sign convention.

10.3.9.19 sysTimeSource: the value of the timeSource attribute of the ClockMaster entity of this PTP Instance (see 8.6.2.7). The data type for sysTimeSource is TimeSource.

10.3.9.20 systemPriority: the systemPriority vector for this PTP Instance. The data type for systemPriority is UInteger224 (see 10.3.5).

10.3.9.21 gmPriority: the current gmPriorityVector for the PTP Instance. The data type for gmPriority is UInteger224 (see 10.3.5).

10.3.9.22 lastGmPriority: the previous gmPriorityVector for the PTP Instance, prior to the most recent invocation of the PortStateSelection state machine. The data type for lastGmPriority is UInteger224 (see 10.3.4). lastGmPriority is used only by the BMCA, i.e., not by the explicit port state configuration option.

10.3.9.23 pathTrace: an array that contains the clockIdentities of the successive PTP Instances that receive, process, and send Announce messages. The data type for pathTrace is ClockIdentity[N], where N is the number of PTP Instances, including the grandmaster, that the Announce information has traversed. This variable is used both by the BMCA and the explicit port state configuration option.

NOTE 1—N is equal to stepsRemoved+1 (see 10.6.3.2.6). The size of the pathTrace array can change after each reception of an Announce message, up to the maximum size for the respective medium. For example, the maximum value of N for a full-duplex, IEEE 802.3 medium, is 179. This is obtained from the fact that the number of PTP octets in an Announce message is 68 + 8N, where N is the number of entries in the pathTrace array (see 10.6.3.1 and Table 10-11) and the maximum payload size for full-duplex, IEEE 802.3 media is 1500 octets. Setting 68 + 8N = 1500 and solving for N gives N = 179.

NOTE 2—The current behavior for the path trace feature is documented in 10.3.11.2.1 and 10.3.16.2.1, and is as follows:

— Item (c) of the description of the qualifyAnnounce() function of the PortAnnounceReceive state machine (see 10.3.11.2.1) indicates that if a path trace TLV is present and one of the elements of the pathSequence array field is equal to the clockIdentity of the clock where the TLV is being processed, the Announce message is not qualified.
— Item (d) of the description of the qualifyAnnounce() function indicates that if the Announce message is qualified and a path trace TLV is present, the pathSequence array of the TLV is copied to the pathTrace array (described here) and the clockIdentity of the PTP Instance that processes the Announce message is appended to the array; however, if a path trace TLV is not present, the path trace array is empty.
— Item (f) of the description of the txAnnounce() function of the PortAnnounceTransmit state machine (see 10.3.16.2.1) indicates that a path trace TLV is constructed and appended to an Announce message just before the

Announce message is transmitted only if the pathTrace array is not empty and appending the TLV does not cause the media-dependent layer frame to exceed any respective maximum size. If appending the TLV does cause a respective maximum frame size to be exceeded, or if the pathTrace array is empty, the TLV is not appended.

— These behaviors of the qualifyAnnounce() and txAnnounce() functions result in the path trace feature not being used, i.e., a path trace TLV is not appended to an Announce message and the pathTrace array is empty, once appending a clockIdentity to the TLV would cause the frame carrying the Announce message to exceed its maximum size.

NOTE 3—Once the value of stepsRemoved of an Announce message reaches 255, the Announce message is not qualified (see item (b) of 10.3.11.2.1).

10.3.9.24 externalPortConfigurationEnabled: a variable whose value indicates whether port states are externally configured or determined by the BMCA. The data type shall be Boolean. The value TRUE indicates that the port states are externally configured; the value FALSE indicates that the port states are determined by the BMCA. This variable is used both by the BMCA and the external port configuration option.

10.3.9.25 lastAnnouncePort: the port number of the port on which the most recent Announce message was received. This variable is used by the PortAnnounceInformationExt and PortStateSettingExt state machines, for the external port configuration option. This variable is not used by the BMCA. The data type for this variable is UInteger16.

10.3.10 Per-port global variables

10.3.10.1 announceReceiptTimeoutTimeInterval: the time interval after which announce receipt timeout occurs if an Announce message has not been received during the interval. The value of announceReceiptTimeoutTimeInterval is equal to announceReceiptTimeout (see 10.7.3.2) multiplied by the announceInterval (see 10.3.10.8) for the port at the other end of the link to which this port is attached. The value of announceInterval for the port at the other end of the link is computed from logMessageInterval of the received Announce message (see 10.6.2.2.14). The data type for announceReceiptTimeoutTimeInterval is UScaledNs. The variable infoIs is used only by the BMCA, i.e., not by the explicit port state configuration option.

10.3.10.2 announceSlowdown: a Boolean that is set to TRUE if the AnnounceIntervalSetting state machine (see Figure 10-19) receives a TLV that requests a larger Announce message transmission interval (see 10.7.2.2), and FALSE otherwise. When announceSlowdown is set to TRUE, the PortAnnounceTransmit state machine (see Figure 10-18) continues to send Announce messages at the old (i.e., faster) rate until a number of Announce messages equal to announceReceiptTimeout (see 10.7.3.2) have been sent, but with the logMessageInterval field of the PTP common header set equal to the new announce interval (i.e., corresponding to the slower rate). After announceReceiptTimeout Announce messages have been sent, subsequent Announce messages are sent at the new (i.e., slower) rate, and with the logMessageInterval field of the PTP common header set to the new announce interval. This variable is used both by the BMCA and the explicit port state configuration option. When announceSlowdown is set to FALSE, the PortAnnounceTransmit state machine immediately sends Announce messages at the new (i.e., slower) rate.

NOTE—If a receiver of Announce messages requests a slower rate, the receiver will continue to use the upstream announceInterval value, which it obtains from the logMessageInterval field of received Announce messages, until it receives an Announce message where that value has changed. If, immediately after requesting a slower Announce message rate, up to announceReceiptTimeout consecutive Announce messages sent to the receiver are lost, announce receipt timeout could occur if the sender had changed to the slower rate immediately. Delaying the slowing down of the sending rate of Announce messages for announceReceiptTimeout messages prevents this from happening.

10.3.10.3 oldAnnounceInterval: the saved value of the previous announce interval, when a new announce interval is requested via a Signaling message that contains a message interval request TLV. The data type for oldAnnounceInterval is UScaledNs. This variable is used both by the BMCA and the explicit port state configuration option.

10.3.10.4 infoIs: an Enumeration2 that takes the values Received, Mine, Aged, or Disabled, to indicate the origin and state of the port's time-synchronization spanning tree information:

a) If infoIs is Received, the port has received current information (i.e., announce receipt timeout has not occurred and, if gmPresent is TRUE, sync receipt timeout also has not occurred) from the master PTP Instance for the attached gPTP communication path.
b) If infoIs is Mine, information for the port has been derived from the SlavePort for the PTP Instance (with the addition of SlavePort stepsRemoved). This includes the possibility that the SlavePort is the port whose portNumber is 0, i.e., the PTP Instance is the root of the gPTP domain.
c) If infoIs is Aged, announce receipt timeout or, in the case where gmPresent is TRUE, sync receipt timeout have occurred.
d) If portOper, ptpPortEnabled, and asCapable are not all TRUE, infoIs is Disabled.

The variable infoIs is used only by the BMCA, i.e., not by the explicit port state configuration option.

10.3.10.5 masterPriority: the masterPriorityVector for the port. The data type for masterPriority is UInteger224 (see 10.3.4). This variable is used both by the BMCA and the explicit port state configuration option.

10.3.10.6 currentLogAnnounceInterval: the current value of the logarithm to base 2 of the mean time interval, in seconds, between the sending of successive Announce messages (see 10.7.2.2). This value is set in the AnnounceIntervalSetting state machine (see 10.3.17). The data type for currentLogAnnounceInterval is Integer8. This variable is used both by the BMCA and the explicit port state configuration option.

10.3.10.7 initialLogAnnounceInterval: the initial value of the logarithm to base 2 of the mean time interval, in seconds, between the sending of successive Announce messages (see 10.7.2.2). The data type for initialLogAnnounceInterval is Integer8. This variable is used both by the BMCA and the explicit port state configuration option.

10.3.10.8 announceInterval: a variable containing the mean Announce message transmission interval for the port. This value is set in the AnnounceIntervalSetting state machine (see 10.3.17). The data type for announceInterval is UScaledNs. This variable is used both by the BMCA and the explicit port state configuration option.

10.3.10.9 messageStepsRemoved: the value of stepsRemoved contained in the received Announce information. The data type for messageStepsRemoved is UInteger16. This variable is used both by the BMCA and the explicit port state configuration option.

10.3.10.10 newInfo: a Boolean variable that is set to cause a port to transmit Announce information; specifically, it is set when an announce interval has elapsed (see Figure 10-18), port states have been updated, and portPriority and portStepsRemoved information has been updated with newly determined masterPriority and masterStepsRemoved information. This variable is used both by the BMCA and the explicit port state configuration option.

10.3.10.11 portPriority: the portPriorityVector for the port. The data type for portPriority is UInteger224 (see 10.3.4). This variable is used only by the BMCA, i.e., not by the explicit port state configuration option.

10.3.10.12 portStepsRemoved: the value of stepsRemoved for the port. portStepsRemoved is set equal to masterStepsRemoved (see 10.3.9.3) after masterStepsRemoved is updated. The data type for portStepsRemoved is UInteger16. This variable is used both by the BMCA and the explicit port state configuration option.

10.3.10.13 rcvdAnnouncePtr: a pointer to a structure that contains the fields of a received Announce message. This variable is used both by the BMCA and the explicit port state configuration option.

10.3.10.14 rcvdMsg: a Boolean variable that is TRUE if a received Announce message is qualified, and FALSE if it is not qualified. This variable is used only by the BMCA, i.e., not by the explicit port state configuration option.

10.3.10.15 updtInfo: a Boolean variable that is set to TRUE to indicate that the PortAnnounceInformation state machine (see 10.3.12) should copy the newly determined masterPriority and masterStepsRemoved to portPriority and portStepsRemoved, respectively. This variable is used both by the BMCA and the explicit port state configuration option; however, its value does not impact the explicit port state configuration option (see the NOTE in 10.3.16.3, just after Figure 10-18).

10.3.10.16 annLeap61: a global variable in which the leap61 flag (see 10.6.2.2.8) of a received Announce message is saved. The data type for annLeap61 is Boolean. This variable is used both by the BMCA and the explicit port state configuration option.

10.3.10.17 annLeap59: a global variable in which the leap59 flag (see 10.6.2.2.8) of a received Announce message is saved. The data type for annLeap59 is Boolean. This variable is used both by the BMCA and the explicit port state configuration option.

10.3.10.18 annCurrentUtcOffsetValid: a global variable in which the currentUtcOffsetValid flag (see 10.6.2.2.8) of a received Announce message is saved. The data type for annCurrentUtcOffsetValid is Boolean. This variable is used both by the BMCA and the explicit port state configuration option.

10.3.10.19 annPtpTimescale: a global variable in which the ptpTimescale flag (see 10.6.2.2.8) of a received Announce message is saved. The data type for annPtpTimescale is Boolean. This variable is used both by the BMCA and the explicit port state configuration option.

10.3.10.20 annTimeTraceable: a global variable in which the timeTraceable flag (see 10.6.2.2.8) of a received Announce message is saved. The data type for annTimeTraceable is Boolean. This variable is used both by the BMCA and the explicit port state configuration option.

10.3.10.21 annFrequencyTraceable: a global variable in which the frequencyTraceable flag (see 10.6.2.2.8) of a received Announce message is saved. The data type for annFrequencyTraceable is Boolean. This variable is used both by the BMCA and the explicit port state configuration option.

10.3.10.22 annCurrentUtcOffset: a global variable in which the currentUtcOffset field (see 10.6.3.2.1) of a received Announce message is saved. The data type for annCurrentUtcOffset is Integer16. This variable is used both by the BMCA and the explicit port state configuration option.

10.3.10.23 annTimeSource: a global variable in which the timeSource field (see 10.6.3.2.1) of a received Announce message is saved. The data type for annTimeSource is TimeSource (see 8.6.2.7). This variable is used both by the BMCA and the explicit port state configuration option.

10.3.11 PortAnnounceReceive state machine

10.3.11.1 State machine variables

The following variables are used in the state diagram of 10.3.11.3:

10.3.11.1.1 rcvdAnnouncePAR: a Boolean variable that notifies the current state machine when Announce message information is received from the MD entity of the same port. This variable is reset by this state machine.

10.3.11.2 State machine functions

10.3.11.2.1 qualifyAnnounce (rcvdAnnouncePtr): qualifies the received Announce message pointed to by rcvdAnnouncePtr as follows:

a) If the Announce message was sent by the current PTP Instance, i.e., if sourcePortIdentity.clockIdentity (see 10.6.2.2.11 and 8.5.2) is equal to thisClock (see 10.2.4.22), the Announce message is not qualified and FALSE is returned;
b) If the stepsRemoved field is greater than or equal to 255, the Announce message is not qualified and FALSE is returned;
c) If a path trace TLV is present and one of the elements of the pathSequence array field of the path trace TLV is equal to thisClock (i.e., the clockIdentity of the current PTP Instance, see 10.2.4.22), the Announce message is not qualified and FALSE is returned;
d) Otherwise, the Announce message is qualified and TRUE is returned. If a path trace TLV is present and the portState of the port is SlavePort, the pathSequence array field of the TLV is copied to the global array pathTrace, and thisClock is appended to pathTrace (i.e., is added to the end of the array). If a path trace TLV is not present, the pathTrace array is set to the empty array (i.e., an array of zero elements). See 10.3.9.23 for a description of the path trace feature.

10.3.11.3 State diagram

The PortAnnounceReceive state machine shall implement the function specified by the state diagram in Figure 10-13, the local variables specified in 10.3.11.1, the functions specified in 10.3.11.2, and the relevant global variables specified in 10.2.4, 10.2.5, 10.3.10, and 11.2.13. The state machine is not used if externalPortConfigurationEnabled is TRUE. The state machine receives Announce information from the MD entity of the same port, determines if the Announce message is qualified, and if so, sets the rcvdMsg variable.

10.3.12 PortAnnounceInformation state machine

10.3.12.1 State machine variables

The following variables are used in the state diagram of 10.3.12.3:

10.3.12.1.1 announceReceiptTimeoutTime: a variable used to save the time at which announce receipt timeout occurs. The data type for announceReceiptTimeoutTime is UScaledNs.

10.3.12.1.2 messagePriorityPAI: the messagePriorityVector corresponding to the received Announce information. The data type for messagePriorityPAI is UInteger224 (see 10.3.4).

10.3.12.1.3 rcvdInfo: an Enumeration2 that holds the value returned by rcvInfo() (see 10.3.12.2.1).

10.3.12.2 State machine functions

10.3.12.2.1 rcvInfo (rcvdAnnouncePtr): decodes the messagePriorityVector (see 10.3.4 and 10.3.5) and stepsRemoved 10.6.3.2.6) field from the Announce information pointed to by rcvdAnnouncePtr (see 10.3.10.13), and then does the following:

a) Stores the messagePriorityVector and stepsRemoved field value in messagePriorityPAI and messageStepsRemoved, respectively, and then
    1) Returns RepeatedMasterInfo if the received message conveys the port state MasterPort, and the messagePriorityVector is the same as the portPriorityVector of the port, else
    2) Returns SuperiorMasterInfo if the received message conveys the port state MasterPort, and the messagePriorityVector is superior to the portPriorityVector of the port, else
    3) Returns InferiorMasterInfo if the received message conveys the port state MasterPort, and the messagePriorityVector is worse than the portPriorityVector of the port, else

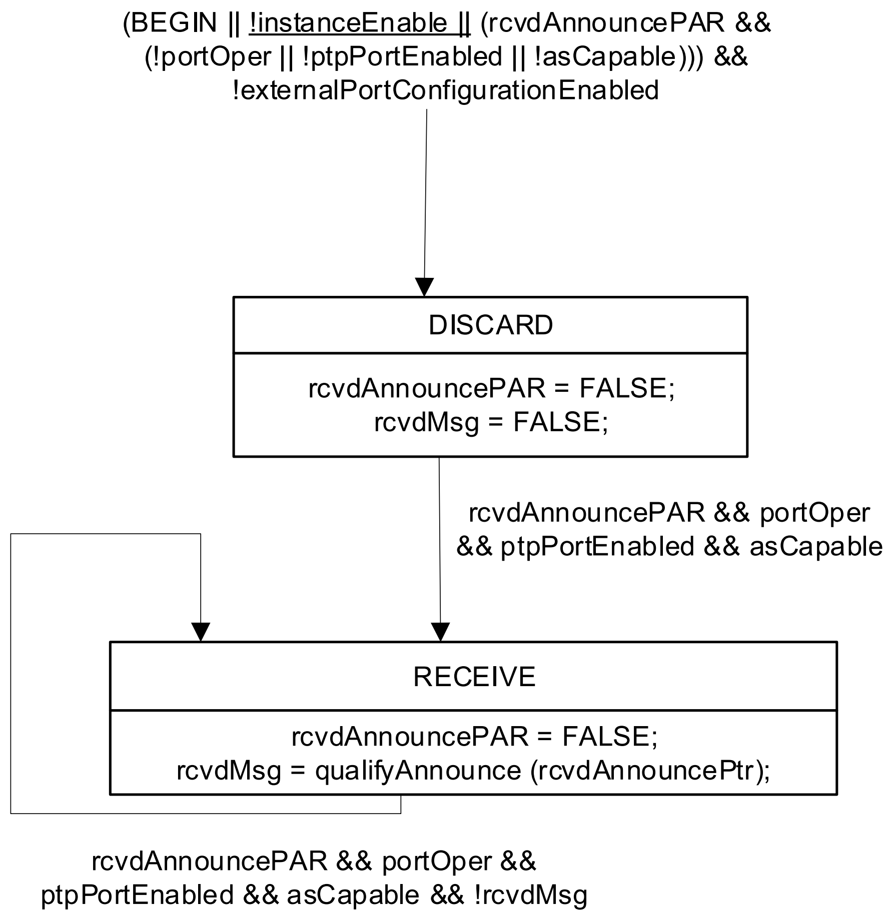


Figure 10-13—PortAnnounceReceive state machine

4) Returns OtherInfo.

NOTE—In accordance with 10.3.5, the messagePriorityVector is superior to the portPriorityVector of the port if, and only if, the messagePriorityVector is better than the portPriorityVector, or the Announce message has been transmitted from the same master PTP Instance and MasterPort as the portPriorityVector. In (1) - (4) above, rcvInfo() first checks whether the messagePriorityVector and portPriorityVector are the same (and the received message conveys the port state MasterPort), before checking whether the messagePriorityVector is superior to the portPriorityVector. This is because RepeatedMasterInfo needs to be returned if the messagePriorityVector and portPriorityVector are the same, while SuperiorMasterInfo needs to be returned in other instances where the Announce message has been transmitted from the same master PTP Instance and MasterPort as the portPriorityVector (if the test for SuperiorMasterInfo were done before the test for RepeatedMasterInfo, SuperiorMasterInfo would be returned for the cases where RepeatedMasterInfo is desired).

10.3.12.2.2 recordOtherAnnounceInfo(): saves the flags leap61, leap59, currentUtcOffsetValid,ptpTimescale, timeTraceable, and frequencyTraceable, and the fields currentUtcOffset and timeSource, of the received Announce message for this port. The values are saved in the per-port global variables annLeap61, annLeap59, annCurrentUtcOffsetValid, annPtpTimescale, annTimeTraceable, annFrequencyTraceable, annCurrentUtcOffset, and annTimeSource (see 10.3.10.16 through 10.3.10.23).

10.3.12.3 State diagram

The PortAnnounceInformation state machine shall implement the function specified by the state diagram in Figure 10-14, the local variables specified in 10.3.12.1, the functions specified in 10.3.12.2, and the relevant global variables specified in 10.2.4, 10.2.5, 10.3.8, 10.3.10, and 11.2.13. This state machine is used only if externalPortConfigurationEnabled is FALSE (if this variable is TRUE, the PortAnnounceInformationExt state machine is used instead). The state machine receives new qualified Announce information from the PortAnnounceReceive state machine (see 10.3.11) of the same port and determines if the Announce information is better than the current best master information it knows about. The state machine also updates the current best master information when it receives updated port state information from the PortStateSelection state machine (see 10.3.13), and when announce receipt timeout or, in the case where gmPresent is TRUE, sync receipt timeout occurs.

10.3.13 PortStateSelection state machine

10.3.13.1 State machine variables

The following variables are used in the state diagram of 10.3.13.3:

10.3.13.1.1 systemIdentityChange: a Boolean variable that notifies the current state machine when the systemIdentity (see 10.3.2) of this PTP Instance has changed. This variable is reset by this state machine.

The systemIdentity changes when at least one of the attributes priority1, clockClass, clockAccuracy, offsetScaledLogVariance, and priority2 changes, e.g., due to management action, degradation or loss of the ClockSource, etc. The systemIdentity also includes the attribute clockIdentity, but this attribute does not change.

10.3.13.1.2 asymmetryMeasurementModeChange: a Boolean variable that notifies the current state machine when the per-port variable asymmetryMeasurementMode (see 10.2.5.2) changes on any port. This variable is reset by this state machine. There is one instance of asymmetryMeasurementModeChange for all the domains (per port). The variable is accessible by all the domains.

10.3.13.2 State machine functions

10.3.13.2.1 updtStateDisabledTree(): sets all the elements of the selectedState array (see 10.2.4.20) to DisabledPort. Sets lastGmPriority to all ones. Sets the pathTrace array (see 10.3.9.23) to contain the single element thisClock (see 10.2.4.22).

10.3.13.2.2 clearReselectTree(): sets all the elements of the reselect array (see 10.3.9.1) to FALSE.

10.3.13.2.3 setSelectedTree(): sets all the elements of the selected array (see 10.3.9.2) to TRUE.

10.3.13.2.4 updtStatesTree(): performs the following operations (see 10.3.4 and 10.3.5 for details on the priority vectors):

a) Computes the gmPathPriorityVector for each port that has a portPriorityVector and for which neither announce receipt timeout nor, if gmPresent is TRUE, sync receipt timeout have occurred,
b) Saves gmPriority (see 10.3.9.21) in lastGmPriority (see 10.3.9.22), computes the gmPriorityVector for the PTP Instance and saves it in gmPriority, chosen as the best of the set consisting of the systemPriorityVector (for this PTP Instance) and the gmPathPriorityVector for each port for which the clockIdentity of the master port is not equal to thisClock (see 10.2.4.22),

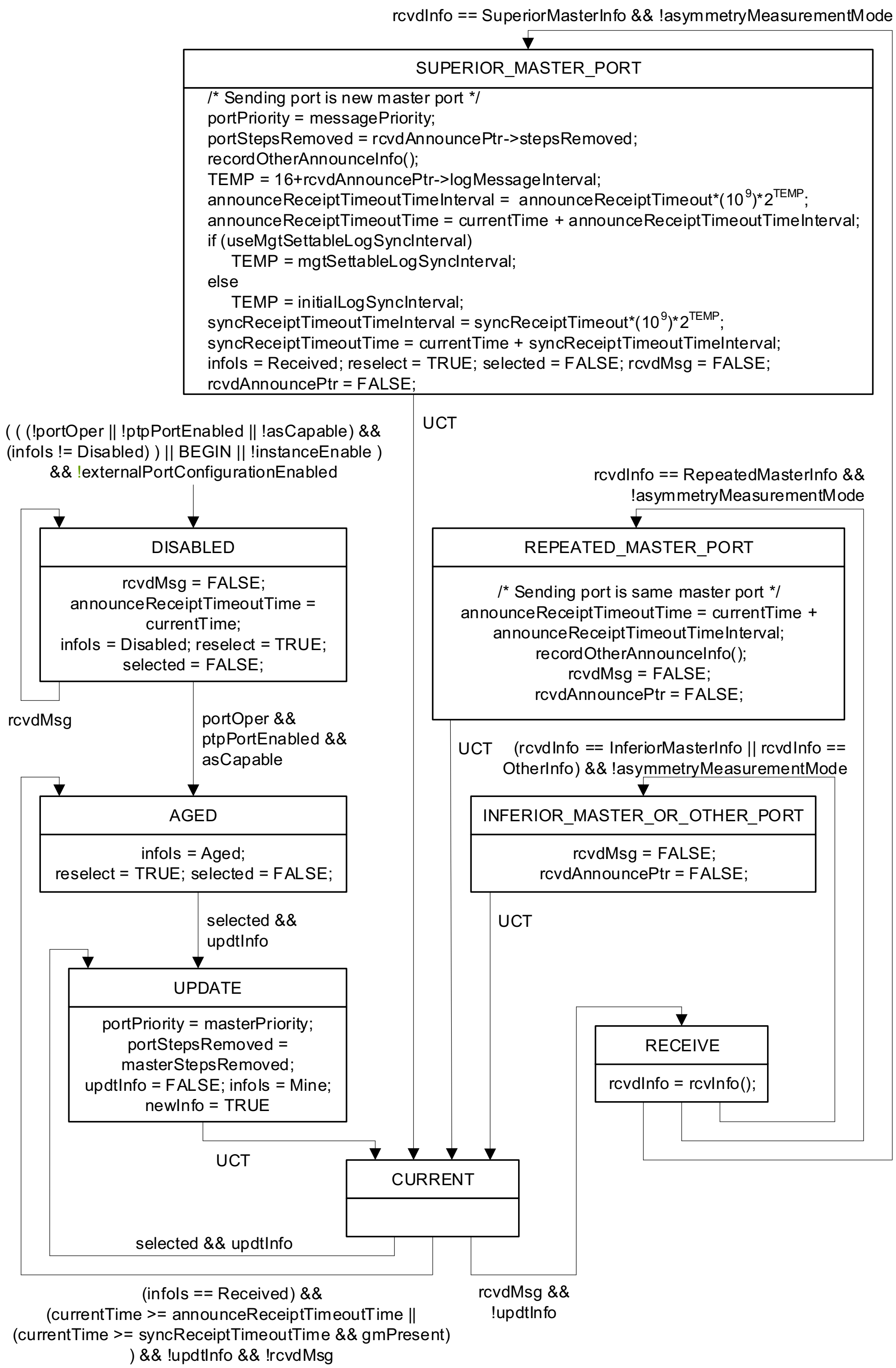


Figure 10-14—PortAnnounceInformation state machine

c) Sets the per PTP Instance global variables leap61, leap59, currentUtcOffsetValid, ptpTimescale, timeTraceable, frequencyTraceable, currentUtcOffset, and timeSource as follows:
   1) If the gmPriorityVector was set to the gmPathPriorityVector of one of the ports, then leap61, leap59, currentUtcOffsetValid,ptpTimescale, timeTraceable, frequencyTraceable, currentUtcOffset, and timeSource are set to annLeap61, annLeap59, annCurrentUtcOffsetValid, annPtpTimescale, annTimeTraceable, annFrequencyTraceable, annCurrentUtcOffset, and annTimeSource, respectively, for that port.
   2) If the gmPriorityVector was set to the systemPriorityVector, then leap61, leap59, currentUtcOffsetValid,ptpTimescale, timeTraceable, frequencyTraceable, currentUtcOffset, and timeSource are set to sysLeap61, sysLeap59, sysCurrentUtcOffsetValid,sysPtpTimescale, sysTimeTraceable, sysFrequencyTraceable, sysCurrentUtcOffset, and sysTimeSource, respectively.
d) Computes the masterPriorityVector for each port.
e) Computes masterStepsRemoved, which is equal to:
   1) messageStepsRemoved (see 10.3.10.9) for the port associated with the gmPriorityVector, incremented by 1, if the gmPriorityVector is not the systemPriorityVector, or
   2) 0 if the gmPriorityVector is the systemPriorityVector,
f) assigns the port state for port j and sets selectedState[j] equal to this port state, as follows, for j = 1, 2, ..., numberPorts:
   1) If the port is disabled (infoIs == Disabled), selectedState[j] is set to DisabledPort.
   2) If asymmetryMeasurementMode is TRUE, selectedState[j] is set to PassivePort and updtInfo is set to FALSE.
   3) If announce receipt timeout, or sync receipt timeout with gmPresent set to TRUE, have occurred (infoIs = Aged), updtInfo is set to TRUE and selectedState[j] is set to MasterPort.
   4) If the portPriorityVector was derived from another port on the PTP Instance or from the PTP Instance itself as the root (infoIs == Mine), selectedState[j] is set to MasterPort. In addition, updtInfo is set to TRUE if the portPriorityVector differs from the masterPriorityVector or portStepsRemoved differs from masterStepsRemoved.
   5) If the portPriorityVector was received in an Announce message and announce receipt timeout, or sync receipt timeout with gmPresent TRUE, have not occurred (infoIs == Received), and the gmPriorityVector is now derived from the portPriorityVector, selectedState[j] is set to SlavePort and updtInfo is set to FALSE.
   6) If the portPriorityVector was received in an Announce message and announce receipt timeout, or sync receipt timeout with gmPresent TRUE, have not occurred (infoIs == Received), the gmPriorityVector is not now derived from the portPriorityVector, the masterPriorityVector is not better than the portPriorityVector, and the sourcePortIdentity component of the portPriorityVector *does not* reflect another port on the PTP Instance, selectedState[j] is set to PassivePort and updtInfo is set to FALSE.
   7) If the portPriorityVector was received in an Announce message and announce receipt timeout, or sync receipt timeout with gmPresent TRUE, have not occurred (infoIs == Received), the gmPriorityVector is not now derived from the portPriorityVector, the masterPriorityVector is not better than the portPriorityVector, and the sourcePortIdentity component of the portPriorityVector *does* reflect another port on the PTP Instance, selectedState[j] set to PassivePort and updtInfo is set to FALSE.
   8) If the portPriorityVector was received in an Announce message and announce receipt timeout, or sync receipt timeout with gmPresent TRUE, have not occurred (infoIs == Received), the gmPriorityVector is not now derived from the portPriorityVector, and the masterPriorityVector is better than the portPriorityVector, selectedState[j]is set to MasterPort and updtInfo is set to TRUE.
g) Updates gmPresent as follows:
   1) gmPresent is set to TRUE if the priority1 field of the rootSystemIdentity of the gmPriorityVector is less than 255.
   2) gmPresent is set to FALSE if the priority1 field of the rootSystemIdentity of the gmPriorityVector is equal to 255.

h) Assigns the port state for port 0 (see 8.5.2.3), and sets selectedState[0], as follows:
   1) if selectedState[j] is set to SlavePort for any port with portNumber j, j = 1, 2, ..., numberPorts, selectedState[0] is set to PassivePort.
   2) if selectedState[j] is *not* set to SlavePort for any port with portNumber j, j = 1, 2, ..., numberPorts, selectedState[0] is set to SlavePort.

i) If the clockIdentity member of the systemIdentity (see 10.3.2) member of gmPriority (see 10.3.9.21) is equal to thisClock (see 10.2.4.22), i.e., if the current PTP Instance is the grandmaster, the pathTrace array is set to contain the single element thisClock (see 10.2.4.22).

10.3.13.3 State diagram

The PortStateSelection state machine shall implement the function specified by the state diagram in Figure 10-15, the functions specified in 10.3.13.1, and the relevant global variables specified in 10.2.4, 10.2.5, 10.3.8, 10.3.10, and 11.2.13. This state machine is used only if externalPortConfigurationEnabled is FALSE (if this variable is TRUE, the PortStateSettingExt state machine is used instead). The state machine updates the gmPathPriority vector for each port of the PTP Instance, the gmPriorityVector for the PTP Instance, and the masterPriorityVector for each port of the PTP Instance. The state machine determines the port state for each port and updates gmPresent.

10.3.14 PortAnnounceInformationExt state machine

10.3.14.1 State machine variables

The following variables are used in the state diagram of 10.3.14.3:

10.3.14.1.1 rcvdAnnouncePAIE: a Boolean variable that notifies the current state machine when Announce message information is received from the MD entity of the same port. This variable is reset by this state machine.

10.3.14.1.2 messagePriorityPAIE: the messagePriorityVector corresponding to the received Announce information. The data type for messagePriorityPAIE is UInteger224 (see 10.3.4).

10.3.14.2 State machine functions

10.3.14.2.1 rcvInfoExt (rcvdAnnouncePtr): decodes the messagePriorityVector (see 10.3.4 and 10.3.5) and stepsRemoved 10.6.3.2.6) field from the Announce information pointed to by rcvdAnnouncePtr (see 10.3.10.13), and then stores the messagePriorityVector and stepsRemoved field value in messagePriorityPAIE and messageStepsRemoved, respectively. If a path trace TLV is present in the Announce message and the portState of the port is SlavePort, the pathSequence array field of the TLV is copied to the global array pathTrace, and thisClock is appended to pathTrace (i.e., is added to the end of the array).

10.3.14.2.2 recordOtherAnnounceInfo(): saves the flags leap61, leap59, currentUtcOffsetValid,ptpTimescale, timeTraceable, and frequencyTraceable, and the fields currentUtcOffset and timeSource, of the received Announce message for this port. The values are saved in the per-port global variables annLeap61, annLeap59, annCurrentUtcOffsetValid, annPtpTimescale, annTimeTraceable, annFrequencyTraceable, annCurrentUtcOffset, and annTimeSource (see 10.3.10.16 through 10.3.10.23).

10.3.14.3 State diagram

The PortAnnounceInformationExt state machine shall implement the function specified by the state diagram in Figure 10-16, the local variables specified in 10.3.14.1, the functions specified in 10.3.14.2, and the relevant global variables specified in 10.2.4, 10.2.5, 10.3.8, 10.3.10, and 11.2.13. This state machine is used

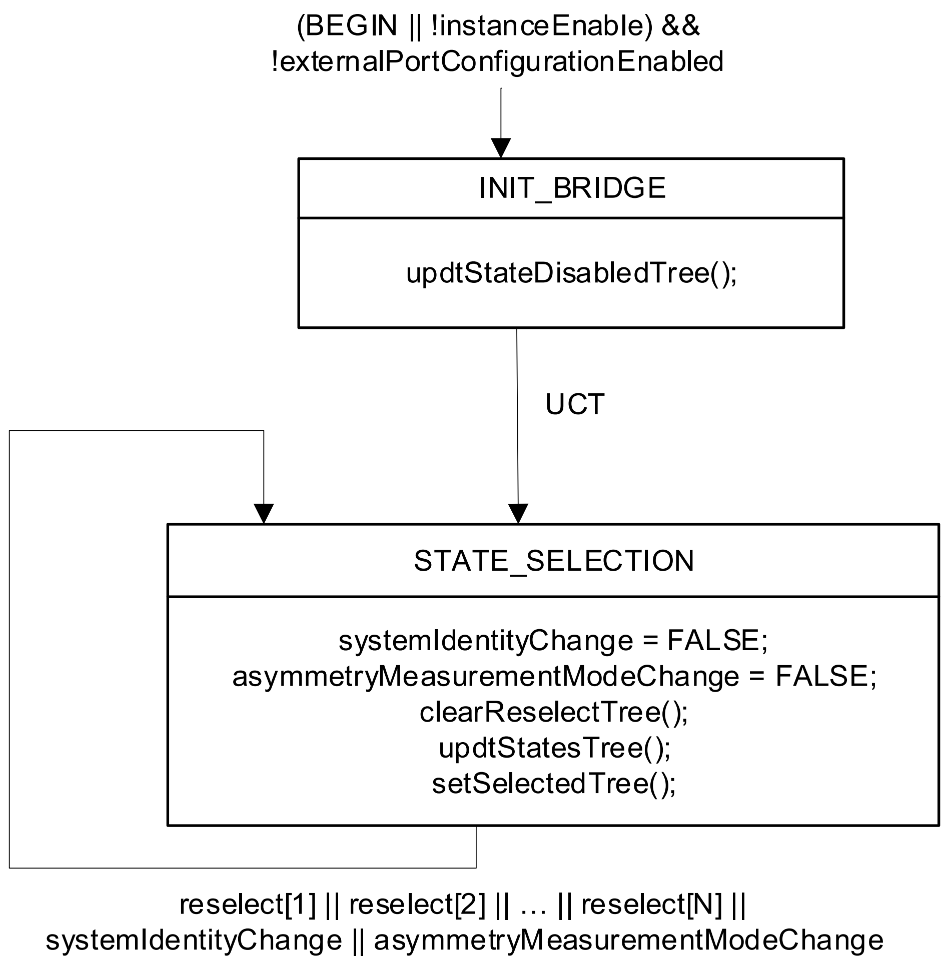


Figure 10-15—PortStateSelection state machine

only if externalPortConfigurationEnabled is TRUE (if this variable is FALSE, the PortAnnounceInformation state machine of 10.3.12.3 is used instead). The state machine receives Announce information from the MD entity of the same port and saves the information.

10.3.15 PortStateSettingExt state machine

10.3.15.1 State machine variables

The following variables are used in the state diagram of 10.3.15.3:

10.3.15.1.1 disabledExt: a Boolean variable that notifies the current state machine (i.e., when it is set to TRUE) when at least one of the variables portOper, ptpPortEnabled, or asCapable, for this port, has changed from TRUE to FALSE. This variable is reset by this state machine.

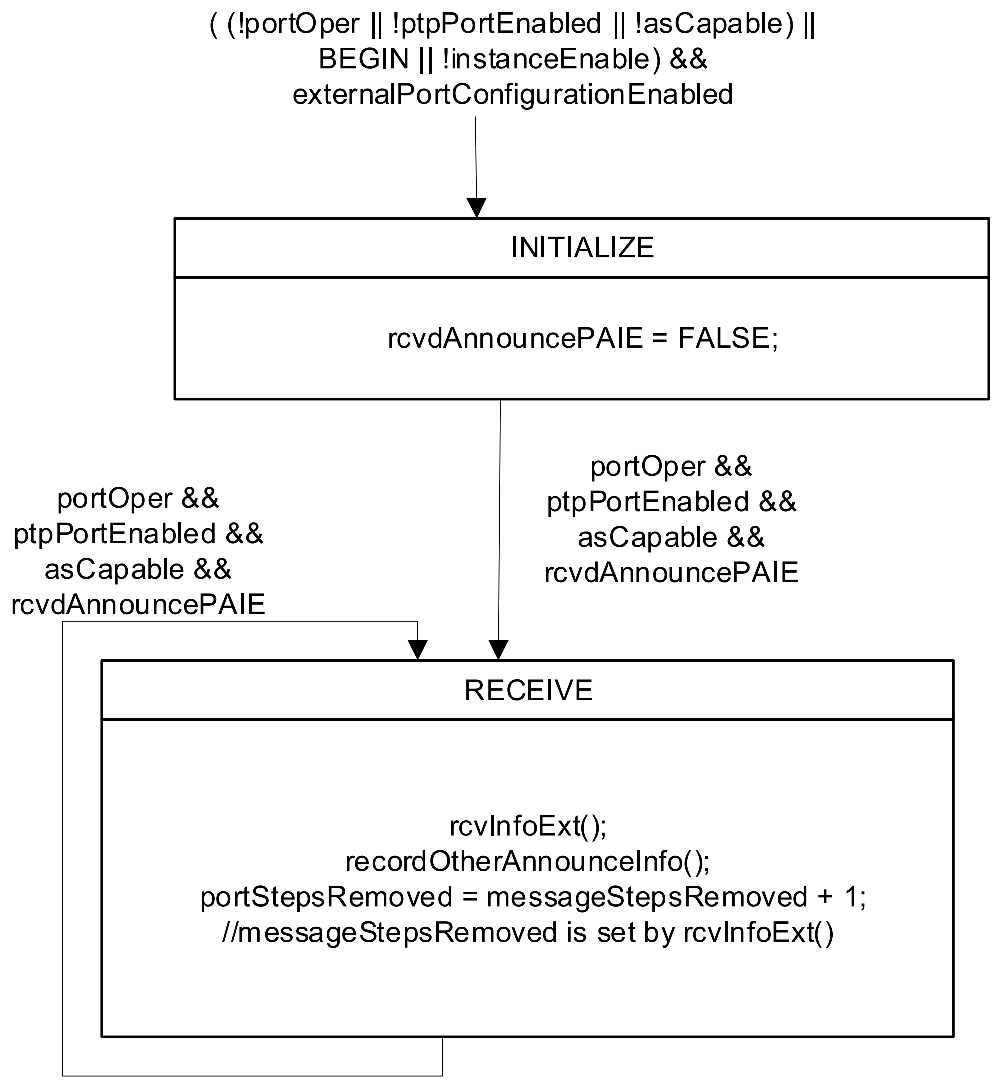


Figure 10-16—PortAnnounceInformationExt state machine

10.3.15.1.2 reenabledExt: a Boolean variable that notifies the current state machine (i.e., when it is set to TRUE) when all of the variables portOper, ptpPortEnabled, and asCapable, for this port, that are FALSE have changed to TRUE. This variable is reset by this state machine.

10.3.15.1.3 a**symmetryMeasurementModeChangeThisPort:** a Boolean variable that notifies the current state machine when the per-port variable asymmetryMeasurementMode (see 10.2.5.2) changes on this port. This variable is reset by this state machine. There is one instance of asymmetryMeasurementModeChangeThisPort for all the domains (per port). The variable is accessible by all the domains.

10.3.15.1.4 rcvdPortStateInd: a Boolean variable that notifies the current state machine (i.e., when it is set to TRUE) when the port state of this port has been externally set. This variable is reset by this state machine.

10.3.15.1.5 portStateInd: an Enumeration2 that indicates the port state that has been set. The values are MasterPort, SlavePort, PassivePort.

NOTE—The port state can be externally set to DisabledPort by setting portOper or ptpPortEnabled to FALSE. The port state is set to DisabledPort when asCapable becomes FALSE.

10.3.15.2 State machine functions

10.3.15.2.1 resetStateTree(j): sets selectedState[j] (see 10.2.4.20) to externalPortConfigurationPortDS.desiredState. Sets the pathTrace array (see 10.3.9.23) to contain the single element thisClock (see 10.2.4.22) if no port of the PTP Instance has the port state SlavePort.

10.3.15.2.2 updtPortState(j): performs the following operations for port j (see 10.3.4 and 10.3.5 for details on the priority vectors):

a) Sets the per PTP Instance global variables leap61, leap59, currentUtcOffsetValid, ptpTimescale, timeTraceable, frequencyTraceable, currentUtcOffset, and timeSource as follows:
   1) If the port state of any port of this PTP Instance other than port 0 (see 8.5.2.3) is SlavePort, then leap61, leap59, currentUtcOffsetValid, ptpTimescale, timeTraceable, frequencyTraceable, currentUtcOffset, and timeSource are set to annLeap61, annLeap59, annCurrentUtcOffsetValid,annPtpTimescale, annTimeTraceable, annFrequencyTraceable, annCurrentUtcOffset, and annTimeSource, respectively, for that port.
   2) If no port of this PTP Instance other than port 0 has the port state SlavePort, then leap61, leap59, currentUtcOffsetValid,ptpTimescale, timeTraceable, frequencyTraceable, currentUtcOffset, and timeSource are set to sysLeap61, sysLeap59, sysCurrentUtcOffsetValid,sysPtpTimescale, sysTimeTraceable, sysFrequencyTraceable, sysCurrentUtcOffset, and sysTimeSource, respectively.
b) Computes masterStepsRemoved as follows:
   1) If the port state of any port of this PTP Instance other than port 0 is SlavePort, then masterStepsRemoved is set equal to portStepsRemoved for that port.
   2) If no port of this PTP Instance other than port 0 has the port state SlavePort, then masterStepsRemoved is set equal to 0.
c) Assigns the port state for port j and sets selectedState[j] equal to this port state, as follows:
   1) If disabledExt is TRUE, selectedState[j] is set to DisabledPort, else
   2) If asymmetryMeasurementMode is TRUE, selectedState[j] is set to PassivePort, else
   3) selectedState[j] is set to portStateInd.
d) Updates gmPresent as follows:
   1) If the port state of any port of this PTP Instance other than port 0 is SlavePort and the priority1 field of the rootSystemIdentity of the messagePriorityPAIE of the slave port is less than 255, gmPresent is set to TRUE, else
   2) If the port state of any port of this PTP Instance other than port 0 is SlavePort and the priority1 field of the rootSystemIdentity of the messagePriorityPAIE of the slave port is equal to 255, gmPresent is set to FALSE, else
   3) If no port of this PTP Instance other than port 0 has the port state SlavePort, gmPresent is set to TRUE if priority1 for this PTP Instance is less than 255 and FALSE if priority1 for this PTP Instance is equal to 255.
e) Assigns the port state for port 0, and sets selectedState[0], as follows:
   1) if selectedState[j] is set to SlavePort, selectedState[0] is set to PassivePort.
   2) if selectedState[j] is *not* set to SlavePort and selectedState[k] is not equal to SlavePort for every k not equal to 0 or j, selectedState[0] is set to SlavePort.
f) Computes the gmPriorityVector as follows:

1) if selectedState[j] is set to SlavePort, the gmPriorityVector is set equal to messagePriorityPAIE for port j.
2) if selectedState[j] is *not* set to SlavePort and selectedState[k] is not equal to SlavePort for every k not equal to 0 or j, the gmPriorityVector is set equal to the systemPriorityVector.

g) Computes the masterPriorityVector for port j.
h) If no port of this PTP Instance has the port state SlavePort, the pathTrace array is set to contain the single element thisClock (see 10.2.4.22).

10.3.15.3 State diagram

The PortStateSettingExt state machine shall implement the function specified by the state diagram in Figure 10-17, the local variables specified in 10.3.15.1, the functions specified in 10.3.15.2, and the relevant global variables specified in 10.2.4, 10.2.5, 10.3.8, 10.3.10, and 11.2.13. This state machine is used only if externalPortConfigurationEnabled is TRUE (if this variable is FALSE, the PortStateSelection state machine of 10.3.13.3 is used instead). A separate instance of this state machine runs on each port (unlike the PortStateSelection state machine, for which a single instance runs in the PTP Instance and performs operations on all the ports).

The state machine updates the gmPriorityVector for the PTP Instance and the masterPriorityVector for each port of the PTP Instance. The state machine determines the port state for each port and updates gmPresent.

NOTE—It is possible to use the external port configuration mechanism to misconfigure the network, e.g., to produce a

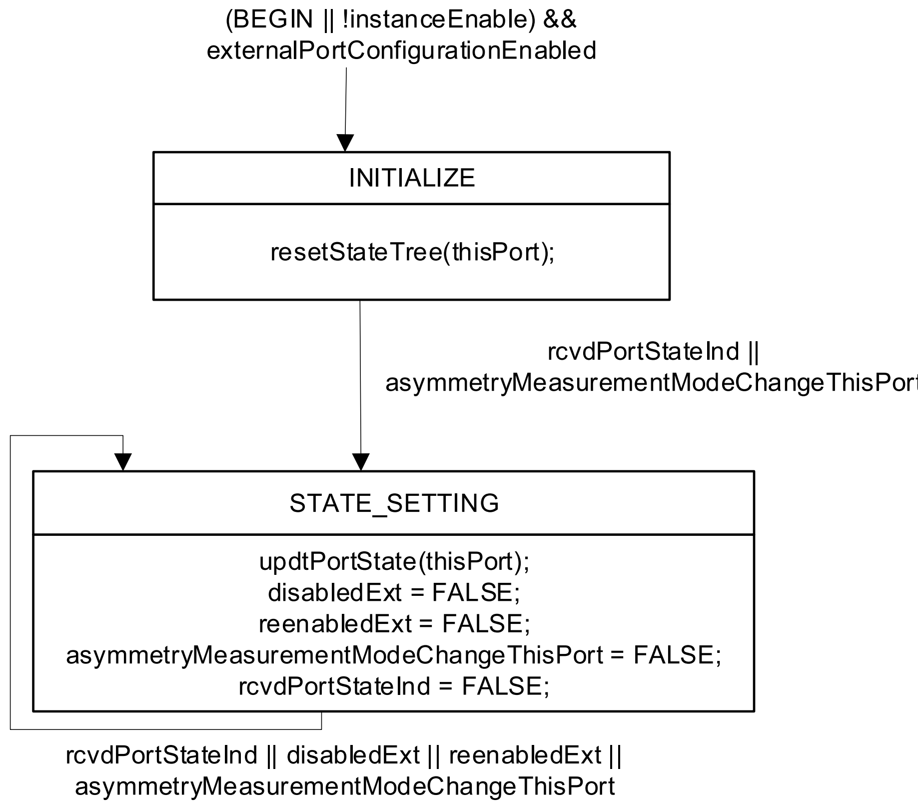


Figure 10-17—PortStateSettingExt state machine

configuration where one or more PTP Instances have more than one slave port. Detecting and correcting misconfiguations is outside the scope of this standard.

10.3.16 PortAnnounceTransmit state machine

10.3.16.1 State machine variables

The following variables are used in the state diagram of 10.3.16.3:

10.3.16.1.1 announceSendTime: the time, relative to the LocalClock, at which the next transmission of Announce information is to occur. The data type for announceSendTime is UScaledNs.

10.3.16.1.2 numberAnnounceTransmissions: a count of the number of consecutive Announce message transmissions after the AnnounceIntervalSetting state machine (see Figure 10-19) has set announceSlowdown (see 10.3.10.2) to TRUE. The data type for numberAnnounceTransmissions is UInteger8.

10.3.16.1.3 interval2: a local variable that holds either announceInterval or oldAnnounceInterval. The data type for interval2 is UScaledNs.

10.3.16.2 State machine functions

10.3.16.2.1 txAnnounce (): transmits Announce information to the MD entity of this port. The Announce information is set as follows:

a) The components of the messagePriorityVector are set to the values of the respective components of the masterPriorityVector of this port.
b) The grandmasterIdentity, grandmasterClockQuality, grandmasterPriority1, and grandmasterPriority2 fields of the Announce message are set equal to the corresponding components of the messagePriorityVector.
c) The value of the stepsRemoved field of the Announce message is set equal to masterStepsRemoved.
d) The Announce message flags leap61, leap59, currentUtcOffsetValid,ptpTimescale, timeTraceable, and frequencyTraceable, and the Announce message fields currentUtcOffset and timeSource, are set equal to the values of the global variables leap61, leap59, currentUtcOffsetValid, ptpTimescale, timeTraceable, frequencyTraceable, currentUtcOffset, and timeSource, respectively (see 10.3.9.4 through 10.3.9.11).
e) The sequenceId field of the Announce message is set in accordance with 10.5.7.
f) A path trace TLV (see 10.6.3.3) is constructed, with its pathSequence field (see 10.6.3.3.4) set equal to the pathTrace array (see 10.3.9.23). If appending the path trace TLV to the Announce message does not cause the media-dependent layer frame to exceed any respective maximum size, the path trace TLV is appended to the Announce message; otherwise, it is not appended. If the pathTrace array is empty, the path trace TLV is not appended. See 10.3.9.23 for a description of the path trace feature.

10.3.16.3 State diagram

The PortAnnounceTransmit state machine shall implement the function specified by the state diagram in Figure 10-18, the local variables specified in 10.3.16.1, the function specified in 10.3.16.2, and the relevant global variables specified in 10.2.4, 10.2.5, 10.3.10, and 11.2.13. The state machine transmits Announce information to the MD entity when an announce interval has elapsed, port states have been updated, and portPriority and portStepsRemoved information has been updated with newly determined masterPriority and masterStepsRemoved information.

NOTE—When the external port configuration option is used (i.e., externalPortConfigurationEnabled is TRUE, see 10.3.9.24) the values of the variables updtInfo and selected do not affect the operation of the PortAnnounceTransmit state machine because the terms of the conditions in which they appear, i.e., (selected && !updtInfo) || externalPortConfiguartionEnabled == TRUE, evaluate to TRUE when externalPortConfigurationEnabled is TRUE.

10.3.17 AnnounceIntervalSetting state machine

10.3.17.1 State machine variables

The following variables are used in the state diagram of 10.3.17.3:

10.3.17.1.1 rcvdSignalingMsg2: a Boolean variable that notifies the current state machine when a Signaling message that contains a Message Interval Request TLV (see 10.6.4.3) is received. This variable is reset by the current state machine.

10.3.17.1.2 rcvdSignalingPtrAIS: a pointer to a structure whose members contain the values of the fields of the received Signaling message that contains a Message Interval Request TLV (see 10.6.4.3).

10.3.17.1.3 logSupportedAnnounceIntervalMax: the maximum supported logarithm to base 2 of the announce interval. The data type for logSupportedAnnounceIntervalMax is Integer8.

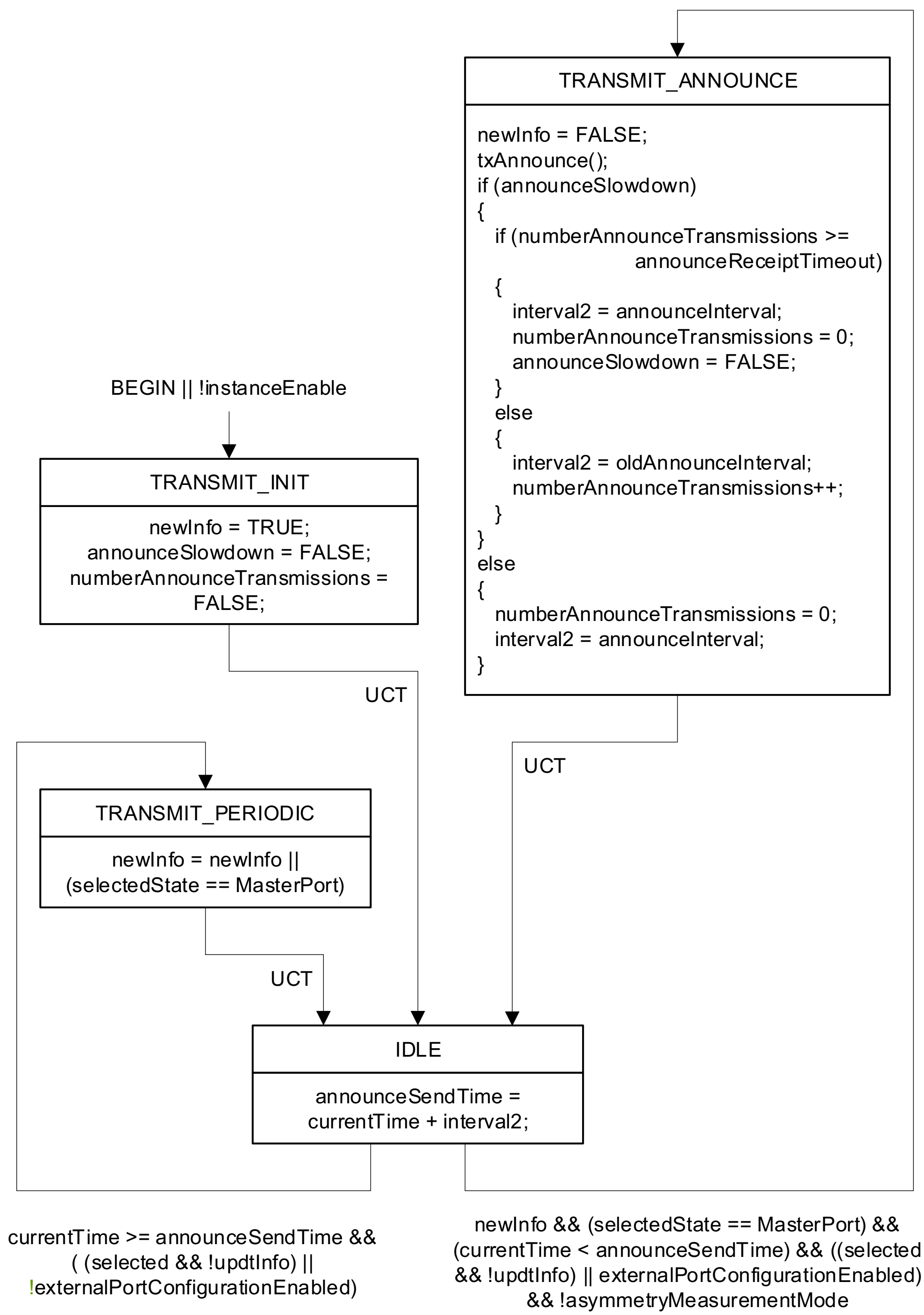


Figure 10-18—PortAnnounceTransmit state machine

10.3.17.1.4 logSupportedClosestLongerAnnounceInterval: the logarithm to base 2 of the announce interval, such that logSupportedClosestLongerAnnounceInterval > logRequestedAnnounceInterval, that is numerically closest to logRequestedAnnounceInterval, where logRequestedAnnounceInterval is the argument of the function computeLogAnnounceInterval() (see 10.3.17.2.2). The data type for logSupportedClosestLongerAnnounceInterval is Integer8.

10.3.17.1.5 computedLogAnnounceInterval: a variable used to hold the result of the function computeLogAnnounceInterval(). The data type for computedLogAnnounceInterval is Integer8.

10.3.17.2 State machine functions

10.3.17.2.1 isSupportedLogAnnounceInterval (logAnnounceInterval): a Boolean function that returns TRUE if the announce interval given by the argument logAnnounceInterval is supported by the gPTP Port, and FALSE if the announce interval is not supported by the gPTP Port. The argument logAnnounceInterval has the same data type and format as the field logAnnounceInterval of the message interval request TLV (see 10.6.4.3.8).

10.3.17.2.2 computeLogAnnounceInterval (logRequestedAnnounceInterval): An Integer8 function that computes and returns the logAnnounceInterval, based on the logRequestedAnnounceInterval. This function is defined as indicated below. It is defined so that the detailed code that it invokes does not need to be placed into the state machine diagram.

```
Integer8 computeLogAnnounceInterval (logRequestedAnounceInterval)
Integer8 logRequestedAnnounceInterval;
{
        Integer8 logSupportedAnnounceIntervalMax,
                        logSupportedClosestLongerAnnounceInterval;
        if (isSupportedLogAnnounceInterval (logRequestedAnnounceInterval))
                // The requested Announce Interval is supported and returned
                return (logRequestedAnnounceInterval)
        else
        {
                if (logRequestedAnnounceInterval > logSupportedAnnounceIntervalMax)
                        // Return the fasted supported rate, even if faster than the requested rate
                        return (logSupportedAnnounceIntervalMax);
                else
                        // Return the fastest supported rate that is still slower than
                        // the requested rate.
                        return (logSupportedClosestLongerAnnounceInterval);
        }
}
```

10.3.17.3 State diagram

The AnnounceIntervalSetting state machine shall implement the function specified by the state diagram in Figure 10-19, the local variables specified in 10.3.17.1, the messages specified in 10.6, the relevant global variables specified in 10.2.5, the relevant managed objects specified in 14.8, and the relevant timing attributes specified in 10.7. This state machine is responsible for setting the global variables that give the duration of the mean interval between successive Announce messages, both at initialization and in response to the receipt of a Signaling message that contains a Message Interval Request TLV (see 10.6.4.3).

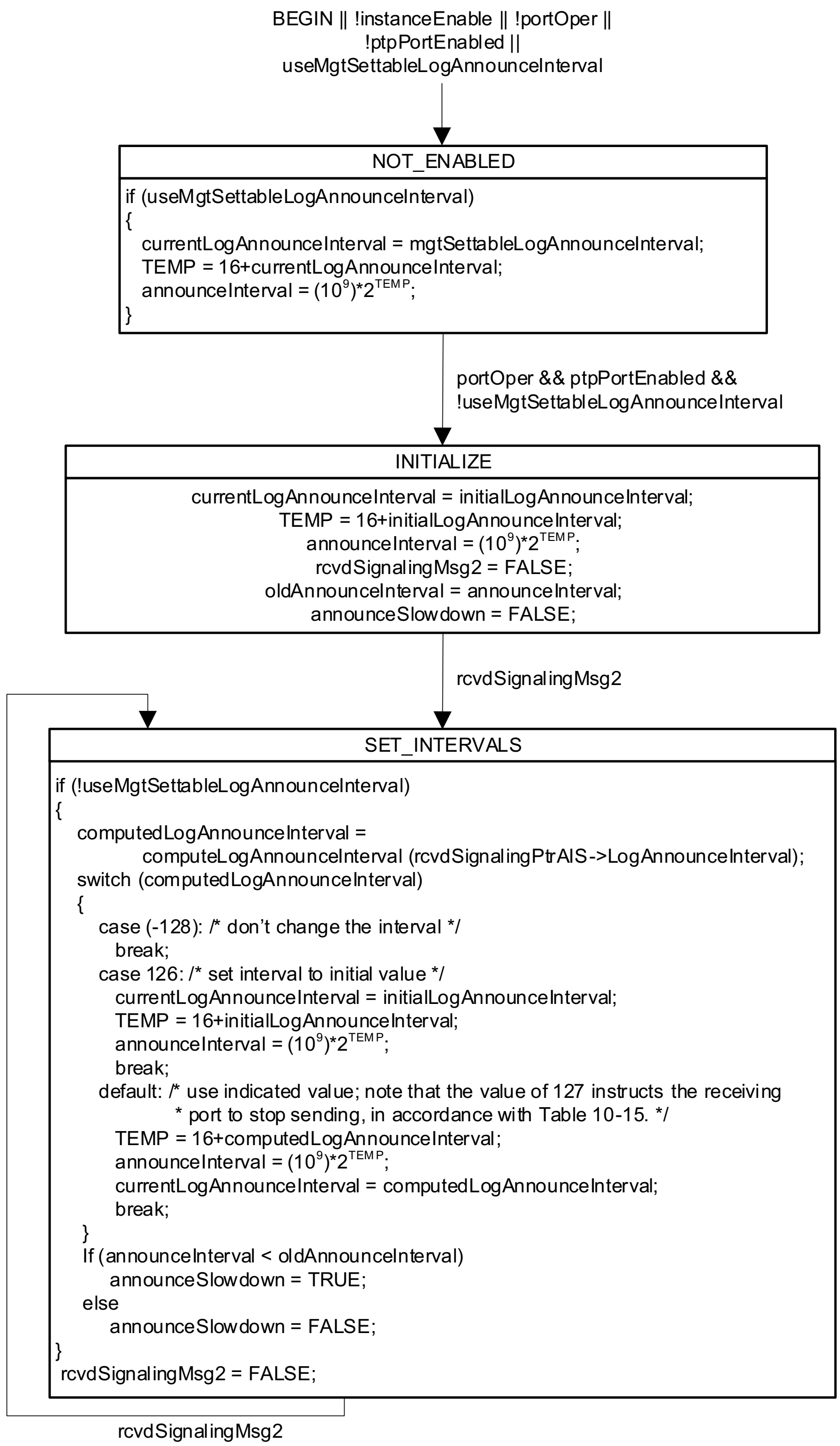


Figure 10-19—AnnounceIntervalSetting state machine

10.3.18 SyncIntervalSetting state machine

10.3.18.1 State machine variables

The following variables are used in the state diagram of 10.3.18.3:

10.3.18.1.1 rcvdSignalingMsg3: a Boolean variable that notifies the current state machine when a Signaling message that contains a Message Interval Request TLV (see 10.6.4.3) is received. This variable is reset by the current state machine.

10.3.18.1.2 rcvdSignalingPtrSIS: a pointer to a structure whose members contain the values of the fields of the received Signaling message that contains a Message Interval Request TLV (see 10.6.4.3).

10.3.18.1.3 logSupportedSyncIntervalMax: the maximum supported logarithm to base 2 of the sync interval. The data type for logSupportedSyncIntervalMax is Integer8.

10.3.18.1.4 logSupportedClosestLongerSyncInterval: the logarithm to base 2 of the sync interval, such that logSupportedClosestLongerSyncInterval > logRequestedSyncInterval, that is numerically closest to logRequestedSyncInterval, where logRequestedSyncInterval is the argument of the function computeLogSyncInterval() (see 10.3.18.2.2). The data type for logSupportedClosestLongerSyncInterval is Integer8.

10.3.18.1.5 computedLogSyncInterval: a variable used to hold the result of the function computeLogSyncInterval(). The data type for computedLogSyncInterval is Integer8.

10.3.18.2 State machine functions

10.3.18.2.1 isSupportedLogSyncInterval (logSyncInterval): a Boolean function that returns TRUE if the sync interval given by the argument logSyncInterval is supported by the gPTP Port, and FALSE if the sync interval is not supported by the gPTP Port. The argument logSyncInterval has the same data type and format as the field syncInterval of the message interval request TLV (see 10.6.4.3.7).

10.3.18.2.2 computeLogSyncInterval (logRequestedSyncInterval): An Integer8 function that computes and returns the logSyncInterval, based on the logRequestedSyncInterval. This function is defined as indicated below. It is defined so that the detailed code that it invokes does not need to be placed into the state machine diagram.

```
Integer8 computeLogSyncInterval (logRequestedSyncInterval)
Integer8 logRequestedSyncInterval;
{
	Integer8 logSupportedSyncIntervalMax, logSupportedClosestLongerSyncInterval;
	if (isSupportedLogSyncInterval (logRequestedSyncInterval))
		// The requested Sync Interval is supported and returned
		return (logRequestedSyncInterval)
	else
	{
		if (logRequestedSyncInterval > logSupportedSyncIntervalMax)
			// Return the fasted supported rate, even if faster than the requested rate
			return (logSupportedSyncIntervalMax);
		else
			// Return the fastest supported rate that is still slower than
			// the requested rate.
			return (logSupportedClosestLongerSyncInterval);
	}
```

}

10.3.18.3 State diagram

The SyncIntervalSetting state machine shall implement the function specified by the state diagram in Figure 10-20, the local variables specified in 10.3.18.1, the messages specified in 10.6, the relevant global variables specified in 10.2.5 the relevant managed objects specified in 14.8, and the relevant timing attributes specified in 10.7. This state machine is responsible for setting the global variables that give the duration of the mean intervals between successive Sync messages, both at initialization and in response to the receipt of a Signaling message that contains a Message Interval Request TLV (see 10.6.4.3).

10.4 State machines related to signaling gPTP protocol capability

10.4.1 GptpCapableTransmit state machine

10.4.1.1 State machine variables

The following variables are used in the state diagram of 10.4.1.3:

10.4.1.1.1 intervalTimer: a variable used to save the time at which the gPTP capable message interval timer is set (see Figure 10-21). A Signaling message containing a gPTP capable TLV is sent when this timer expires. The data type for intervalTimer is UScaledNs.

10.4.1.1.2 txSignalingMsgPtr: a pointer to a structure whose members contain the values of the fields of a Signaling message to be transmitted, which contains a gPTP capable TLV (see 10.6.4.4).

10.4.1.1.3 interval3: a local variable that holds either gPtpCapableMessageInterval or oldGptpCapableMessageInterval. The data type for interval3 is UScaledNs.

10.4.1.1.4 numberGptpCapableMessageTransmissions: a count of the number of consecutive transmissions of Signaling messages that contain a gPTP capable TLV, after the GptpCapableIntervalSetting state machine (see Figure 10-23) has set gPtpCapableMessageSlowdown (see 10.2.5.19) to TRUE. The data type for numberGptpCapableMessageTransmissions is UInteger8.

10.4.1.2 State machine functions

10.4.1.2.1 setGptpCapableTlv(): creates a structure containing the parameters of a Signaling message that contains a gPTP capable TLV, to be transmitted (see 10.6.4), and returns a pointer to this structure. The parameters are set as follows:

a) logGptpCapableMessageInterval is set to the value of the managed object logGptpCapableMessageInterval (see 14.8.41), and
b) remaining parameters are set as specified in 10.6.4 and 10.6.4.4.

10.4.1.3 State diagram

The GptpCapableTransmit state machine shall implement the function specified by the state diagram in Figure 10-21, the local variables specified in 10.4.1.1, the functions specified in 10.4.1.2, the relevant parameters specified in 10.6.4 and 10.6.4.4, and the relevant timing attributes specified in 10.7. This state machine is responsible for setting the parameters of each Signaling message that contains the gPTP capable TLV, and causing these Signaling messages to be transmitted at a regular rate.

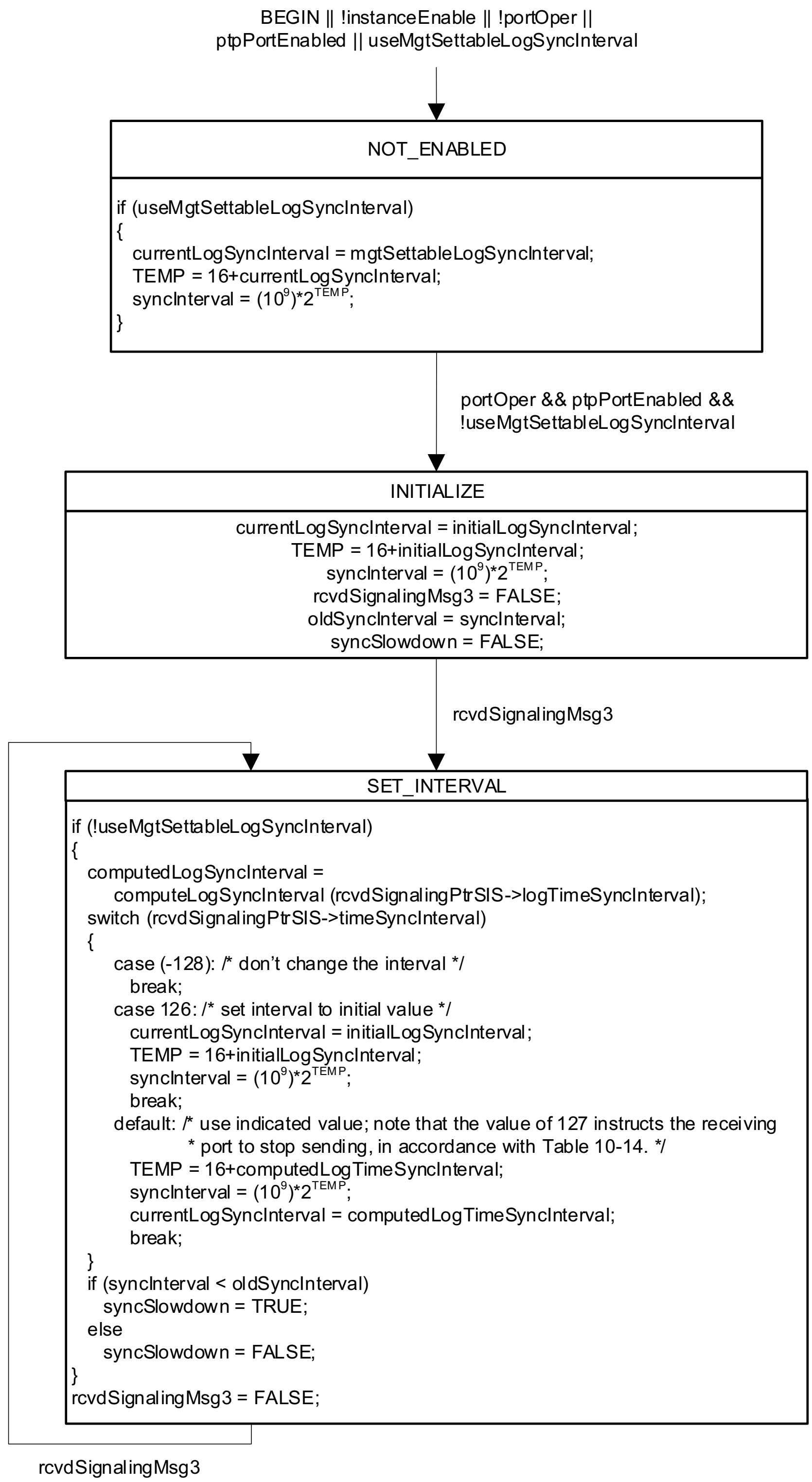


Figure 10-20—SyncIntervalSetting state machine

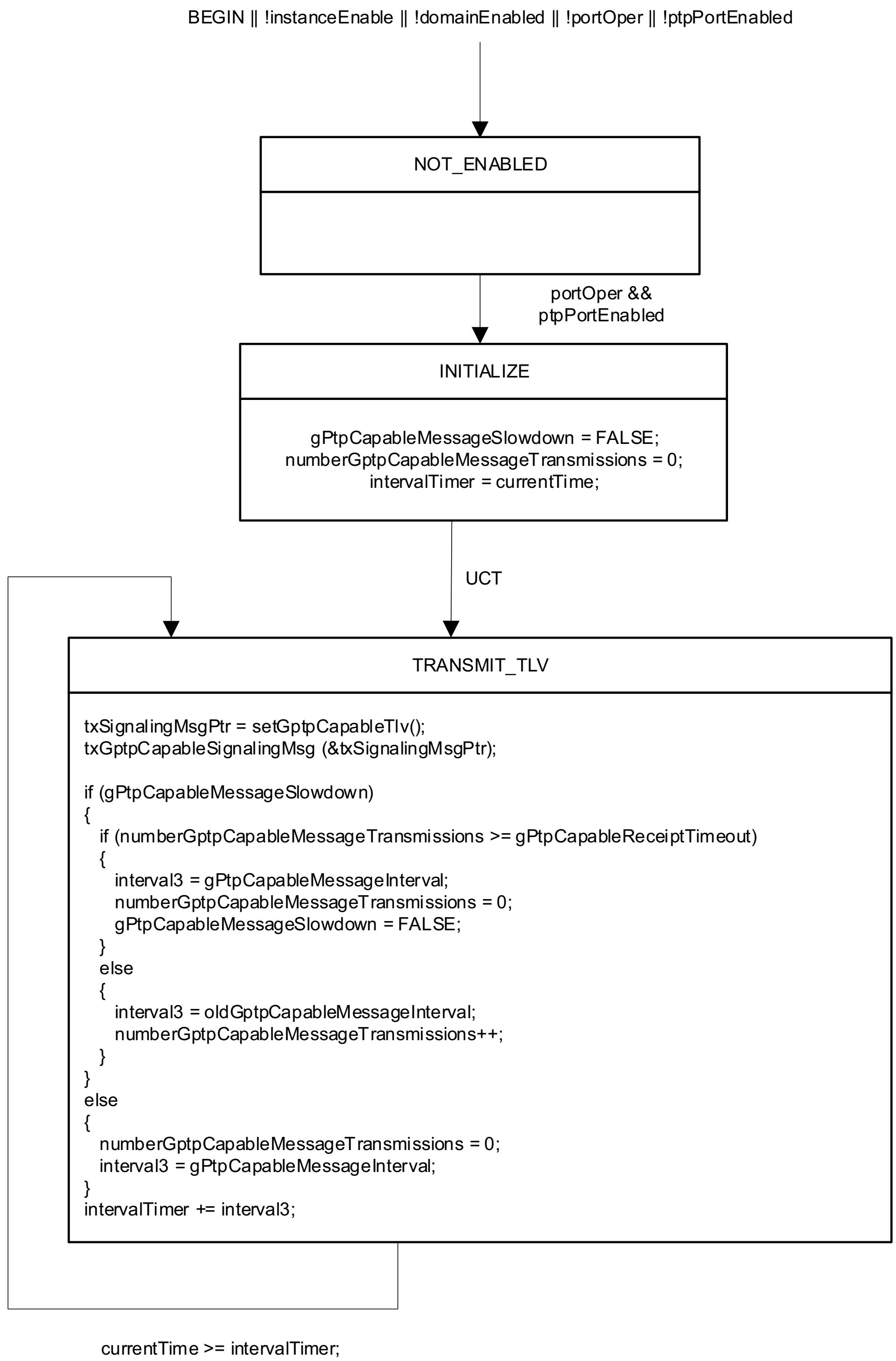


Figure 10-21—GptpCapableTransmit state machine

10.4.2 GptpCapableReceive state machine

10.4.2.1 State machine variables

The following variables are used in the state diagram of 10.4.2.2:

10.4.2.1.1 rcvdGptpCapableTlv: a Boolean variable that notifies the current state machine when a Signaling message containing a gPTP capable TLV is received. This variable is reset by the current state machine.

10.4.2.1.2 rcvdSignalingMsgPtr: a pointer to a structure whose members contain the values of the fields of a Signaling message whose receipt is indicated by rcvdGptpCapableTlv (see 10.4.2.1.1).

10.4.2.1.3 gPtpCapableReceiptTimeoutTimeInterval: the time interval after which, if a Signaling message containing a gPTP capable TLV is not received, the neighbor of this port is considered to no longer be invoking the gPTP protocol. The data type for gPtpCapableReceiptTimeoutTimeInterval is UScaledNs.

10.4.2.1.4 timeoutTime: a variable used to save the time at which the neighbor of this port is considered to no longer be invoking the gPTP protocol if a Signaling message containing a gPTP capable TLV is not received. The data type for timeoutTime is UScaledNs.

10.4.2.2 State diagram

The GptpCapableReceive state machine shall implement the function specified by the state diagram in Figure 10-22, the local variables specified in 10.4.2.1, the relevant parameters specified in 10.6.4 and 10.6.4.4, and the relevant timing attributes specified in 10.7. This state machine is responsible for setting neighborGptpCapable to TRUE on receipt of a Signaling message containing the gPTP capable TLV, and setting the timeout time after which neighborGptpCapable is set to FALSE.

10.4.3 GptpCapableIntervalSetting state machine

10.4.3.1 State machine variables

The following variables are used in the state diagram of 10.4.3.3:

10.4.3.1.1 rcvdSignalingMsg4: a Boolean variable that notifies the current state machine when a Signaling message that contains a gPTP capable Message Interval Request TLV (see 10.6.4.5) is received. This variable is reset by the current state machine.

10.4.3.1.2 rcvdSignalingPtrGIS: a pointer to a structure whose members contain the values of the fields of the received Signaling message that contains a gPTP capable Message Interval Request TLV (see 10.6.4.5).

10.4.3.1.3 logSupportedGptpCapableMessageIntervalMax: the maximum supported logarithm to base 2 of the gPTP capable message interval. The data type for logSupportedGptpCapableMessageIntervalMax is Integer8.

10.4.3.1.4 logSupportedClosestLongerGptpCapableMessageInterval: the logarithm to base 2 of the gPTP capable message interval, such that logSupportedClosestLongerGptpCapableMessageInterval > logRequestedGptpCapableMessageInterval, that is numerically closest to logRequestedGptpCapableMessageInterval, where logRequestedGptpCapableMessageInterval is the argument of the function computeLogGptpCapableMessageInterval() (see 10.4.3.2.2). The data type for logSupportedClosestLongerGptpCapableMessageInterval is Integer8.

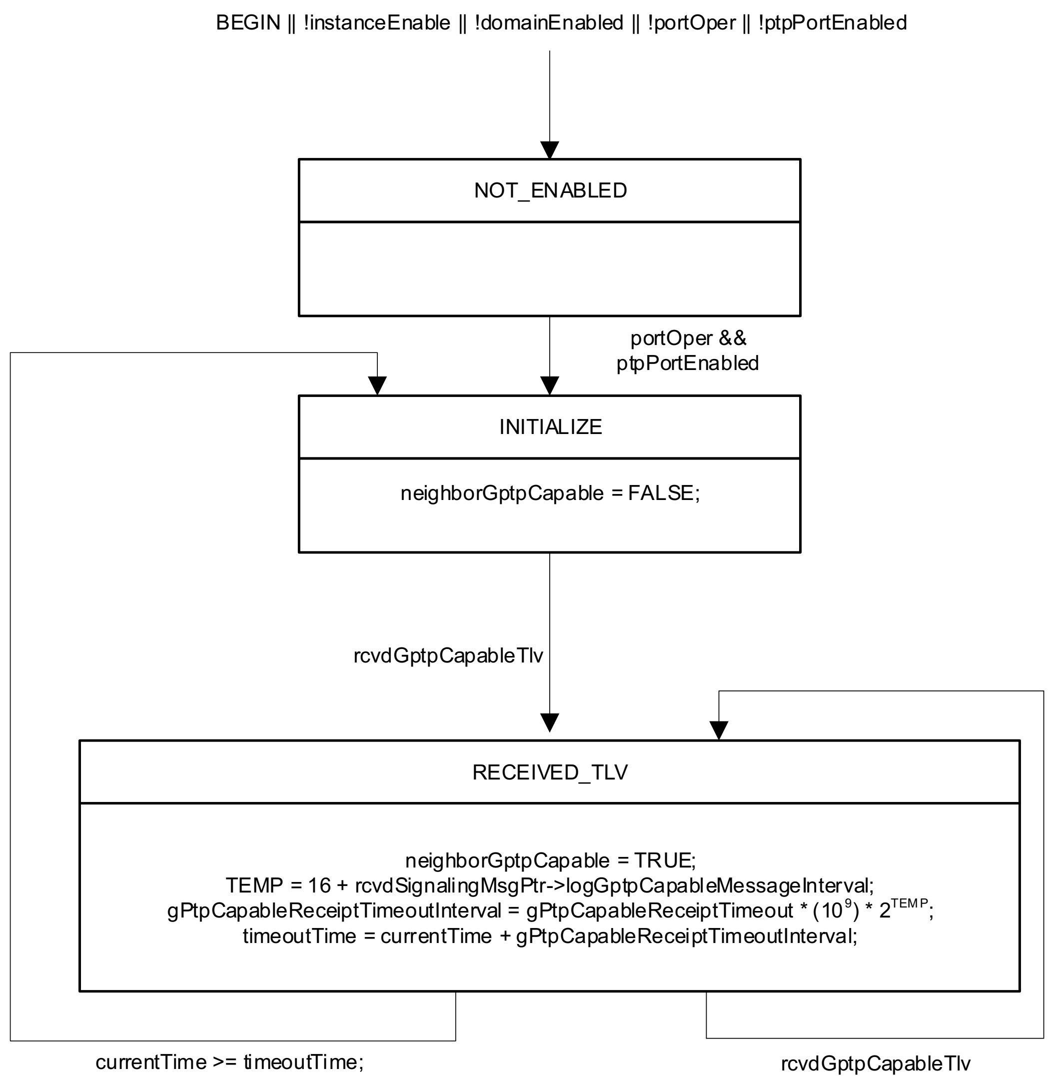


Figure 10-22—GptpCapableReceive state machine

10.4.3.1.5 computedLogGptpCapableMessageInterval: a variable used to hold the result of the function computeLogGptpCapableMessageInterval(). The data type for computedLogGptpCapableMessageInterval is Integer8.

10.4.3.2 State machine functions

10.4.3.2.1 isSupportedLogGptpCapableMessageInterval (logGptpCapableMessageInterval): a Boolean function that returns TRUE if the gPTP capable message interval given by the argument logGptpCapableMessageInterval is supported by the gPTP Port, and FALSE if the gPTP capable message interval is not supported by the gPTP Port. The argument logGptpCapableMessageInterval has the same data type and format as the field logGptpCapableMessageInterval of the gPTP capable message interval request TLV (see 10.6.4.5.6).

10.4.3.2.2 computeLogGptpCapableMessageInterval (logRequestedGptpCapableMessageInterval): An Integer8 function that computes and returns the logGptpCapableMessageInterval, based on the logRequestedGptpCapableMessageInterval. This function is defined as indicated below. It is defined so that the detailed code that it invokes does not need to be placed into the state machine diagram.

```
Integer8 computeLogGptpCapableMessageInterval (logRequestedGptpCapableMessageInterval)
Integer8 logRequestedGptpCapableMessageInterval;
{
        Integer8 logSupportedGptpCapableMessageIntervalMax,
                        logSupportedClosestLongerGptpCapableMessageInterval;
        if (isSupportedLogGptpCapableMessageInterval
                                (logRequestedGptpCapableMessageInterval))
                // The requested gPTP capable Message Interval is supported and returned
                return (logRequestedGptpCapableMessageInterval)
        else
        {
                if (logRequestedGptpCapableMessageInterval >
                                logSupportedGptpCapableMessageIntervalMax)
                        // Return the fasted supported rate, even if faster than the requested rate
                        return (logSupportedGptpCapableMessageIntervalMax);
                else
                        // Return the fastest supported rate that is still slower than
                        // the requested rate.
                        return (logSupportedClosestLongerGptpCapableMessageInterval);
        }
}
```

10.4.3.3 State diagram

The GptpCapableIntervalSetting state machine shall implement the function specified by the state diagram in Figure 10-23, the local variables specified in 10.4.3.1, the messages specified in 10.6, the relevant global variables specified in 10.2.5 the relevant managed objects specified in 14.8, and the relevant timing attributes specified in 10.7. This state machine is responsible for setting the global variables that give the duration of the mean intervals between successive Signaling messages containing the gPTP capable TLV, both at initialization and in response to the receipt of a Signaling message that contains a gPTP capable Message Interval Request TLV (see 10.6.4.5).

10.5 Message attributes

10.5.1 General

This subclause describes media-independent attributes of the Announce message and the Signaling message that are not described in 8.4.2, and whose descriptions are not generic to all messages used in this standard. This subclause also describes media-independent attributes of all time-synchronization event messages.

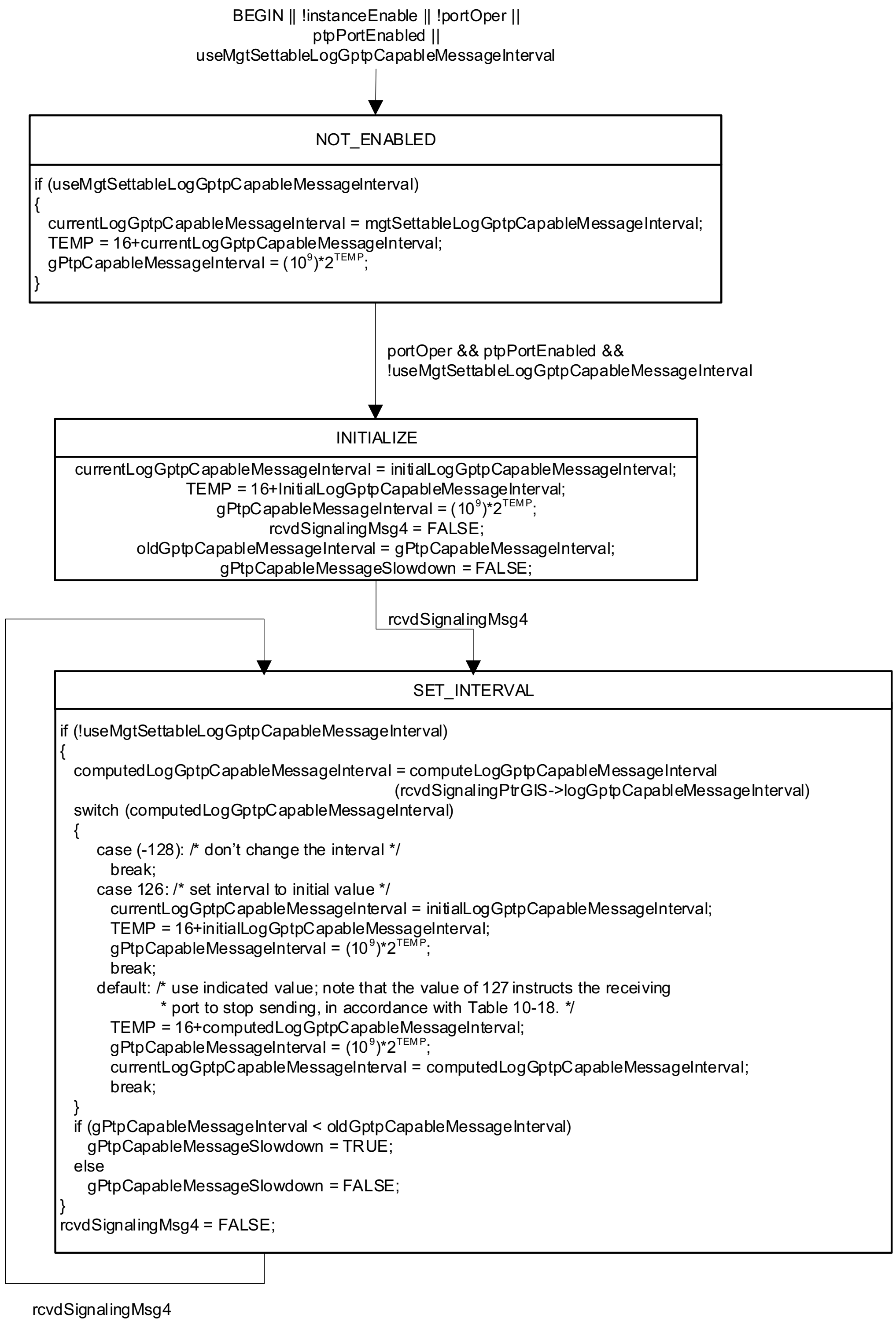


Figure 10-23—GptpCapableIntervalSetting state machine

10.5.2 Message class

The Announce message is a general message, i.e., it is not timestamped. An Announce message provides status and characterization information of the PTP Instance that transmitted the message and the grandmaster. This information is used by the receiving PTP Instance when executing the BMCA.

The Signaling message is a general message, i.e., it is not timestamped. A Signaling message carries information, requests, and/or commands between PTP Instances, via one or more TLVs.

NOTE—In this standard, the Signaling message is used by a port of a PTP Instance to request that the port at the other end of the link send time-synchronization event messages, link delay measurement messages, or Announce messages at desired intervals, to indicate whether the port at the other end of the link should compute neighborRateRatio and/or meanLinkDelay, and to indicate whether a port can receive and correctly process one-step Syncs.. The *message interval request TLV,* is defined to carry this information (see 10.6.4.3). One usage of this functionality is to allow a time-aware system in power-saving mode to remain connected to a gPTP domain via the port on which the Signaling message is sent.

10.5.3 Addresses

The destination address of the Announce and Signaling messages shall be the reserved multicast address given in Table 10-4 unless otherwise specified in a media-dependent clause (see 12.2 and 16.2).

Table 10-4—Destination address for Announce and Signaling messages

Destination address
01-80-C2-00-00-0E
NOTE—This address is taken from Table 8-1, Table 8-2, and Table 8-3 of IEEE Std 802.1Q-2018.

NOTE—Frames whose destination address is the address of Table 10-4 are never forwarded, according to IEEE 802.1Q. Use of this address is shared by IEEE 802.1AS and other IEEE 802.1 protocols.

10.5.4 EtherType

The EtherType of the Announce and Signaling messages shall be the EtherType given in Table 10-5.

Table 10-5—EtherType for Announce and Signaling messages

EtherType
88-F7

NOTE—This EtherType is used for all PTP messages.

10.5.5 Subtype

The subtype of the Announce and Signaling messages is indicated by the majorSdoId field (see 10.6.2.2.1).

NOTE—The subtype for all PTP messages is indicated by the majorSdoId field.

10.5.6 Source port identity

The Announce message, Signaling message, and all time-synchronization messages contain a sourcePortIdentity field (see 10.6.2.2.11), which identifies the egress port (see 8.5) on which the respective message is sent.

10.5.7 Sequence number

Each PortSync entity of a PTP Instance maintains a separate sequenceId pool for each of the message types Announce and Signaling, respectively, transmitted by the MD entity of the port.

Each Announce and Signaling message contains a sequenceId field (see 10.6.2.2.12), that carries the message sequence number. The sequenceId of an Announce message shall be one greater than the sequenceId of the previous Announce message sent by the transmitting port, subject to the constraints of the rollover of the UInteger16 data type used for the sequenceId field. The sequenceId of a Signaling message shall be one greater than the sequenceId of the previous Signaling message sent by the transmitting port, subject to the constraints of the rollover of the UInteger16 data type used for the sequenceId field.

10.6 Message formats

10.6.1 General

The PTP messages Announce and Signaling each have a header, body, and, if present, a suffix that contains one or more TLVs (see 10.6.2, 10.6.3, 10.6.4 of this standard, and Clause 14 of IEEE Std 1588-2019). Reserved fields shall be transmitted with all bits of the field 0 and ignored by the receiver, unless otherwise specified. The data type of the field shall be the type indicated in brackets in the title of each subclause.

Subclause 10.6 defines the path trace TLV, which is carried by the Announce message (see 10.6.3.2.8), and the message interval request TLV, which is carried by the Signaling message (see 10.6.4.3).

Table 10-6 — Forwarding TLVs of IEEE Std 1588-2019

tlvType values	Value (hex)	PTP Message the TLV is attached to	Forwarding rules contained in	Defined in
MANAGEMENT	0001	Management[a]	15.3.3 of IEEE Std 1588-2019	15.5.2 of IEEE Std 1588-2019
MANAGEMENT ERROR STATUS	0002	Management[a]	15.3.3 of IEEE Std 1588-2019	15.5.2 of IEEE Std 1588-2019
PATH_TRACE	0008	Announce	16.2 of IEEE Std 1588-2019, and required by 10.6.3.3 of this standard	16.2 of IEEE Std 1588-2019, and required by 10.6.3.3 of this standard
ALTERNATE_TIME_OFFSET_INDICATOR	0009	Announce	16.3 of IEEE Std 1588-2019	16.3 of IEEE Std 1588-2019
CUMULATIVE_RATE_RATIO	000A	Sync or Follow_Up	16.10 of IEEE Std 1588-2019	16.10 of IEEE Std 1588-2019
ENHANCED_ACCURACY_METRICS	000B	Announce	16.12 of IEEE Std 1588-2019	16.12 of IEEE Std 1588-2019
PAD	000C	Any message	14.3 of IEEE Std 1588-2019	14.3 of IEEE Std 1588-2019
ORGANIZATION_EXTENSION_FORWARDING	4000	Announce	14.1.1.2 of IEEE Std 1588-2019	14.2 of IEEE Std 1588-2019
Reserved range for IEEE Std 1588 forwarding TLVs	4001-7FFF	Announce	14.1.1.2 of IEEE Std 1588-2019	

[a]PTP Management Messages are not used in this standard. They are specified in IEEE Std 1588-2019.

IEEE Std 1588-2019 specifies various optional features that have associated TLVs. These optional features, including the associated TLVs, may be supported by an implementation of this standard. IEEE Std 1588-2019 also specifies that certain TLVs are forwarding (i.e., forwardable), and specifies respective forwarding rules. The TLV forwarding requirement in IEEE Std 1588-2019 means that a PTP Relay Instance forwards the TLV, but the PTP Relay Instance is not required to change the contents of that TLV. If the corresponding optional feature is supported by the PTP Relay Instance, the PTP Relay Instance shall change the contents of the TLV according to the reference to IEEE Std 1588-2019 in the "Forwarding rules contained in" column of Table 10-6. If the corresponding optional feature is not supported by the PTP Relay Instance, the PTP Relay Instance shall forward the TLV unchanged

If a PTP Instance cannot parse a non-forwarding TLV, it shall ignore it and attempt to parse the next TLV (see 14.1 of IEEE Std 1588-2019).

NOTE 2—Any overhead specific to the respective medium is added to each message.

10.6.2 Header

10.6.2.1 General header specifications

The common header for all PTP messages shall be as specified in Table 10-7 and 10.6.2.2 and its subclauses.

Table 10-7—PTP message header

Bits 7 6 5 4	Bits 3 2 1 0	Octets	Offset
majorSdoId	messageType	1	0
minorVersionPTP	versionPTP	1	1
messageLength		2	2
domainNumber		1	4
minorSdoId		1	5
flags		2	6
correctionField		8	8
messageTypeSpecific		4	16
sourcePortIdentity		10	20
sequenceId		2	30
controlField		1	32
logMessageInterval		1	33

10.6.2.2 Header field specifications

10.6.2.2.1 majorSdoId (Nibble)

The value is specified in 8.1 for all transmitted PTP messages of a gPTP domain. The value is specified in 11.2.17 for all transmitted PTP messages of the Common Mean Link Delay Service. Any PTP message received for which the value is not one of the values specified in those subclauses shall be ignored.

10.6.2.2.2 messageType (Enumeration4)

The value indicates the type of the message, as defined in Table 10-8.

The most significant bit of the messageType field divides this field in half between event and general messages, i.e., it is 0 for event messages and 1 for general messages.

Table 10-8—Values for messageType field

Message type	Message class	Value
Announce	General	0xB
Signaling	General	0xC
NOTE—Values for the messageType field for other PTP messages that are used only for specific media are defined in the respective media-dependent clause(s).		

10.6.2.2.3 minorVersionPTP (UInteger4)

For transmitted messages, the value shall be 1 (see 7.5.5 and 13.3.2.5 of IEEE Std 1588). For received messages, the value is ignored.

NOTE—minorVersionPTP indicates the minor version number of IEEE 1588 PTP used in the PTP profile contained in this standard for information only.

10.6.2.2.4 versionPTP (UInteger4)

For transmitted messages, the value shall be 2 (see 7.5.5 and 13.3.2.4 of IEEE Std 1588). For received messages, if the value is not 2, the entire message shall be ignored.

NOTE—VersionPTP indicates the version number of IEEE 1588 PTP used in the PTP profile contained in this standard.

10.6.2.2.5 messageLength (UInteger16)

The value is the total number of octets that form the PTP message. The counted octets start with and include the first octet of the header and terminate with and include the last octet of the last TLV or, if there are no TLVs, with the last octet of the message as defined in this clause.

NOTE—For example, the Follow_Up message (see 11.4.4) contains a PTP header (34 octets), preciseOriginTimestamp (10 octets), and Follow_Up information TLV (32 octets). The value of the messageLength field is 34+10+32 = 76.

10.6.2.2.6 domainNumber (UInteger8)

The value is the gPTP domain number specified in 8.1.

10.6.2.2.7 minorSdoId (UInteger8)

The value is specified in 8.1 for all transmitted PTP messages of a gPTP domain. The value is specified in 11.2.17 for all transmitted PTP messages of the Common Mean Link Delay Service. Any PTP message received for which the value is not one of the values specified in those subclauses shall be ignored.

10.6.2.2.8 flags (Octet2)

The value of the bits of the array are defined in Table 10-9. For message types where the bit is not defined in Table 10-9, the value of the bit is set to FALSE.

10.6.2.2.9 correctionField (Integer64)

The value is 0.

10.6.2.2.10 messageTypeSpecific (Octet4)

The value of the messageTypeSpecific field varies, based on the value of the messageType field, as described in Table 10-10.

For Event messages only, the four octets of the messageTypeSpecific field may be used for internal implementation of a PTP Instance and its ports. For example, if the clock consists of multiple hardware components that are not synchronized, messageTypeSpecific can be used to transfer an internal timestamp between components (e.g. a physical layer chip and the clock's processor).

The messageTypeSpecific field is not used for features of this standard, and it has no meaning from one clock to another. In the on-the-wire format at each PTP Port, for all messageType values, the messageTypeSpecific field is transmitted with all bits of the field 0, and ignored on receive.

Table 10-9—Values of flag bits

Octet	Bit	Message Types	Name	Value
0	0	All	alternateMasterFlag in Announce, Sync, Follow_Up, and Delay_Resp messages	Not used in this standard, reserved as FALSE and ignored on reception
0	1	Sync, Pdelay_Resp	twoStepFlag	*For Sync messages:* a) For a one-step transmitting port (see 11.1.3 and 11.2.13.9), the value is FALSE. b) For a two-step transmitting port, the value is TRUE. *For Pdelay_Resp messages:* The value is transmitted as TRUE and ignored on reception.
0	2	All	unicastFlag	Not used in this standard, reserved as FALSE and ignored on reception
0	3	All	Reserved	Not used by IEEE Std 1588 - 2008, reserved as FALSE and ignored on reception
0	4	All	Reserved	Not used by IEEE Std 1588 - 2008, reserved as FALSE and ignored on reception
0	5	All	PTP profileSpecific 1	Not used in this standard, reserved as FALSE and ignored on reception

Table 10-9—Values of flag bits

Octet	Bit	Message Types	Name	Value
0	6	All	PTP profileSpecific 2	Not used in this standard, reserved as FALSE and ignored on reception
0	7	All	Reserved	Not used in this standard, reserved as FALSE and ignored on reception
1	0	Announce	leap61	The value of the global variable leap61 (see 10.3.9.4)
1	1	Announce	leap59	The value of the global variable leap59 (see 10.3.9.5)
1	2	Announce	currentUtcOffsetValid	The value of the global variable currentUtcOffsetValid (see 10.3.9.6)
1	3	Announce	ptpTimescale	The value of the global variable ptpTimescale (see 10.3.9.7)
1	4	Announce	timeTraceable	The value of the global variable timeTraceable (see 10.3.9.8)
1	5	Announce	frequencyTraceable	The value of the global variable frequencyTraceable (see 10.3.9.9)
1	6	All	Reserved	Not used by IEEE Std 1588 - 2008, reserved as FALSE and ignored on reception
1	7	All	Reserved	Not used by IEEE Std 1588 - 2008, reserved as FALSE and ignored on reception

Table 10-10 — messageTypeSpecific semantics

Value of messageType	Description
Follow_Up, Pdelay_Resp_Follow_Up, Announce, Signaling, Management	For the General message class, this field is reserved; it is transmitted as 0 and ignored on reception.
Sync, Delay_Req, Pdelay_Resp	For the Event message class, this field may be used for internal implementation as specified in this subclause.

10.6.2.2.11 sourcePortIdentity (PortIdentity)

The value is the port identity attribute (see 8.5.2) of the port that transmits the PTP message.

10.6.2.2.12 sequenceId (UInteger16)

The sequenceId field is assigned as specified in 10.5.7.

10.6.2.2.13 controlField (UInteger8)

The value is 0.

10.6.2.2.14 logMessageInterval (Integer8)

For an Announce message, the value is the value of currentLogAnnounceInterval (see 10.3.10.6) for the port that transmits the Announce message. For a Signaling message, the value is transmitted as 0x7F and ignored on reception.

10.6.3 Announce message

10.6.3.1 General Announce message specifications

The fields of the body of the Announce message shall be as specified in Table 10-11 and 10.6.3.2 and its subclauses.

Table 10-11—Announce message fields

Bits (7 6 5 4 3 2 1 0)	Octets	Offset
header (see 10.6.2)	34	0
reserved	10	34
currentUtcOffset	2	44
reserved	1	46
grandmasterPriority1	1	47
grandmasterClockQuality	4	48
grandmasterPriority2	1	52
grandmasterIdentity	8	53
stepsRemoved	2	61
timeSource	1	63
path trace TLV	4+8N	64

10.6.3.2 Announce message field specifications

10.6.3.2.1 currentUtcOffset (Integer16)

The value is the value of currentUtcOffset (see 10.3.9.10) for the PTP Instance that transmits the Announce message.

10.6.3.2.2 grandmasterPriority1 (UInteger8)

The value is the value of the priority1 component of the rootSystemIdentity of the gmPriorityVector (see 10.3.5) of the PTP Instance that transmits the Announce message.

10.6.3.2.3 grandmasterClockQuality (ClockQuality)

The value is the clockQuality formed by the clockClass, clockAccuracy, and offsetScaledLogVariance of the rootSystemIdentity of the gmPriorityVector (see 10.3.5) of the PTP Instance that transmits the Announce message.

10.6.3.2.4 grandmasterPriority2 (UInteger8)

The value is the value of the priority2 component of the rootSystemIdentity of the gmPriorityVector (see 10.3.5) of the PTP Instance that transmits the Announce message.

10.6.3.2.5 grandmasterIdentity (ClockIdentity)

The value is the value of the clockIdentity component of the rootSystemIdentity of the gmPriorityVector (see 10.3.5) of the PTP Instance that transmits the Announce message.

10.6.3.2.6 stepsRemoved (UInteger16)

The value is the value of masterStepsRemoved (see 10.3.9.3) for the PTP Instance that transmits the Announce message.

10.6.3.2.7 timeSource (TimeSource)

The value is the value of timeSource (see 8.6.2.7 and 10.3.9.11) for the PTP Instance that transmits the Announce message.

10.6.3.2.8 Path trace TLV

The Announce message carries the path trace TLV, defined in 10.6.3.3.

10.6.3.3 Path trace TLV definition

10.6.3.3.1 General

The fields of the path-trace TLV shall be as specified in Table 10-12 and in 10.6.4.3.2 through 10.6.4.3.9. This TLV, and its use, are defined in IEEE Std 1588-2019 (see 16.2 and Table 52 of IEEE Std 1588-2019).

Table 10-12—Path trace TLV

Bits								Octets	Offset from start of TLV
7	6	5	4	3	2	1	0		
tlvType								2	0
lengthField								2	2
pathSequence								8N	4

10.6.3.3.2 tlvType (Enumeration16)

The value of the tlvType field is 0x8.

NOTE—This is the value that indicates the TLV is a path trace TLV, as specified in 16.2.5.1 and Table 50 of IEEE Std 1588-2019. The value is specified there as PATH_TRACE, whose value is 0x8.

10.6.3.3.3 lengthField (UInteger16)

The value of the lengthField is 8N.

10.6.3.3.4 pathSequence (ClockIdentity[N])

The value of pathSequence is a ClockIdentity array. The array elements are the clockIdentities of the successive PTP Instances that receive and send an Announce message. The quantity N is the number of PTP Instances, including the grandmaster, that the Announce information has traversed.

NOTE—N is equal to stepsRemoved+1 (see 10.6.3.2.6). The size of the pathSequence array increases by 1 for each PTP Instance that the Announce information traverses.

10.6.4 Signaling message

10.6.4.1 General Signaling message specifications

The fields of the body of the Signaling message shall be as specified in Table 10-13 and 10.6.4.2 and its subclauses.

Table 10-13—Signaling message fields

Bits (7 6 5 4 3 2 1 0)	Octets	Offset
header (see 10.6.2)	34	0
targetPortIdentity	10	34
message interval request TLV, gPTP capable TLV, or gPTP capable message interval request TLV	16	44

10.6.4.2 Signaling message field specifications

10.6.4.2.1 targetPortIdentity (PortIdentity)

The value of targetPortIdentity.clock identity is all ones, i.e., 0xFFFFFFFFFFFFFFFF. The value of targetPortIdentity.portNumber is all ones, i.e., 0xFFFF.

10.6.4.2.2 Message interval request TLV or gPTP capable TLV

The Signaling message carrieseither the message interval request TLV, defined in 10.6.4.3 or the gPTP capable TLV, defined in 10.6.4.4, but not both. If it is desired to send both TLVs, two Signaling messages must be sent.

10.6.4.3 Message interval request TLV definition

10.6.4.3.1 General

The fields of the message interval request TLV shall be as specified in Table 10-14 and in 10.6.4.3.2 through 10.6.4.3.9. This TLV is a standard organization extension TLV for the Signaling message, as specified in 14.2 of IEEE Std 1588-2019.

Table 10-14—Message interval request TLV

Bits (7 6 5 4 3 2 1 0)	Octets	Offset from start of TLV
tlvType	2	0
lengthField	2	2
organizationId	3	4
organizationSubType	3	7
logLinkDelayInterval	1	10
logTimeSyncInterval	1	11
logAnnounceInterval	1	12
flags	1	13
reserved	2	14

10.6.4.3.2 tlvType (Enumeration16)

The value of the tlvType field is 0x3.

NOTE—This is the value that indicates the TLV is a vendor and standard organization extension TLV, as specified in 14.2.2.1 and Table 52 of IEEE Std 1588-2019. The value is specified there as ORGANIZATION_EXTENSION, whose value is 0x3.

10.6.4.3.3 lengthField (UInteger16)

The value of the lengthField is 12.

10.6.4.3.4 organizationId (Octet3)

The value of organizationId is 00-80-C2.

10.6.4.3.5 organizationSubType (Enumeration24)

The value of organizationSubType is 2.

10.6.4.3.6 logLinkDelayInterval (Integer8)

The value is the logarithm to base 2 of the mean time interval, desired by the port that sends this TLV, between successive Pdelay_Req messages sent by the port at the other end of the link. The format and allowed values of logLinkDelayInterval are the same as the format and allowed values of initialLogPdelayReqInterval (see 11.5.2.2).

The values 127, 126, and –128 are interpreted as defined in Table 10-15.

Table 10-15—Interpretation of special values of logLinkDelayInterval

Value	Instruction to PTP Instance that receives this TLV
127	Instructs the port that receives this TLV to stop sending link delay measurement messages.
126	Instructs the port that receives this TLV to set currentLogPdelayReqInterval to the value of initialLogPdelayReqInterval (see 11.5.2.2).
–128	Instructs the port that receives this TLV not to change the mean time interval between successive Pdelay_Req messages.
All values in the ranges [-127, -25] and [25, 125] are reserved.	

10.6.4.3.7 logTimeSyncInterval (Integer8)

The value is the logarithm to base 2 of the mean time interval, desired by the port that sends this TLV, between successive time-synchronization event messages sent by the port at the other end of the link. The format and allowed values of logTimeSyncInterval are the same as the format and allowed values of initialLogSyncInterval (see 10.7.2.3, 11.5.2.3, 12.8, and 13.8.2).

The values 127, 126, and –128 are interpreted as defined in Table 10-16.

When a signaling message that contains this TLV is sent by a port, the value of syncReceiptTimeoutTimeInterval for that port (see 10.2.5.3) shall be set equal to syncReceiptTimeout (see 10.7.3.1) multiplied by the value of the interval, in seconds, reflected by logTimeSyncInterval .

10.6.4.3.8 logAnnounceInterval (Integer8)

The value is the logarithm to base 2 of the mean time interval, desired by the port that sends this TLV, between successive Announce messages sent by the port at the other end of the link. The format and allowed values of logAnnounceInterval are the same as the format and allowed values of initialLogAnnounceInterval (see 10.7.2.2).

The values 127, 126, and –128 are interpreted as defined in Table 10-17.

When a signaling message that contains this TLV is sent by a port, the value of announceReceiptTimeoutTimeInterval for that port (see 10.3.10.1) shall be set equal to announceReceiptTimeout (see 10.7.3.2) multiplied by the value of the interval, in seconds, reflected by logAnnounceInterval.

Table 10-16—Interpretation of special values of logTimeSyncInterval

Value	Instruction to PTP Instance that receives this TLV
127	Instructs the port that receives this TLV to stop sending time-synchronization event messages.
126	Instructs the port that receives this TLV to set currentLogSyncInterval to the value of initialLogSyncInterval (see 10.7.2.3, 11.5.2.3, 12.8, and 13.8.2).
–128	Instructs the port that receives this TLV not to change the mean time interval between successive time-synchronization event messages.
All values in the ranges [-127, -25] and [25, 125] are reserved.	

Table 10-17—Interpretation of special values of logAnnounceInterval

Value	Instruction to PTP Instance that receives this TLV
127	Instructs the port that receives this TLV to stop sending Announce messages.
126	Instructs the port that receives this TLV to set currentLogAnnounceInterval to the value of initialLogAnnounceInterval (see 10.7.2.2).
–128	Instructs the port that receives this TLV not to change the mean time interval between successive Announce messages.
All values in the ranges [-127, -25] and [25, 125] are reserved.	

10.6.4.3.9 flags (Octet)

Bits 0 through 2 of the octet are defined in Table 10-18 and take on the values TRUE and FALSE. Bits not

Table 10-18—Definitions of bits of flags field of message interval request TLV

Bit	Name
0	computeNeighborRateRatio
1	computeMeanLinkDelay
2	oneStepReceiveCapable

defined in Table 10-18 are set to FALSE and ignored on receipt.

NOTE—For full-duplex, point-to-point links (see Clause 11), it is expected that the port sending this TLV will set bits 0 and/or 1 to FALSE if this port will not provide valid timing information in its subsequent responses (Pdelay_Resp and Pdelay_Resp_Follow_Up) to Pdelay_Req messages. Similarly, it is expected that the port sending this TLV will set bit 2 to TRUE if it is capable of receiving and correctly processing one-step Sync messages.

10.6.4.4 gPTP capable TLV definition

10.6.4.4.1 General

The fields of the gPTP capable TLV shall be as specified in Table 10-19 and in 10.6.4.4.2 through 10.6.4.4.7. This TLV is a standard organization extension TLV for the Signaling message, as specified in 14.2 of IEEE Std 1588-2019.

Table 10-19—gPTP capable TLV

Bits (7 6 5 4 3 2 1 0)	Octets	Offset from start of TLV
tlvType	2	0
lengthField	2	2
organizationId	3	4
organizationSubType	3	7
logGptpCapableMessageInterval	1	10
flags	1	11
reserved	4	12

10.6.4.4.2 tlvType (Enumeration16)

The value of the tlvType field is 0x3.

NOTE—This is the value that indicates the TLV is a vendor and standard organization extension TLV, as specified in 14.2.2.1 and Table 52 of IEEE Std 1588-2019. The value is specified there as ORGANIZATION_EXTENSION, whose value is 0x3.

10.6.4.4.3 lengthField (UInteger16)

The value of the lengthField is 12.

10.6.4.4.4 organizationId (Octet3)

The value of organizationId is 00-80-C2.

10.6.4.4.5 organizationSubType (Enumeration24)

The value of organizationSubType is 4.

10.6.4.4.6 logGptpCapableMessageInterval (Integer8)

The value of logGptpCapableMessageInterval is the logarithm to base 2 of the mean gPTP capable message interval in seconds (see 10.7.2.1 and 10.7.2.5)

10.6.4.4.7 flags (Octet)

The flag bits shall be transmitted as FALSE and ignored on receipt.

10.6.4.5 gPTP capable message interval request TLV definition

10.6.4.5.1 General

The fields of the gPTP capable message interval request TLV shall be as specified in Table 10-20 and in 10.6.4.3.2 through 10.6.4.3.9. This TLV is a standard organization extension TLV for the Signaling message, as specified in 14.2 of IEEE Std 1588-2019.

10.6.4.5.2 tlvType (Enumeration16)

The value of the tlvType field is 0x3.

NOTE—This is the value that indicates the TLV is a vendor and standard organization extension TLV, as specified in 14.2.2.1 and Table 52 of IEEE Std 1588-2019. The value is specified there as ORGANIZATION_EXTENSION, whose value is 0x3.

10.6.4.5.3 lengthField (UInteger16)

The value of the lengthField is 10.

10.6.4.5.4 organizationId (Octet3)

The value of organizationId is 00-80-C2.

Table 10-20—gPTP capable message interval request TLV

Bits (7 6 5 4 3 2 1 0)	Octets	Offset from start of TLV
tlvType	2	0
lengthField	2	2
organizationId	3	4
organizationSubType	3	7
logGptpCapableMessageInterval	1	10
reserved	3	11

10.6.4.5.5 organizationSubType (Enumeration24)

The value of organizationSubType is 5.

10.6.4.5.6 logGptpCapableMessageInterval (Integer8)

The value is the logarithm to base 2 of the mean time interval, desired by the port that sends this TLV, between successive Signaling messages that contain the gPTP capable TLV (see 10.6.4.4), sent by the port at the other end of the link. The format and allowed values of logGptpCapableMessageInterval are the same as the format and allowed values of initialLogGptpCapableMessageInterval (see 10.7.2.5).

The values 127, 126, and –128 are interpreted as defined in Table 10-21..

Table 10-21—Interpretation of special values of logGptpCapableMessageInterval

Value	Instruction to PTP Instance that receives this TLV
127	Instructs the port that receives this TLV to stop sending Signaling messages that contain the gPTP capable TLV.
126	Instructs the port that receives this TLV to set currentlogGptpCapableMessageInterval to the value of initialLogGptpCapableMessageInterval (see 10.7.2.4).
–128	Instructs the port that receives this TLV not to change the mean time interval between successive Signaling messages that contain the gPTP capable TLV.
All values in the ranges [-127, -25] and [25, 125] are reserved.	

When a signaling message that contains this TLV is sent by a port, the value of gPtpCapableReceiptTimeoutTimeInterval for that port (see 10.3.10.1) shall be set equal to pPtpCapableReceiptTimeout (see 10.7.3.3) multiplied by the value of the interval, in seconds, reflected by logGptpCapableMessageInterval.

10.7 Protocol timing characterization

10.7.1 General

This subclause specifies timing and timeout attributes for the media-independent sublayer state machines.

10.7.2 Message transmission intervals

10.7.2.1 General interval specification

The mean time interval between the sending of successive Announce messages, referred to as the *announce interval,* shall be as specified in 10.7.2.2.

The mean time interval between the sending of successive time-synchronization event messages for full-duplex point-to-point, IEEE 802.11, and CSN links, and successive general messages containing time-synchronization information for IEEE 802.3 EPON links, is referred to as the sync interval. The sync interval shall be as specified in 10.7.2.3.

The mean time interval between the sending of successive Signaling messages that contain the gPTP capable TLV, referred to as the *gPTP capable message interval,* shall be as specified in 10.7.2.5.

10.7.2.2 Announce message transmission interval

The logarithm to the base 2 of the announce interval (in seconds) is carried in the logMessageInterval field of the Announce message.

When useMgtSettableLogAnnounceInterval (see 14.8.14) is FALSE, the initialLogAnnounceInterval specifies the announce interval when the port is initialized and the value the announce interval is set to when a message interval request TLV is received with the logAnnounceInterval field set to 126 (see the AnnounceIntervalSetting state machine, 10.3.17). The currentLogAnnounceInterval specifies the current value of the announce interval. The default value of initialLogAnnounceInterval is 0. Every port supports the value 127; the port does not send Announce messages when currentLogAnnounceInterval has this value (see 10.3.17). A port may support other values, except for the reserved values indicated in Table 10-17. A port ignores requests for unsupported values (see 10.3.17). The initialLogAnnounceInterval and currentLogAnnounceInterval are per-port attributes.

When useMgtSettableLogAnnounceInterval is TRUE, currentLogAnnounceInterval is set equal to mgtSettableLogAnnounceInterval (see 14.8.15) and initialLogAnnounceInterval is ignored.

Announce messages shall be transmitted such that the value of the arithmetic mean of the intervals, in seconds, between message transmissions is within ±30% of $2^{\text{currentLogAnnounceInterval}}$. In addition, a gPTP Port shall transmit Announce messages such that at least 90% of the inter-message intervals are within ±30% of the value of $2^{\text{currentLogAnnounceInterval}}$. The interval between successive Announce messages should not exceed twice the value of $2^{\text{portDS.logAnnounceInterval}}$, in order to prevent causing an announceReceiptTimeout event. The PortAnnounceTransmit state machine (see 10.3.16) is consistent with these requirements, i.e., the requirements here and the requirements of the PortAnnounceTransmit state machine can be met simultaneously.

NOTE 1—A minimum number of inter-message intervals is necessary in order to verify that a gPTP Port meets these requirements. The arithmetic mean is the sum of the inter-message interval samples divided by the number of samples. For more detailed discussion of statistical analyses, see [B18].

NOTE 2—If useMgtSettableLogAnnounceInterval is FALSE, the value of initialLogAnnounceInterval is the value of the mean time interval between successive Announce messages when the port is initialized. The value of the mean time interval between successive Announce messages can be changed, e.g., if the port receives a Signaling message that carries a message interval request TLV (see 10.6.4.3), and the current value is stored in currentLogAnnounceInterval. The value of the mean time interval between successive Announce messages can be reset to the initial value, e.g., by a message interval request TLV for which the value of the field logAnnounceInterval is 126 (see 10.6.4.3.8).

NOTE 3—A port that requests (using a Signaling message that contains a message interval request TLV, see 10.6.4 and 10.3.17) that the port at the other end of the attached link set its currentLogAnnounceInterval to a specific value can determine if the request was honored by examining the logMessageInterval field of subsequent received Announce messages.

10.7.2.3 Time-synchronization event message transmission interval

The logarithm to the base 2 of the sync interval (in seconds) is carried in the logMessageInterval field of the time-synchronization messages.

When useMgtSettableLogSyncInterval (see 14.8.19) is FALSE, the initialLogSyncInterval specifies the sync interval when the port is initialized and the value the sync interval is set to when a message interval request TLV is received with the logTimeSyncInterval field set to 126 (see the SyncIntervalSetting state machine, 10.3.18). The default value is media-dependent; the value is specified in the respective media-dependent clauses. The initialLogSyncInterval is a per-port attribute.

The currentLogSyncInterval specifies the current value of the sync interval, and is a per-port attribute.

When useMgtSettableLogSyncInterval is TRUE, currentLogSyncInterval is set equal to mgtSettableLogSyncInterval (see 14.8.20) and initialLogSyncInterval is ignored.

When the value of syncLocked is FALSE, time-synchronization messages shall be transmitted such that the value of the arithmetic mean of the intervals, in seconds, between message transmissions is within ±30% of $2^{\text{currentLogSyncInterval}}$. In addition, a gPTP Port shall transmit time-synchronization messages such that at least 90% of the inter-message intervals are within ±30% of the value of $2^{\text{currentLogSyncInterval}}$. The interval between successive time-synchronization messages should not exceed twice the value of $2^{\text{portDS.logSyncInterval}}$, in order to prevent causing a syncReceiptTimeout event. The PortSyncSyncSend state machine (see 10.2.12) is consistent with these requirements, i.e., the requirements here and the requirements of the PortSyncSyncSend state machine can be met simultaneously.

NOTE 1—A minimum number of inter-message intervals is necessary in order to verify that a gPTP Port meets these requirements. The arithmetic mean is the sum of the inter-message interval samples divided by the number of samples. For more detailed discussion of statistical analyses, see [B18]

NOTE 2—If useMgtSettableLogSyncInterval is FALSE the value of initialLogSyncInterval is the value of the sync interval when the port is initialized. The value of the sync interval can be changed, e.g., if the port receives a Signaling message that carries a message interval request TLV (see 10.6.4.3), and the current value is stored in currentLogSyncInterval. The value of the sync interval can be reset to the initial value, e.g., by a message interval request TLV for which the value of the field logTimeSyncInterval is 126 (see 10.6.4.3.7).

10.7.2.4 Interval for providing synchronization information by ClockMaster entity

The clockMasterLogSyncInterval specifies the mean time interval between successive instants at which the ClockMaster entity provides time-synchronization information to the SiteSync entity. The value is less than or equal to the smallest currentLogSyncInterval (see 10.7.2.3) value for all the ports of the PTP Instance. The clockMasterLogSyncInterval is an internal, per PTP Instance variable.

10.7.2.5 Interval for sending the gPTP capable TLV Signaling message

The logarithm to the base 2 of the gPTP capable message interval (in seconds) is carried in the logGptpCapableMessageInterval field of the gPTP capable TLV. The default value shall be 0. The range shall be -24 through 24. Other values in the range [-128, 127] shall be reserved.

When useMgtSettableLogGptpCapableMessageInterval (see 14.8.14) is FALSE, the initialLogGptpCapableMessageInterval specifies the following:

a) the mean gPTP capable message interval when the port is initialized, and
b) the value the gPTP capable message interval is set to when a gPTP capable TLV is received with the logGptpCapableMessageInterval field set to 126 (see the GptpCapableMessageIntervalSetting state machine, 10.4.3).

The currentLogGptpCapableMessageInterval specifies the current value of the gPTP capable message interval. Every port supports the value 127; the port does not send Signaling messages containing a gPTP capable TLV when currentLogGptpCapableMessageInterval has this value (see 10.4.3). A port may support other values, except for the reserved values indicated in Table 10-21. A port shall ignore requests for unsupported values (see 10.4.3). The initialLogGptpCapableMessageInterval and currentLogGptpCapableMessageInterval are per-port attributes.

When useMgtSettableLogGptpCapableMessageInterval is TRUE, currentLogGptpCapableMessageInterval is set equal to mgtSettableLogGptpCapableMessageInterval (see 14.8.15) and initialLogGptpCapableMessageInterval is ignored.

NOTE 1—If useMgtSettableLogGptpCapableMessageInterval is FALSE, the value of initialLogGptpCapableMessageInterval is the value of the mean time interval between successive Signaling messages containing the gPTP capable TLV when the port is initialized. The value of the mean time interval between successive Signaling messages containing the gPTP capable TLV can be changed, e.g., if the port receives a Signaling message that carries a gPTP capable TLV (see 10.6.4.4), and the current value is stored in currentLogGptpCapableMessageInterval. The value of the mean time interval between successive Signaling messages containing the gPTP capable TLV can be reset to the initial value, e.g., by a Signaling message containing a gPTP capable message interval request TLV for which the value of the field logGptpCapableMessageInterval is 126 (see 10.6.4.5.6).

NOTE 2—A port that requests (using a Signaling message that contains a gPTP capable message interval request TLV, see 10.6.4 and 10.6.4.5) that the port at the other end of the attached link set its currentLogGptpCapableMessageInterval to a specific value can determine if the request was honored by examining the logGptpCapableMessageInterval field of subsequent received Signaling messages containing the gPTP capable TLV.

NOTE 3—The GptpCapableTransmit state machine ensures that the times between transmission of successive Signaling messages, containing the gPTP capable TLV, expressed in seconds, are not smaller than $2^{\text{currentLogGptpCapableMessaggeInterval}}$. This is consistent with the manner in which Pdelay_Req messages are transmitted (see NOTE 3 of 11.5.2.2).

10.7.3 Timeouts

10.7.3.1 syncReceiptTimeout

The value of this attribute tells a slave port the number of sync intervals to wait without receiving synchronization information, before assuming that the master is no longer transmitting synchronization information, and that the BMCA needs to be run, if appropriate. The condition of the slave port not receiving synchronization information for syncReceiptTimeout sync intervals is referred to as *sync receipt timeout.*

The default value shall be 3. The syncReceiptTimeout is a per-port attribute.

10.7.3.2 announceReceiptTimeout

The value of this attribute tells a slave port the number of announce intervals to wait without receiving an Announce message, before assuming that the master is no longer transmitting Announce messages, and that the BMCA needs to be run, if appropriate. The condition of the slave port not receiving an Announce message for announceReceiptTimeout announce intervals is referred to as *announce receipt timeout.*

The default value shall be 3. The announceReceiptTimeout is a per-port attribute.

10.7.3.3 gPtpCapableReceiptTimeout

The value of this attribute tells a port the number of gPTP capable message intervals to wait without receiving from its neighbor a Signaling message containing a gPTP capable TLV, before determining that its neighbor is no longer invoking the gPTP protocol.

NOTE—A determination that its neighbor is no longer invoking the gPTP protocol will cause the port to set asCapable to FALSE.

The default value shall be 9. The range shall be 1-255.

11. Media-dependent layer specification for full-duplex, point-to-point links

11.1 Overview

11.1.1 General

A port attached to a full-duplex, point-to-point link uses the PTP peer delay protocol to measure propagation delay on the link. An overview of the propagation delay measurement is given in 11.1.2. Synchronization information is transported using the PTP messages Sync and Follow_Up. An overview of the transport of synchronization information is given in 11.1.3. An overview of the MD entity model for a full-duplex, point-to-point medium is given in 11.1.4.

11.1.2 Propagation delay measurement

The measurement of propagation delay on a full-duplex, point-to-point link using the peer-to-peer delay mechanism is illustrated in Figure 11-1. The mechanism is the same as the peer-to-peer delay mechanism described in IEEE Std 1588-2019, specialized to a two-step PTP Port[14] and sending the requestReceiptTimestamp and the responseOriginTimestamp separately [see 11.4.2 of IEEE Std 1588-2019, item (c)(8)]. The measurement is made by each port at the end of every full-duplex, point-to-point link. Thus, both ports sharing a link will independently make the measurement, and both ports will know the propagation delay as a result. This allows the time-synchronization information described in 11.1.3 to be transported irrespective of the direction taken by a Sync message. The propagation delay measurement is made on ports otherwise blocked by non-PTP algorithms (e.g., Rapid Spanning Tree Protocol) used to eliminate cyclic topologies. This enables either no loss of synchronization or faster resynchronization, after a reconfiguration, because propagation delays are already known and do not have to be initially measured when the reconfiguration occurs.

Since, as will be explained shortly, the propagation delay measurement is made using timestamps relative to the LocalClock entities at each port at the ends of the link and the resulting mean delay is expressed in the responder timebase (see 11.2.19.3.4), there is no need to measure the mean delay for the link in each domain, because the mean delay is the same in each domain. In addition, the quantity neighborRateRatio (see 10.2.5.7) is the ratio of the responder to requester LocalClock frequency, and is also the same in all domains. Therefore, the propagation delay and neighborRateRatio measurements are domain-independent. Single instances of the respective state machines that cause these measurements to be made are invoked, rather than one instance per domain, and the results are available to all domains. The PTP messages used for the measurements (i.e., Pdelay_Req, Pdelay_Resp, and Pdelay_Resp_Follow_Up, see 11.4.5 through 11.4.7) carry 0 in the domainNumber field, but this value is not used.

In Figure 11-1, the propagation delay measurement is initiated by a time-aware system at one end of a link; this time-aware system is referred to as the *peer delay initiator*. For purposes of the measurement, the other time-aware system is the *peer delay responder.* A similar measurement occurs in the opposite direction, with the initiator and responder interchanged and the directions of the messages in Figure 11-1 reversed.

The propagation delay measurement starts with the initiator issuing a Pdelay_Req message and generating a timestamp, t_1. The responder receives this message and timestamps it with time t_2. The responder returns a Pdelay_Resp message and timestamps it with time t_3. The responder returns the time t_2 in the Pdelay_Resp message, and the time t_3 in a Pdelay_Resp_Follow_Up message. The initiator generates a timestamp, t_4, upon receiving the Pdelay_Resp message. The initiator then uses these four timestamps to compute the mean propagation delay (i.e., meanLinkDelay, see 8.3) as shown in Equation (11-1):

[14] See 3.1.86 of IEEE Std 1588-2019 for the definition of a *two-step PTP Port*

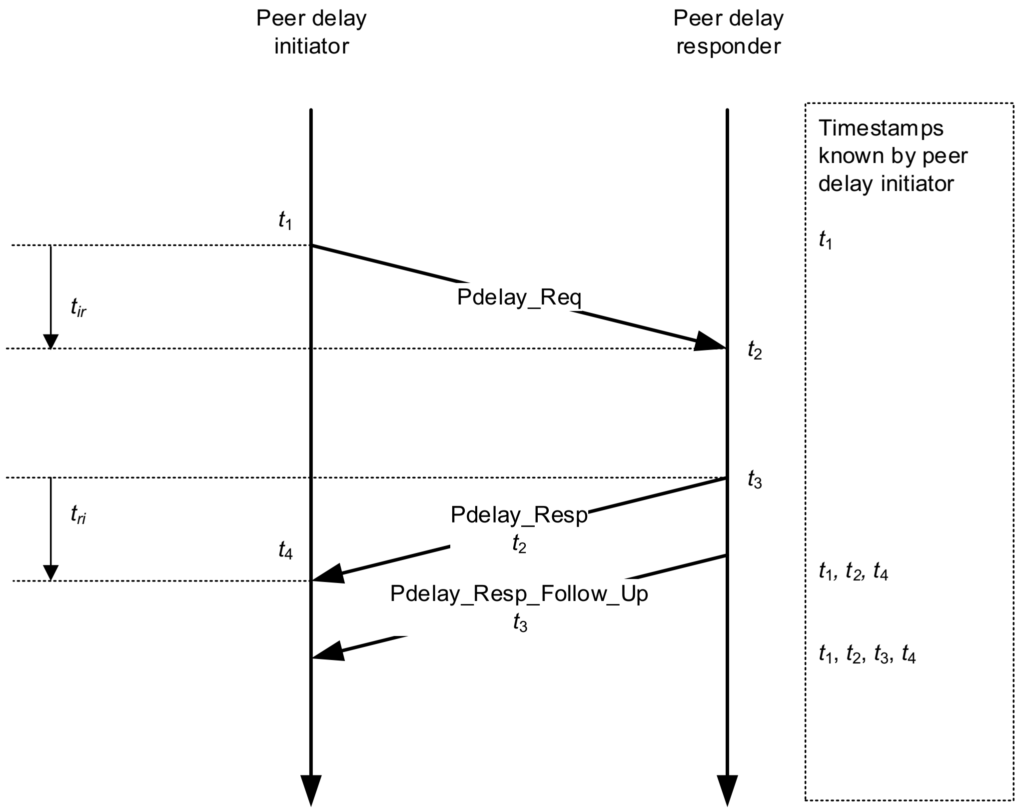


Figure 11-1—Propagation delay measurement using peer-to-peer delay mechanism

$$
\begin{aligned}
t_{ir} &= t_2 - t_1 \\
t_{ri} &= t_4 - t_3 \\
D &= \frac{t_{ir} + t_{ri}}{2} = \frac{(t_4 - t_1) - (t_3 - t_2)}{2}
\end{aligned} \tag{11-1}
$$

where D is the measured mean propagation delay and the other quantities are defined in Figure 11-1.

Note that it is the mean propagation delay that is measured here. Any link asymmetry is modeled as described in 8.3. Any asymmetry that is not corrected for introduces an error in the transported synchronized time value.

The accuracy of the mean propagation delay measurement depends on how accurately the times t_1, t_2, t_3, and t_4 are measured. In addition, Equation (11-1) assumes that the initiator and responder timestamps are taken relative to clocks that have the same frequency. In practice, t_1 and t_4 are measured relative to the LocalClock entity of the initiator time-aware system, and t_2 and t_3 are measured relative to the LocalClock entity of the responder time-aware system. If the propagation delay measurement is desired relative to the responder time base, the term ($t_4 - t_1$) in Equation (11-1) must be multiplied by the rate ratio of the responder relative to the initiator, otherwise there will be an error equal to $0.5y(t_4 - t_1)$, where y is the frequency offset of the responder relative to the initiator. Likewise, if the propagation delay measurement is desired relative to the initiator time base, the term ($t_3 - t_2$) in Equation (11-1) must be multiplied by the rate ratio of the initiator relative to the responder, otherwise there will be an error equal to $0.5y(t_3 - t_2)$, where y is the frequency offset of the initiator relative to the responder. Finally, if the propagation delay measurement is desired relative to the grandmaster time base, each term must be multiplied by the rate ratio of the grandmaster relative to the time base that term is expressed in.

There can also be an error in measured propagation delay due to time measurement granularity (see B.1.2). For example, if the time measurement granularity is 40 ns (as specified in B.1.2), the timestamps t_1, t_2, t_3, and/or $t4$ can undergo 40 ns step changes. When this occurs, the measured propagation delay, D, will change by 20 ns (or by a multiple of 20 ns if more than one of the timestamps has undergone a 40 ns step change). The actual propagation delay has not changed by 20 ns; the effect is due to time measurement granularity. The effect can be reduced, and the accuracy improved, by averaging successive measured propagation delay values. For example, an exponential averaging filter can be used, i.e., as shown in Equation (11-2):

$$D_{avg,k} = aD_{avg,k-1} + (1-a)D_{k-1} \quad (11\text{-}2)$$

where D_k is the k^{th} propagation delay measurement, $D_{avg,k}$ is the k^{th} computed average propagation delay, and k is an index for the propagation delay measurements (i.e., peer delay message exchange). The quantity a is the exponential weighting factor; it can be set so that the weight of a past propagation delay measurement is $1/e$ after M measurements, i.e., as shown in Equation (11-3):

$$a = e^{-\frac{1}{M}} \quad (11\text{-}3)$$

The above averager must be initialized. One method is to use a simple average (i.e., the sum of the sample values divided by the number of samples) of the measurements made up to the current measurement until a window of M measurements has been accumulated. In this case, Equation (11-2) is used only for $k > M$. For $k \le M$, the averaged propagation delay is given by Equation (11-4):

$$D_{avg,k} = \frac{(k-1)D_{avg,k-1} + D_{k-1}}{k} \quad (11\text{-}4)$$

The rate ratio of the responder relative to the initiator is the quantity neighborRateRatio (see 10.2.5.7). It is computed by the function computePdelayRateRatio() (see 11.2.19.3.3) of the MDPdelayReq state machine (see 11.2.19) using successive values of t_3 and t_4. As indicated in the description of computePdelayRateRatio(), any scheme that uses this information is acceptable as long as the performance requirements of B.2.4 are met. One example scheme is given in NOTE 1 of 11.2.19.3.3.

11.1.3 Transport of time-synchronization information

The transport of time-synchronization information by a PTP Instance, using Sync and Follow_Up (or just Sync) messages, is illustrated in Figure 11-2. The mechanism is mathematically equivalent to the mechanism described in IEEE Std 1588-2019 for a peer-to-peer transparent clock that is syntonized (see 11.4.5.1, 11.5.1, and 11.5.2.2 of IEEE Std 1588-2019). However, as will be seen shortly, the processes of transporting synchronization by a peer-to-peer transparent clock that is syntonized and by a boundary clock are mathematically and functionally equivalent. The main functional difference between the two types of clocks is that the boundary clock participates in best master selection and invokes the BMCA, while the peer-to-peer transparent clock does not participate in best master selection and does not invoke the BMCA (and implementations of the two types of clocks can be different).

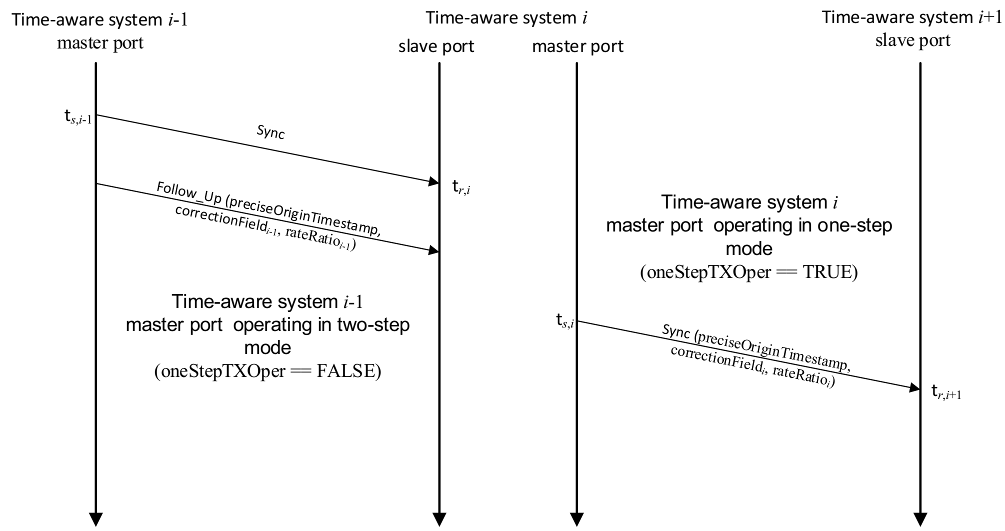


Figure 11-2—Transport of time-synchronization information

Figure 11-2 shows three adjacent PTP Instances, indexed *i*–1, *i*, and *i*+1. Synchronization is transported from PTP Instance *i*–1 to PTP Instance *i* using two-step time transport, and then to PTP Instance *i*+1 using one-step time transport. PTP Instance *i*–1 sends a Sync message to PTP Instance *i* at time $t_{s,i-1}$, relative to the LocalClock entity of PTP Instance *i*–1. At a later time (since it is using two-step time transport), PTP Instance *i*–1 sends an associated Follow_Up message to PTP Instance *i*, which contains a preciseOriginTimestamp, $\text{correctionField}_{i-1}$, and rateRatio_{i-1}. The preciseOriginTimestamp contains the time of the grandmaster when it originally sent this synchronization information. It is not indexed here because it normally does not change as the Sync and Follow_Up messages traverse the network. The quantity $\text{correctionField}_{i-1}$ contains the difference between the synchronized time when the Sync message is sent (i.e., the synchronized time that corresponds to the local time $t_{s,i-1}$) and the preciseOriginTimestamp. The sum of preciseOriginTimestamp and $\text{correctionField}_{i-1}$ gives the synchronized time that corresponds to $t_{s,i-1}$. The quantity rateRatio_{i-1} is the ratio of the grandmaster frequency to the frequency of the LocalClock entity of PTP Instance *i*–1.

PTP Instance *i* receives the Sync message from PTP Instance *i*–1 at time $t_{r,i}$, relative to its LocalClock entity. It timestamps the receipt of the Sync message, and the timestamp value is $t_{r,i}$. It receives the associated Follow_Up message some time later.

PTP Instance *i* will eventually send a new Sync message at time $t_{s,i}$, relative to its LocalClock entity. The time $t_{s,i}$ occurs after the Follow_Up message is received from PTP Instance *i*–1. It will have to compute correctionField_i, i.e., the difference between the synchronized time that corresponds to $t_{s,i}$ and the preciseOriginTimestamp. To do this, it must compute the value of the time interval between $t_{s,i-1}$ and $t_{s,i}$, expressed in the grandmaster time base. This interval is equal to the sum of the following quantities:

a) The propagation delay on the link between PTP Instances *i*–1 and *i*, expressed in the grandmaster time base, and
b) The difference between $t_{s,i}$ and $t_{r,i}$ (i.e., the residence time), expressed in the grandmaster time base.

The mean propagation delay on the link between PTP Instances i–1 and i, relative to the LocalClock entity of PTP Instance i–1, is equal to meanLinkDelay (see 10.2.5.8). This must be multiplied by rateRatio_{i-1} to express it in the grandmaster time base. The total propagation delay is equal to the mean propagation delay plus the quantity delayAsymmetry (see 8.3 and 10.2.5.9); delayAsymmetry is already expressed in the grandmaster time base. The residence time, $t_{s,i}–t_{r,i}$, must be multiplied by rateRatio_i to express it in the grandmaster time base.

The preceding computation is organized slightly differently in the state machines of 11.2.14 and 11.2.15. Rather than explicitly expressing the link propagation delay in the grandmaster time base, the local time at PTP Instance i that corresponds to $t_{s,i–1}$ is computed; this is the upstreamTxTime member of the MDSyncReceive structure (see 10.2.2.2.7; recall that $t_{s,i–1}$ is relative to the LocalClock entity of PTP Instance i–1). upstreamTxTime is equal to the quantity $t_{r,i}$ minus the link propagation delay expressed relative to the LocalClock entity of PTP Instance i. The link propagation delay expressed relative to the LocalClock entity of PTP Instance i is equal to the sum of the following:

c) The quantity meanLinkDelay (see 10.2.5.8) divided by neighborRateRatio (see 10.2.5.7), and
d) The quantity delayAsymmetry (see 10.2.5.9) divided by rateRatio_i.

The division of delayAsymmetry by rateRatio_i is performed after rateRatio_i has been updated, as described shortly. The computation of upstreamTxTime is done by the MDSyncReceiveSM state machine in the function setMDSyncReceiveMDSR() (see 11.2.14.2.1). When PTP Instance i sends a Sync message to PTP Instance i+1, it computes the sum of the link propagation delay and residence time, expressed in the grandmaster time base, as:

e) The quantity ($t_{s,i}$ – upstreamTxTime)(rateRatio_i).

As in item d) above, this computation is performed after rateRatio_i has been updated, as described shortly. The quantity of item e) is added to $\text{correctionField}_{i–1}$ to obtain correctionField_i. The computation of item e) and correctionField_i is done by the MDSyncSendSM state machine in the function setFollowUp() (see 11.2.15.2.3). The quantity correctionField_i is inserted in the Sync message sent by PTP Instance i.

The difference between mean propagation delay relative to the grandmaster time base and relative to the time bases of the PTP Instance at the other end of the attached link or of the current PTP Instance is usually negligible. The former can be obtained from the latter by multiplying the latter by the ratio of the grandmaster frequency to the frequency of the LocalClock entity of the PTP Instance at the other end of the link attached to this port. This ratio differs from 1 by 200 ppm or less. For example, for a worst-case frequency offset of the LocalClock entity of the PTP Instance at the other end of the link, relative to the grandmaster, of 200 ppm, and a measured propagation time of 100 ns, the difference in D relative to the two time bases is 20 ps. The corresponding difference for link delay asymmetry in this example is also negligible because the magnitude of the link delay asymmetry is of the same order of magnitude as the mean propagation time, or less. However, the difference is usually not negligible for residence time, because residence time can be much larger (see B.2.2).

It was previously indicated that the processes of transporting synchronization by a peer-to-peer transparent clock that is syntonized and by a boundary clock are mathematically and functionally equivalent. This is because the computations described above compute the synchronized time when the Sync message is sent by the PTP Instance. The same computations are done if PTP Instance i sends a Sync message without having received a new Sync message, i.e., if Sync receipt timeout occurs (see 10.7.3.1). In this case, PTP Instance i uses the most recently received time-synchronization information from PTP Instance i–1, which would be prior to PTP Instance i having sent its most recent Sync message. The synchronized time corresponding to the sending of a Sync message is equal to the sum of the preciseOriginTimestamp and correctionField. Normally a boundary clock places this entire value, except for any sub-nanosecond portion, in the preciseOriginTimestamp, while a transparent clock retains the preciseOriginTimestamp and updates the correctionField. However, the sum of the two fields is equal to the synchronized time when the Sync message is sent in both cases.

The ratio of the grandmaster frequency to the frequency of the LocalClock entity at PTP Instance *i*, rateRatio_i, is equal to the same quantity at PTP Instance *i*–1, rateRatio_{i-1}, multiplied by the ratio of the frequency of the LocalClock entity at PTP Instance *i*–1 to the frequency of the LocalClock entity at PTP Instance *i*, neighborRateRatio (see 10.2.5.7). If neighborRateRatio is sufficiently small, this is approximately equal to the sum of rateRatio_{i-1} and the quantity neighborRateRatio–1, which is the frequency offset of PTP Instance *i*–1 relative to PTP Instance *i*. This computation is done by the PortSyncSyncReceive state machine (see 10.2.8).

NOTE—The sending of time-synchronization information by the master ports of a PTP Instance might or might not be tightly synchronized with the receipt of time-synchronization information by the slave port. If a master port has the same logMessageInterval as the slave port, it will transmit timing event messages as soon as possible after the slave port has received the corresponding timing event messages, and the master port is operating in "syncLocked" mode (see 10.2.5.15). If a master port and slave port have different logMessageInterval values, then the master port can send timing event messages without any synchronization with the slave port.

11.1.4 Model of operation

A PTP Instance contains one MD entity per PTP Instance, per port. This entity contains functions generic to all media, which are described in Clause 10, and functions specific to the respective medium for the link. Functions specific to full-duplex, point-to-point links are described in the current clause.

NOTE—IEEE 802.3 full-duplex, point-to-point links are in the category of links specified in this clause.

The model for a PTP Instance of a time-aware system with full-duplex, point-to-point links is shown in Figure 11-3. It assumes the presence of one full-duplex, point-to-point MD entity per port. The media-independent entities shown in Figure 11-3 are described in 10.1.1.

A general, media-independent description of the generation of timestamps is given in 8.4.3. A more specific description for PTP event messages is given in 11.3.2.1. A PTP event message is timestamped relative to the LocalClock entity when the message timestamp point (see 3.12) crosses the timestamp measurement plane (see 3.28). The timestamp is corrected for any ingressLatency or egressLatency (see 8.4.3) to produce a timestamp relative to the reference plane (see 3.21). The corrected timestamp value is provided to the MD entity.

The MD entity behavior and detailed state machines specific to full-duplex, point-to-point links are described in 11.2. The behavior of the MD entity that is generic to all media is described in Clause 10.

11.2 State machines for MD entity specific to full-duplex, point-to-point links

11.2.1 General

This subclause describes the media-dependent state machines for an MD entity, for the case of full-duplex, point-to-point links. The state machines are all per port, because an instance of each is associated with an MD entity. The state machines are as follows:

a) MDSyncReceiveSM (shown in Figure 10-2, and in more detail in Figure 11-4): receives Sync and, if the received information is two-step, Follow_Up messages, and sends the time-synchronization information carried in these messages to the PortSync entity of the same port. There is one instance of this state machine per port, per domain.

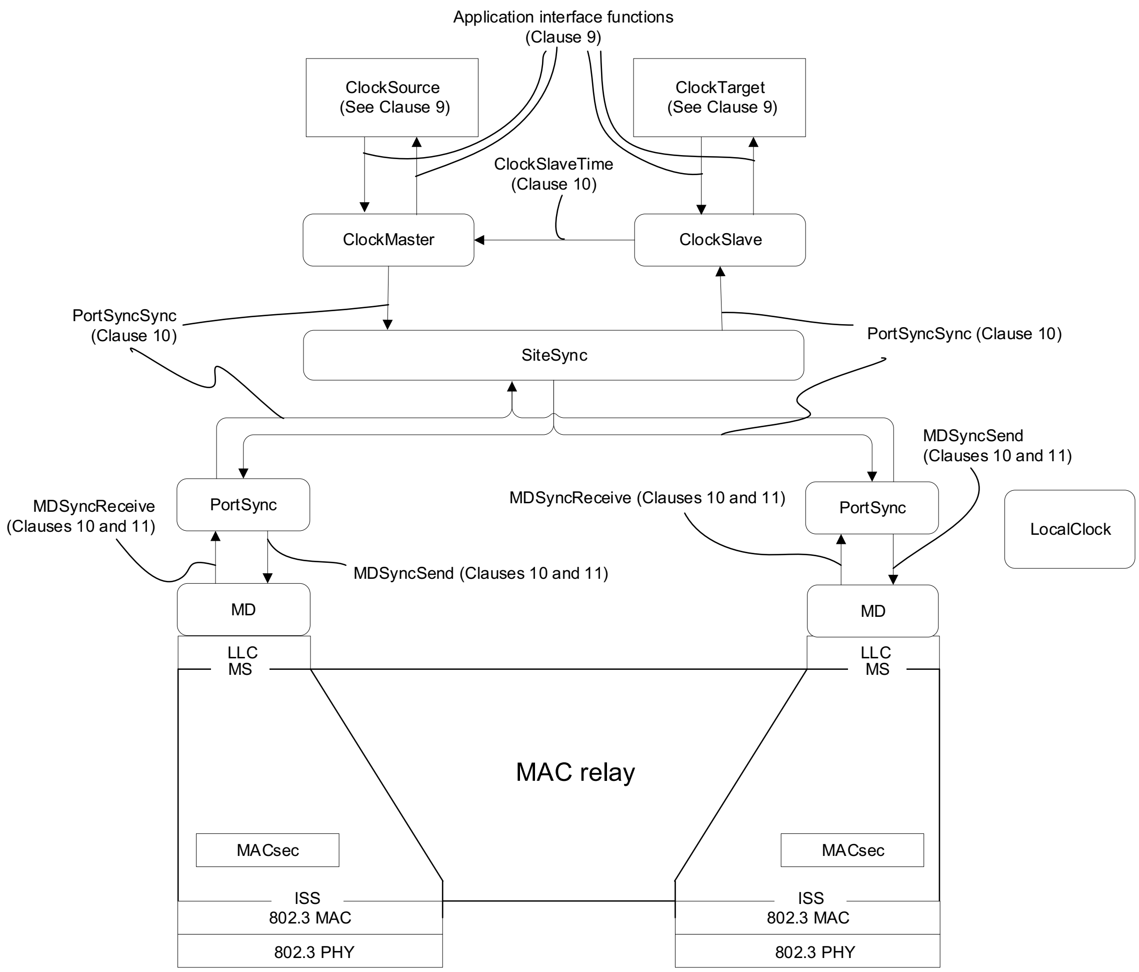


Figure 11-3—Model for a PTP Instance of a time-aware system with full-duplex, point-to-point links

b) MDSyncSendSM (shown in Figure 10-2, and in more detail in Figure 11-4): receives an MDSyncSend structure from the PortSync entity of the same port, transmits a Sync message, uses the syncEventEgressTimestamp, corrected for egressLatency, and information contained in the MDSyncSend structure to compute information needed for the Sync message if the port is currently operating as a one-step port and for the corresponding Follow_Up message if the port is currently operating as a two-step port, and transmits the Follow_Up message if the port is two-step. There is one instance of this state machine per port, per domain.

c) MDPdelayReq (shown in Figure 11-5): transmits a Pdelay_Req message, receives Pdelay_Resp and Pdelay_Resp_Follow_Up messages corresponding to the transmitted Pdelay_Req message, uses the information contained in successive Pdelay_Resp and Pdelay_Resp_Follow_Up messages to compute the ratio of the frequency of the LocalClock entity in the PTP Instance at the other end of the attached link to the frequency of the LocalClock entity in this PTP Instance, and uses the information obtained from the message exchange and the computed frequency ratio to compute propagation delay on the attached link. There is one instance of this state machine for all the domains, per port.

d) MDPdelayResp (shown in Figure 11-5): receives a Pdelay_Req message from the MD entity at the other end of the attached link, and responds with Pdelay_Resp and Pdelay_Resp_Follow_Up messages. There is one instance of this state machine for all the domains, per port.

e) SyncIntervalSetting state machine (not shown): receives a Signaling message that contains a Message Interval Request TLV (see 10.6.4.3), and sets the global variables that give the duration of the mean intervals between successive Sync messages. There is one instance of this state machine

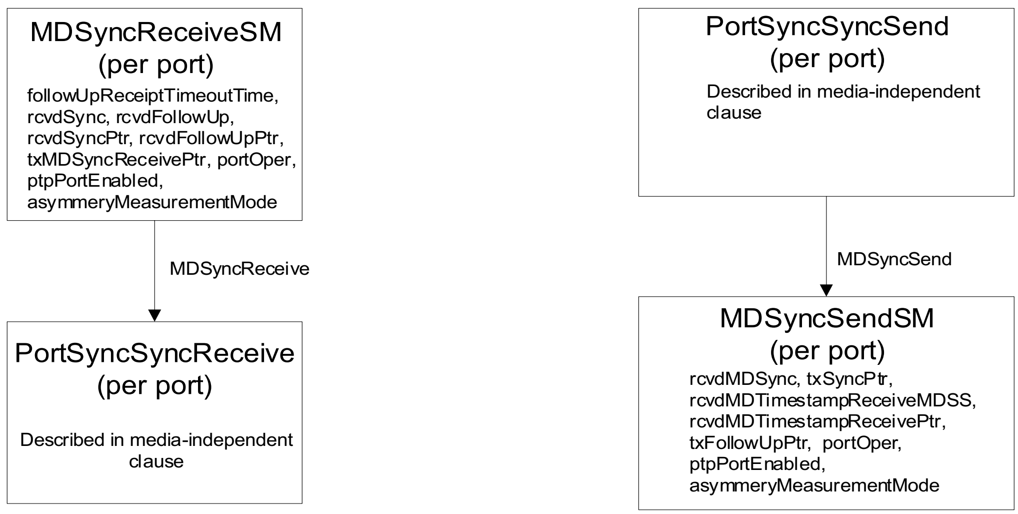


Figure 11-4—Detail of MD entity time-synchronization state machines for full-duplex, point-to-point links

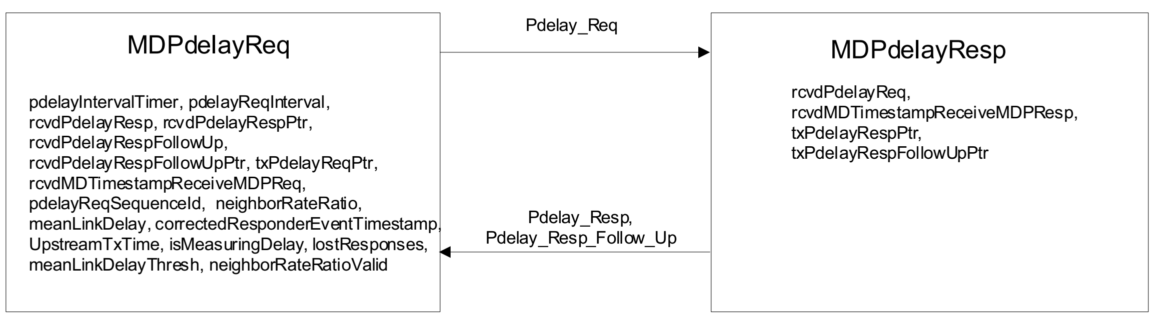


Figure 11-5—peer-to-peer delay mechanism state machines—overview and interrelationships

per port, per domain.

f) LinkDelayIntervalSetting state machine (not shown): receives a Signaling message that contains a Message Interval Request TLV (see 10.6.4.3), and sets the global variables that give the duration of the mean intervals between successive Pdelay_Req messages. There is one instance of this state machine for all the domains, per port.

Figure 10-2, Figure 11-4, and Figure 11-5 are not themselves state machines, but illustrate the machines, their interrelationships, the principle variables and messages used to communicate between them, their local variables, and performance parameters. The figures do not show the service interface primitives between the media-dependent layer and the LLC. Figure 11-5 is analogous to Figure 10-2; while Figure 10-2 applies to the general time-synchronization protocol, Figure 11-5 is limited to the peer-to-peer delay mechanism for measurement of propagation delay in full-duplex, point-to-point links. Figure 11-4 shows greater detail of the MDSyncReceiveSM and MDSyncSendSM state machines than Figure 10-2, for the case of full-duplex, point-to-point links.

The state machines described in 11.2 and its subclauses use some of the global per PTP Instance system variables defined in 10.2.4, the global per-port variables defined in 10.2.5, and the functions defined in 10.2.6.

11.2.2 Determination of asCapable and asCapableAcrossDomains

There is one instance of the global variable asCapable (see 10.2.5.1) per port, per domain. There is one instance of the global variable asCapableAcrossDomains (see 11.2.13.12, per port, that is common across and accessible by all the domains.

The per-port global variable asCapable (see 10.2.5.1) indicates whether or not the IEEE 802.1AS protocol is operating, in this domain, on the link attached to this port, and can provide the time-synchronization performance described in B.3. asCapable is used by the PortSync entity, which is media-independent; however, the determination of asCapable is media-dependent.

The per-port global variable asCapableAcrossDomains is set by the MDPdelayReq state machine (see 11.2.19 and Figure 11-9). For a port attached to a full-duplex, point-to-point link, asCapableAcrossDomains shall be set to TRUE if and only if it is determined, via the peer-to-peer delay mechanism, that the following conditions hold for the port:

a) The port is exchanging peer delay messages with its neighbor,
b) The measured delay does not exceed meanLinkDelayThresh,
c) The port does not receive multiple Pdelay_Resp or Pdelay_Resp_Follow_Up messages in response to a single Pdelay_Req message, and
d) The port does not receive a response from itself or another port of the same PTP Instance.

NOTE—If a PTP Instance implements only domain 0 and the MDPdelayReq and MDPdelayResp state machines are invoked on domain 0 (see 11.2.19), asCapableAcrossDomains is still set by the MDPdelayReq state machine.

The default value of meanLinkDelayThresh shall be set as specified in Table 11-1.

The per-port, per-domain global variable asCapable shall be set to TRUE if and only if the following conditions hold:

e) The value of asCapableAcrossDomains is TRUE, and
f) One of the following conditions holds:
   1) The value of neighborGptpCapable for this port is TRUE, or
   2) The value of domainNumber is zero, and the value of sdoId for peer delay messages received on this port is 0x100.

NOTE—Condition (f)(2) ensures backward compatibility with the 2011 edition of this standard. A PTP Instance compliant with the current edition of this standard that is attached, via a full-duplex, point-to-point link, to a PTP Instance compliant with the 2011 edition of this standard will not receive Signaling messages that contain the gPTP capable TLV and will not set neighborGptpCapable to TRUE. However, condition (f)(2) ensures that asCapable for this port and domain (i.e., domain 0) will still be set to TRUE if condition (e) holds, because the peer delay messages received from the time-aware system compliant with the 2011 edition of this standard will have sdoId set to 0x100.

Table 11-1—Value of meanLinkDelayThresh for various links

Link	Value of meanLinkDelayThresh (ns) (see NOTE)
100BASE-TX, 1000BASE-T	800_{10}
100BASE-FX, 1000BASE-X	FFFF FFFF FFFF FFFF FFFF $FFFF_{16}$

NOTE—The actual propagation delay for 100BASE-TX and 1000BASE-T links is expected to be smaller than the above respective threshold. If the measured mean propagation delay (i.e., meanLinkDelay, see 10.2.5.8) exceeds this threshold, it is assumed that this is due to the presence of equipment that does not implement gPTP. For 100BASE-FX and 1000BASE-X links, the actual propagation delay can be on the order of or larger than the delay produced by equipment that does not implement gPTP and, therefore, such equipment cannot be detected by comparing measured propagation delay with a threshold. Therefore, in this case meanLinkDelayThresh is set to the largest possible value (i.e., all 1s).

11.2.3 Use of MAC Control PAUSE operation

A PTP Instance shall not use the MAC Control PAUSE operation.

11.2.4 Use of priority-based flow control

A PTP Instance that implements priority-based flow control shall neither transmit nor honor upon receipt priority-based flow control messages that act on the IEEE 802.1AS message priority code point (see 8.4.4).

11.2.5 Use of link aggregation

gPTP PDUs, when used with physical ports that are attached to an IEEE 802.1AX Link Aggregation Group, shall be transmitted and received over the physical ports (Aggregation Ports), not the Aggregator Port. Using the Parser/Multiplexer functions described in Clauses 6.1.3 and 6.2.7 of IEEE 802.1AX-2014, received gPTP PDUs are recognized as control frames, separated from the incoming data stream, and directed to the gPTP layers. gPTP PDUs output from the gPTP layers are integrated into the port's data stream by the Parser/Multiplexer.

Assuming that ptpPortEnabled is true for all the physical links (Aggregation Links) in a Link Aggregation Group, and that all the physical links are connected to the same two systems, gPTP will measure the delay on each physical link, and the IEEE 802.1AS protocol will choose one of the physical links for transmitting time from the master clock.

11.2.6 Service interface primitives and data structures communicated between state machines

The following subclauses describe the service primitives and data structures communicated between the time-synchronization state machines of the MD entity. First the service primitives are described, followed by the data structures.

11.2.7 DL-UNITDATA.request

This service primitive is used by an MD entity to request to the associated LLC the transmission of a Sync, Follow_Up, Pdelay_Req, Pdelay_Resp, or Pdelay_Resp_Follow_Up message. The primitive is described in 2.2.1.1.1 of ISO/IEC 8802-2 [B7].

11.2.8 DL-UNITDATA.indication

This service primitive is used by the LLC to indicate to the associated MD entity the receipt of a Sync, Follow_Up, Pdelay_Req, Pdelay_Resp, or Pdelay_Resp_Follow_Up message. The primitive is described in 2.2.1.1.1 of ISO/IEC 8802-2 [B7].

11.2.9 MDTimestampReceive

11.2.9.1 General

This structure provides the timestamp, relative to the timestamp measurement plane, of the event message that was just sent or just received. The structure is received by the MD entity of the port. The MD entity corrects this timestamp for any ingressLatency or egressLatency (see 8.4.3) before using it and/or passing it to higher layer entities.

The structure is:

```
MDTimestampReceive{
        timestamp
}
```

The member of the structure is defined in the following subclause.

11.2.9.2 timestamp (UScaledNs)

The timestamp is the value of the timestamp of the event message (i.e., Sync, Pdelay_Req, Pdelay_Resp) that was just transmitted or received. This timestamp is taken relative to the timestamp measurement plane.

11.2.10 MDSyncReceive

This structure is specified in 10.2.2.2.

11.2.11 MDSyncSend

This structure is specified in 10.2.2.1.

11.2.12 Overview of MD entity global variables

The subclause that follows this subclause, i.e., 11.2.13, defines global variables used by the MD entity state machines whose scopes are per PTP Instance, common across all PTP Instances (e.g., used by CMLDS, see

11.2.17), per port per PTP Instance, and per port but common across all PTP Instances. Table 11-2 summarizes the scope of each global variable of Table 11.2.13.

Table 11-2 — Summary of scope of global variables used by time synchronization state machines (see 10.2.4 and 10.2.5)

Variable name	Subclause of definition	Per PTP Instance (i.e., per domain)	Per PTP Instance, per PTP Port	Instance used by CMLDS (see 11.2.17), i.e., that is common across all LinkPorts	Instance used by CMLDS, per LinkPort
currentLogPdelayReqInterval	11.2.13.1	No	Yes[a]	No	Yes
initialLogPdelayReqInterval	11.2.13.2	No	Yes[a]	No	Yes
pdelayReqInterval	11.2.13.3	No	Yes[a]	No	Yes
allowedLostResponses	11.2.13.4	No	Yes[a]	No	Yes
allowedFaults	11.2.13.5	No	Yes[a]	No	Yes
isMeasuringDelay	11.2.13.6	No	Yes[a]	No	Yes
meanLinkDelayThresh	11.2.13.7	No	Yes[a]	No	Yes
syncSequenceId	11.2.13.8	No	Yes	No	No
oneStepReceive	11.2.13.9	No	Yes	No	No
oneStepTransmit	11.2.13.10	No	Yes	No	No
oneStepTxOper	11.2.13.11	No	Yes	No	No
asCapableAcrossDomains	11.2.13.12	No	No	No	Yes

[a]The instance of this variable that is per PTP Instance, per PTP Port exists only for domain 0.

11.2.13 MD entity global variables

11.2.13.1 currentLogPdelayReqInterval: the current value of the logarithm to base 2 of the mean time interval, in seconds, between the sending of successive Pdelay_Req messages (see 11.5.2.2). This value is set in the LinkDelayIntervalSetting state machine (see 11.2.21). The data type for currentLogPdelayReqInterval is Integer8. There is one instance of this variable for all the domains (per port). The variable is accessible by all the domains.

11.2.13.2 initialLogPdelayReqInterval: the initial value of the logarithm to base 2 of the mean time interval, in seconds, between the sending of successive Pdelay_Req messages (see 11.5.2.2). The data type for initialLogPdelayReqInterval is Integer8. There is one instance of this variable for all the domains (per port). The variable is accessible by all the domains.

11.2.13.3 pdelayReqInterval: a variable containing the mean Pdelay_Req message transmission interval for the port corresponding to this MD entity. The value is set in the LinkDelayIntervalSetting state machine (see 11.2.21). The data type for pdelayReqInterval is UScaledNs. There is one instance of this variable for all the domains (per port). The variable is accessible by all the domains.

11.2.13.4 allowedLostResponses: the number of Pdelay_Req messages for which a valid response is not received, above which a port is considered to not be exchanging peer delay messages with its neighbor. The data type for allowedLostResponses is UInteger8. The default value and range of allowedLostResponses is given in 11.5.3. There is one instance of this variable for all the domains (per port). The variable is accessible by all the domains.

11.2.13.5 allowedFaults: the number of instances where meanLinkDelay (see 10.2.5.8) exceeds meanLinkDelayThresh (see 11.2.13.7) and/or the computation of neighborRateRatio is invalid (see 11.2.19.2.10), above which asCapableAcrossDomains is set to FALSE, i.e., the port will be considered to not be capable of interoperating with its neighbor via the IEEE 802.1AS protocol (see 10.2.5.1). The data type for allowedFaults is UInteger8. The default value and range of allowedFaults is given in 11.5.4. There is one instance of this variable for all the domains (per port). The variable is accessible by all the domains.

11.2.13.6 isMeasuringDelay: a Boolean that is TRUE if the port is measuring link propagation delay. For a full-duplex, point-to-point link, the port is measuring link propagation delay if it is receiving Pdelay_Resp and Pdelay_Resp_Follow_Up messages from the port at the other end of the link (i.e., it performs the measurement using the peer-to-peer delay mechanism). There is one instance of this variable for all the domains (per port). The variable is accessible by all the domains.

11.2.13.7 meanLinkDelayThresh: the propagation time threshold, above which a port is not considered capable of participating in the IEEE 802.1AS protocol. If meanLinkDelay (see 10.2.5.8) exceeds meanLinkDelayThresh, then asCapableAcrossDomains (see 11.2.13.12) is set to FALSE. The data type for meanLinkDelayThresh is UScaledNs. There is one instance of this variable for all the domains (per port). The variable is accessible by all the domains.

NOTE—The variable meanLinkDelayThresh was named neighborPropDelayThresh in the 2011 edition of this standard.

11.2.13.8 syncSequenceId: the sequenceId (see 11.4.2.7) for the next Sync message to be sent by this MD entity. The data type for syncSequenceId is UInteger16.

11.2.13.9 oneStepReceive: a Boolean variable that is TRUE if the port is capable of receiving one-step Sync messages.

11.2.13.10 oneStepTransmit: a Boolean variable that is TRUE if the port is capable of transmitting one-step Sync messages.

11.2.13.11 oneStepTxOper: a Boolean variable that is TRUE if the port will be transmitting one-step Sync messages (see 11.1.3).

11.2.13.12 asCapableAcrossDomains: a Boolean that isTRUE if and only if conditions (a)-(d) of 11.2.2 are satisfied. This Boolean is set by the MDPdelayReq state machine, and is used in determining asCapable for a port (see 11.2.2). There is one instance of this variable for all the domains (per port). The variable is accessible by all the domains. In the case where only one domain is active, asCapableAcrossDomains is equivalent to the variable asCapable (see 10.2.5.1)

11.2.14 MDSyncReceiveSM state machine

11.2.14.1 State machine variables

The following variables are used in the state diagram of 11.2.14.3:

11.2.14.1.1 followUpReceiptTimeoutTime: a variable used to save the time at which the information conveyed by a received Sync message will be discarded if the associated Follow_Up message is not received by then. The data type for followUpReceiptTimeoutTime is UScaledNs.

11.2.14.1.2 rcvdSync: a Boolean variable that notifies the current state machine when a Sync message is received. This variable is reset by the current state machine.

11.2.14.1.3 rcvdFollowUp: a Boolean variable that notifies the current state machine when a Follow_Up message is received. This variable is reset by the current state machine.

11.2.14.1.4 rcvdSyncPtr: a pointer to a structure whose members contain the values of the fields of the Sync message whose receipt is indicated by rcvdSync (see 11.2.14.1.2).

11.2.14.1.5 rcvdFollowUpPtr: a pointer to a structure whose members contain the values of the fields of the Follow_Up message whose receipt is indicated by rcvdFollowUp (see 11.2.14.1.3).

11.2.14.1.6 txMDSyncReceivePtrMDSR: a pointer to a structure whose members contain the values of the parameters of an MDSyncReceive structure to be transmitted.

11.2.14.1.7 upstreamSyncInterval: the sync interval (see 10.7.2.1) for the upstream port that sent the received Sync message. The data type for upstreamSyncInterval is UScaledNs.

11.2.14.2 State machine functions

The following functions are used in the state diagram of 11.2.14.3:

11.2.14.2.1 setMDSyncReceiveMDSR(): creates an MDSyncReceive structure, and returns a pointer to this structure. The members of this structure are set as follows:

a) If twoStepFlag of the most recently received Sync message is TRUE, followUpCorrectionField is set equal to the sum of the correctionField (see 11.4.2.5) of the most recently received Sync message, plus the correctionField of the most recently received Follow_Up message; else followUpCorrectionField is set equal to the correctionField of the most recently received Sync message,
b) sourcePortIdentity is set equal to the sourcePortIdentity (see 11.4.2.6) of the most recently received Sync message (see 11.4.3),
c) logMessageInterval is set equal to the logMessageInterval (see 11.4.2.8) of the most recently received Sync message (see 11.4.3),
d) If twoStepFlag of the most recently received Sync message is TRUE, preciseOriginTimestamp is set equal to the preciseOriginTimestamp (see 11.4.4.2.1) of the most recently received Follow_Up message; else preciseOriginTimestamp is set equal to the originTimestamp (see 11.4.3.2.1) of the most recently received Sync message,
e) rateRatio is set equal to the quantity (cumulativeScaledRateOffset$\times 2^{-41}$)+1.0, where the cumulativeScaledRateOffset field is for the most recently received Follow_Up information TLV (see 11.4.4.3.6),
f) upstreamTxTime is set equal to the syncEventIngressTimestamp for the most recently received Sync message (see 11.4.3), minus the mean propagation time on the link attached to this port (meanLinkDelay, see 10.2.5.8) divided by neighborRateRatio (see 10.2.5.7), and, if and only if the state machine is invoked by the instance-specific peer-to-peer delay mechanism, minus delayAsymmetry (see 10.2.5.9) for this port divided by rateRatio [see e) above]. The syncEventIngressTimestamp is equal to the timestamp value measured relative to the timestamp measurement plane, minus any ingressLatency (see 8.4.3). The upstreamTxTime can be written:

   State machine invoked by instance-specific peer-to-peer delay mechanism:
   upstreamTxTime = syncEventIngressTimestamp − (meanLinkDelay/RateRatio) − (delayAsymmetry/neighborRateRatio)

   State machine invoked by CMLDS:
   upstreamTxTime = syncEventIngressTimestamp − (meanLinkDelay/neighborRateRatio)

NOTE 1—The mean propagation time is divided by neighborRateRatio to convert it from the time base of the PTP Instance at the other end of the attached link to the time base of the current PTP Instance. If the instance-specific peer-to-peer delay mechanism is used (i.e., portDS.delayMechanism is P2P), delayAsymmetry is divided by rateRatio to convert it from the time base of the grandmaster to the time base of the current PTP Instance. The first quotient is then subtracted from syncEventIngressTimestamp, and the second quotient is subtracted from syncEventIngressTimestamp if the instance-specific peer-to-peer delay mechanism is used. The syncEventIngressTimestamp is measured relative to the time base of the current PTP Instance. See 11.2.17.2 for more detail.

NOTE 2—The difference between the mean propagation time in the grandmaster time base, the time base of the PTP Instance at the other end of the link, and the time base of the current PTP Instance is usually negligible. The same is true of any delayAsymmetry (see NOTE 2 of 11.2.19.3.4).

g) gmTimeBaseIndicator is set equal to the gmTimeBaseIndicator of the most recently received Follow_Up information TLV (see 11.4.4.3.7),
h) lastGmPhaseChange is set equal to the lastGmPhaseChange of the most recently received Follow_Up information TLV (see 11.4.4.3.8),
i) lastGmFreqChange is set equal to the scaledLastGmFreqChange of the most recently received Follow_Up information TLV (see 11.4.4.3.9), multiplied by 2^{-41}, and
j) domainNumber is set equal to the domainNumber of the most recently received Sync message (see 11.4.3).

11.2.14.2.2 txMDSyncReceive (txMDSyncReceivePtrMDSR): transmits an MDSyncReceive structure to the PortSyncSyncReceive state machine of the PortSync entity of this port.

11.2.14.3 State diagram

The MDSyncReceiveSM state machine shall implement the function specified by the state diagram in Figure 11-6, the local variables specified in 11.2.14.1, the functions specified in 11.2.14.2, the structure specified in 11.2.10 and 10.2.2.1, the messages specified in 11.4, and the relevant global variables and functions specified in 10.2.4 through 10.2.6. The state machine receives Sync and, if the received information is two-step, Follow_Up messages, places the time-synchronization information in an MDSyncReceive structure, and sends the structure to the PortSyncSyncReceive state machine of the PortSync entity of this port.

11.2.15 MDSyncSendSM state machine

11.2.15.1 State machine variables

The following variables are used in the state diagram of 11.2.15.3:

11.2.15.1.1 rcvdMDSyncMDSS: a Boolean variable that notifies the current state machine when an MDSyncSend structure is received. This variable is reset by the current state machine.

11.2.15.1.2 txSyncPtr: a pointer to a structure whose members contain the values of the fields of a Sync message to be transmitted.

11.2.15.1.3 rcvdMDTimestampReceiveMDSS: a Boolean variable that notifies the current state machine when the syncEventEgressTimestamp (see 11.3.2.1) for a transmitted Sync message is received. This variable is reset by the current state machine.

11.2.15.1.4 rcvdMDTimestampReceivePtr: a pointer to the received MDTimestampReceive structure (see 11.2.9).

11.2.15.1.5 txFollowUpPtr: a pointer to a structure whose members contain the values of the fields of a Follow_Up message to be transmitted.

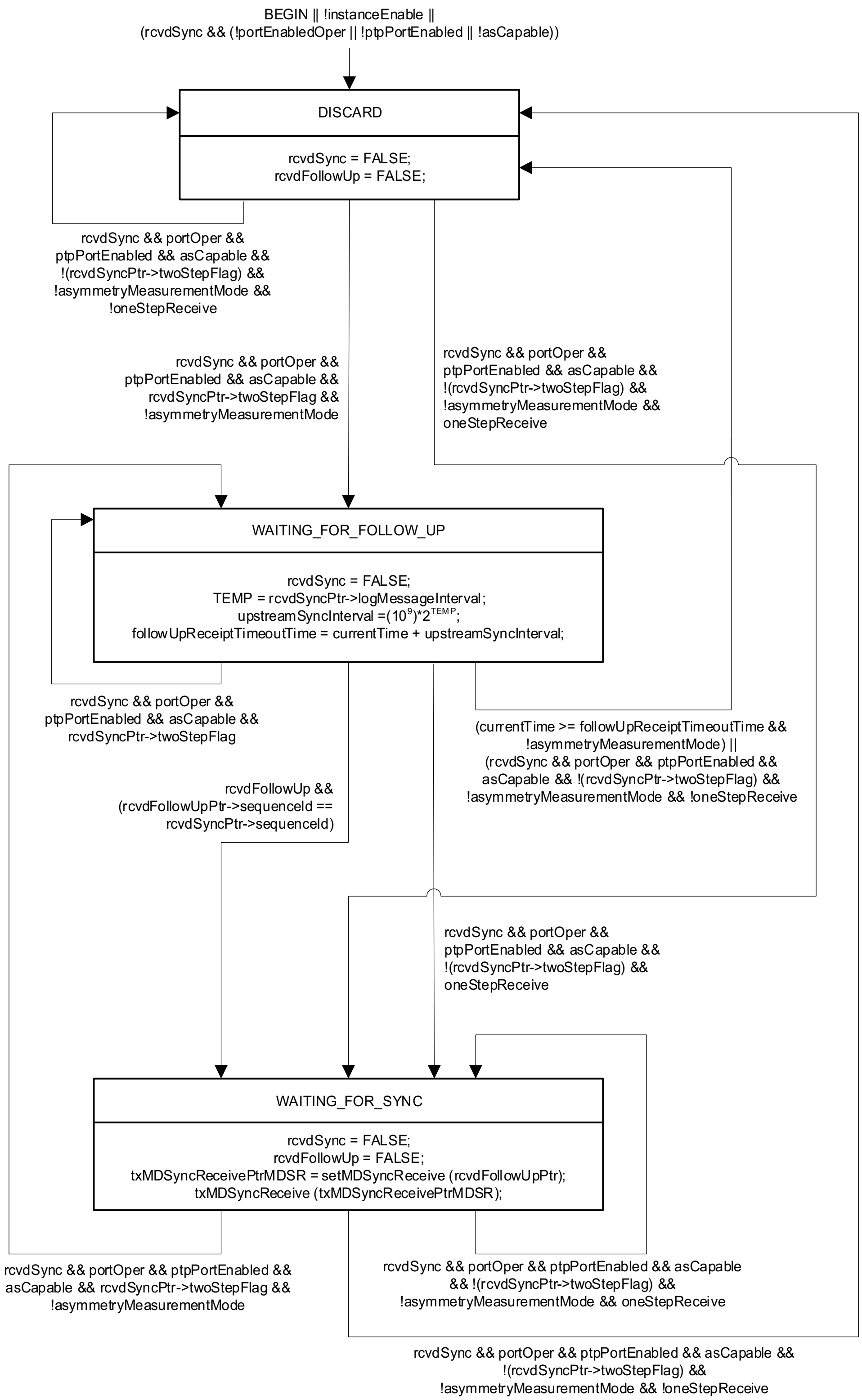


Figure 11-6—MDSyncReceiveSM state machine

11.2.15.2 State machine functions

The following functions are used in the state diagram of 11.2.15.3:

11.2.15.2.1 setSyncTwoStep(): creates a structure whose parameters contain the fields (see 11.4 and its subclauses) of a Sync message to be transmitted, and returns a pointer to this structure. The parameters are set as follows:

a) correctionField is set equal to 0,
b) sourcePortIdentity is set equal to the sourcePortIdentity member of the most recently received MDSyncSend structure (see 10.2.2.1 and 11.2.11),
c) sequenceId is set equal to syncSequenceId (see 11.2.13.8),
d) logMessageInterval is set equal to the logMessageInterval member of the most recently received MDSyncSend structure (see 10.2.2.1 and 11.2.11), and
e) remaining fields are set as specified in 11.4.2 and 11.4.3, for the case where twoStepFlag is TRUE.

11.2.15.2.2 txSync (txSyncPtr): transmits a Sync message from this MD entity, whose fields contain the parameters in the structure pointed to by txSyncPtr (see 11.2.15.1.2).

11.2.15.2.3 setFollowUp(): creates a structure whose parameters contain the fields (see 11.4 and its subclauses) of a Follow_Up message to be transmitted, and returns a pointer to this structure. The parameters are set as follows:

a) followUpCorrectionField is set equal to the sum of:
   1) the followUpCorrectionField member of the most recently received MDSyncSend structure (see 10.2.2.1 and 11.2.11)
   2) the quantity

   rateRatio × (syncEventEgressTimestamp − upstreamTxTime),

   where rateRatio is the rateRatio member of the most recently received MDSyncSend structure (see 10.2.2.1 and 11.2.11), upstreamTxTime is the upstreamTxTime member of the most recently received MDSyncSend structure (see 10.2.2.1 and 11.2.11), and syncEventEgressTimestamp is the timestamp pointed to by rcvdMDTimestampReceivePtr, corrected for egressLatency (see 8.4.3).

NOTE—If the PTP Instance that contains this PortSync entity is a PTP Relay Instance, the quantity

syncEventEgressTimestamp − upstreamTxTime

is the sum of the residence time and link propagation delay on the upstream link, relative to the LocalClock entity, and the quantity

rateRatio × (syncEventEgressTimestamp − upstreamTxTime)

is the sum of the residence time and link propagation delay on the upstream link, relative to the grandmaster (see 11.2.14.2.1 for details on the setting of upstreamTxTime).

b) sourcePortIdentity is set equal to the sourcePortIdentity member of the most recently received MDSyncSend structure (see 10.2.2.1 and 11.2.11),
c) sequenceId is set equal to syncSequenceId (see 11.2.13.8),
d) logMessageInterval is set equal to the logMessageInterval member of the most recently received MDSyncSend structure (see 10.2.2.1 and 11.2.11),
e) preciseOriginTimestamp is set equal to the preciseOriginTimestamp member of the most recently received MDSyncSend structure (see 10.2.2.1 and 11.2.11),
f) rateRatio is set equal to the rateRatio member of the most recently received MDSyncSend structure (see 10.2.2.1 and 11.2.11),

g) gmTimeBaseIndicator is set equal to the gmTimeBaseIndicator member of the most recently received MDSyncSend structure (see 10.2.2.1 and 11.2.11),
h) lastGmPhaseChange is set equal to the lastGmPhaseChange member of the most recently received MDSyncSend structure (see 10.2.2.1 and 11.2.11),
i) lastGmFreqChange is set equal to the scaledLastGmFreqChange member of the most recently received MDSyncSend structure (see 10.2.2.1 and 11.2.11), multiplied by 2^{41},
j) domainNumber is set equal to the domainNumber of the most recently received MDSyncSend structure (see 10.2.2.1 and 11.2.11), and
k) remaining fields are set as specified in 11.4.2 and 11.4.4.

11.2.15.2.4 txFollowUp (txFollowUpPtr): transmits a Follow_Up message from this MD entity, whose fields contain the parameters in the structure pointed to by txFollowUpPtr (see 11.2.15.1.5).

11.2.15.2.5 setSyncOneStep(): creates a structure whose parameters contain the fields (see 11.4 and its subclauses) of a Sync message to be transmitted, and returns a pointer to this structure. The parameters are set as follows:

a) correctionField is set equal to the followUpCorrectionField member of the most recently received MDSyncSend structure (see 10.2.2.1 and 11.2.11)
b) sourcePortIdentity is set equal to the sourcePortIdentity member of the most recently received MDSyncSend structure (see 10.2.2.1 and 11.2.11),
c) sequenceId is set equal to syncSequenceId (see 11.2.13.8),
d) logMessageInterval is set equal to the logMessageInterval member of the most recently received MDSyncSend structure (see 10.2.2.1 and 11.2.11),
e) originTimestamp is set equal to the preciseOriginTimestamp member of the most recently received MDSyncSend structure (see 10.2.2.1 and 11.2.11),
f) rateRatio is set equal to the rateRatio member of the most recently received MDSyncSend structure (see 10.2.2.1 and 11.2.11),
g) gmTimeBaseIndicator is set equal to the gmTimeBaseIndicator member of the most recently received MDSyncSend structure (see 10.2.2.1 and 11.2.11),
h) lastGmPhaseChange is set equal to the lastGmPhaseChange member of the most recently received MDSyncSend structure (see 10.2.2.1 and 11.2.11),
i) lastGmFreqChange is set equal to the scaledLastGmFreqChange member of the most recently received MDSyncSend structure (see 10.2.2.1 and 11.2.11), multiplied by 2^{41},
j) domainNumber is set equal to the domainNumber of the most recently received MDSyncSend structure (see 10.2.2.1 and 11.2.11), and
k) remaining fields are set as specified in 11.4.2 and 11.4.3, for the case where twoStepFlag is FALSE.

11.2.15.2.6 modifySync(): adds to the correctionField of the transmitted Sync message, after the syncEventEgressTimestamp is taken and known, the quantity:

$$\text{rateRatio} \times (\text{syncEventEgressTimestamp} - \text{upstreamTxTime}),$$

where rateRatio is the rateRatio member of the most recently received MDSyncSend structure (see 10.2.2.1 and 11.2.11), upstreamTxTime is the upstreamTxTime member of the most recently received MDSyncSend structure (see 10.2.2.1 and 11.2.11), and syncEventEgressTimestamp is the timestamp pointed to by rcvdMDTimestampReceivePtr, corrected for egressLatency (see 8.4.3).

NOTE—If the PTP Instance that contains this PortSync entity is a PTP Relay Instance, the quantity

$$\text{syncEventEgressTimestamp} - \text{upstreamTxTime}$$

is the sum of the residence time and link propagation delay on the upstream link, relative to the LocalClock entity, and the quantity rateRatio × (syncEventEgressTimestamp – upstreamTxTime)

is the sum of the residence time and link propagation delay on the upstream link, relative to the grandmaster (see 11.2.14.2.1 for details on the setting of upstreamTxTime).

11.2.15.3 State diagram

The MDSyncSendSM state machine shall implement the function specified by the state diagram in Figure 11-7; the local variables specified in 11.2.15.1; the functions specified in 11.2.15.2; the structures specified in 11.2.9, 11.2.11 and 10.2.2.1; the messages specified in 11.4; the MD entity global variables specified in 11.2.13; and the relevant global variables and functions specified in 10.2.4 through 10.2.6. The state machine receives an MDSyncSend structure from the PortSyncSyncSend state machine of the PortSync entity of this port and transmits a Sync and corresponding Follow_Up message.

11.2.16 OneStepTxOperSetting state machine

11.2.16.1 State machine variables

The following variables are used in the state diagram of 11.2.16.2:

11.2.16.1.1 rcvdSignalingMsg4: a Boolean variable that notifies the current state machine when a Signaling message that contains a Message Interval Request TLV (see 10.6.4.3) is received. This variable is reset by the current state machine.

11.2.16.1.2 rcvdSignalingPtrOSTOS: a pointer to a structure whose members contain the values of the fields of the received Signaling message that contains a Message Interval Request TLV (see 10.6.4.3).

11.2.16.2 State diagram

The OneStepTxOperSetting state machine shall implement the function specified by the state diagram in Figure 11-8, the local variables specified in 11.2.16.1, the messages specified in 10.6 and 11.4, the relevant global variables specified in 10.2.5 and 11.2.13, and the relevant managed objects specified in 14.8. This state machine is responsible for setting the relevant global variables and managed objects pertaining to one-step/two-step operation.

11.2.17 Common Mean Link Delay Service

11.2.17.1 General

Each port of a time-aware system invokes a single instance of the MDPdelayReq state machine (see 11.2.19) and the MDPdelayResp state machine (see 11.2.20). If the time-aware system implements more than one domain, these two state machines shall provide a Common Mean Link Delay Service (CMLDS), as described in this subclause, that measures mean propagation delay on the link attached to the port and the neighbor rate ratio for the port (i.e., the ratio of the frequency of the LocalClock entity of the time-aware system at the other end of the link attached to this port, to the frequency of the LocalClock entity of this time-aware system). The CMLDS makes the mean propagation delay and neighbor rate ratio available to all active domains. If the time-aware system implements one domain (the domainNumber of this domain is 0, see 8.1), these two state machines may provide the CMLDS; however, if they do not provide the CMLDS (i.e., if only the PTP Instance-specific peer delay mechanism is provided), they shall be invoked on domain 0. This means that if the domain number is not 0, portDS.delayMechanism (see 14.8.5 and Table 14-8) must not be P2P.

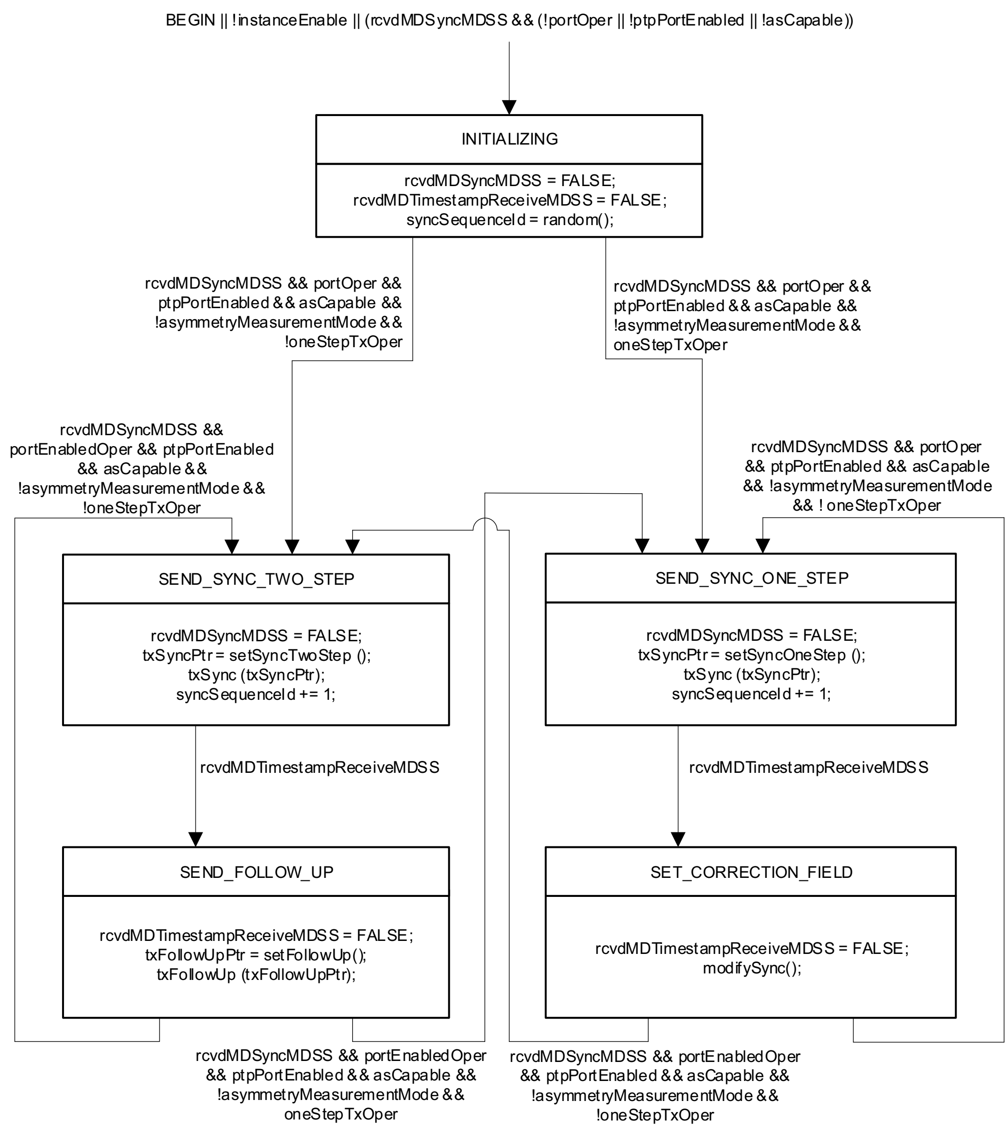


Figure 11-7—MDSyncSendSM state machine

NOTE 1—In the above sentence, the condition that the time-aware system implements only one domain implicitly assumes that IEEE 802.1AS is the only PTP profile present on the respective port of the time-aware system, i.e., no other PTP profiles are implemented on that port. If other PTP profiles that use the CMLDS are present on the port, the CMLDS must be provided.

In accordance with IEEE Std 1588-2019, the term Link Port is used to refer to a port of the CMLDS. A gPTP Port for which portDS.delayMechanism is COMMON_P2P uses the CMLDS provided by the Link Port whose cmldsLinkPortDS.portIdentity.portNumber (see 14.16.2) is equal to the commonServicesPortDS.cmldsLinkPortPortNumber (see 14.14.2) for this gPTP port.

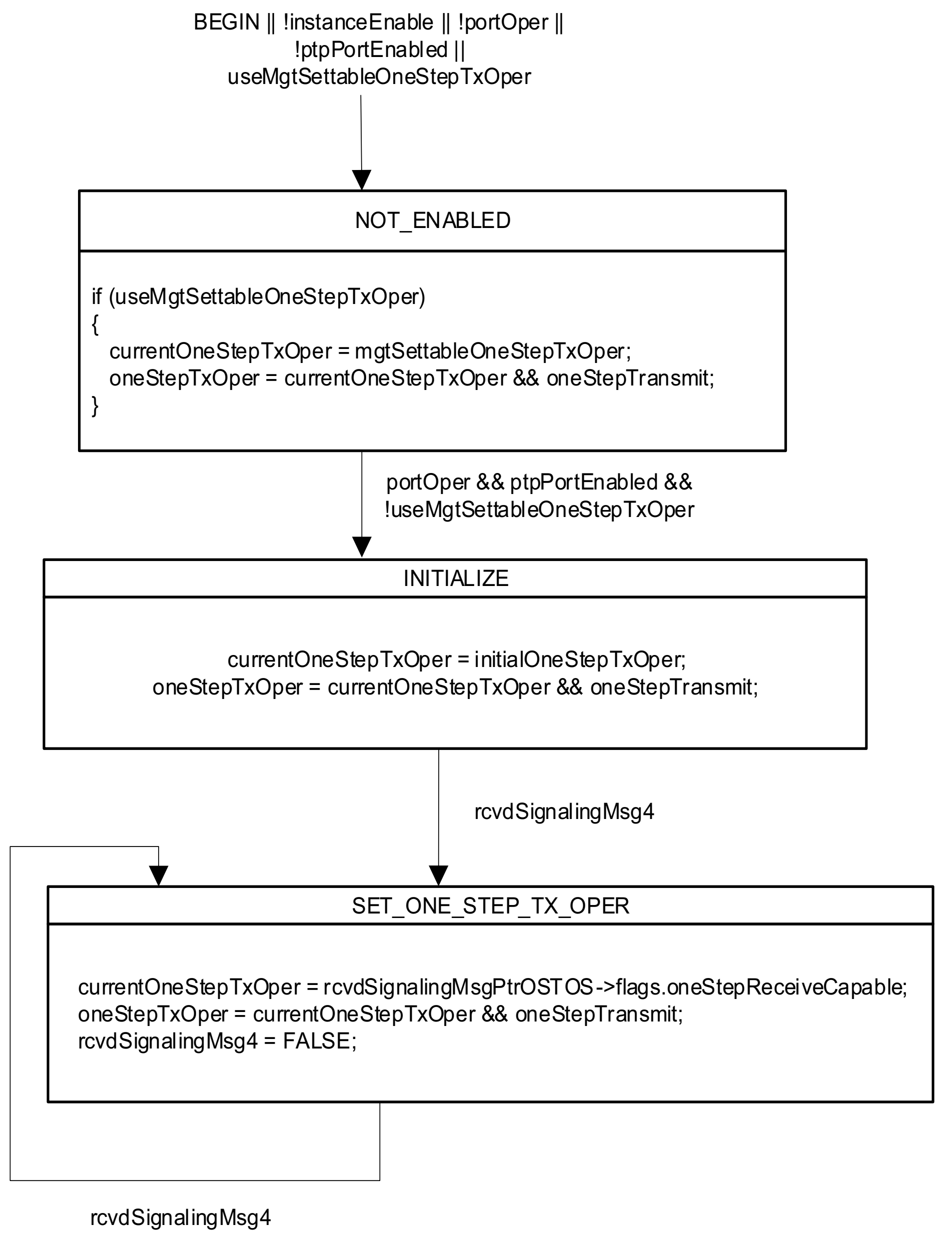


Figure 11-8—OneStepTxOperSetting state machine

The value of majorSdoId for the CMLDS shall be 0x2. The value of minorSdoId for the Common Mean Link Delay Service shall be 0x00. This means that the value of sdoId for the Common Mean Link Delay Service is 0x200.

NOTE 2—The above requirements for majorSdoId and minorSdoId are for the CMLDS. The requirements for gPTP domains, including instance-specific peer delay messages, are given in 8.1.

If a gPTP Port that invokes the CMLDS receives a Pdelay_Req message with majorSdoId value of 0x1, minorSdoId value of 0x00, and domainNumber value of 0, the port shall respond with PTP Instance-specific peer delay messages (i.e., the Pdelay_Resp and Pdelay_Resp_Follow_Up corresponding to this Pdelay_Req) using the instance-specific peer-to-peer delay mechanism. These instance-specific messages have majorSdoId value of 0x1, minorSdoId value of 0x00, and domainNumber value of 0.

NOTE 3—The above requirement ensures (a) backward compatibility with time-aware systems that comply with the 2011 version of this standard, and (b) compatibility with time-aware systems that implement only one domain and invoke the MDPdelayReq and MDPdelayResp state machines on domain 0.

NOTE 4—In general, a port can receive (i) peer delay messages of the CMLDS, (ii) PTP Instance-specific peer delay messages of domain 0 (with sdoId of 0x100), and, (iii) if there are other PTP profiles on the neighbor port that use instance-specific peer delay, peer delay messages of those profiles. The port responds to the messages of (i) if it invokes CMLDS, the messages of (ii) if it invokes gPTP domain 0, and the messages of (iii) if it invokes the respective other PTP profiles.

The CMLDS shall be enabled on a Link Port if the value of portDS.delayMechanism (see 14.8.5) is COMMON_P2P for at least one PTP Port that is enabled (i.e., for which portOper and ptpPortEnabled are both TRUE) and corresponds to the same physical port as the Link Port (see 14.1). The value of cmldsLinkPortEnabled is TRUE if the CMLDS is enabled on the Link Port, and FALSE if the CMLDS is not enabled on the Link Port.

11.2.17.2 Differences between instance-specific peer-to-peer delay mechanism and CMLDS in computations of mean link delay and effect of delayAsymmetry

The MDPdelayReq state machine (see 11.2.19), MDPdelay_Resp state machine (see 11.2.20), and MDSyncReceiveSM state machine (see 11.2.14) perform various computations of mean link delay and of the effect of delayAsymmetry differently, depending on whether the respective computations are done using the instance-specific peer-to-peer delay mechanism or using CMLDS. The differences are:

a) Instance-specific peer delay computes mean link delay averaged over the two directions, and adds delayAsymmetry separately when computing upstreamTxTime by the setMDSyncReceiveMDSR() function of the MDSyncReceive state machine, while CMLDS corrects the computed mean link delay for delayAsymmetry, and therefore does not need to add it separately (see 11.2.14.2.1).
b) Instance-specific peer delay sets the correctionField of a transmitted Pdelay_Req message to 0, while CMLDS sets it to -delayAsymmetry (see 11.2.19.3.1).
c) Instance-specific peer delay sets the correctionField of Pdelay_Resp equal to the fractional nanosecond portion of the pdelayReqEventIngressTimestamp of the corresponding Pdelay_Req, while CMLDS sets the correctionField of Pdelay_Resp equal to minus the fractional nanoseconds portion of the pdelayReqEventIngressTimestamp of the corresponding Pdelay_Req (see 11.2.20.3.1).
d) Instance-specific peer delay sets the correctionField of Pdelay_Resp_Follow_Up equal to the fractional nanosecond portion of the pdelayRespEventEgressTimestamp, while CMLDS sets the correctionField of Pdelay_Resp_Follow_Up equal to the sum of the correctionField of the corresponding Pdelay_Req and the fractional nanosecond portion of the pdelayRespEventEgressTimestamp (see 11.2.20.3.3).
e) When computing mean link delay (i.e., the quantity D in Eq. (11-5)), the correctionField of the Pdelay_Resp message, divided by 2^{16}, is added when computing the quantity $t2$ if instance-specific peer delay is used, while it is subtracted if CMLDS is used (see 11.2.19.3.4).
f) When computing neighborRateRatio, the computation of the correctResponderEventTimestamp must be corrected for delayAsymmetry if, and only if, CMLDS is used. This is because, in the case of CMLDS, delayAsymmetry was subtracted from the Pdelay_Req correctionField, and then the Pdelay_Resp_Follow_Up correctionField was set equal to the sum of the Pdelay_Req correctionField and the fractional nanoseconds portion of the PdelayRespEventEgressTimestamp,

while in the case of instance-specific peer delay, the correctionField of Pdelay_Req was set equal to 0 (see 11.2.19.3.1, 11.2.19.3.3, and 11.2.20.3.3).

The computations in this standard for the case of the instance-specific peer-to-peer delay mechanism are the same as in IEEE 802.1AS-2011, for backward compatibility. However, the computations in this standard for the case of CMLDS must be consistent with IEEE Std 1588-2019, because CMLDS can be used by other PTP profiles, in addition to the PTP profile included in IEEE 802.1AS, that might be present in a gPTP node. Therefore, the computations for the cases of the instance-specific peer-to-peer delay mechanism and CMLDS are different.

11.2.18 Common Mean Link Delay Service global variables

11.2.18.1 cmldsLinkPortEnabled: a per-Link-Port Boolean that is TRUE if both the value of portDS.delayMechanism is Common_P2P and the value of portDS.ptpPortEnabled is TRUE, for at least one PTP Port that uses the CMLDS that is invoked on the Link Port; otherwise, the value is FALSE.

11.2.19 MDPdelayReq state machine

11.2.19.1 General

This state machine is invoked as part of the Common Mean Link Delay Service. There is one instance of this state machine for all the domains (per port). This means that there also is one instance of each of the state machine variables of 11.2.19.2, state machine functions of 11.2.19.3, and relevant global variables of 10.2.5 and 11.2.13 for all the domains (per port). None of the variables used or functions invoked in this state machine are specific to a single domain. However, the single instances of all of these objects or entities are accessible to all the domains.

NOTE—This state machine uses the variable asCapableAcrossDomains (see 11.2.13.12). In the case where only one domain is active, asCapableAcrossDomains is equivalent to the variable asCapable.

11.2.19.2 State machine variables

The following variables are used in the state diagram of 11.2.19.4:

11.2.19.2.1 pdelayIntervalTimer: a variable used to save the time at which the Pdelay_Req interval timer is started (see Figure 11-9). A Pdelay_Req message is sent when this timer expires. The data type for pdelayIntervalTimer is UScaledNs.

11.2.19.2.2 rcvdPdelayResp: a Boolean variable that notifies the current state machine when a Pdelay_Resp message is received. This variable is reset by the current state machine.

11.2.19.2.3 rcvdPdelayRespPtr: a pointer to a structure whose members contain the values of the fields of the Pdelay_Resp message whose receipt is indicated by rcvdPdelayResp (see 11.2.19.2.2).

11.2.19.2.4 rcvdPdelayRespFollowUp: a Boolean variable that notifies the current state machine when a Pdelay_Resp_Follow_Up message is received. This variable is reset by the current state machine.

11.2.19.2.5 rcvdPdelayRespFollowUpPtr: a pointer to a structure whose members contain the values of the fields of the Pdelay_Resp_Follow_Up message whose receipt is indicated by rcvdPdelayRespFollowUp (see 11.2.19.2.4).

11.2.19.2.6 txPdelayReqPtr: a pointer to a structure whose members contain the values of the fields of a Pdelay_Req message to be transmitted.

11.2.19.2.7 rcvdMDTimestampReceiveMDPReq: a Boolean variable that notifies the current state machine when the pdelayReqEventEgressTimestamp (see 11.3.2.1) for a transmitted Pdelay_Req message is received. This variable is reset by the current state machine.

11.2.19.2.8 pdelayReqSequenceId: a variable that holds the sequenceId for the next Pdelay_Req message to be transmitted by this MD entity. The data type for pdelayReqSequenceId is UInteger16.

11.2.19.2.9 lostResponses: a count of the number of consecutive Pdelay_Req messages sent by the port, for which Pdelay_Resp and/or Pdelay_Resp_Follow_Up messages are not received. The data type for lostResponses is UInteger16.

11.2.19.2.10 neighborRateRatioValid: a Boolean variable that indicates whether or not the function computePdelayRateRatio() (see 11.2.19.3.3) successfully computed neighborRateRatio (see 10.2.5.7).

11.2.19.2.11 detectedFaults: a count of the number of consecutive faults (see 11.5.4 for the definition of a fault).

11.2.19.2.12 portEnabled0: a Boolean variable whose value is equal to ptpPortEnabled (see 10.2.5.13) if this state machine is invoked by the instance-specific peer-to-peer delay mechanism, and is equal to cmldsLinkPortEnabled (see 11.2.18.1) if this state machine is invoked by the CMLDS.

11.2.19.2.13 s: a variable whose value is +1 if this state machine is invoked by the instance-specific peer-to-peer delay mechanism, and -1 if this state machine is invoked by the CMLDS. The data type for s is Integer8.

11.2.19.3 State machine functions

The following functions are used in the state diagram of 11.2.19.4:

11.2.19.3.1 setPdelayReq(): creates a structure containing the parameters (see 11.4 and its subclauses) of a Pdelay_Req message to be transmitted, and returns a pointer, txPdelayReqPtr (see 11.2.19.2.6), to this structure. The parameters are set as follows:

a) sourcePortIdentity is set equal to the port identity of the port corresponding to this MD entity (see 8.5.2),
b) sequenceId is set equal to pdelayReqSequenceId (see 11.2.19.2.8),
c) correctionField is set to
    1) 0 if this state machine is invoked by the instance-specific peer-to-peer delay mechanism, and
    2) -delayAsymmetry (i.e., the negative of delayAsymmetry) if this state machine is invoked by the CMLDS, and
d) remaining parameters are set as specified in 11.4.2 and 11.4.5.

11.2.19.3.2 txPdelayReq(txPdelayReqPtr): transmits a Pdelay_Req message from the MD entity, containing the parameters in the structure pointed to by txPdelayReqPtr (see 11.2.19.2.6).

11.2.19.3.3 computePdelayRateRatio(): computes neighborRateRatio (see 10.2.5.7) using the following information conveyed by successive Pdelay_Resp and Pdelay_Resp_Follow_Up messages:

a) The pdelayRespEventIngressTimestamp (see 11.3.2.1) values for the respective Pdelay_Resp messages
b) The correctedResponderEventTimestamp values, whose data type is UScaledNs, obtained by adding the following fields of the received Pdelay_Resp_Follow_Up message:
    1) The seconds field of the responseOriginTimestamp field, multiplied by 10^9,
    2) The nanoseconds field of the responseOriginTimestamp parameter,
    3) The correctionField, divided by 2^{16}, and
    4) delayAsymmetry, if and only if this state machine is invoked by CMLDS.

NOTE—If delayAsymmetry does not change during the time interval over which neighborRateRatio is computed, it is not necessary to subtract it if this state machine is invoked by CMLDS, because in that case it will cancel when computing the difference between earlier and later correctedResponderEventTimestamps.

Any scheme that uses the preceding information, along with any other information conveyed by the successive Pdelay_Resp and Pdelay_Resp_Follow_Up messages, to compute neighborRateRatio is acceptable as long as the performance requirements specified in B.2.4 are met. If neighborRateRatio is successfully computed, the Boolean neighborRateRatioValid (see 11.2.19.2.10) is set to TRUE. If neighborRateRatio is not successfully computed (e.g., if the MD entity has not yet exchanged a sufficient number of peer delay messages with its peer), the Boolean neighborRateRatioValid is set to FALSE.

NOTE 1—As one example, neighborRateRatio can be estimated as the ratio of the elapsed time of the LocalClock entity of the time-aware system at the other end of the link attached to this port, to the elapsed time of the LocalClock entity of this time-aware system. This ratio can be computed for the time interval between a set of received Pdelay_Resp and Pdelay_Resp_Follow_Up messages and a second set of received Pdelay_Resp and Pdelay_Resp_Follow_Up messages some number of Pdelay_Req message transmission intervals later, i.e.,

$$\frac{\text{correctedResponderEventTimestamp}_N - \text{correctedResponderEventTimestamp}_0}{\text{pdelayRespEventIngressTimestamp}_N - \text{pdelayRespEventIngressTimestamp}_0}$$

where N is the number of Pdelay_Req message transmission intervals separating the first set of received Pdelay_Resp and Pdelay_Resp_Follow_Up messages and the second set, and the successive sets of received Pdelay_Resp and Pdelay_Resp_Follow_Up messages are indexed from 0 to N with the first set indexed 0.

NOTE 2—This function must account for non-receipt of Pdelay_Resp and/or Pdelay_Resp_Follow_Up for a Pdelay_Req message, and also for receipt of multiple Pdelay_Resp messages within one Pdelay_Req message transmission interval.

11.2.19.3.4 computePropTime(): computes the mean propagation delay on the link attached to this MD entity, D, and returns this value. D is given by Equation (11-5):

$$D = \frac{r \cdot (t_4 - t_1) - (t_3 - t_2)}{2} \tag{11-5}$$

where

t_4 = pdelayRespEventIngressTimestamp (see 11.3.2.1) for the Pdelay_Resp message received in response to the Pdelay_Req message sent by the MD entity, expressed in ns; the pdelayRespEventIngressTimestamp is equal to the timestamp value measured relative to the timestamp measurement plane, minus any ingressLatency (see 8.4.3)

t_1 = pdelayReqEventEgressTimestamp (see 11.3.2.1) for the Pdelay_Req message sent by the P2PPort entity, expressed in ns

t_2 = sum of (1) the ns field of the requestReceiptTimestamp, (2) the seconds field of the requestReceiptTimestamp multiplied by 10^9, and (3) the correctionField multiplied by s (see 11.2.19.2.13) and then divided by 2^{16} (i.e., the correctionField is expressed in ns plus fractional ns), of the Pdelay_Resp message received in response to the Pdelay_Req message sent by the MD entity

t_3 = sum of (1) the ns field of the responseOriginTimestamp, (2) the seconds field of the responseOriginTimestamp multiplied by 10^9, and (3) the correctionField divided by 2^{16} (i.e., the correctionField is expressed in ns plus fractional ns), of the Pdelay_Resp_Follow_Up message received in response to the Pdelay_Req message sent by the MD entity

r = current value of neighborRateRatio for this MD entity (see 10.2.5.7)

Propagation delay averaging may be performed, as described in 11.1.2 by Equation (11-2), Equation (11-3), and Equation (11-4). In this case, the successive values of propagation delay computed using Equation (11-5) are input to either Equation (11-2) or Equation (11-4), and the computed average propagation delay is returned by this function.

NOTE 1—Equation (11-5) defines *D* as the mean propagation delay relative to the time base of the time-aware system at the other end of the attached link. It is divided by neighborRateRatio (see 10.2.5.7) to convert it to the time base of the current time-aware system when subtracting from syncEventIngressTimestamp to compute upstreamTxTime [see 11.2.14.2.1 f)].

NOTE 2—The difference between mean propagation delay relative to the grandmaster time base and relative to the time base of the time-aware system at the other end of the attached link is usually negligible. To see this, note that the former can be obtained from the latter by multiplying the latter by the ratio of the grandmaster frequency to the frequency of the LocalClock entity of the time-aware system at the other end of the link attached to this port. This ratio differs from 1 by 200 ppm or less. For example, for a worst-case frequency offset of the LocalClock entity of the time-aware system at the other end of the link, relative to the grandmaster, of 200 ppm, and a measured propagation time of 100 ns, the difference in *D* relative to the two time bases is 20 ps.

NOTE 3—In IEEE Std 1588-2019, the computation of Eq. (11-5) is organized differently from the organization used in IEEE Std 802.1AS-2019. Using the definitions of t_2 and t_3 above, Eq. (11-5) can be rewritten:

$$D = [r\cdot(t_4 - t_1) - (\text{responseOriginTimestamp} - \text{requestReceiptTimestamp}) + (\text{correctionField of Pdelay\_Resp}) - (\text{correctionField of Pdelay\_Resp\_Follow\_Up})] / 2 \quad (11\text{-}6)$$

where each term is expressed in units of ns as described in the definitions of t_1, t_2, t_3, and t_4 above. In IEEE Std 1588-2019, the fractional ns portion of t_2 is subtracted from the correctionField of Pdelay_Resp, rather than added as in this standard; however, the correctionField of Pdelay_Resp is then subtracted in Eq. (11-6) rather than added, and the two minus signs cancel. The computations of *D* in IEEE Std 802.1AS-2019 and IEEE 1588-2019 are mathematically equivalent. The organization of the computation used in IEEE Std 1588 must be used in the case of CMLDS for interoperability with IEEE Std 1588 (see 11.2.17.2).

11.2.19.4 State diagram

The MDPdelayReq state machine shall implement the function specified by the state diagram in Figure 11-9, the local variables specified in 11.2.19.2, the functions specified in 11.2.19.3, the messages specified in 11.4, and the relevant global variables and functions specified in 11.2.13 and a through 10.2.6. This state machine is responsible for the following:

a) Sending Pdelay_Req messages and restarting the pdelayIntervalTimer,
b) Detecting that the peer mechanism is running,
c) Detecting if Pdelay_Resp and/or Pdelay_Resp_Follow_Up messages corresponding to a Pdelay_Req message sent are not received,
d) Detecting whether more than one Pdelay_Resp is received within one Pdelay_Req message transmission interval (see 11.5.2.2),
e) Computing propagation time on the attached link when Pdelay_Resp and Pdelay_Resp_Follow_Up messages are received, and
f) Computing the ratio of the frequency of the LocalClock entity of the time-aware system at the other end of the attached link to the frequency of the LocalClock entity of the current time-aware system.

NOTE 1—The ratio of the frequency of the LocalClock entity of the time-aware system at the other end of the attached link to the frequency of the LocalClock entity of the current time-aware system, neighborRateRatio, retains its most recent value when a Pdelay_Resp and/or Pdelay_Resp_Follow_Up message is lost.

NOTE 2—Normally, Pdelay_Resp should be received within a time interval after sending Pdelay_Req that is less than or equal to the Pdelay turnaround time plus twice the mean propagation delay. However, while receiving Pdelay_Resp after a time interval that is longer than this can result in worse time-synchronization performance (see 11.1.2 and B.2.3), the peer delay protocol will still operate. It is expected that a peer delay initiator can receive and process Pdelay_Resp and Pdelay_Resp_Follow_Up messages within a time interval after sending Pdelay_Req that is as large as the current Pdelay Req message transmission interval (see 11.5.2.2).

11.2.20 MDPdelayResp state machine

11.2.20.1 General

This state machine is invoked as part of the Common Mean Link Delay Service. There is one instance of this state machine for all the domains (per port). This means that there also is one instance of each of the state machine variables of 11.2.20.2, state machine functions of 11.2.20.3, and relevant global variables of 10.2.5 and 11.2.13 for all the domains (per port). None of the variables used or functions invoked in this state machine are specific to a single domain. However, the single instances of all of these objects or entities are accessible to all the domains.

11.2.20.2 State machine variables

The following variables are used in the state diagram of 11.2.20.4:

11.2.20.2.1 rcvdPdelayReq: a Boolean variable that notifies the current state machine when a Pdelay_Req message is received. This variable is reset by the current state machine.

11.2.20.2.2 rcvdMDTimestampReceiveMDPResp: a Boolean variable that notifies the current state machine when the pdelayRespEventEgressTimestamp (see 11.3.2.1) for a transmitted Pdelay_Resp message is received. This variable is reset by the current state machine.

11.2.20.2.3 txPdelayRespPtr: a pointer to a structure whose members contain the values of the fields of a Pdelay_Resp message to be transmitted.

11.2.20.2.4 txPdelayRespFollowUpPtr: a pointer to a structure whose members contain the values of the fields of a Pdelay_Resp_Follow_Up message to be transmitted.

11.2.20.2.5 portEnabled1: a Boolean variable whose value is equal to ptpPortEnabled (see 10.2.5.13) if this state machine is invoked by the instance-specific peer-to-peer delay mechanism, and is equal to cmldsLinkPortEnabled (see 11.2.18.1) if this state machine is invoked by the CMLDS.

11.2.20.3 State machine functions

The following functions are used in the state diagram of 11.2.20.4:

11.2.20.3.1 setPdelayResp(): creates a structure containing the parameters (see 11.4 and its subclauses) of a Pdelay_Resp message to be transmitted, and returns a pointer, txPdelayRespPtr (see 11.2.20.2.3), to this structure. The parameters are set as follows:

a) sourcePortIdentity is set equal to the port identity of the port corresponding to this MD entity (see 8.5.2),
b) sequenceId is set equal to the sequenceId field of the corresponding Pdelay_Req message,
c) requestReceiptTimestamp is set equal to the pdelayReqEventIngressTimestamp (see 11.3.2) of the corresponding Pdelay_Req message, with any fractional ns portion truncated,
d) correctionField is set equal to:
    1) the fractional ns portion of the pdelayReqEventIngressTimestamp of the corresponding Pdelay_Req message if this state machine is invoked by the instance-specific peer-to-peer delay mechanism, and
    2) minus the fractional ns portion of the pdelayReqEventIngressTimestamp of the corresponding Pdelay_Req message if this state machine is invoked by CMLDS,
e) requestingPortIdentity is set equal to the sourcePortIdentity field of the corresponding Pdelay_Req message, and
f) remaining parameters are set as specified in 11.4.2 and 11.4.6.

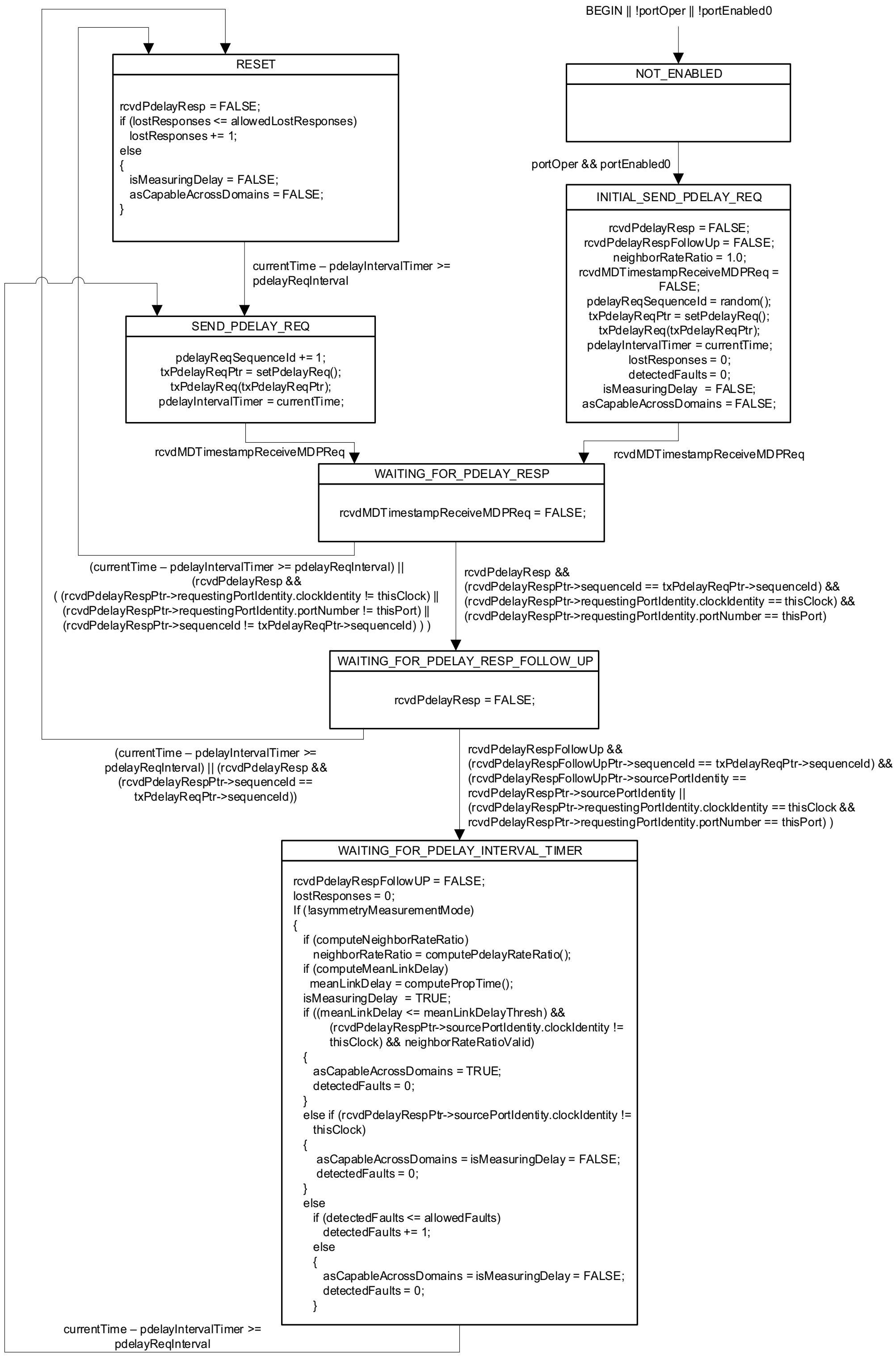


Figure 11-9—MDPdelayReq state machine

11.2.20.3.2 txPdelayResp(txPdelayRespPtr): transmits a Pdelay_Resp message from the MD entity, containing the parameters in the structure pointed to by txPdelayRespPtr (see 11.2.20.2.3).

11.2.20.3.3 setPdelayRespFollowUp(): creates a structure containing the parameters (see 11.4 and its subclauses) of a Pdelay_Resp_Follow_Up message to be transmitted, and returns a pointer, txPdelayRespFollowUpPtr (see 11.2.20.2.4), to this structure. The parameters are set as follows:

a) sourcePortIdentity is set equal to the port identity of the port corresponding to this MD entity (see 8.5.2),
b) sequenceId is set equal to the sequenceId field of the corresponding Pdelay_Req message,
c) responseOriginTimestamp is set equal to the pdelayRespEventEgressTimestamp (see 11.3.2) of the corresponding Pdelay_Resp message, with any fractional ns truncated,
d) correctionField is set equal to:
    1) the fractional ns portion of the pdelayRespEventEgressTimestamp of the corresponding Pdelay_Resp message if this state machine is invoked by the instance-specific peer-to-peer delay mechanism, and
    2) the sum of the correctionField of the corresponding Pdelay_Req message and the fractional ns portion of the pdelayRespEventEgressTimestamp of the corresponding Pdelay_Resp message if this state machine is invoked by CMLDS,
e) requestingPortIdentity is set equal to the sourcePortIdentity field of the corresponding Pdelay_Req message, and
f) remaining parameters are set as specified in 11.4.2 and 11.4.6.

11.2.20.3.4 txPdelayRespFollowUp(txPdelayRespFollowUpPtr): transmits a Pdelay_Resp_Follow_Up message from the P2PPort entity containing the parameters in the structure pointed to by txPdelayRespFollowUpPtr (see 11.2.20.2.4).

11.2.20.4 State diagram

The MDPdelayResp state machine shall implement the function specified by the state diagram in Figure 11-10, the local variables specified in 11.2.20.2, the functions specified in 11.2.20.4, the messages specified in 11.4, and the relevant global variables and functions specified in 10.2.4 through 10.2.6. This state machine is responsible for responding to Pdelay_Req messages, received from the MD entity at the other end of the attached link, with Pdelay_Resp and Pdelay_Resp_Follow_Up messages.

11.2.21 LinkDelayIntervalSetting state machine

11.2.21.1 General

This state machine is part of the Common Mean Link Delay Service. There is one instance of this state machine per port, for the Common Service of the time-aware system.

11.2.21.2 State machine variables

The following variables are used in the state diagram of 11.2.21.2:

11.2.21.2.1 rcvdSignalingMsg1: a Boolean variable that notifies the current state machine when a Signaling message that contains a Message Interval Request TLV (see 10.6.4.3) is received. This variable is reset by the current state machine.

11.2.21.2.2 rcvdSignalingPtrLDIS: a pointer to a structure whose members contain the values of the fields of the received Signaling message that contains a Message Interval Request TLV (see 10.6.4.3).

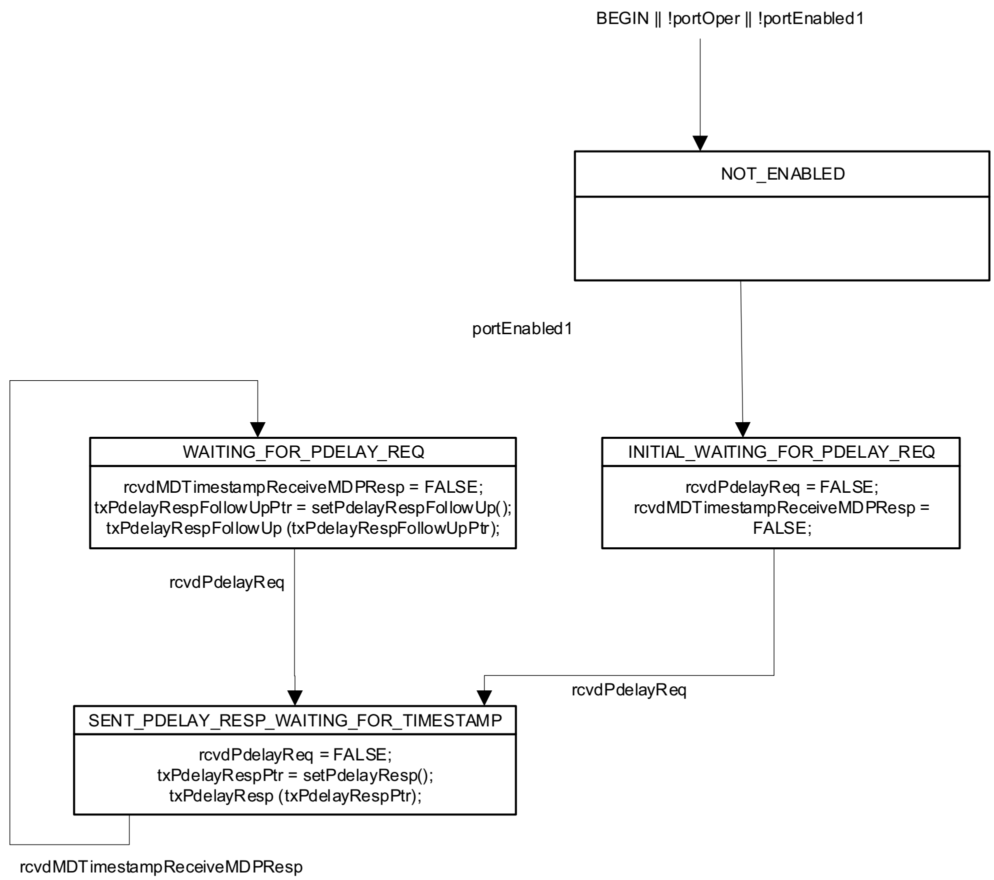


Figure 11-10—MDPdelayResp state machine

11.2.21.2.3 portEnabled3: a Boolean variable whose value is equal to ptpPortEnabled (see 10.2.5.13) if this state machine is invoked by the instance-specific peer-to-peer delay mechanism, and is equal to cmldsLinkPortEnabled (see 11.2.18.1) if this state machine is invoked by the CMLDS.

11.2.21.2.4 logSupportedPdelayReqIntervalMax: the maximum supported logarithm to base 2 of the Pdelay_Req interval. The data type for logSupportedPdelayReqIntervalMax is Integer8.

11.2.21.2.5 logSupportedClosestLongerPdelayReqInterval: the logarithm to base 2 of the Pdelay_Req interval, such that logSupportedClosestLongerPdelayReqInterval > logRequestedPdelayReqInterval, that is numerically closest to logRequestedPdelayReqInterval, where logRequestedPdelayReqInterval is the argument of the function computeLogPdelayReqInterval() (see 11.2.21.3.2). The data type for logSupportedClosestLongerPdelayReqInterval is Integer8.

11.2.21.2.6 computedLogPdelayReqInterval: a variable used to hold the result of the function computeLogPdelayReqInterval(). The data type for computedLogPdelayReqInterval is Integer8.

11.2.21.3 State machine functions

11.2.21.3.1 isSupportedLogPdelayReqInterval (logPdelayReqInterval): a Boolean function that returns TRUE if the Pdelay_Req interval given by the argument logPdelaytReqInterval is supported by the gPTP Port, and FALSE if the Pdelay_Req interval is not supported by the gPTP Port. The argument logPdelayReqInterval has the same data type and format as the field logLinkDelayInterval of the message interval request TLV (see 10.6.4.3.6).

11.2.21.3.2 computeLogPdelayReqInterval (logRequestedPdelayReqInterval): An Integer8 function that computes and returns the logPdelayReqInterval, based on the logRequestedPdelayReqInterval. This function is defined as indicated below. It is defined so that the detailed code that it invokes does not need to be placed into the state machine diagram.

```
Integer8 computeLogPdelayReqInterval (logRequestedPdelayReqInterval)
Integer8 logRequestedPdelayReqInterval;
{
	Integer8 logSupportedPdelayReqIntervalMax,
				logSupportedClosestLongerPdelayReqInterval;
	if (isSupportedLogPdelayReqInterval (logRequestedPdelayReqInterval))
		// The requested Pdelay_Req Interval is supported and returned
		return (logRequestedPdelayReqInterval)
	else
	{
		if (logRequestedPdelayReqInterval > logSupportedPdelayReqIntervalMax)
			// Return the fasted supported rate, even if faster than the requested rate
			return (logSupportedPdelayReqIntervalMax);
		else
			// Return the fastest supported rate that is still slower than
			// the requested rate.
			return (logSupportedClosestLongerPdelayReqInterval);
	}
}
```

11.2.21.4 State diagram

The LinkDelayIntervalSetting state machine shall implement the function specified by the state diagram in Figure 11-11, the local variables specified in 11.2.21.2, the messages specified in 10.6 and 11.4, the relevant global variables specified in 10.2.5 and 11.2.13, the relevant managed objects specified in 14.8 and 14.14, and the relevant timing attributes specified in 10.7 and 11.5. This state machine is responsible for setting the global variables that give the duration of the mean interval between successive Pdelay_Req messages, and also the global variables that control whether meanLinkDelay and neighborRateRatio are computed, both at initialization and in response to the receipt of a Signaling message that contains a Message Interval Request TLV (see 10.6.4.3).

NOTE—A signaling message received by this state machine, which carries the message interval request TLV (see 10.6.4.3), is ignored if multiple profiles are present and the Signaling message is directed to the CMLDS (i.e., if the value of sdoId of the Signaling message is 0x200).

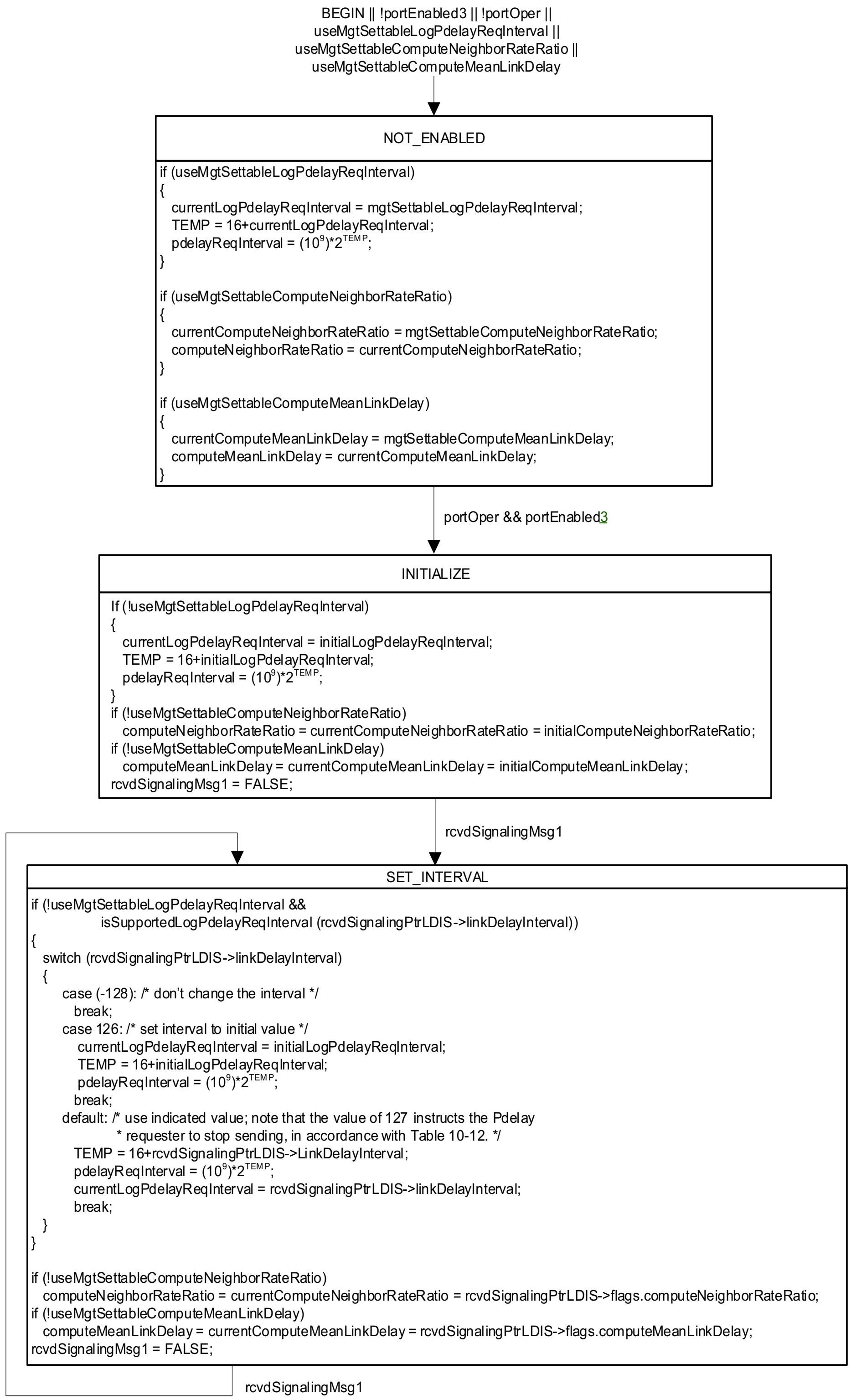


Figure 11-11—LinkDelayIntervalSetting state machine

11.3 Message attributes

11.3.1 General

This subclause describes attributes of the Sync, Follow_Up, Pdelay_Req, Pdelay_Resp, and Pdelay_Resp_Follow_Up messages that are not described in 8.4.2.

11.3.2 Message types contained in each message class

11.3.2.1 Event message class

The event message class contains the following message types:

a) Sync: A Sync message contains time-synchronization information that originates at a ClockMaster entity. The appearance of a Sync message at the reference plane of the port corresponding to an MD entity is an event to which the LocalClock assigns a timestamp, the syncEventIngressTimestamp or syncEventEgressTimestamp, based on the time of the LocalClock. The syncEventIngressTimestamp and syncEventEgressTimestamp are measured relative to the timestamp measurement plane; the MD entity corrects them for ingress and egress latencies, respectively (see 8.4.3). The Sync message is followed by a Follow_Up message containing synchronization information that is based in part on the sum of the syncEventEgressTimestamp and any egressLatency (see 8.4.3).
b) Pdelay_Req: A Pdelay_Req message is transmitted by an MD entity to another MD entity as part of the peer-to-peer delay mechanism (see 11.2.19 and Figure 11-9) to determine the delay on the link between them. The appearance of a Pdelay_Req message at the reference plane of the port of an MD entity is an event to which the LocalClock assigns a timestamp, the pdelayReqEventIngressTimestamp or pdelayReqEventEgressTimestamp, based on the time of the LocalClock. The pdelayReqEventIngressTimestamp and pdelayReqEventEgressTimestamp are measured relative to the timestamp measurement plane; the MD entity corrects them for ingress and egress latencies, respectively (see 8.4.3).
c) Pdelay_Resp: A Pdelay_Resp message is transmitted by an MD entity to another MD entity in response to the receipt of a Pdelay_Req message. The Pdelay_Resp message contains the pdelayReqEventIngressTimestamp of the Pdelay_Req message that it is transmitted in response to. The appearance of a Pdelay_Resp message at the reference plane of the port of an MD entity is an event to which the LocalClock assigns a timestamp, the pdelayRespEventIngressTimestamp or pdelayRespEventEgressTimestamp, based on the time of the LocalClock. The pdelayRespEventIngressTimestamp and pdelayRespEventEgressTimestamp are measured relative to the timestamp measurement plane; the MD entity corrects them for ingress and egress latencies, respectively (see 8.4.3). The Pdelay_Resp message is followed by a Pdelay_Resp_Follow_Up message containing the sum of the pdelayRespEventEgressTimestamp and any egressLatency (see 8.4.3).

Event messages shall be assigned the timestamps previously defined, in accordance with 8.4.3.

11.3.2.2 General message class

The general message class contains the following message types:

a) Follow_Up: A Follow_Up message communicates the value of the syncEventEgressTimestamp for the associated Sync message.
b) Pdelay_Resp_Follow_Up: A Pdelay_Resp_Follow_Up message communicates the value of the PdelayRespEventEgressTimestamp for the associated Pdelay_Resp message.

General messages are not required to be timestamped.

11.3.3 VLAN tag

A frame that carries an IEEE 802.1AS message shall not have a VLAN tag nor a priority tag (see 3.184 of IEEE Std 802.1Q-2018).

11.3.4 Addresses

The destination address of Sync, Follow_Up, Pdelay_Req, Pdelay_Resp, and Pdelay_Resp_Follow_Up messages shall be the reserved multicast address given in Table 11-3.

Table 11-3—Destination address for Sync, Follow_Up, Pdelay_Req, Pdelay_Resp, and Pdelay_Resp_Follow_Up messages

Destination address
01-80-C2-00-00-0E
NOTE—This address is taken from Table 8-1, Table 8-2, and Table 8-3 of IEEE Std 802.1Q-2018.

All Sync, Follow_Up, Pdelay_Req, Pdelay_Resp, and Pdelay_Resp_Follow_Up messages shall use the MAC address of the respective egress physical port as the source address.

11.3.5 EtherType

The EtherType of Sync, Follow_Up, Pdelay_Req, Pdelay_Resp, and Pdelay_Resp_Follow_Up messages shall be the EtherType given in Table 11-4.

Table 11-4—EtherType for Sync, Follow_Up, Pdelay_Req, Pdelay_Resp, and Pdelay_Resp_Follow_Up messages

EtherType
88-F7

11.3.6 Subtype

The subtype of the Sync, Follow_Up, Pdelay_Req, Pdelay_Resp, and Pdelay_Resp_Follow_Up messages is indicated by the majorSdoId field (see 10.6.2.2.1)

NOTE—The subtype for all PTP messages is indicated by the majorSdoId field.

11.3.7 Source port identity

TheSync, Follow_Up, Pdelay_Req, Pdelay_Resp, and Pdelay_Resp_Follow_Up messages each contain a sourcePortIdentity field (see 11.4.2.6) that identifies the egress port (see 8.5) on which the respective message is sent.

11.3.8 Sequence number

Each MD entity shall maintain a separate sequenceId pool for each of the message types Sync and Pdelay_Req, respectively.

Each Sync and Pdelay_Req message contains a sequenceId field (see 11.4.2.7), that carries the message sequence number. The sequenceId of a Sync message shall be one greater than the sequenceId of the previous Sync message sent by the transmitting port, subject to the constraints of the rollover of the

UInteger16 data type used for the sequenceId field. The sequenceId of a Pdelay_Req message shall be one greater than the sequenceId of the previous Pdelay_Req message sent by the transmitting port, subject to the constraints of the rollover of the UInteger16 data type used for the sequenceId field.

Separate pools of sequenceId are not maintained for the following message types:

a) Pdelay_Resp
b) Follow_Up
c) Pdelay_Resp_Follow_Up

For these exceptions, the sequenceId value is specified in 11.4.2.7, Table 11-7.

11.3.9 Event message timestamp point

The message timestamp point for a PTP event message shall be the beginning of the first symbol following the start of frame delimiter.

11.4 Message formats

11.4.1 General

The PTP messages Sync, Follow_Up, Pdelay_Req, Pdelay_Resp, and Pdelay_Resp_Follow_Up shall each have a header, body, and if present, a suffix that contains one or more TLVs (see 10.6.2, 11.4.3, 11.4.4, 11.4.5, 11.4.6, 11.4.7 of this standard, and Clause 14 of IEEE 1588-2019). Reserved fields shall be transmitted with all bits of the field 0 and ignored by the receiver, unless otherwise specified. The data type of the field shall be the type indicated in brackets in the title of each subclause.

Subclause 11.4.4.3 defines the Follow_Up information TLV, which is carried by the Follow_Up message if the corresponding Sync message is two-step (i.e., twoStepFlag of the Sync message, see 10.6.2.2.8, is TRUE) and by the Sync message if the message is one-step (i.e., twoStepFlag is FALSE). The Follow_Up information TLV is the first TLV of the Follow_Up message or Sync message. The requirements for the parsing and forwarding of TLVs are given in 10.6.1.

NOTE—The standard Ethernet header and FCS (18 bytes total) must be added to each message.

11.4.2 Header

The common header for the PTP messages Sync, Follow_Up, Pdelay_Req, Pdelay_Resp, and Pdelay_Resp_Follow_Up shall be as specified in 10.6.2 and its subclauses, except as noted in the following subclauses.

11.4.2.1 messageType (Enumeration4)

The value indicates the type of the message, as defined in Table 11-5.

The most significant bit of the message ID field divides this field in half between event and general messages, i.e., it is 0 for event messages and 1 for general messages.

NOTE—The reserved nibble immediately following messageType is reserved for future expansion of the messageType field.

Table 11-5—Values for messageType field

Message type	Message class	Value
Sync	Event	0x0
Pdelay_Req	Event	0x2
Pdelay_Resp	Event	0x3
Follow_Up	General	0x8
Pdelay_Resp_Follow_Up	General	0xA
NOTE—Other values for the messageType field, except for 0xB that is used for the Announce message and 0xC that is used for the Signaling message (see 10.6.2.2.2), are not used in this standard.		

11.4.2.2 messageLength (UInteger16)

The value is the total number of octets that form the PTP message. The counted octets start with and include the first octet of the header and terminate with and include the last octet of the last TLV or, if there are no TLVs, with the last octet of the message as defined in this subclause.

NOTE—See 10.6.2.2.5 for an example.

11.4.2.3 domainNumber (UInteger8)

The domainNumber for Pdelay_Req, Pdelay_Resp, and Pdelay_Resp_Follow_Up messages shall be 0. The domainNumber for all other PTP messages is as specified in 10.6.2.2.6.

11.4.2.4 flags (Octet2)

The value of the bits of the array are defined in Table 10-9.

11.4.2.5 correctionField (Integer64)

The correctionField is the value of the correction as specified in Table 11-6, measured in nanoseconds and multiplied by 2^{16}. For example, 2.5 ns is represented as 0x0000000000028000.

A value of one in all bits, except the most significant, of the field, indicates that the correction is too big to be represented.

11.4.2.6 sourcePortIdentity (PortIdentity)

The value is the portIdentity of the egress port (see 8.5.2) on which the respective message is sent.

11.4.2.7 sequenceId (UInteger16)

The value is assigned by the originator of the message in conformance with 11.3.8, except in the case of Follow_Up, Pdelay_Resp, and Pdelay_Resp_Follow_Up messages. The sequenceId field values for these exceptions are defined in the state diagrams given in the figures referenced in Table 11-7.

Table 11-6—Value of correctionField

Message type	Value
Follow_Up Sync (sent by a one-step port, see 11.1.3 and 11.2.13.9)	Corrections for fractional nanoseconds (see 10.2.9 and Figure 10-5), difference between preciseOriginTimestamp field (if sent by a two-step port) or originTimestamp field (if sent by a one-step port) and current synchronized time (see 11.2.15.2.3 and Figure 11-7), and asymmetry corrections (see 8.3, 11.2.14.2.1, and 11.2.15.2.3; the quantity delayAsymmetry is used in the computation of upstreamTxTime in 11.2.14.2.1, and upstreamTxTime is used in computing an addition to the correctionField in 11.2.15.2.3)
Pdelay_Resp, Pdelay_Resp_Follow_Up	Corrections for fractional nanoseconds (see Figure 11-9 and Figure 11-10) if the message is sent by the instance-specific peer-to-peer delay mechanism; or For Pdelay_Resp, minus the corrections for fractional nanoseconds (see 11.2.20.3.1, Figure 11-9 and Figure 11-10), and for Pdelay_Resp_Follow_Up, the sum of the correctionField of the corresponding Pdelay_Req message and the fractional ns portion of the pdelayRespEventEgressTimestamp of the corresponding Pdelay_Resp message, if this state machine is invoked by CMLDS
Sync (sent by a two-step port), Pdelay_Req, Announce, Signaling	The value is 0 (see 10.6.2.2.9) if the message is sent by the instance-specific peer-to-peer delay mechanism, or The value is 0 for Sync (sent by a two-step port), Announce, and Signaling, and -delayAsymmetry for Pdelay_Req (see 11.2.19.3.1), if the message is sent by CMLDS
NOTE—IEEE Std 1588-2019 describes asymmetry corrections for the Pdelay_Req and Pdelay_Resp messages. However, the peer-to-peer delay mechanism computes the mean propagation delay. In the case here where the gPTP communication path is a full-duplex, point-to-point link, these corrections cancel in the mean propagation delay computation and therefore are not needed.	

Table 11-7—References for sequenceId value exceptions

Message type	Reference
Follow_Up	See 11.2.15 and Figure 11-7
Pdelay_Resp	See 11.2.20 and Figure 11-10
Pdelay_Resp_Follow_Up	See 11.2.20 and Figure 11-10

11.4.2.8 logMessageInterval (Integer8)

For Sync and Follow_Up messages, the value is the value of currentLogSyncInterval (see 10.2.5.4 and 10.7.2.3). For Pdelay_Req messages, the value is the value of currentLogPdelayReqInterval. For Pdelay_Resp and Pdelay_Resp_Follow_Up messages, the value is transmitted as 0x7F and ignored on reception.

11.4.3 Sync message

11.4.3.1 General Sync message specifications

If the twoStep flag of the PTP common header (see Table 10-9) of the Sync message is TRUE, the fields of the Sync message shall be as specified in Table 11-8. If the twoStep flag of the PTP common header of the Sync message is FALSE, the fields of the Sync message shall be as specified in Table 11-9 and 11.4.3.2 and its subclauses.

Table 11-8—Sync message fields if twoStep flag is TRUE

Bits (7 6 5 4 3 2 1 0)	Octets	Offset
header (see 11.4.2)	34	0
reserved	10	34

Table 11-9—Sync message fields if twoStep flag is FALSE

Bits (7 6 5 4 3 2 1 0)	Octets	Offset
header (see 11.4.2)	34	0
originTimestamp	10	34
Follow_Up information TLV	32	44

11.4.3.2 Sync message field specifications if twoStep flag is FALSE

11.4.3.2.1 originTimestamp (Timestamp)

The value of the originTimestamp field is the sourceTime of the ClockMaster entity of the grandmaster, when the Sync message was sent by that grandmaster, with any fractional nanoseconds truncated (see 10.2.9).

The sum of the correctionField and the originTimestamp field of the Sync message is the value of the synchronized time corresponding to the syncEventEgressTimestamp at the PTP Instance that sent the Sync message, including any fractional nanoseconds.

11.4.3.2.2 Follow_Up information TLV

The Sync message carries the Follow_Up information TLV, defined in 11.4.4.3. This TLV shall be the first TLV after the fixed fields.

11.4.4 Follow_Up message

11.4.4.1 General Follow_Up message specifications

The fields of the Follow_Up message shall be as specified in Table 11-10 and 11.4.4.2 and its subclauses.

Table 11-10—Follow_Up message fields

Bits (7 6 5 4 3 2 1 0)	Octets	Offset
header (see 11.4.2)	34	0
preciseOriginTimestamp	10	34
Follow_Up information TLV	32	44

11.4.4.2 Follow_Up message field specifications

11.4.4.2.1 preciseOriginTimestamp (Timestamp)

The value of the preciseOriginTimestamp field is the sourceTime of the ClockMaster entity of the grandmaster, when the associated Sync message was sent by that grandmaster, with any fractional nanoseconds truncated (see 10.2.9).

The sum of the correctionFields in the Follow_Up and associated Sync messages, added to the preciseOriginTimestamp field of the Follow_Up message, is the value of the synchronized time corresponding to the syncEventEgressTimestamp at the PTP Instance that sent the associated Sync message, including any fractional nanoseconds.

11.4.4.2.2 Follow_Up information TLV

The Follow_Up message carries the Follow_Up information TLV, defined in 11.4.4.3. This TLV shall be the first TLV after the fixed fields.

11.4.4.3 Follow_Up information TLV definition

11.4.4.3.1 General

The fields of the Follow_Up information TLV shall be as specified in Table 11-11 and in 11.4.4.3.2 through 11.4.4.3.9. This TLV is a standard organization extension TLV for the Follow_Up message, as specified in 14.2 of IEEE Std 1588-2019.

NOTE—The Follow_Up information TLV is different from the CUMULATIVE_RATE_RATIO TLV of IEEE Std 1588-2019 (see 16.10 and Table 52 of IEEE Std 1588-2019). While both TLVs carry cumulative rate offset information, the Follow_Up information TLV also carries information on the grandmaster time base, most recent phase change, and most recent frequency change. The CUMULATIVE_RATE_RATIO TLV is not used by gPTP.

Table 11-11—Follow_Up information TLV

Bits (7 6 5 4 3 2 1 0)	Octets	Offset
tlvType	2	0
lengthField	2	2
organizationId	3	4
organizationSubType	3	7
cumulativeScaledRateOffset	4	10
gmTimeBaseIndicator	2	14
lastGmPhaseChange	12	16
scaledLastGmFreqChange	4	28

11.4.4.3.2 tlvType (Enumeration16)

The value of the tlvType field is 0x3.

NOTE—This is the value that indicates the TLV is a vendor and standard organization extension TLV, as specified in 14.2.2.1 and Table 52 of IEEE Std 1588-2019. The value is specified there as ORGANIZATION_EXTENSION, whose value is 0x3.

11.4.4.3.3 lengthField (UInteger16)

The value of the lengthField is 28.

11.4.4.3.4 organizationId (Octet3)

The value of organizationId is 00-80-C2.

11.4.4.3.5 organizationSubType (Enumeration24)

The value of organizationSubType is 1.

11.4.4.3.6 cumulativeScaledRateOffset (Integer32)

The value of cumulativeScaledRateOffset is equal to (rateRatio – 1.0) $\times$ (2^{41}), truncated to the next smaller signed integer, where rateRatio is the ratio of the frequency of the grandMaster to the frequency of the LocalClock entity in the PTP Instance that sends the message.

NOTE—The above scaling allows the representation of fractional frequency offsets in the range $[-(2^{-10} - 2^{-41}), 2^{-10} - 2^{-41}]$, with granularity of 2^{-41}. This range is approximately $[-9.766 \times 10^{-4}, 9.766 \times 10^{-4}]$.

11.4.4.3.7 gmTimeBaseIndicator (UInteger16)

The value of gmTimeBaseIndicator is the timeBaseIndicator of the ClockSource entity for the current grandmaster (see 9.2.2.3).

NOTE—The timeBaseIndicator is supplied by the ClockSource entity to the ClockMaster entity via the ClockSourceTime.invoke function (see 9.2.2.3).

11.4.4.3.8 lastGmPhaseChange (ScaledNs)

The value of lastGmPhaseChange is the time of the current grandmaster minus the time of the previous grandmaster, at the time that the current grandmaster became grandmaster. The value is copied from the lastGmPhaseChange member of the MDSyncSend structure whose receipt causes the MD entity to send the Follow_Up message (see 11.2.11).

11.4.4.3.9 scaledLastGmFreqChange (Integer32)

The value of scaledLastGmFreqChange is the fractional frequency offset of the current grandmaster relative to the previous grandmaster, at the time that the current grandmaster became grandmaster, or relative to itself prior to the last change in gmTimeBaseIndicator, multiplied by 2^{41} and truncated to the next smaller signed integer. The value is obtained by multiplying the lastGmFreqChange member of MDSyncSend whose receipt causes the MD entity to send the Follow_Up message (see 11.2.11) by 2^{41}, and truncating to the next smaller signed integer.

NOTE—The above scaling allows the representation of fractional frequency offsets in the range $[-(2^{-10} - 2^{-41}), 2^{-10} - 2^{-41}]$, with granularity of 2^{-41}. This range is approximately $[-9.766 \times 10^{-4}, 9.766 \times 10^{-4}]$.

11.4.5 Pdelay_Req message

The fields of the Pdelay_Req message shall be as specified in Table 11-12.

Table 11-12—Pdelay_Req message fields

Bits (7 6 5 4 3 2 1 0)	Octets	Offset
header (see 11.4.2)	34	0
reserved	10	34
reserved	10	44

11.4.6 Pdelay_Resp message

11.4.6.1 General Pdelay_Resp message specifications

The fields of the Pdelay_Resp message shall be as specified in Table 11-13 and 11.4.6.2 and its subclauses.

11.4.6.2 Pdelay_Resp message field specifications

11.4.6.2.1 requestReceiptTimestamp (Timestamp)

The value is the seconds and nanoseconds portion of the pdelayReqEventIngressTimestamp of the associated Pdelay_Req message (see 11.2.19).

Table 11-13—Pdelay_Resp message fields

Bits (7 6 5 4 3 2 1 0)	Octets	Offset
header (see 11.4.2)	34	0
requestReceiptTimestamp	10	34
requestingPortIdentity	10	44

11.4.6.2.2 requestingPortIdentity (PortIdentity)

The value is the value of the sourcePortIdentity field of the associated Pdelay_Req message (see 11.4.5).

11.4.7 Pdelay_Resp_Follow_Up message

11.4.7.1 General Pdelay_Resp_Follow_Up message specifications

The fields of the Pdelay_Resp_Follow_Up message shall be as specified in Table 11-14 and 11.4.7.2 and its subclauses.

Table 11-14—Pdelay_Resp_Follow_Up message fields

Bits (7 6 5 4 3 2 1 0)	Octets	Offset
header (see 11.4.2)	34	0
responseOriginTimestamp	10	34
requestingPortIdentity	10	44

11.4.7.2 Pdelay_Resp_Follow_Up message field specifications

11.4.7.2.1 responseOriginTimestamp (Timestamp)

The value is the seconds and nanoseconds portion of the pdelayRespEventEgressTimestamp of the associated Pdelay_Resp message (see 11.4.6.2).

11.4.7.2.2 requestingPortIdentity (PortIdentity)

The value is the value of the sourcePortIdentity field of the associated Pdelay_Req message (see 11.4.5).

11.5 Protocol timing characterization

11.5.1 General

This subclause specifies timing attributes for the media-dependent sublayer specified in this clause.

11.5.2 Message transmission intervals

11.5.2.1 General interval specification

The mean time interval between successive Pdelay_Req messages is represented as the logarithm to the base 2 of this time interval measured in seconds. The value of this logarithmic attribute shall be as specified in 11.5.2.2.

The mean time interval between successive Sync messages shall be as specified in 10.7.2.1, 10.7.2.3, and 11.5.2.3.

11.5.2.2 Pdelay_Req message transmission interval

When useMgtSettableLogPdelayReqInterval (see 14.16.12) is FALSE, the initialLogPdelayReqInterval specifies the following:

a) The mean time interval between successive Pdelay_Req messages sent over a link when the port is initialized, and
b) The value the mean time interval between successive Pdelay_Req messages is set to when a message interval request TLV is received with the logLinkDelayIntervalField set to 126 (see 11.2.21).

The currentLogPdelayReqInterval specifies the current value of the mean time interval between successive Pdelay_Req messages. The default value of initialLogPdelayReqInterval is 0. Every port supports the value 127; the port does not send Pdelay_Req messages when currentLogPdelayReqInterval has this value (see 11.2.21). A port may support other values, except for the reserved values indicated in Table 10-15. A port shall ignore requests for unsupported values (see 11.2.21). The initialLogPdelayReqInterval and currentLogPdelayReqInterval are per-port attributes.

When useMgtSettableLogPdelayReqInterval is TRUE, currentLogSyncInterval is set equal to mgtSettableLogPdelayReqInterval (see 14.16.13) and initialLogPdelayReqInterval is ignored.

NOTE 1—If useMgtSettableLogPdelayReqInterval is FALSE, the value of initialLogPdelayReqInterval is the value of the mean time interval between successive Pdelay_Req messages when the port is initialized. The value of the mean time interval between successive Pdelay_Req messages can be changed, e.g., if the port receives a Signaling message that carries a message interval request TLV (see 10.6.4.3) and the current value is stored in currentLogPdelayReqInterval. The value of the mean time interval between successive Pdelay_Req messages can be reset to the initial value, e.g., by a message interval request TLV for which the value of the field logLinkDelayInterval is 126 (see 10.6.4.3.6).

NOTE 2—A port that requests (using a Signaling message that contains a message interval request TLV, see 10.6.4 and 11.2.21) that the port at the other end of the attached link set its currentLogPdelayReqInterval to a specific value can determine if the request was honored by examining the logMessageInterval field of subsequent received Pdelay_Req messages.

NOTE 3—The MDPdelayReq state machine ensures that the times between transmission of successive Pdelay_Req messages, expressed in seconds, are not smaller than $2^{currentLogPdelayReqInterval}$. This is consistent with IEEE Std 1588, which requires that the logarithm to the base 2 of the mean value of the interval in seconds between Pdelay_Req message transmissions is no smaller than the interval computed from the value of the portDS.logMinPdelayReqInterval member of the data set of the transmitting PTP Instance. The sending of Pdelay_Req messages is governed by the LocalClock, and not the synchronized time (i.e., the estimate of the grandmaster time). Since the LocalClock frequency can be slightly larger than the grandmaster frequency (e.g., by 100 ppm, which is the specified frequency accuracy of the LocalClock, see Annex B.1.1), it is possible for the time intervals between successive Pdelay_Req messages to be slightly less than $2^{currentLogPdelayReqInterval}$ when measured relative to the synchronized time.

11.5.2.3 Sync message transmission interval default value

The default value of initialLogSyncInterval (see 10.7.2.3) is –3. Every port supports the value 127; the port does not send Sync messages when currentLogSyncInterval has this value (see 10.3.18). A port may support other values, except for the reserved values indicated in Table 10-16. A port ignores requests for unsupported values (see 10.3.18).

NOTE—A port that requests (using a Signaling message that contains a message interval request TLV, see 10.6.4 and 10.3.18) that the port at the other end of the attached link set its currentLogSyncInterval to a specific value can determine if the request was honored by examining the logMessageInterval field of subsequent received Sync messages.

11.5.3 allowedLostResponses

The variable allowedLostResopnse (see 11.2.13.4) is the number of Pdelay_Req messages for which a valid response is not received, above which a port is considered to not be exchanging peer delay messages with its neighbor. The default value of allowedLostResponses shall be 9. The range shall be 1 - 255.

11.5.4 allowedFaults

The variable allowedFaults (see 11.2.13.5) is the number of faults, i.e., the number of

a) instances where the computed mean propagation delay, i.e., meanLinkDelay (see 10.2.5.8), exceeds the threshold, meanLinkDelayThresh (see 11.2.13.7), and/or
b) instances where the computation of neighborRateRatio is invalid (see 11.2.19.2.10)

above which asCapableAcrossDomains is set to FALSE, i.e., the port will be considered to not be capable of interoperating with its neighbor via the IEEE 802.1AS protocol (see 10.2.5.1). The default value of allowedFaults shall be 9. The range shall be 1 - 255.

NOTE—The above description of allowedFaults uses the variable asCapableAcrossDomains (see 11.2.13.12). In the case where only one domain is active, asCapableAcrossDomains is equivalent to the variable asCapable.

11.6 Control of computation of neighborRateRatio

The variable computeNeighborRateRatio (see 10.2.5.10) indicates whether neighborRateRatio is to be computed by this port when the peer-to-peer delay mechanism is invoked.

When useMgtSettableComputeNeighborRateRatio (see 14.16.16) is FALSE, computeNeighborRateRatio is initialized to the value of initialComputeNeighborRateRatio.

The currentComputeNeighborRateRatio specifies the current value of computeNeighborRateRatio. The default value of initialComputeNeighborRateRatio is TRUE. The initialComputeNeighborRateRatio and currentComputeNeighborRateRatio are per-port attributes.

When useMgtSettableComputeNeighborRateRatio is TRUE, currentComputeNeighborRateRatio is set equal to mgtSettableComputeNeighborRateRatio (see 14.16.17) and initialComputeNeighborRateRatio is ignored.

NOTE 1—If useMgtSettableComputeNeighborRateRatio is FALSE, the value of initialComputeNeighborRateRatio determines whether neighborRateRatio is computed by the peer delay mechanism when the port is initialized. The value of computeNeighborRateRatio can be changed, e.g., if the port receives a Signaling message that carries a message interval request TLV (see 10.6.4.3) and the current value is stored in currentComputeNeighborRateRatio.

11.7 Control of computation of meanLinkDelay

The variable computeMeanLinkDelay (see 10.2.5.10) indicates whether meanLinkDelay is to be computed by this port when the peer-to-peer delay mechanism is invoked.

When useMgtSettableComputeMeanLinkDelay (see 14.16.20) is FALSE, computeMeanLinkDelay is initialized to the value of initialComputeMeanLinkDelay.

The currentComputeMeanLinkDelay specifies the current value of computeMeanLinkDelay. The default value of initialComputeMeanLinkDelay is TRUE. The initialComputeMeanLinkDelay and currentComputeMeanLinkDelay are per-port attributes.

When useMgtSettableComputeMeanLinkDelay is TRUE, currentComputeMeanLinkDelay is set equal to mgtSettableComputeMeanLinkDelay (see 14.16.21) and initialComputeMeanLinkDelay is ignored.

NOTE 1—If useMgtSettableComputeMeanLinkDelay is FALSE, the value of initialComputeMeanLinkDelay determines whether meanLinkDelay is computed by the peer delay mechanism when the port is initialized. The value of computeMeanLinkDelay can be changed, e.g., if the port receives a Signaling message that carries a message interval request TLV (see 10.6.4.3) and the current value is stored in currentComputeMeanLinkDelay.

12. Media-dependent layer specification for IEEE 802.11 links

12.1 Overview

Accurate synchronized time is distributed across a domain through time measurements between adjacent PTP Instances in a packet network. Time is communicated from the root of the clock spanning tree (i.e., the grandmaster) to the leaves of the tree, by recursively propagating time from a leaf-facing “master” port to some number of root-facing “slave” ports in devices at the next level of the tree through measurements made across the links connecting the devices. While the time semantics are consistent across the time-aware packet network, the method for communicating synchronized time from a master port to the immediate downstream link partner varies depending on the type of link interconnecting the two systems.

This clause specifies the interface primitives and state machines that provide accurate synchronized time across wireless IEEE 802.11 links as part of a packet network. This clause builds upon time measurement features defined in IEEE Std 802.11, and makes no distinction between stations with an Access Point function and stations without an Access Point function.

12.1.1 IEEE802.11 timing measurement procedure and fine timing measurement procedure

IEEE Std 802.11 defines a family of wireless measurements, including both “timing measurement” (TM) and “fine timing measurement” (FTM) which captures timestamps of the transmit time and receive time of a round-trip message exchange between associated WLAN stations.

In contrast to the protocol defined for full-duplex point-to-point links, this clause does not define any new frames nor the transmission of any frames. Rather, it makes use of a MAC Layer Management Entity (MLME) interface, which causes the IEEE 802.11 layer to not only take timestamps of measurement frames as they are transmitted and received, but to also *generate* and *consume* the measurement frames, all within the IEEE 802.11 MLME layer, and then to provide timestamp information from the MLME to this media-dependent layer through a set of well-defined service primitives. However, as an aid to the reader, the protocol and frames used by the IEEE 802.11 MLME for both timing measurement and fine timing measurement are described briefly as follows and illustrated in Figure 12-1 and Figure 12-2, respectively.

Both timing measurementand fine timing measurement are accomplished through a round-trip frame exchange. For timing measurement, the first frame of the round-trip measurement is generated by the master within the IEEE 802.11 MLME when the MLME-TIMINGMSMT.request primitive is invoked. For fine timing measurement, an initial fine timing measurement request frame is generated by the slave within the IEEE 802.11 MLME when the MLME-FINETIMINGMSMTRQ.request primitive is invoked. After this frame is successfully received by the master, the first frame of the round-trip measurement is generated by the master within the IEEE 802.11 MLME when the MLME-FINETIMINGMSMT.request primitive is invoked. As defined by IEEE Std 802.11, upon receipt of the resulting timing measurement or fine timing measurement frame frame, the slave station transmits an IEEE 802.11 Ack control frame to the master station. Four timestamps are captured during this two-frame exchange, as follows:

a) *t1* is when (in the master station’s time base) the request frame is transmitted
b) *t2* is when (in the slave station’s time base) the request frame is received
c) *t3* is when (in the slave station’s time base) the Ack control frame is transmitted
d) *t4* is when (in the master station’s time base) the Ack control frame is received

When the master sends either a fine timing measurement or a timing measurement frame, it passes the *t1* and *t4* timestamps (and other end-to-end synchronization information) and FollowUpInformation, from the previous measurement to the slave. A pair of tokens is passed in each timing or fine timing measurement frame, one to identify the current measurement and the other to allow the slave to associate the timestamp information with the previous measurement.

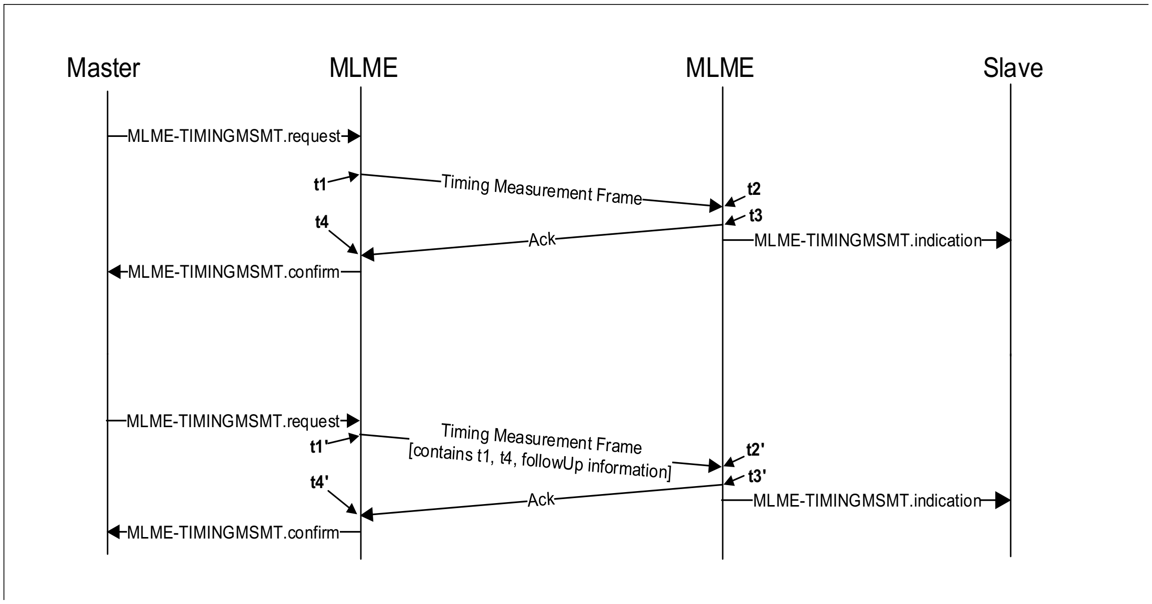


Figure 12-1—Timing measurement procedure for IEEE 802.11 links

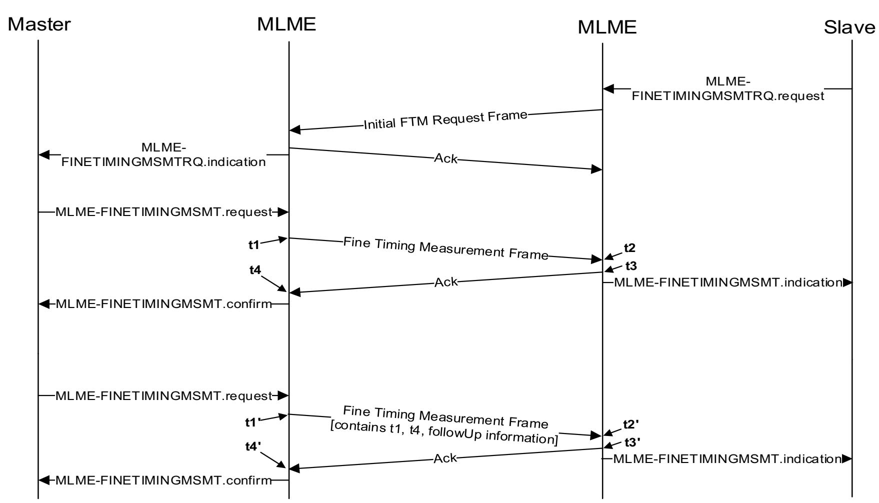


Figure 12-2—Fine timing measurement procedure for IEEE 802.11 links

Note that, unlike point-to-point full-duplex ports, IEEE 802.11 ports do not compute the link delay measurements in both directions since only a port in the Slave state makes use of that information. As a result, a port that transitions from the Master state to the Slave state (e.g., due to selection of a new grandmaster), might collect a number of link delay measurements and perform averaging or other filtering before achieving the desired accuracy.

NOTE 1—Fine timing measurement can be used for time synchronization as described in this standard; however, it can also be used for location services as defined in IEEE Std 802.11. Since FTM supports only one configuration at a time it is possible that setting that single configuration for time synchronization might disable its previous configuration for use by another application for location services or, conversely, an application that configures FTM for location services could disable this standard's use for time synchronization. Implementations that use FTM for two or more purposes might need to be aware of this behavior, and could choose to implement a shim layer that tracks these multiple uses and coordinates that shared use such that no single use disables the other uses.

In timing measurement as used in this standard, the master generates MLME-TIMINGMSMT. request primitives in a manner such that the requirements of 10.7.2.3 and 12.8 for the time synchronization message interval are satisfied. This results in timing measurement frames being sent from the master to the slave continuously, at a rate that satisfies the requirements of these two subclauses. It is not necessary for the slave to continually request timing information from the master. In contrast, in fine timing measurement the slave must request timing information from the master. Since, as explained above, a fine timing measurement frame carries timestamp information for a previous measurement, the slave actually requests a burst of fine timing measurement frames from the master. Actually, the example of Figure 12-2 shows a burst of fine timing measurement frames (there are two frames in that example); however, that aspect was not emphasized in the discussion above. The fine timing measurement process is described in more detail as follows, and is illustrated in Figure 12-3. In this discussion, the focus is on the transmission of the frames. To simplify the figure, the service primitives are omitted from Figure 12-3.

Figure 12-3 is adapted from Figure 11-37 of IEEE Std 802.11-2016. Additional details on the FTM procedure are given in 11.24.6 of IEEE Std 802.11-2016. The example of this figure is for the case where the initiating station (STA), i.e., the slave requests a single burst of three FTM frames from the responding STA, i.e., the master, as soon as possible. The slave makes this request by sending an initial FTM Request to the master with respective parameters set to appropriate values. The FTM parameters that are relevant to time synchronization in this standard are described in 12.6, and all the FTM parameters are described in more detail in 9.4.2.168 of IEEE Std 802.11-2016. However, in the example here, the parameter ASAP is set to 1 to indicate to the master that the FTM frames are desired as soon as possible, and the Number of Bursts Exponent parameter is set to 0 to indicate a single burst. As mentioned above, Figure 12-3 is a simplified view; the slave causes the frame to be sent by invoking the MLME-FINETIMINGMSMTRQ.request primitive, which includes the FTM parameter values. The master sends an acknowledgment (Ack) frame to the slave to indicate it received the initial FTM request. The master then sends an initial FTM frame at a time that is recommended to be no more than 10 ms later than the receipt of the initial FTM request. The initial FTM frame indicates to the slave whether the master was able to grant the values of the FTM parameters that the slave requested. If the requested parameters are granted, the procedure continues (Figure 12-3 illustrates this case). If the requested parameters are not granted, the slave sends a new initial FTM request for a burst of two FTM frames. If the new request is granted, the procedure continues. If the new request is not granted, the slave and master use TM if they both support TM. If, at this point, either the slave, the master, or both do not support TM, the procedure terminates and asCapable is set to FALSE (see 12.4).

NOTE 2—IEEE Std 802.11-2016 allows various options in the case the master does not grant the request. The above procedure is used here.

The initial FTM frame (initial FTM_1 in Figure 12-3) sent by the master is timestamped with the value t1_1 on transmission from the master, and timestamped with the value t2_1 on receipt by the slave. The initial FTM frame has fields that carry the t1 and t4 timestamps of the previous FTM frame and corresponding Ack; however, since this is the first FTM frame of the burst, these fields are set to zero. The slave responds to the master with an Ack, which is timestamped with the value t3_1 on transmission from the slave and with the value t4_1 on receipt by the master.

The master sends the second FTM frame (FTM_2 in Figure 12-3) after a time interval since sending the initial FTM frame that is greater than or equal to the Min Delta FTM parameter that was requested by the slave. This FTM frame is timestamped with t1_2 on transmission and t2_2 on reception. This FTM frame also carries the values of the timestamps t1_1 and t4_1 of the initial FTM frame and corresponding Ack. On receipt of the second FTM frame, the slave sends an Ack frame to the master; this frame is timestamped with t3_2 on transmission and t4_2 on reception. Finally, the master sends the third FTM frame (FTM_3 in Figure 12-3) to the slave, also at a time interval since sending FTM_2 that is greater than or equal to the Min Delta FTM parameter that was requested by the slave. As with FTM_2, this frame is timestamped with t1_3 on transmission and t2_3 on reception. This FTM frame also carries the values of the timestamps t1_2 and t4_2 of the second FTM frame and corresponding Ack. On receipt of the third FTM frame, the slave sends an Ack frame to the master; this frame is timestamped with t3_3 on transmission and t4_3 on reception.

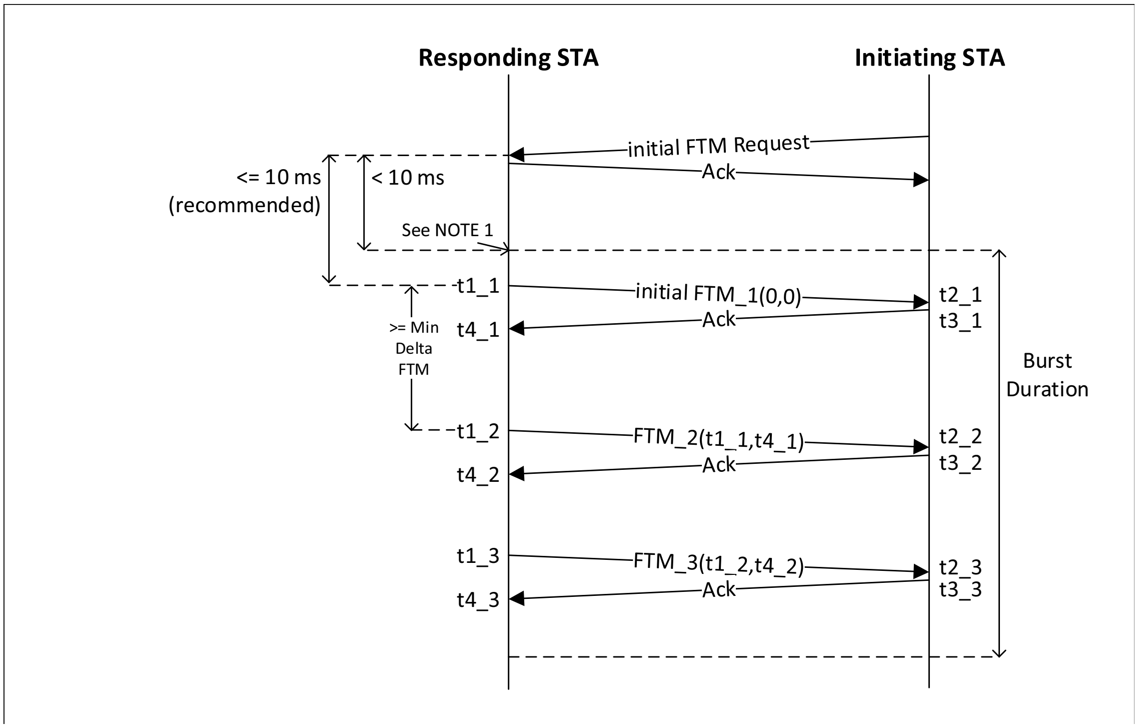


Figure 12-3—Illustration of fine timing measurement burst

On completion of the above exchanges of FTM frames and corresponding acknowledgments, the slave knows the transmission and reception times for the initial FTM frame (t1_1, t2_1, t3_1, and t4_1) and second FTM frame (t1_2, t2_2, t3_2, and t4_2), and reception time for the third FTM frame (t2_3) and transmission time for the corresponding Ack (t3_3). The slave can use this information (along with FollowUpInformation contained in the VendorSpecific information element, see 12.7) to synchronize to the master. In this standard, timestamps for the minimum delay FTM frames are used. Specifically, the slave computes the quantities D1 = t2_1 - t1_1, and D2 = t2_2 - t1_2, and uses the timestamps t1_*i* and t2_*i*, where *i* = 1 if D1 < D2 and *i* = 2 if D1 ≥ D2, to compute the respective members of the MDSyncReceive structure . The timestamps of FTM_3 and its corresponding Ack are not used; FTM_3 is used only to convey the timestamps of FTM_2 and its corresponding Ack.

If the master does not grant the parameters requested initially by the slave, i.e., for a burst of three FTM frames, but it does grant the subsequent request for a burst of two FTM frames, the slave has a full set of timestamps for only the initial FTM_1 frame. In this case, the slave uses the timestamps t1_1, t2_1, t3_1, and t4_1 to compute the respective members of the MDSyncReceive structure.

With the above procedure for FTM, the slave controls the rate at which time synchronization information is sent from the master. This is different from TM, full-duplex IEEE 802.3, IEEE 802.3 EPON, and CSN transports. In those cases, the sending of time synchronization information from the master to the slave is controlled by the master; this is true for syncLocked (see 10.2.5.15) TRUE, in which case the information is sent as soon as it is received from further upstream, and syncLocked FALSE, in which case it is sent independently of information received from further upstream. In the case of FTM, the slave requests time synchronization information from the master at an average rate equal to the inverse of the current synchronization message interval currentLogSyncInterval (see 12.8 and 14.8.18). In addition, the actual intervals between successive requests by the slave for time synchronization information meet the requirements of 10.7.2.3. This also means that the value of syncLocked at the master port will not affect the sending of time synchronization information from the master to the slave; the requests for time synchronization information from the slave are asynchronous to the receipt of time synchronization information from upstream at the node that contains the master port.

12.1.2 Layering for IEEE 802.11 links

The *media-dependent* (MD) entity is tailored to the link technology and is responsible for translating the PortSync entity's media-independent actions to media-dependent PDUs or primitives as necessary for communicating synchronized time from the master port over the link to a single slave port. In the case of an IEEE 802.11 link, this one-to-one relationship between the MD entities of the master and slave implies that if the one physical IEEE 802.11 port is associated with multiple stations, each association requires its own instantiation of the IEEE 802.1AS PortSync entity and MD entity. The MLME-TIMINGMSMT and MLME-FINETIMINGMSMT service primitives defined in IEEE Std 802.11-2016, are used to perform timing measurements and fine timing measurements, respectively, between a master IEEE 802.11 station and associated IEEE 802.11 slave stations. Figure 12-4 illustrates how the MD entity interacts with the higher and lower layers.

12.2 Messages

All media-dependent frames are generated and consumed by the lower-layer IEEE 802.11 MLME and thus none are defined here. Also, since the IEEE 802.11 event messages are timestamped by the MAC/PHY, the timestamp point is defined in IEEE Std 802.11 as well. Media-independent messages, i.e., Announce and Signaling messages, are transmitted using the unicast address of the WLAN station instead of the group address defined in 10.5.3.

12.3 Determination of timing measurement and fine timing measurement capability

The bits of the per-port global variable tmFtmSupport (see 12.5.1.5) shall be set as indicated in Table 12-1.

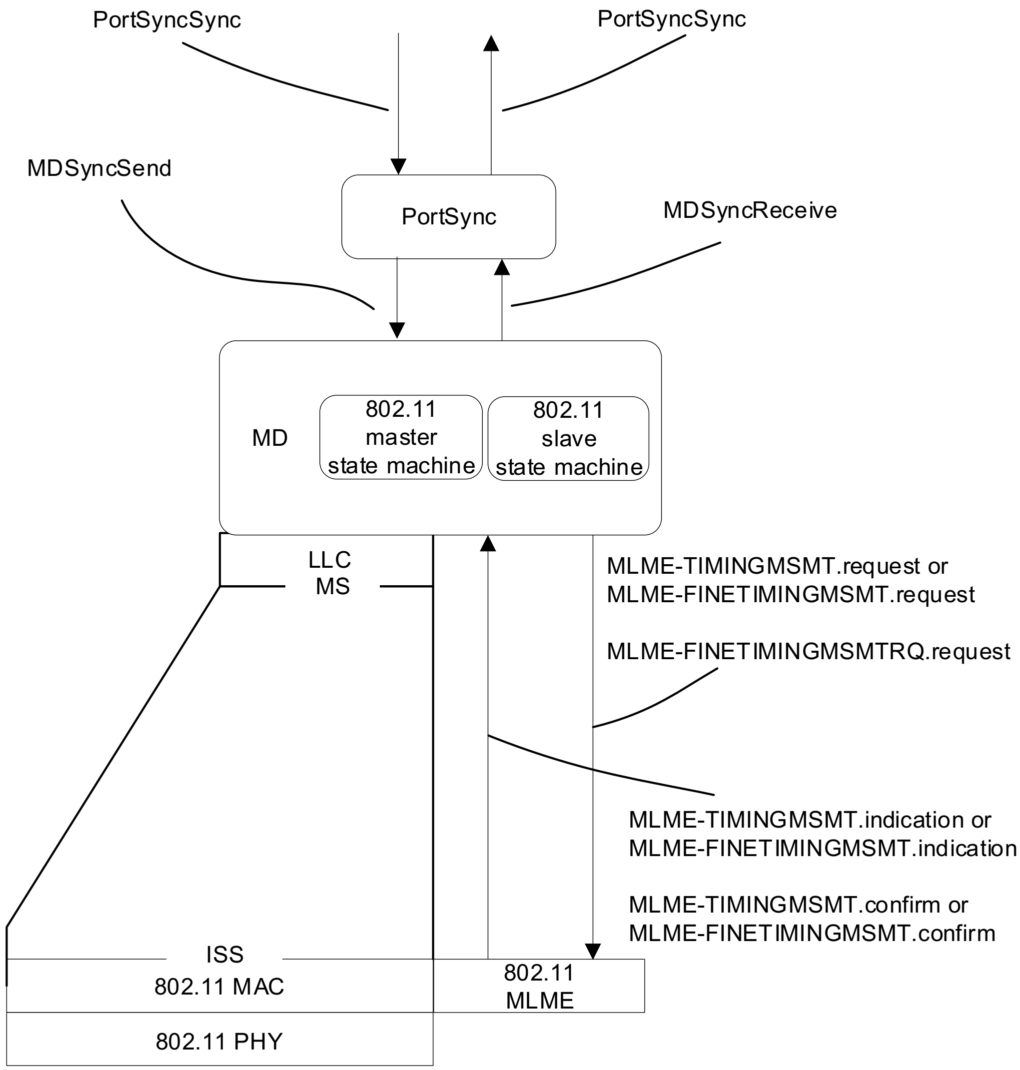


Figure 12-4—Media-dependent and lower entities in stations with IEEE 802.11 links

Table 12-1 — Values of bits of tmFtmSupport

Bit	Value
0	TRUE if (a) the port supports timing measurement, and (b) the timing measurement bit in the Extended Capabilities information element defined in Table 9-135 of IEEE Std 802.11-2016 indicates that the peer 802.11 station is capable of participating in the timing measurement protocol FALSE otherwise
1	TRUE if (a) the port supports fine timing measurement, and (b) the fine timing measurement responder and initiator bits in the Extended Capabilities information element defined in Table 9-135 of 802.11-2016 indicate that the peer 802.11 station is capable of participating in the fine timing measurement protocol FALSE otherwise
2 - 7	Reserved as FALSE

12.4 Determination of asCapable

The per-port, per-domain instance of the global variable asCapable (see 10.2.5.1) is set to TRUE if the following conditions hold (see 12.5.1 and 12.5.2):

a) The value of tmFtmSupport is not zero,
b) neighborGptpCapable is TRUE ~~and the value of domainNumber is not zero~~, and
c) at least one of the following conditions hold:
   1) bit 0 of tmFtmSupport is TRUE,
   2) bit 1 of tmFtmSupport is TRUE and, if the port is a master port, it can support (i.e., grant) the parameters requested by the slave with either FTMs per burst equal to 3 or FTMs per burst equal to 2
   3) bit 1 of tmFtmSupport is TRUE and, if the port is a slave port, the master port at the other end of the link can support (i.e., grant) the parameters requested by the slave with either FTMs per burst equal to 3 or FTMs per burst equal to 2.

If the value of domainNumber is zero (which is required for support of the 2011 edition of this standard) and bit 0 of tmFtmSupport is TRUE, asCapable can be set to TRUE. In all other instances, asCapable shall be set to FALSE.

NOTE—The above conditions ensure backward compatibility with the 2011 edition of this standard. A time-aware system that is compliant with the 2011 edition of this standard will not process the gPTP capable TLV, and asCapable will be determined as specified in the 2011 edition. A PTP Instance of a time-aware system compliant with the current edition of this standard that is attached, via an 802.11 link, to a node compliant with the 2011 edition of this standard will not receive Signaling messages that contain the gPTP capable TLV and will not set neighborGptpCapable to TRUE; however, the first sentence immediately after (b) ensures that asCapable for this port and domain (i.e., domain 0) will still be set in a manner consistent with that of the 2011 edition of this standard.

12.5 State machines

12.5.1 Media-dependent master state machines

12.5.1.1 Overview

The MD entity of an IEEE 802.11 port whose port state is MasterPort (see Table 10-2) shall behave in a manner that is indistinguishable, relative to an observer external to a system, from a strict implementation of the master state machines in Figure 12-5 and Figure 12-6 (referred to as master state machine A and master state machine B, respectively), the local variables specified in , the functions specified in 12.5.1.4, the shared variables specified in 12.5.1.5, and the primitives defined in 12.5.1.6.

In the case of timing measurement, master state machine A is responsible for initiating a time measurement whenever the PortSync entity requests it do so, as indicated by the rcvdMDSyncDot11MasterA Boolean (see 12.5.1.3 (h)). Master state machine A invokes the IEEE 802.11 MLME-TIMINGMSMT.request primitive and waits for the subsequent MLME-TIMINGMSMT.confirm primitive. It collects local timestamp information from the measurement (*t1* and *t4*, provided by the confirm primitive) and includes the information in the subsequent request. See 8.4.3 for more information on timestamps. Master state machine B is not used for timing measurement.

For fine timing measurement, master state machine A receives and stores information from the PortSync entity. Master state machine B receives the MLME-FINETIMINGMSMTRQ.indication caused by the initial FTM request from the slave. It sets asCapable as specified in 12.4. It then generates successive MLME-FINETIMINGMSMT.request primitives to indicate to the slave whether it can grant the requested parameters, and also to cause information saved by master state machine A to be sent to the slave. It receives MLME-FINETIMINGMSMT.confirm primitives caused by ACKs received from the slave. It collects local timestamp information from the current measurement (*t1* and *t4,* provided by the confirm primitive) and includes the information in the subsequent MLME-FINETIMINGMSMT.request.

12.5.1.2 State diagrams

NOTE—In the computation of the burstDuration in master state machine B, the burst duration parameter from 802.11-2016 is converted to UScaledNs (i.e., units of 2^{-16} ns, see 6.3.3.2 of IEEE Std 802.11-2016). The quantity initReqParamsDot11MasterB.burstDuration–2 is the logarithm to base 2 of the burst duration in µs. Also, it is assumed that the burst duration starts when the initial FTM request is received. In actuality, the timer begins by the partial TSF timer value indicated in the initial FTM frame, which is slightly after the initial FTM request is received.

12.5.1.3 State machine local variables

a) **dialogToken:** an unsigned 8-bit integer used to identify a measurement from among those preceding and following it.
b) **followUpInfoValid:** a Boolean variable indicating whether the FollowUpInformation (e.g., timestamps and rateRatio, see 12.7) and the link partner are unchanged since the last timing measurement or fine timing measurement.
c) **requestParams:** a structure whose members contain the values of the fields of either the MLME-TIMINGMSMT.requestor MLME-FINETIMINGMSMT.request primitive.
d) **paramsFromConfirm:** a structure whose members contain the values of the fields of either the MLME-TIMINGMSMT.confirmor MLME-FINETIMINGMSMT.confirm primitive.
e) **dot11SlaveMac:** the MAC address of the station associated with the current port.
f) **slaveMacOfLastRequest:** the MAC address of the station of the previous request, used to validate FollowUpInformation.
g) **residenceTime:** a temporary variable that holds the computation of the time between receipt of the last synchronization information and transmission of synchronization information.
h) **rcvdMDSyncDot11MasterA:** a Boolean variable that is set to TRUE when an MDSyncSend structure is provided by the PortSync entity.
i) **rcvdConfirm:** a Boolean variable that is set to TRUE when either the MLME-TIMINGMSMT.confirm or MLME-FINETIMINGMSMT.confirm primitive is received.
j) **nframesSent:** an unsigned 8-bit integer used to count the number of frames sent by the slave (the number of indications received is counted).
k) **rcvdInitIndication:** a Boolean variable that is set to TRUE when the initial MLME-FINETIMINGMSMTRQ.indication primitive is received.
l) **initReqParamsDot11MasterB:** a structure whose members contain the values of the fields of an MLME-FINETIMINGMSMTRQ.indication primitive.

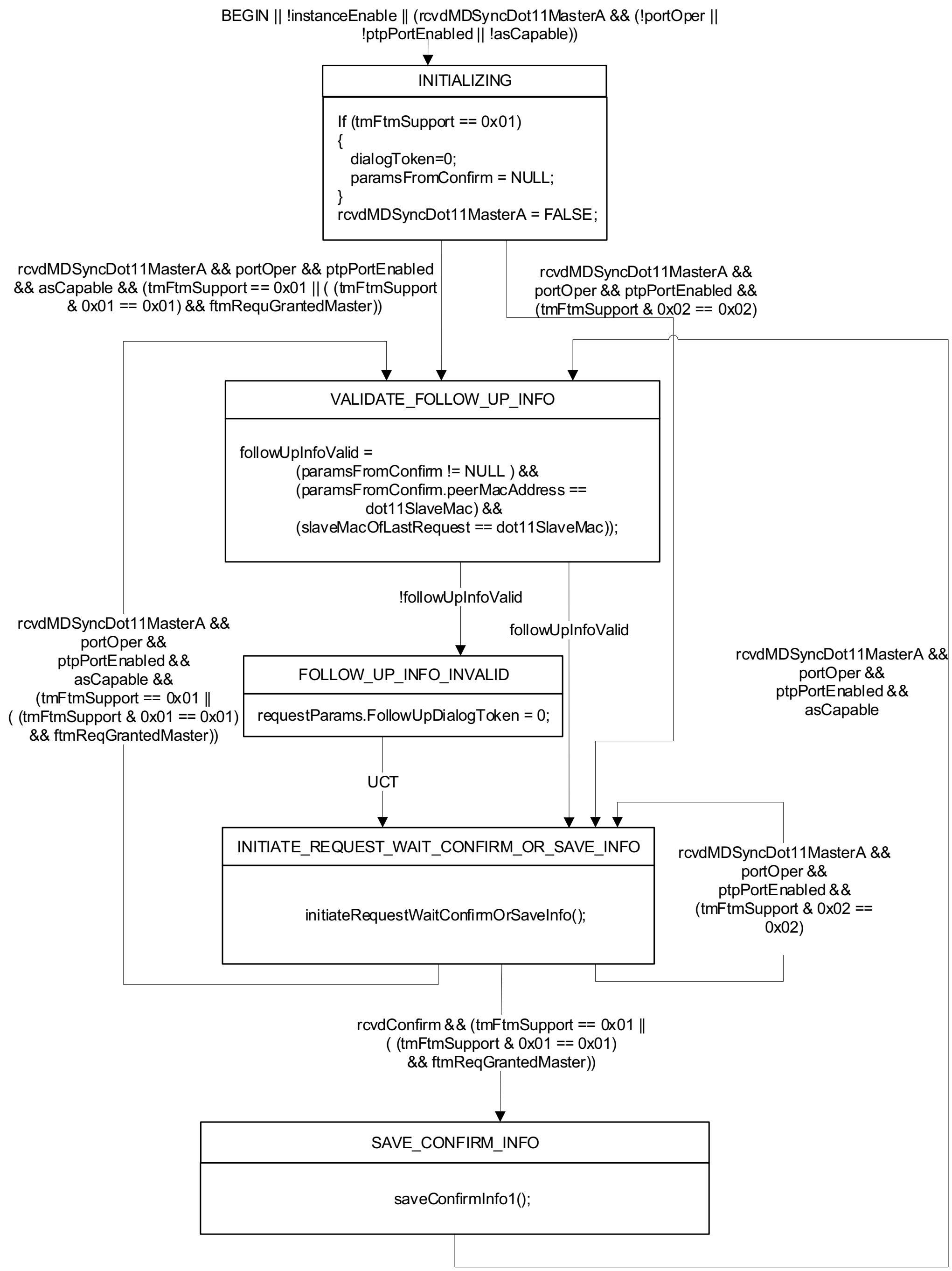


Figure 12-5—Master state machine A. (a) For TM, receives information from the PortSync entity and sends to slave, and (b) for FTM, receives and stores information from the Port-Sync entity.

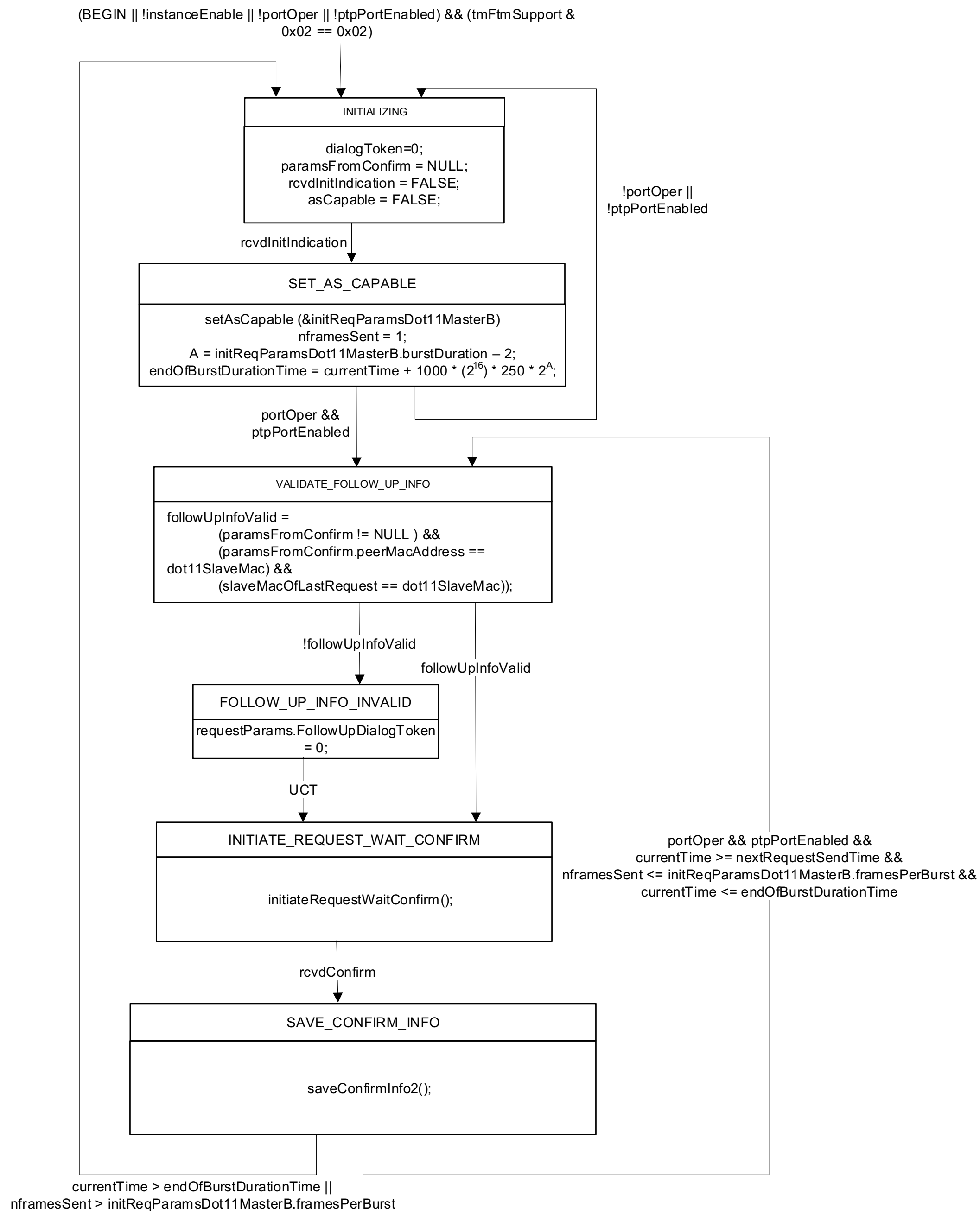


Figure 12-6—Master state machine B. (a) For TM, not invoked, and (b) for FTM, receives initial FTM request from slave and sends information received from upstream to slave in successive FTM frames

m) **endOfBurstDurationTime:** a UScaledNs variable whose value is the time at which the current burst ends.

n) **nextRequestSendTime:** a UScaledNs variable whose value is the expected time that the next MLME-FINETIMINGMSMTRQ.indication primitive, for the next request from the slave, will be received.

12.5.1.4 State machine functions

a) **setRequestParams(&requestParams, MDSyncSend):** assigns values to the parameters of the request primitive of either MLME-TIMINGMSMT or MLME-FINETIMINGMSMT (see 12.5.1.6) as follows:
   4) Members of the FollowUpInformation member of the VendorSpecific information element, as defined in 12.7, are assigned as defined in 11.4.4 with the exception of the correctionField, which is assigned as shown in the Master state machine in 12.5.1.2.
   5) The other fields of the VendorSpecific information element are assigned as follows:
      i) ElementID is assigned the value 221 as defined in Table 8-54 (Element IDs) of IEEE Std 802.11-2012, indicating that the information element is of type Vendor Specific.
      ii) The Length field is set to 80.

NOTE—This is equal to the length of the Follow_Up payload defined in 11.4.4 (including the common header) plus the length of the OUI or CID field and the Type field (see Figure 12-8).

      iii) The OUI or CID field is set to 00-80-C2.
      iv) The Type field is set to 0.
   6) Max t1 Error, Maxt4 Error are set to zero.
   7) For Fine Timing Measurement frames, the location configuration information (LCI) Report, Location Civic Report are not present. All other members are left unchanged.

b) **setAsCapable (&initReqParamsDot11MasterB):** determines the value of asCapable consistent with 12.4 and whether the master is able to grant the parameters requested by the slave. This function is used only in the case of FTM.

c) **initiateRequestWaitConfirmOrSaveInfo():** This function is defined as indicated below. It is used in Master state machine A. It is defined so that the detailed code that it invokes does not need to be placed into the state machine diagram.

```
initiateRequestWaitConfirmOrSaveInfo()
{
        rcvdMDSyncDot11MasterA = FALSE;

        If (tmFtmSupport == 0x01)
        {
                if ((++dialogToken % 256) == 0) dialogToken++;
                requestParams.DialogToken=dialogToken;
                requestParams.PeerMACAddress = dot11SlaveMac;
                setRequestParams(&requestParams, MDSyncSend);
                MLME-TIMINGMSMT.request(requestParams);
                requestParams.FollowUpDialogToken = 0;
                                        //In case no confirm is received
                slaveMacOfLastRequest = dot11SlaveMac;
        }
}
```

d) **saveConfirmInfo1():** This function is defined as indicated below. It is used in Master state machine A. It is defined so that the detailed code that it invokes does not need to be placed into the state machine diagram.

```
saveConfirmInfo1()
```

```
{
        MLME-TIMINGMSMT.confirm(&paramsFromConfirm);

        requestParams.FollowUpDialogToken = paramsFromConfirm.DialogToken;
        requestParams.T1 = paramsFromConfirm.T1;
        requestParams.T4 = paramsFromConfirm.T4;

        // NOTE: In Timing Measurement, T1 is in units of 10 ns.
        // upstreamTxTime is units of 2^-16 nanoseconds.

        K = 1;
        // K is 1 for Timing Measurement.
        residenceTime = MDSyncSend.rateRatio *
                (paramsFromConfirm.T1 * 10^K*(2^16) - MDSyncSend.upstreamTxTime);

        requestParams.VendorSpecific.correctionField =
                residenceTime + MDSyncSend.followUpCorrectionField;
        // NOTE: T1 and T4 are timestamps from a single
        // local clock source.  The roll-over of the 32-bit timestamps returned by
        // MLME-TIMINGMSMT.request and MLME-TIMINGMSMT.indication
        //        must be accounted for.
}
```

e) **initiateRequestWaitConfirm():** This function is defined as indicated below. It is used in Master state machine B. It is defined so that the detailed code that it invokes does not need to be placed into the state machine diagram.

```
initiateRequestWaitConfirm()
{
        If ((++dialogToken % 256) == 0) dialogToken++;
        If (nframesSent == initReqParamsDot11MasterB.framesPerBurst)
                dialogToken = 0;

        requestParams.DialogToken=dialogToken;
        requestParams.PeerMACAddress = dot11SlaveMac;
        setRequestParams(&requestParams, MDSyncSend);
        // In the following statement, MinDeltaFTM, which is in units of 100
        //    microseconds, is converted to UScaledNs (i.e., units of 2^-16 ns, see 6.3.3.2)
        nextRequestSendTime = currentTime +
                initReqParamsDot11MasterB.MinDeltaFTM * (65536 x 10^5);
        MLME-FINETIMINGMSMT.request(requestParams);
        requestParams.FollowUpDialogToken = 0;    //In case no confirm is received
        slaveMacOfLastRequest = dot11SlaveMac;
}
```

f) **saveConfirmInfo2():** This function is defined as indicated below. It is used in Master state machine B. It is defined so that the detailed code that it invokes does not need to be placed into the state machine diagram.

```
saveConfirmInfo2()
{
        MLME-FINETIMINGMSMT.confirm(&paramsFromConfirm);

        requestParams.FollowUpDialogToken = paramsFromConfirm.DialogToken;
        requestParams.T1 = paramsFromConfirm.T1;
        requestParams.T4 = paramsFromConfirm.T4;
```

```
        // NOTE: In Fine Timing Measurement, T1 is in units of 0.1 ns.
        // upstreamTxTime is units of 2^-16 nanoseconds.

        K = -3;
        // K is 1 for Timing Measurement and -3 for Fine Timing Measurement.
        residenceTime = MDSyncSend.rateRatio *
                (paramsFromConfirm.T1 * 10^K*(2^16) - MDSyncSend.upstreamTxTime);

        requestParams.VendorSpecific.correctionField =
        residenceTime + MDSyncSend.followUpCorrectionField;
        // NOTE: T1 and T4 are timestamps from a single
        // local clock source. The roll-over of the 48-bit timestamps returned by
        // MLME-FINETIMINGMSMT.request and MLME-FINETIMINGMSMT.indication
        // must be accounted for.

        // A frame is only counted as being sent, for purposes of number of frames in a
        // burst, if a confirm is received. It is up to IEEE Std 802.11 to handle the case
        // where a confirm is not received.
        nframesSent += 1;
    }
```

12.5.1.5 Shared Variables

a) **MDSyncSend:** a structure as defined in 10.2.2.1.
b) **portOper:** a Boolean as defined in 10.2.5.12.
c) **ptpPortEnabled:** a Boolean as defined in 10.2.5.13.
d) **asCapable:** a Boolean whose value is specified in 12.4 and 10.2.5.1.
e) **tmFtmSupport:** an Octet whose bits are interpreted as Booleans, and whose values are specified in 12.3, Table 12-1.
f) **ftmReqGrantedMaster:** a Boolean whose value is TRUE if the master grants the current initial FTM request, and FALSE otherwise.

12.5.1.6 Master primitives

12.5.1.6.1 MLME-TIMINGMSMT.request

The MLME-TIMINGMSMT.request primitive is used by a master station to initiate a timing measurement and also communicates timestamps *t1* and *t4* captured by the master during a previous measurement. The primitive and its parameters are specified in 6.3.57.2 of IEEE Std 802.11-2016.

12.5.1.6.2 MLME-TIMINGMSMT.confirm

The MLME-TIMINGMSMT.confirm primitive indicates that a timing measurement request has completed. The primitive and its parameters are specified in 6.3.57.3 of IEEE Std 802.11-2016.

12.5.1.6.3 MLME-FINETIMINGMSMT.request

The MLME-FINETIMINGMSMT request primitive is used by a master station to initiate a fine timing measurement, and also communicates timestamps *t1* and *t4* captured by the master during a previous measurement. The primitive and its parameters are specified in 6.3.58.2 of IEEE Std 802.11-2016.

12.5.1.6.4 MLME-FINETIMINGMSMT.confirm

The MLME-FINETIMINGMSMT request primitive indicates that a fine timing measurement request has completed. The primitive and its parameters are specified in 6.3.58.3 of IEEE Std 802.11-2016.

12.5.1.6.5 MLME-FINETIMINGMSMTRQ.indication

The MLME-FINETIMINGMSMTRQ.indication primitive indicates to a master station that the slave is requesting a burst of FTM frames with the indicated parameters. The primitive and its parameters are specified in 6.3.70.3 of IEEE Std 802.11-2016.

12.5.2 Media-dependent slave state machine

12.5.2.1 Overview

The MD entity of an IEEE 802.11 port whose port state is SlavePort or PassivePort (see 10.3.6) shall behave in a manner that is indistinguishable, relative to an observer external to a system, from a strict implementation of the slave state machine in 12.5.2.2, the local variables specified in 12.5.2.3, the functions specified in 12.5.2.4, the shared variables specified in 12.5.2.5, and the primitives defined in 12.5.2.6.

The slave state machine is responsible for collecting information fromthe Timing measurement or Fine Timing measurement indications, constructing an MDSyncReceive structure with the relevant information, and passing the structure to the PortSync entity for further processing. In order to do this, the state machine saves locally captured timestamps (i.e., *t2* and *t3*) received in the indication, associating them with the timestamps sent from the master port in a future indication (i.e., *t1* and *t4)*. In addition, in the case of Fine Timing measurement, the slave state machine is responsible for generating the MLME-FINETIMINGMSMTRQ.request primitive, which causes the initial FTM request frame to be sent to the master.

12.5.2.2 State diagram

While quantities are shown to be computed from information in consecutive indications, an implementation can choose to compute over longer intervals as long as the clock performance requirements of Annex B are met.

12.5.2.3 State machine local variables

a) **indParams:** a structure whose members contain the values of the fields of the MLME-TIMINGMSMT.indication primitive or MLME-FINETIMINGMSMT.indication primitive, as defined in 12.5.2.6, depending on whether TM or FTM, respectively, is used.
b) **previousIndParams:** a structure with members identical to those of indParams, used to save parameters from the previous indication.
c) **rcvdIndication:** a Boolean that is set to TRUE when either the MLME-TIMINGMSMT.indication or MLME-FINETIMINGMSMT.indication primitive is received.
d) **RESTART:** a Boolean that indicates, in the case FTM is being used, that a new burst should be initiated.
e) **rcvdIndicationTimeoutTime:** a UScaledNs variable whose value is the time after which the state machine will not wait any longer for the next MLME-FINETIMINGMSMT.indication, and a new burst is initiated.
f) **nframesRcvd:** an unsigned 8-bit integer used to count the number of frames received by the master in the burst (the number of indications received from the master are counted).
g) **initReqParamsDot11Slave:** a structure whose members contain the values of the fields of an MLME-FINETIMINGMSMTRQ.indication primitive.

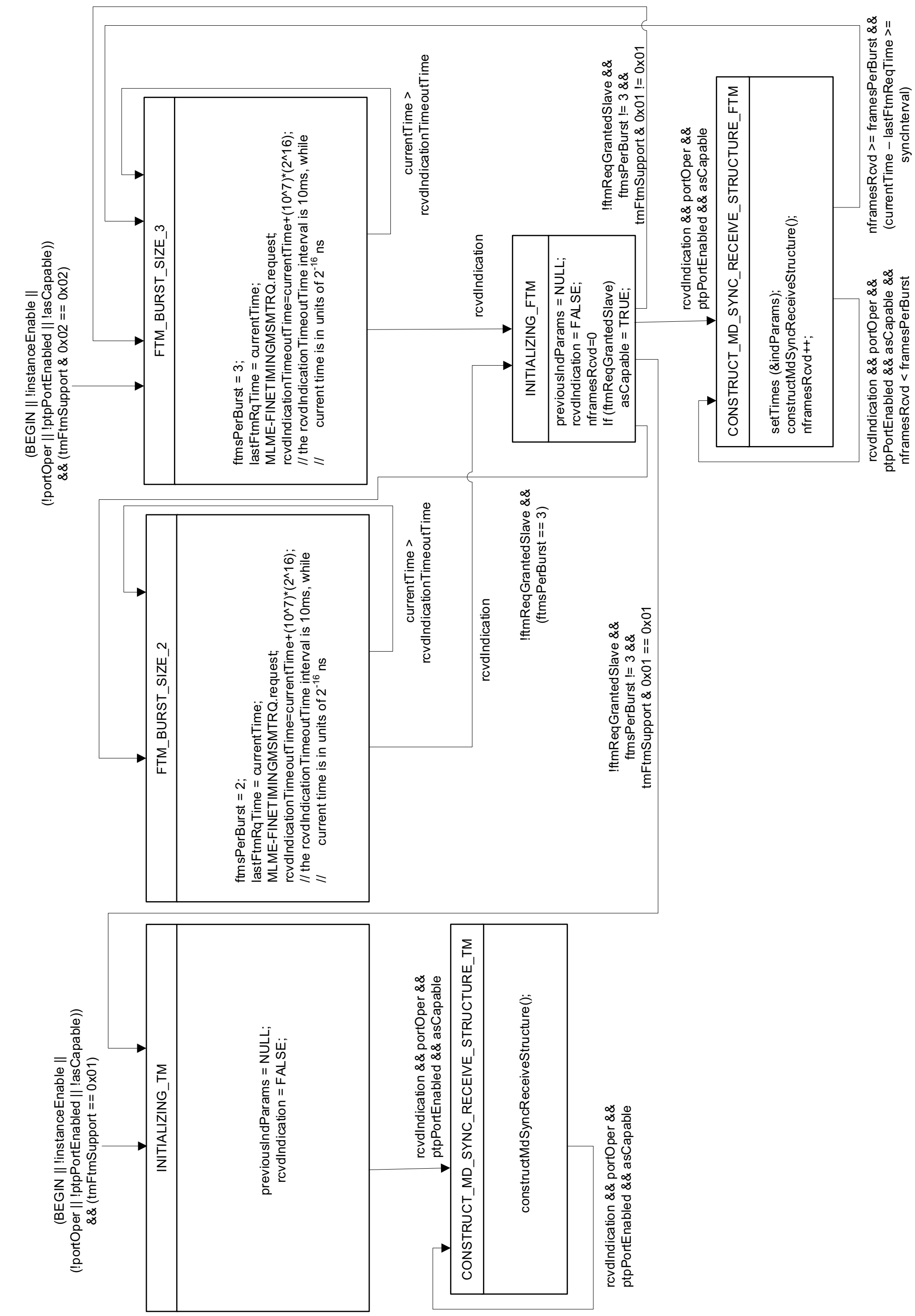


Figure 12-7—Slave state machine

h) **ftmsPerBurst:** the value of the FTM parameter 'FTMs per burst' (see 12.6), i.e., the number of FTM frames in the burst granted by the master.

i) **ftmReqGrantedSlave:** a Boolean that is TRUE if the master has granted the respective request for a burst, and FALSE otherwise.

j) **t1_1, t1_2, t2_1, t2_2, t2_3, t3_1, t3_2, t3_3, t4_1, t4_2:** temporary local variables used to hold the values of the timestamps T1, T2, T3, and T4 returned by the MLME-FINETIMINGMSMT.indication primitive in the structure indParams.

k) D1, D2: temporary local variables used to hold the values of delay computed for the first two FTM frames and corresponding Acks in the case where the master grants the request for three FTM frames.

12.5.2.4 State machine functions

a) **setMDSyncReceiveDot11Slave(indParams):** creates an MDSyncReceive structure and returns the structure. All fields are assigned from FollowUpInformation (contained in the VendorSpecific information element) of indParams as in 11.2.14.2.1.

b) **passMDSyncReceiveToPortSync():** passes an MDSyncReceive structure to the PortSync entity of this port.

c) **setTimes (&indParams):**This function extracts the timestamp values from the successive indParams structures returned by the multiple FTM frame exchanges of a burst, and places the correct times (corresponding to minimum delay frames and Acks) in the final indParams structure. This final indParams structure is then used in the function constructMdSyncReceiveStructure(). This procedure is needed for the case where the master grants three FTM frames for the burst. If the master grants only two FTM frames for the burst, the timestamp values returned in the indication primitive of the second frame are used. The function setTimes is used in the Slave state machine, and is defined so that the detailed code that it invokes does not need to be placed into the state machine diagram.

```
setTimes (&indParams)
{
        if (nframesRcvd == 1)
        {
                t2_1 = indParams.T2;
                t3_1 = indParams.T3;
        }
        if (nframesRcvd == 2)
        {
                t1_1 = indParams.T1;
                t2_2 = indParams.T2;
                t3_2 = indParams.T3;
                t4_1 = indParams.T4
        }
        if (nframesRcvd == 3)
        {
                t1_2 = indParams.T1;
                t2_3 = indParams.T2;
                t3_3= indParams.T3;
                t4_2 = indParams.T4;
                if (ftmsPerBurst == 3)
                {
                        D1 = t2_1 - t1_1;
                        D2 = t2_2 - t1_2
                        if (D2 <= D1)
                        {
```

```
                        indParams.T2 = t2_2;
                        indParams.T1 = t1_2;
                    }
                    else
                    {
                        indParams.T2 = t2_1;
                        indParams.T1 = t1_1;
                    }
                    D1 = t4_1 - t3_1;
                    D2 = t4_2 - t3_2
                    if (D2 <= D1)
                    {
                        indParams.T4 = t4_2;
                        indParams.T3 = t3_2;
                    }
                    else
                    {
                        indParams.T4 = t4_1;
                        indParams.T3 = t3_1;
                    }

                }
            }



    }
```

d) **constructMdSyncReceiveStructure():** This function constructs the MDSyncReceive structure, and is defined as indicated below. It is used in the Slave state machine. It is defined so that the detailed code that it invokes does not need to be placed into the state machine diagram.

```
constructMdSyncReceiveStructure()
{
    if (tmFtmSupport == 0x01)
        MLME-TIMINGMSMT.indication(&indParams);
    else if (tmFtmSupport & 0x02 == 0x02)
    {
        MLME-FINETIMINGMSMT.indication(&indParams);
        nframesRcvd++;
        if (nframesRcvd == initReqParamsDot11Slave.framesPerBurst) ||
            (currentTime > endofBurstDurationTime)
                RESTART=1;
    }

    if ((previousIndParams != NULL) &&
        (previousIndParams.PeerMacAddress == dot11SlaveMac) &&
        (indParams.FollowUpDialogToken != 0))
    {

        neighborRateRatio =
            (indParams.T1-previousIndParams.T1) /
            (indParams.T2-previousIndParams.T2);
```

```
                //NOTE: Other methods of computing neighborRateRatio
                                can be used.

                if (tmFtmSupport == 0x01)
                        K = 1;
                else if (tmFtmSupport & 0x02 == 0x02)
                        K = -3;
                //K = 1 for Timing Measurement and K = -3 for Fine Timing Measurement
                meanLinkDelay =
                        (((indParams.T4 - indParams.T1) -
                        neighborRateRatio * (indParams.T3 - indParams.T2)) /
                                        (2.0)) * (10^K);

                //NOTE: Other methods of computing meanLinkDelay
                                ~~may~~ can be used.

                MDSyncReceive = setMDSyncReceiveDot11Slave(indParams);
                MDSyncReceive.VendorSpecific.rateRatio +=
                        (neighborRateRatio – 1);
                MDSyncReceive.VendorSpecific.upstreamTxTime =
                                indParams.T2* (2^16) *(10^K) -
                                meanLinkDelay*(2^16)/neighborRateRatio;
                //NOTE: Actions performed with the timestampError
                        parameters of indParams are implementation independent.

                passMDSyncReceiveToPortSync(&MDSyncReceive);
        }
        previousIndParams = indParams;
        rcvdIndication = FALSE;
}
```

12.5.2.5 State machine shared variables

a) **MDSyncReceive:** a structure used for passing information between MD and PortSync, as defined in 10.2.2.2.
b) **portOper:** a Boolean as defined in 10.2.5.12.
c) **ptpPortEnabled:** a Boolean as defined in 10.2.5.
d) **asCapable:** a Boolean as defined in 12.4 and 10.2.5.1.
e) **meanLinkDelay:** the delay over the link to the associated WLAN station as defined in 10.2.5.8.
f) **neighborRateRatio:** the measured ratio of the frequency of the LocalClock entity of the time-aware system at the other end of the link attached to this port, to the frequency of the LocalClock entity of this time-aware system, as defined in 10.2.5.7.
g) **tmFtmSupport:** an Octet whose value is specified in 12.3.

12.5.2.6 Slave primitives

12.5.2.6.1 MLME-TIMINGMSMT.indication

The MLME-TIMINGMSMT.indication primitive is received by a slave station as the natural result of the peer master station issuing the corresponding request primitive, and carries the same parameters plus local timestamp information. The primitive and its parameters are specified in 6.3.57.4 of IEEE Std 802.11-2016.

12.5.2.6.2 MLME-FINETIMINGMSMT.indication

The MLME-FINETIMINGMSMT.indication primitive is received by a slave station as the natural result of the peer master station issuing the corresponding request primitive, and carries the same parameters plus local timestamp information. The primitive and its parameters are specified in 6.3.58.4 of IEEE Std 802.11-2016.

12.5.2.6.3 MLME-FINETIMINGMSMTRQ.request

The MLME-FINETIMINGMSMTRQ.request primitive is used by the slave to request a burst of FTM frames from the master, with respective FTM parameters. The primitive and its parameters are specified in 6.3.70.2 of IEEE Std 802.11-2016.

12.6 FTM parameters

The values of the FTM parameters that are relevant to time synchronization transport in this standard are specified in Table 12-2, along with a brief description of each parameter. These parameter values are carried in the initial FTM Request. A more detailed description of these and other FTM parameters are given in 9.4.2.168 of IEEE Std 802.11-2016. The parameters Burst Duration and Min Delta FTM depend on the time synchronization message interval, i.e., on currentLogSyncInterval (see 10.7.2.3 and 12.8). The values for these parameters are specified in Table 12-3. In this table, the Burst Duration encoding (i.e., value and corresponding duration in ms) is taken from Table 9-257 of IEEE Std 802.11-2016. The Min Delta FTM encoding values are in multiples of 100 µs (see 9.4.2.168 of IEEE Std 802.11-2016).

The values of the FTM parameters given in Table 12-2 shall be used in the MLME-FINETIMINGMSMTRQ.request invoked by the slave STA (i.e., in the initial FTM Request). The values for Burst Duration and Min Delta FTM given in Table 12-3 shall be used in the MLME-FINETIMINGMSMTRQ.request invoked by the slave STA.

Background on the derivation of the FTM parameters of Table 12-2 and Table 12-3 is given in [B26].

12.7 Format of VendorSpecific information element

The IEEE 802.11 MLME request and indication primitives for timing measurement and fine timing measurement support an ability to carry data transparently between stations using the VendorSpecific information element. The Type field within the VendorSpecific Content identifies the type of information that follows the Type field. See Figure 12-8 and Table 12-4.

Table 12-2—FTM parameters relevant to time synchronization transport

Parameter	Value	Description
Number of Bursts Exponent	0	log to base 2 of the number of bursts requested by the slave (value of 0 indicates that one burst is requested)
Burst Duration	see Table 12-3	duration of the burst of burst of FTM frames and their corresponding Acks
Min Delta FTM	see Table 12-3	minimum time between consecutive FTM frames
Partial TSF Timer	1	see 9.4.2.168 of IEEE Std 802.11-2016
Partial TSF Timer No Preference	reserved in the initial FTM request	see 9.4.2.168 of IEEE Std 802.11-2016
ASAP	1	ASAP = 1 indicates that the slave would like the master to respond as soon as possible
ASAP Capable	reserved in the initial FTM request	see 9.4.2.168 of IEEE Std 802.11-2016
FTMs per burst	3 in the first initial FTM Request 2 in the first retry, if the first initial FTM Request is not granted	Desired number of FTM frames and corresponding Acks in the requested burst
Burst Period	reserved when Number of Bursts Exponent is zero	see 9.4.2.168 of IEEE Std 802.11-2016

Table 12-3—Values of Burst Duration and Min Delta FTM, for each value of currentLogSyncInterval

currentLogSyncInterval	nominal message rate (messages/s)	Burst Duration	Min Delta FTM
-24 through -6, inclusive	64 - 16,777,216	6 (4 ms)	6 (0.6 ms)
-5	32	8 (16 ms)	25 (2.5 ms)
-4	16	9 (32ms)	50 (5 ms)
-3	8	10 (64 ms)	100 (10 ms)
-2 through 24, inclusive	4	11 (128 ms)	200 (20 ms)

Table 12-4—Values of the Type field in the VendorSpecific information element

Value	Description
0	The Type field is followed by FollowUp-Information
1–255	Reserved

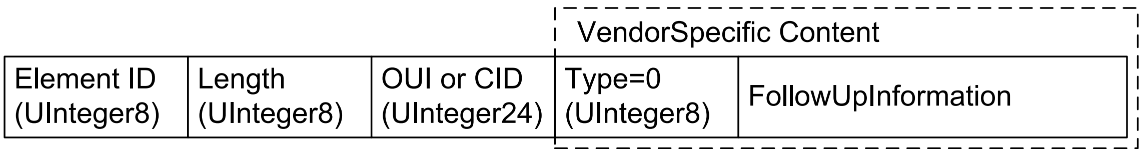


Figure 12-8—Format of VendorSpecific information element when Type = 0

This mechanism shall be used to carry end-to-end link-independent timing information from the master port to the associated slave port, including preciseOriginTimestamp, rateRatio, correctionField, and other fields of the Follow-Up message, as described in 12.5.1.4. For consistency, all of these fields are packed into the FollowUpInformation field using exactly the same format as used for full-duplex point-to-point links. In other words, the master state machine communicates an entire Follow_Up message (i.e., including all the fields of the common header, see 11.4.2 and 10.6.2, the preciseOriginTimestamp, and all the fields of the Follow_Up information TLV, see 11.4.4) using this mechanism. The Type field, illustrated in Figure 12-8, identifies this use of the OUI or CID within the VendorSpecific information element. Table 12-4 lists values for the Type field.

12.8 Synchronization message interval

12.8.1 General synchronization message interval specification

The mean time interval between successive synchronization messages shall be as specified in 10.7.2.1, 10.7.2.3, and 12.8.2.

12.8.2 Synchronization message interval default value

The default value of initialLogSyncInterval (see 10.7.2.3) is –3. Every port supports the value 127; the port does not send Sync messages when currentLogSyncInterval has this value (see 10.3.18). A port may support other values, except for the reserved values indicated in Table 10-16. A port ignores requests (see 10.3.18) for unsupported values.

Processing of the message interval request TLV carried in a Signaling message (see 10.6.4) shall be supported, as specified by the SyncIntervalSetting state machine of 10.3.18 (see Figure 10-20), except that: the logLinkDelayInterval (which is not relevant to IEEE 802.11 ports) is set to –128 by the sender of the Signaling message, the logLinkDelayInterval is ignored by the receiver, and unsupported values of logTimeSyncInterval are ignored by the receiver.

NOTE 1—For TM, a slave port that requests (using a Signaling message that contains a message interval request TLV, see 10.6.4 and 10.3.18) that the port at the other end of the attached link set its currentLogSyncInterval to a specific value can determine if the request was honored by examining the logMessageInterval field of a FollowUpInformation contained in the VendorSpecific information element of a subsequent MLME indication primitive.

NOTE 2—The time interval between every pair of adjacent timing or fine timing measurements is not guaranteed to be precisely the same. Some variation is expected, due to factors such as the MAC protocol (e.g., delay in accessing the medium and/or packet retries) and the power state of the associated station. However, timestamp t1 is not captured when either TIMINGMSMT.request or FINETIMINGMSMT.request is invoked, but only after the frame resulting from the request is actually transmitted.

13. Media-dependent layer specification for interface to IEEE 802.3 Ethernet passive optical network link

13.1 Overview

13.1.1 General

This clause specifies the service interface primitives, state machines, and message formats that provide accurate synchronized time across IEEE 802.3 Ethernet passive optical network (EPON) links, through the use of the timing process and measurements specified in 64.2.1.1, 64.3.2.4, 77.2.1.1, and 76.1.2 of IEEE Std 802.3-2018. For purposes of this clause, an EPON link is an EPON that contains one optical line terminal (OLT) and associated optical network units (ONUs).

A time-aware system contains at most one ONU, but may contain more than one OLT (i.e., thePTP Instances of a time-aware system are clock slaves to at most one EPON link, but may be clock masters to more than one EPON link).

13.1.2 Description of the EPON timing process

The timing process in EPON relies on the 32-bit counters (see 64.2.2.2 and 77.2.2.2 of IEEE Std 802.3-2018) at both the OLT and the ONU. The 32-bit counter used by EPON is the LocalClock entity of the PTP Instance that uses the respective OLT or ONU. These counters increment every time_quantum, which is equal to 16 ns (see 64.2.2.1 and 77.2.2.1 of IEEE Std 802.3-2018). IEEE Std 802.3-2018 defines multipoint control protocol (MPCP), which is one of the protocols that enables MAC clients to communicate over a point-to-multipoint optical network. When either the clock master (OLT) or the clock slave (ONU) transmits an MPCP data unit (MPCPDU), its counter value is mapped into the timestamp field. Clauses 64 and 77 of IEEE Std 802.3-2018 specify the EPON timing mechanism.

13.1.3 Best master selection

13.1.3.1 General

An EPON link contains one OLT and the associated ONUs. The OLT is the clock master and the associated ONUs are clock slaves. The OLT initiates the time synchronization as a requester. The ONUs are the responders of the time synchronization. This means that the invocation of BMCA results in the OLT having the port state MasterPort and the ONU having the port state SlavePort (see 10.3.1 and Table 10-2),for all PTP Instances using these ports, regardless of the attributes of PTP Instances downstream from the ONU. This behavior is achieved using the acceptable master table feature defined in 17.5 of IEEE Std 1588-2019.

A PTP Instance that contains an ONU port shall maintain a configured table, the acceptableMasterTable, and a per-port Boolean variable acceptableMasterTableEnabled. The data type of acceptableMasterTable is AcceptableMasterTable (see 13.1.3.2).

13.1.3.2 AcceptableMasterTable

The AcceptableMasterTable type represents a table of AcceptableMaster entries.

```
struct AcceptableMasterTable{
        UInteger16 maxTableSize;
        UInteger16 actualTableSize;
        AcceptableMaster[actualTableSize] acceptableMaster;
}
```

The maxTableSize member is the maximum size of the AcceptableMasterTable. The actualTableSizeMember is the actual size of the AcceptableMasterTable. The AcceptableMaster array contains a list of AcceptableMaster ports. The value of maxTableSize is implementation specific. actualTableSize shall be less than or equal to maxTableSize.

An AcceptableMasterTable is configurable and may contain a number of AcceptableMaster entries up to maxTableSize.

13.1.3.3 AcceptableMaster

The AcceptableMaster type represents a port that can be considered, in the execution of the BMCA, as a candidate for master.

```
struct AcceptableMaster{
        PortIdentity acceptablePortIdentity;
        UInteger8 alternatePriority1;
}
```

The acceptablePortIdentity member is the PortIdentity of an acceptable master port. The alternatePriority1 member contains an alternate value for the priority1 attribute of the acceptable master port (see 13.1.3.4).

13.1.3.4 Acceptable master table feature

The acceptable master table feature shall modify the operation of the BMCA (see 10.3) as follows:

a) If acceptableMasterTableEnabled for a port is FALSE, the BMCA operates as described in 10.3 and its subclauses.
b) If acceptableMasterTableEnabled for a port is TRUE, then:
    1) The function qualifyAnnounce() of the PortAnnounceReceive state machine (see 10.3.11.2.1) is replaced by the following:

       qualifyAnnounce (rcvdAnnouncePtr): qualifies the received Announce message pointed to by rcvdAnnouncePtr as follows:
        i) if the Announce message was sent by the current PTP Instance, i.e., if sourcePortIdentity.clockIdentity (see 10.6.2.2.11 and 8.5.2) is equal to thisClock (see 10.2.4.22), the Announce message is not qualified and FALSE is returned;
        ii) if the stepsRemoved field is greater than or equal to 255, the Announce message is not qualified and FALSE is returned;
        iii) if the sourcePortIdentity of the Announce message is not equal to the sourcePortIdentity of one of the entries of the acceptableMasterTable, FALSE is returned;
        iv) if a path trace TLV is present and one of the elements of the pathSequence array field of the path trace TLV is equal to thisClock (i.e., the clockIdentity of the current PTP Instance, see 10.2.4.22), the Announce message is not qualified and FALSE is returned; otherwise, the Announce message is qualified and TRUE is returned. If a path trace TLV is present and the portState of the port is SlavePort, the pathSequence array field of the TLV is copied to the global array pathTrace, and thisClock is appended to pathTrace (i.e., is added to the end of the array).
    2) If the alternatePriority1 member of the AcceptableMaster array element that corresponds to the sourcePortIdentity of a received Announce message is 0, the alternatePriority1 member has no effect on the operation of BMCA.
    3) If the alternatePriority1 member of the AcceptableMaster array element that corresponds to the sourcePortIdentity of a received Announce message is greater than 0, the value of the grandmasterPriority1 field of the Announce message is replaced by the value of alternatePriority1 of this AcceptableMaster array element for use in the invocation of BMCA.

13.1.3.5 Default configuration of acceptable master table feature

The default configuration of the acceptable master table feature for a PTP Instance that is attached to an IEEE 802.3 EPON link shall be as follows:

a) If the PTP Instance does not contain an ONU port, the default acceptableMasterTable is empty, i.e., the member actualTableSize is 0 and there are no AcceptableMaster array entries. The variable acceptableMasterTableEnabled for each port is set to FALSE.
b) If the PTP Instance contains an ONU port, the default acceptableMasterTable contains one element in the AcceptableMaster array. The member actualTableSize is 1. The acceptablePortIdentity of that element is set equal to the portIdentity of the OLT port that the ONU port is attached to, and alternatePriority1 set equal to 244. The variable acceptableMasterTableEnabled for each port is set to TRUE.

NOTE—These default settings ensure that, with the default priority1 values of 8.6.2.1, Table 8-1, used for all PTP Instances, the PTP Instance that contains the ONU port will consider Announce messages only from the OLT that the ONU port is attached to when invoking the BMCA. The alternatePriority1 value of 244 ensures that the OLT will be considered better than the ONU in the sense of the BMCA, which will cause the OLT port state to be set to MasterPort and the ONU port state to be set to SlavePort. All other ports of this PTP Instance that are not disabled and for which asCapable is TRUE will have port states of either MasterPort or PassivePort. If all PTP Instances downstream from the ONU have priority1 greater than 244, then the port at the other end of each link attached to each non-ONU port that is not disabled and for which asCapable is TRUE will have port states of either SlavePort or Passive port; in this case, the downstream network portions will get their timing through the EPON. However, if a downstream PTP Instance has priority1 less than 244, or priority1 equal to 244 and is better than the grandmaster information contained in the Announce message received by the ONU based on other attributes, then the portion of the network that is downstream of the ONU and includes that better PTP Instance will get its timing from that better downstream PTP Instance. In this case, the endpoints of the link of that network portion attached to the PTP Instance that contains the ONU will both have port states of MasterPort, and the ports at each end of the link will send Announce messages. However, the Announce messages sent by the downstream PTP Instance will be ignored by the PTP Instance that contains the ONU because the sourcePortIdentity of those Announce messages will not be contained in the acceptableMasterTable. The Announce messages sent by the PTP Instance that contains the ONU will be used in the invocation of the BMCA at the downstream PTP Instance; however, those Announce messages will not reflect the best master because one of the downstream PTP Instances is better.

13.1.4 Time synchronization in EPON

Transmission in the EPON downstream direction (from OLT to ONUs) utilizes time division multiplexing (TDM). In the upstream direction (from ONUs to OLT), time division multiple access (TDMA) is employed. Due to the frame queuing in TDMA, the downstream delay is different from the upstream delay. Asymmetric delay also occurs in the EPON physical layer due to upstream and downstream transmission using different wavelengths. The index of refraction is frequency dependent, which results in the upstream and downstream delays being asymmetric. The accurate time synchronization across the EPON links is operated as follows. It is assumed that the clock master (the OLT) has an accurate synchronized time. The clock master informs the clock slave (the ONU) what the accurate synchronized time will be when the counter of the clock slave reaches a certain value. The information transfer can be accomplished using the organization-specific slow protocol (OSSP) message (see Clause 57 of IEEE Std 802.3-2018).

The following reference process, illustrated schematically in Figure 13-1, will result in the clock slave of an ONU being synchronized to the clock master of the OLT:

a) The clock master selects a value X of the local MPCP counter that is used as the timing reference. Any value can be chosen, provided it is relative to the current epoch of the MPCP counter.
b) The clock master calculates the $ToD_{X,i}$ based on $ToD_{X,o}$ using Equation (13-1).

$$ToD_{X,i} = ToD_{X,o} + RTT_i \cdot \frac{\text{ndown}}{(\text{nup} + \text{ndown})} \cdot \text{rateRatio} \tag{13-1}$$

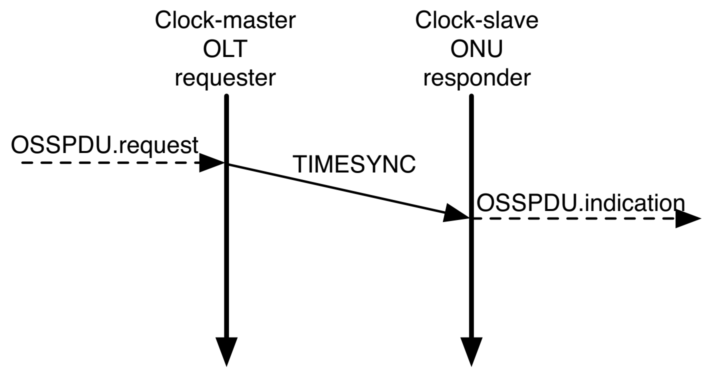


Figure 13-1—IEEE 802.3 EPON time-synchronization interfaces

where $ToD_{X,i}$ is the synchronized time when the MPCP counter at the clock slave *i* reaches a value equal to the timestamp X minus the *onuLatencyFactor*; $ToD_{X,o}$ is the synchronized time when the MPCP counter at the clock master reaches a value equal to the timestamp X plus the *oltLatencyFactor*; RTT_i is the round-trip time measured by the clock master for clock slave *i*, i.e., ONU *i*; nup is the effective refraction index of the light propagating in the upstream channel; ndown is the effective refraction index of the light propagating in the downstream channel; and rateRatio is the rateRatio member of the most recently received MDSyncSend structure. The *onuLatencyFactor* and *oltLatencyFactor* are given in Equation (13-2) and Equation (13-3), respectively. The impact of the worst-case variation in the transmission wavelength for the clock master and clock slave transmitters is examined in VII of ITU-T G.984.3, Amendment 2.

$$onuLatencyFactor = onuIngressLatency - (onuIngressLatency + onuEgressLatency) \cdot \frac{\mathrm{ndown}}{(\mathrm{nup} + \mathrm{ndown})} \cdot \mathrm{rateRatio} \quad (13\text{-}2)$$

$$oltLatencyFactor = oltEgressLatency - (oltIngressLatency + oltEgressLatency) \cdot \frac{\mathrm{ndown}}{(\mathrm{nup} + \mathrm{ndown})} \cdot \mathrm{rateRatio} \quad (13\text{-}3)$$

c) The clock master sends the pair of values (X, $ToD_{X,i}$) to clock slave i via the downstream TIMESYNC message.

NOTE—After the clock slave receives the downstream TIMESYNC message, it can compute the synchronized time, *ToD*, when the value of the local MPCP counter is equal to *S*; *ToD* is given by the following equation:

$$ToD = ToD_{X,i} + [(S - X) \bmod (2^{32})](16\text{ ns}) \cdot \text{rateRatio}$$

where (A) mod (B) is A modulo B.

The OSSP message is a general message (see 3.8), analogous to Follow_Up. Note that the preceding synchronized time values correspond to timestamps that are referenced to the MAC control sublayer. Both the clock master and clock slave are responsible for compensating their processing delays (e.g., the ingressLatency and egressLatency, as described in 8.4.3). RTT_i is measured using MPCPDU timestamps, inserted into the frame structure as specified by 64.2.1.1 and 77.2.1.1 of IEEE Std 802.3-2018.

13.2 Message attributes

13.2.1 Message class

The TIMESYNC message is a general message (see 3.8 and 8.4.2.2). It is transmitted in the downstream direction, from OLT to ONU.

13.3 Message format

13.3.1 TIMESYNC message

13.3.1.1 General TIMESYNC message specifications

The fields of the body of the TIMESYNC message shall be as specified in Table 13-1 and 13.3.1.2 and its subclauses.

13.3.1.2 TIMESYNC message field specifications

13.3.1.2.1 Destination address (Octet6)

The destination address field is equal to 01-80-C2-00-00-02 (see 57A.3 of IEEE Std 802.3-2018).

13.3.1.2.2 Source address (Octet6)

The source address field is the individual MAC address associated with the port through which the TIMESYNC message is transmitted (see 57B.1.1 of IEEE Std 802.3-2018).

13.3.1.2.3 Length/Type (Octet2)

The value of this field is equal to 0x8809 (see 57A.4 of IEEE Std 802.3-2018).

13.3.1.2.4 Subtype (Octet)

The value of this field is equal to 0x0A (see 57A.4 of IEEE Std 802.3-2018).

Table 13-1—TIMESYNC message fields

Bits (7 6 5 4 3 2 1 0)		Octets	Offset
Destination Address		6	0
Source Address		6	6
Length/Type		2	12
Subtype		1	14
OUI or CID		3	15
Message Identifier		2	18
X		4	20
$ToD_{X,i}$		10	24
sourcePortIdentity		10	34
logMessageInterval		1	44
rateRatio		8	45
gmTimeBaseIndicator		2	53
lastGmPhaseChange		12	55
scaledLastGmFreqChange		4	67
domainNumber		1	71
majorSdoId	reserved	1	72
minorSdoId		1	73
reserved		0	74
FCS		4	

13.3.1.2.5 OUI or CID (Octet3)

This field contains the OUI or CID that identifies the Organization-Specific Data. The value is 00-80-C2, i.e., the OUI assigned to IEEE 802.1.

13.3.1.2.6 Message identifier (Octet2)

This field is the TIMESYNC message identifier. The value of this field is 1.

13.3.1.2.7 *X* (UInteger32)

The *X* field is the selected timestamp that will be used as the timing reference as specified in 13.1.4.

13.3.1.2.8 $ToD_{X,i}$ (Timestamp)

$ToD_{X,i}$ is the synchronized time when the MPCP counter at the clock slave *i* reaches a value equal to *X* minus the *onuLatencyFactor* (see 13.1.4). *X* is carried in the respective TIMESYNC message. Synchronization of the MPCP clock is described in detail in 64.2.1.1 and 77.2.1.1 in IEEE Std 802.3-2018, for 1G-EPON 10G-EPON, respectively.

NOTE—Any subnanosecond portion of synchronized time (in this case, time of day), normally transported in a correction field (see 10.2.2.1.2, 10.2.2.2.2, and 10.2.2.3.5), is not transported over EPON.

13.3.1.2.9 sourcePortIdentity (PortIdentity)

This field is specified as the sourcePortIdentity member of the MDSyncSend structure most recently received from the PortSync entity of the OLT (see 10.2.2.1.4).

13.3.1.2.10 logMessageInterval (Integer8)

This field is specified as the logMessageInterval member of the MDSyncSend structure most recently received from the PortSync entity of the OLT (see 10.2.2.1.5). It is the value of the currentLogSyncInterval for this port (see 10.7.2.3).

13.3.1.2.11 rateRatio (Float64)

This field is specified as the rateRatio member of the MDSyncSend structure most recently received from the PortSync entity of the OLT (see 10.2.2.1.8).

13.3.1.2.12 gmTimeBaseIndicator (UInteger16)

This field is specified as the gmTimeBaseIndicator member of the MDSyncSend structure most recently received from the PortSync entity of the OLT (see 10.2.2.1.9).

13.3.1.2.13 lastGmPhaseChange (ScaledNs)

This field is specified as the lastGmPhaseChange member of the MDSyncSend structure most recently received from the PortSync entity of the OLT (see 10.2.2.1.10).

13.3.1.2.14 scaledLastGmFreqChange (Integer32)

The value of scaledLastGmFreqChange is the fractional frequency offset of the current grandmaster relative to the previous grandmaster, at the time that the current grandmaster became grandmaster, or relative to itself prior to the last change in gmTimeBaseIndicator, multiplied by 2^{41} and truncated to the next smaller signed integer. The value is obtained by multiplying the lastGmFreqChange member of MDSyncSend (see 10.2.2.1) whose receipt causes the MD entity to send the TIMESYNC message by 2^{41}, and truncating to the next smaller signed integer.

NOTE—The above scaling allows the representation of fractional frequency offsets in the range $[-(2^{-10} - 2^{-41}), 2^{-10} - 2^{-41}]$, with granularity of 2^{-41}. This range is approximately $[-9.766 \times 10^{-4}, 9.766 \times 10^{-4}]$.

13.3.1.2.15 domainNumber (UInteger8)

This field is specified as the gPTP domain number (see 8.1).

13.3.1.2.16 majorSdoId (Nibble)

The value is the same as the value specified in 8.1 for all transmitted PTP messages of a gPTP domain. Any TIMESYNC message received for which the value is not one of the values specified in 8.1 shall be ignored.

NOTE—The nibble that immediately follows majorSdoId is reserved (see 10.6.1).

13.3.1.2.17 minorSdoId (UInteger8)

The value is the same as the value specified in 8.1 for all transmitted PTP messages of a gPTP domain. Any TIMESYNC message received for which the value is not one of the values specified in 8.1 shall be ignored.

13.3.1.2.18 reserved

The reserved field that follows minorSdoId has variable length. The field shall have zero length on transmission. Any bytes between the minorSdoId and the FCS field shall be ignored on reception.

13.3.1.2.19 FCS (Octet4)

This field is the Frame Check Sequence (see 57B.1.1 of IEEE 802.3-2018).

13.4 Determination of asCapable

The default value of the per-port, per-domain global variable asCapable shall be TRUE.

The per-port, per-domain global variable asCapable shall be set to TRUE if the value of neighborGptpCapable for this port is TRUE.

NOTE—The above conditions ensure backward compatibility with the 2011 edition of this standard. A time-aware system that is compliant with the 2011 edition of this standard will not process the gPTP capable TLV, and asCapable will be determined as specified in the 2011 edition. A PTP Instance of a time-aware system compliant with the current edition of this standard that is attached, via an EPON link, to a node compliant with the 2011 edition of this standard will not receive Signaling messages that contain the gPTP capable TLV and will not set neighborGptpCapable to TRUE. However, the above ensures that asCapable for this port and domain (i.e., domain 0) will still be set in a manner consistent with that of the 2011 edition of this standard, because the default value of asCapable is TRUE in that edition.Layering for IEEE 802.3 EPON links

The MD entity is media-dependent and is responsible for translating the media-independent layer to media-dependent PDUs or primitives as necessary for communicating synchronized time over the EPON link from the OLT to a single ONU. This implies that if one OLT port is associated with multiple ONUs, it will require one IEEE 802.1AS PortSync entity and one MD entity per associated ONU. The OSSPDU primitives are used to communicate synchronized time information. Figure 13-2 illustrates how the MD entity interacts with the OSSP sublayer.

13.5 Service interface definitions

13.5.1 OSSPDU.request

13.5.1.1 General

This service interface primitive is generated periodically by the MD entity of the clock master every sync interval (see 10.7.2.1). It triggers transmission of a TIMESYNC message from the clock master to the clock slave. The values of the parameters of the primitive are sent to the clock slave via the TIMESYNC message.

13.5.1.2 OSSPDU.request parameters

OSSPDU.request {
    X
    $ToD_{X,i}$
    sourcePortIdentity
    logMessageInterval
    rateRatio
    gmTimeBaseIndicator
    lastGmPhaseChange
    scaledLastGmFreqChange

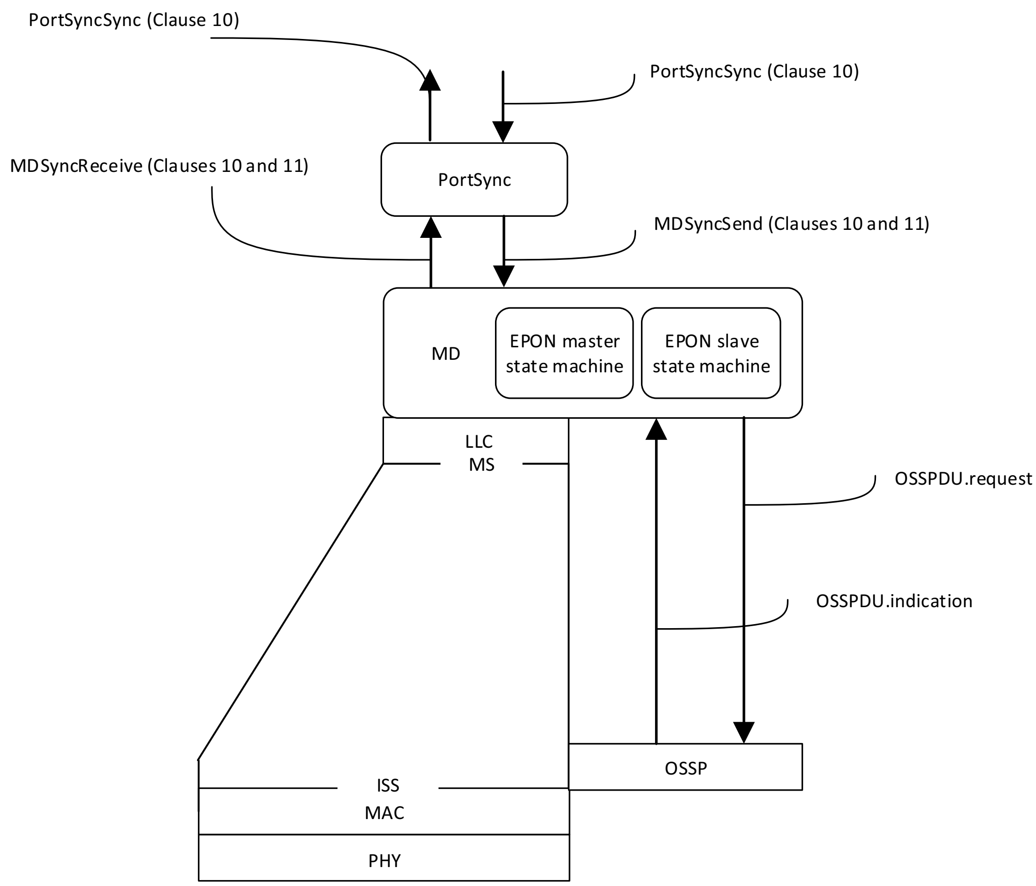


Figure 13-2—IEEE 802.3 EPON interface model

    domainNumber
}

The parameter definitions are as follows:

13.5.1.3 *X* (Integer32)

The *X* field is the selected timestamp that will be used as the timing reference as specified in 13.1.4.

13.5.1.4 $ToD_{X,i}$ (Timestamp)

$ToD_{X,i}$ is the synchronized time when the MPCP counter at the clock slave *i* reaches a value equal to *X* minus the *onuLatencyFactor* (see 13.1.4). *X* is carried in the respective TIMESYNC message. Synchronization of the MPCP clock is described in detail in 64.2.1.1 and 77.2.1.1 in IEEE Std 802.3-2018, for 1G-EPON and 10G-EPON, respectively.

13.5.1.5 sourcePortIdentity (PortIdentity)

This parameter identifies the sourcePortIdentity value for this port (see 13.3.1.2.9).

13.5.1.6 logMessageInterval (Integer8)

This parameter identifies the currentLogSyncInterval value for this port (see 13.3.1.2.10).

13.5.1.7 rateRatio (Float64)

This parameter identifies the rateRatio value for this port (see 13.3.1.2.11).

13.5.1.8 gmTimeBaseIndicator (UInteger16)

This parameter identifies the gmTimeBaseIndicator value for this port (see 13.3.1.2.12).

13.5.1.9 lastGmPhaseChange (ScaledNs)

This parameter identifies the lastGmPhaseChange value for this port (see 13.3.1.2.13).

13.5.1.10 scaledLastGmFreqChange (Integer32)

This parameter identifies the scaledLastGmFreqChange value for this port (see 13.3.1.2.14).

13.5.1.11 domainNumber

This parameter identifies the domainNumber for this instance of gPTP (see 13.3.1.2.15).

13.5.1.12 sdoId

This parameter identifies the sdoId for this instance of gPTP (see 13.3.1.2.16 and 13.3.1.2.17).

13.5.1.13 When generated

This primitive is generated by the clock master every $2^{\text{currentLogSyncInterval}}$ seconds when it is in the MASTER state, as the first phase of synchronized time information transfer.

13.5.1.14 Effect of receipt

Upon receipt of this primitive, a TIMESYNC message is enqueued for transmission.

NOTE—Arrival of the TIMESYNC message at the ONU after the selected time *X* does not impede proper operation of the synchronization mechanism defined in this clause.

13.5.2 OSSPDU.indication

13.5.2.1 General

This service interface primitive is generated on receipt of a TIMESYNC message by the responder, and provides the values contained in the corresponding OSSPDU.request primitive to the clock slave.

13.5.2.2 OSSPDU.indication parameters

OSSPDU.indication {
    X
    $ToD_{X,i}$
    sourcePortIdentity
    logMessageInterval

rateRatio
gmTimeBaseIndicator
lastGmPhaseChange
scaledLastGmFreqChange
domainNumber
sdoId
}

The parameters of the OSSPDU.indication are set equal to the corresponding fields of the most recently received TIMESYNC message. Their definitions are given in 13.3.1.2.7 through 13.3.1.2.15, respectively.

13.5.2.3 When generated

This primitive is generated by the receipt of a TIMESYNC message during the phase of synchronized time information transfer.

13.5.2.4 Effect of receipt

Upon receipt, the OSSPDU.indication parameters are used by the MD entity to compute the parameters of the MDSyncReceive structure that will be transmitted to the PortSync entity of this port.

13.6 MD entity global variables

13.6.1 RTT_i**:** is used only by the OLT MD entity. RTT_i is the RTT between the clock master and clock slave. The data type for RTT_i is UInteger32.

NOTE—RTT is measured and updated by the MPCP using the mechanism specified in IEEE Std 802.3-2018, and stored in RTT_i when measured and updated. RTT_i is not used by the ONU, and is set to zero in an ONU MD entity.

13.7 State machines

13.7.1 Requester state machine

13.7.1.1 Function

This state machine generates and consumes primitives, at the requester, used to provide accurate synchronized time across EPON links to the responder.

13.7.1.2 State machine variables

The following variables are used in the state diagram of 13.7.1.4:

13.7.1.2.1 ndown: the effective index of the light propagating in the downstream channel. The data type for ndown is Float64.

13.7.1.2.2 nup: the effective index of the light propagating in the upstream channel. The data type for ndown is Float64.

13.7.1.2.3 rcvdMDSyncEponReq:a Boolean variable that notifies the current state machine when an MDSyncSend structure is received. This variable is reset by the current state machine.

13.7.1.2.4 rcvdMDSyncPtrEponReq: a pointer to the received MDSyncSend structure.

13.7.1.2.5 registered: a Boolean variable that indicates an ONU has registered to EPON.

13.7.1.2.6 $ToD_{X,i}$: the synchronized time when the MPCP counter at the clock slave *i* reaches a value equal to *X* (see 13.7.1.2.7) minus the *onuLatencyFactor* (see 13.1.4). The data type for $ToD_{X,i}$ is Timestamp.

13.7.1.2.7 $ToD_{X,o}$: the synchronized time when the MPCP counter at the clock master reaches a value equal to *X* (see 13.7.1.2.7) plus the *oltLatencyFactor* (see 13.1.4). The data type for $ToD_{X,o}$ is Timestamp.

13.7.1.2.8 *X*: the value of the timestamp [see 13.1.4a)] that is selected as the reference time. The data type for *X* is UInteger32.

13.7.1.3 State machine functions

The following function is used in the state diagram of 13.7.1.4:

13.7.1.3.1 setToDXo(): computes the state machine variable $ToD_{X,o}$ (see 13.7.1.2.7) as the sum of:

a) The preciseOriginTimestamp member of the most recently received MDSyncSend structure,
b) The followUpCorrectionField of the most recently received MDSyncSend structure, and
c) The quantity

$$\text{rateRatio} \times (X \times (16\ \text{ns}) - \text{upstreamTxTime}) \quad (13\text{-}4)$$

where rateRatio and upstreamTxTime are the rateRatio and upstreamTxTime members, respectively, of the most recently received MDSyncSend structure, and *X* is defined in 13.7.1.2.8 (see 13.7.1.2.7).

13.7.1.4 State diagram

The requester state machine shall implement the function specified by the state diagram in Figure 13-3, the local variables specified in 13.7.1.2, the service interface primitives specified in 13.5, the structure specified in 10.2.2.1, the message specified in 13.3, and the relevant global variables specified in 10.2.5 and 13.6. The state machine receives an MDSyncSend structure from the PortSyncSyncSend state machine of the PortSync entity of this port and transmits an OSSPDU.request primitive to cause a TIMESYNC message to be sent to the responder (ONU).

13.7.2 Responder state machine

13.7.2.1 Function

This state machine responds to EPON-specific primitives generated by receipt of a TIMESYNC message from the requester.

13.7.2.2 State machine variables

The following variables are used in the state diagram of 13.7.2.4:

13.7.2.2.1 rcvdOSSPDUind: a Boolean variable that notifies the responder state machine when a TIMESYNC message is received and the OSSPDU.indication primitive is generated.

13.7.2.2.2 txMDSyncReceivePtrEponResp: a pointer to a structure whose members contain the values of the parameters of an MDSyncReceive structure to be transmitted.

13.7.2.2.3 rcvdOSSPDUptr: a pointer to a structure whose members contain the values of the parameters of the OSSPDU.indication primitive whose receipt is indicated by rcvdOSSPDUind (see 13.7.2.2.1).

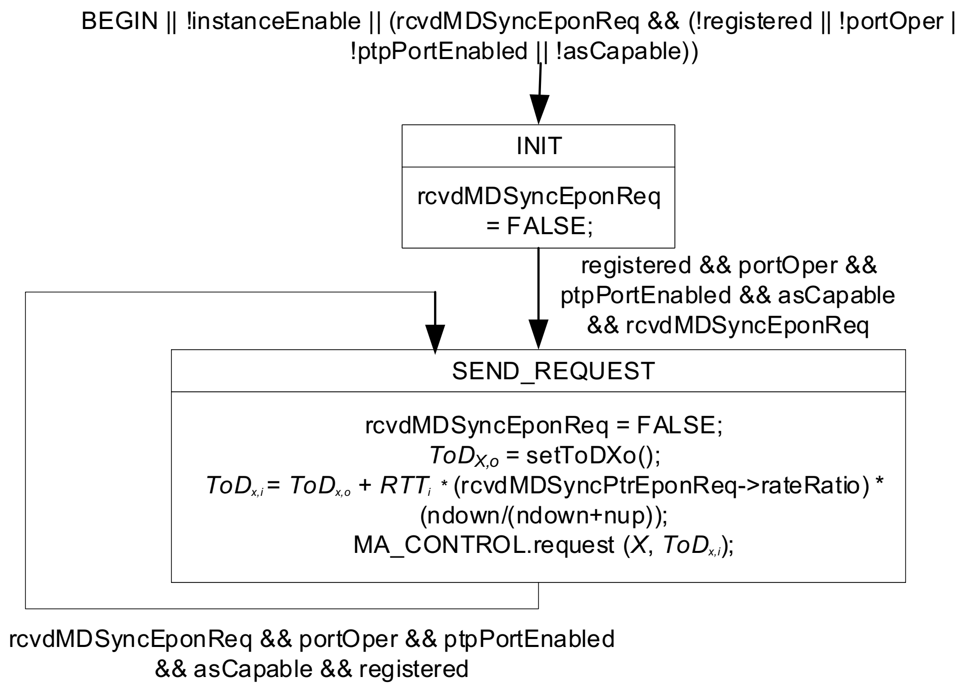


Figure 13-3—State machine for IEEE 802.3 EPON requester

13.7.2.3 State machine functions

The following functions are used in the state diagram of 13.7.2.4:

13.7.2.3.1 setMDSyncReceiveEponResp(): creates an MDSyncReceive structure (see 10.2.2.2) using members of the structure pointed to by rcvdOSSPDUptr (see 13.7.2.2.3), and returns a pointer to this structure. The members of this structure are set as follows:

a) followUpCorrectionField is set equal to 0,
b) sourcePortIdentity is set equal to an 8-byte clockIdentity plus a 2-byte portNumber. The 8-byte clockIdentity is generated by mapping the 6 byte Source Address (see 13.3.1.2.2) of the most recently received TIMESYNC message, which is an EUI-48, to an EUI-64 format (see 8.5.2.2). The 2-byte portNumber is set equal to 1,
c) logMessageInterval is set equal to the logMessageInterval of the most recently received TIMESYNC message (see 13.3.1.2.10),
d) preciseOriginTimestamp is set equal to the $ToD_{X,i}$ field of the most recently received TIMESYNC message (see 13.3.1.2.8),
e) rateRatio is set to the rateRatio of the most recently received TIMESYNC message (see 13.3.1.2.11),
f) upstreamTxTime is set equal to X multiplied by 16 ns, where X is the value of the X field of the most recently received TIMESYNC message (see 13.3.1.2.7),
g) gmTimeBaseIndicator is set equal to the gmTimeBaseIndicator of the most recently received TIMESYNC message (see 13.3.1.2.12),
h) lastGmPhaseChange is set equal to the lastGmPhaseChange of the most recently received TIMESYNC message (see 13.3.1.2.13),

i) lastGmFreqChange is set equal to the scaledLastGmFreqChange of the most recently received TIMESYNC message (see 13.3.1.2.14), divided by 2^{41}, and
j) domainNumber is set equal to the domainNumber of the most recently received TIMESYNC message (see 13.3.1.2.15).

13.7.2.3.2 txMDSyncReceive (txMDSyncReceivePtrEponResp): transmits an MDSyncReceive structure to the PortSyncSyncReceive state machine of the PortSync entity of this port.

13.7.2.4 State diagram

The responder state machine shall implement the function specified by the state diagram in Figure 13-4, the local variables specified in 13.7.2.2, the functions specified in 13.7.2.3, the service interface primitives specified in 13.5, the structure specified in 10.2.2.2, the message specified in 13.3, and the relevant global variables specified in 10.2.5 and 13.6. The state machine receives an OSSPDU.indication primitive in response to its having received a TIMESYNC message from the requester (OLT), and transmits an MDSyncReceive structure to the PortSync entity of this port.

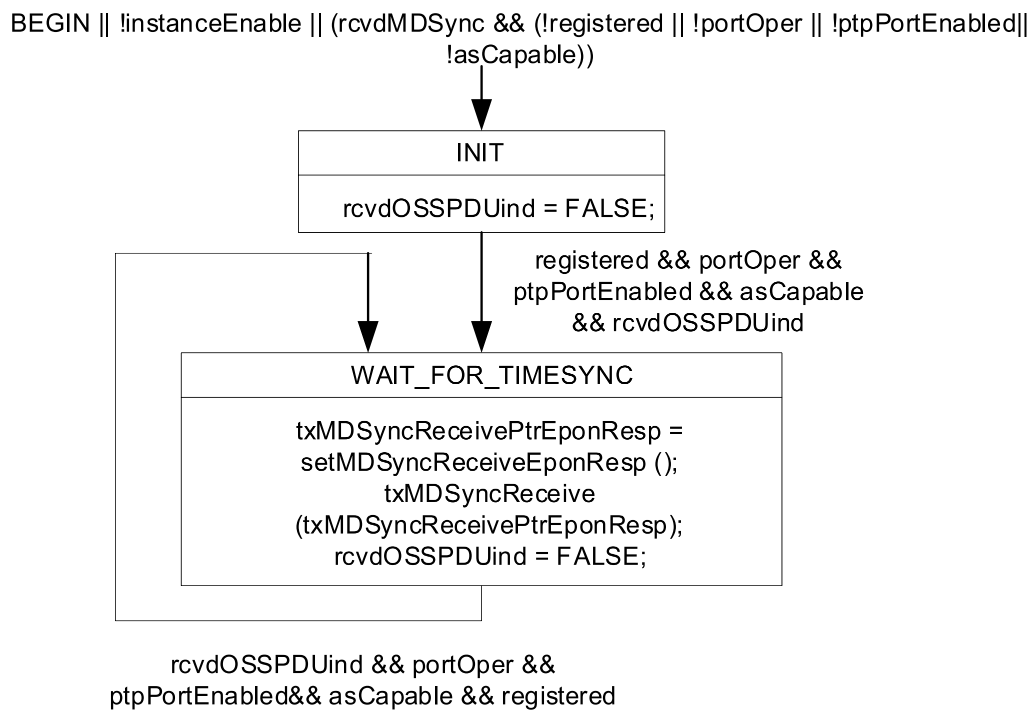


Figure 13-4—State machine for IEEE 802.3 EPON responder

13.8 Message transmission intervals

13.8.1 General interval specification.

The mean time interval between successive TIMESYNC messages shall be as specified in 10.7.2.1, 10.7.2.3, and 13.8.2.

13.8.2 TIMESYNC message transmission interval default value

The default value of initialLogSyncInterval (see 10.7.2.4) is –3. Every port supports the value 127; the port does not send TIMESYNC messages when currentLogSyncInterval has this value. A port may support other values, except for the reserved values indicated in Table 10-16.

Processing of the message interval request TLV carried in a Signaling message (see 10.6.4) shall be supported, as specified by the SyncIntervalSetting state machine of 11.2.2110.3.18 (see Figure 10-20), except that: the logLinkDelayInterval (which is not relevant to IEEE 802.3 EPON ports) is set to 128 by the sender of the Signaling message, the logLinkDelayInterval is ignored by the receiver, and unsupported values of logTimeSyncInterval are ignored by the receiver.

14. Timing and synchronization management

14.1 General

This clause defines the set of managed objects, and their functionality, that allow administrative configuration of clock parameters and timing and synchronization protocols.

Management data models typically represent data for the physical device (i.e. time-aware system). The specifications for discovery, management address, and security for the physical device are typically covered by standards of the management mechanism, which are outside the scope of this standard. For the management information model of this standard, the scope of work is the data contained within a time-aware system. From a management perspective, the time-aware system contains a list of one or more PTP Instances. Each entry in the list is a set of managed data sets for the respective PTP Instance.

Conformance for each managed object is optional. This standard operates correctly using default values, and therefore management is not essential. Since the management mechanism is not limited to remote protocols (e.g., SNMP, NETCONF, etc.), management can use a local mechanism with a simple interface (e.g. DIP switches). Therefore, each product can determine the support of managed objects as appropriate for its management mechanism.

The following hierarchy summarizes the managed data sets within a gPTP Node:

A) instanceList[]
  a) defaultDS
  b) currentDS
  c) parentDS
  d) timePropertiesDS
  e) pathTraceDS
  f) acceptableMasterTableDS
  g) portList[]
    1) portDS
    2) descriptionPortDS
    3) portStatisticsDS
    4) acceptableMasterPortDS
    5) externalPortConfigurationPortDS
    6) asymmetryMeasurementModeDS
    7) commonServicesPortDS

B) commonServices
  a) commonMeanLinkDelayService
    1) cmldsDefaultDS
    2) cmldsLinkPortList[]
      i) cmldsLinkPortDS
      ii) cmldsLinkPortStatisticsDS
      iii) cmldsAsymmetryMeasurementModeDS
  b) future common services can follow

The instanceList is indexed using a number that is unique per PTP Instance within the time-aware system, applicable to the management context only (i.e. not used in PTP messages). The domainNumber of the PTP Instance must not be used as the index to instanceList, since it is possible for a time-aware system to contain multiple PTP Instances using the same domainNumber. The portList is indexed using a number that is unique per logical port (i.e., PTP Port) in the PTP Instance (see 8.5.1). Since the portNumber of a logical port can have any value in the range 1, 2, 3, ..., 0xFFFE (see 8.5.2.3), the portList index and portNumber values for a logical port will not necessarily be the same. PTP Instances and logical ports may be created or deleted dynamically in implementations that support dynamic create/delete of devices. Unless otherwise indicated, the data sets and managed objects under the instanceList[] are maintained separately for each PTP Instance supported by the time-aware system.

Following the instanceList[] and all the data sets of each instanceList[] member is an overall structure for common services. That structure contains one sub-structure for each common service. At present there is only one common service, namely the Common Mean Link Delay Service, and the corresponding sub-structure is the commonMeanLinkDelayService structure. The item “future common services can follow” is a placeholder for any common services that might be defined in the future. The commonMeanLinkDelayService structure contains the data sets and lists that are needed by the Common Mean Link Delay Service.

The commonMeanLinkDelayService structure contains the cmldsLinkPortList, which is a list of CMLDS logical ports, i,e, Link Ports (see 11.2.17), of the time-aware system that will run the common service. The CMLDS must be implemented (i.e., a CMLDS executable must be present) on every physical port for which there is a PTP Port of a PTP Instance that can use the CMLDS (i.e., where portDS.delayMechanism of that PTP Instance can have the value COMMON_P2P). Therefore, the cmldsLinkPortList[] must include Link Ports that correspond to all such physical ports. As is the case for the portList of a PTP Instance, the cmldsLinkPortList is indexed using a number that is unique per Link Port that invokes the CMLDS (see 8.5.1). Since the portNumber of a logical port (i.e., PTP port or CMLDS Link Port) can have any value in the range 1, 2, 3, ..., 0xFFFE (see 8.5.2.3), the cmldsLinkPortList index and cmldsLinkPortDS.portIdentity.portNumber values for a logical port of the Common Mean Link Delay Service will not necessarily be the same. CMLDS Link Ports may be created or deleted dynamically in implementations that support dynamic create/delete of devices.

The Common Mean Link Delay Service Data Sets are not maintained separately for each PTP Instance. Rather, a single copy of the commonServices.cmldsDefaultDS is maintained for the time-aware system, and a single copy of each data set under the cmldsLinkPortList[] is maintained per port of the time-aware system.

A PTP Instance can use the commonServicesPortDS to determine which Link Port it must use when it obtains information provided by the Common Mean Link Delay Service (see 14.14).

NOTE—This hierarchy is intended to support a wide variety of time-aware system implementations. Some examples include a) a time-aware system containing four PTP Relay Instances, each of which use the same physical ports, but different domainNumber values, and b) a time-aware system that represents a chassis with slots for switch/router cards, where each switch/router card is represented as a PTP Instance using distinct physical ports, and all PTP Instances can use the same domainNumber.

The objects that comprise this management resource are as follows:

a) The Default Parameter Data Set (defaultDS in the above hierarchy, see Table 14-1), which represents the native capabilities of a PTP Instance, i.e., a PTP Relay Instance or a PTP End Instance station;
b) The Current Parameter Data Set (currentDS in the above hierarchy, see Table 14-2), which represents the position of a local system and other information, relative to the grandmaster;
c) The Parent Parameter Data Set (parentDS in the above hierarchy, see Table 14-3), which represents capabilities of the up-stream system, toward the grandmaster, as measured at a local system;
d) The Time Properties Parameter Data Set (timePropertiesDS in the above hierarchy, see Table 14-4), which represents capabilities of the grandmaster, as measured at a local system;
e) The Path Trace Parameter Data Set (pathTraceDS in the above hierarchy, see Table 14-5), which represents the current path trace information (see 10.3.9.23) available at the PTP Instance;
f) The Acceptable Master Table Parameter Data Set (acceptableMasterTableDS in the above hierarchy, see Table 14-6), which represents the acceptable master table used when an EPON port is used by a PTP Instance of a time-aware system;

g) The Port Parameter Data Set (portDS in the above hierarchy, see Table 14-10), which represents time-aware capabilities at a given PTP Relay Instance or PTP End Instance port;
h) The Description Port Parameter Data Set (descriptionPortDS in the above hierarchy, see Table 14-11), which contains the profileIdentifier for this PTP profile as specified in F.1;
i) The Port Parameter Statistics Data Set (portStatisticsDS in the above hierarchy, see Table 14-12), which represents statistics and counters associated with time-aware capabilities at a given PTP Relay Instance or PTP End Instance port;
j) The Acceptable Master Port Parameter Data Set (acceptableMasterPortDS in the above hierarchy, see Table 14-13), which represents the capability to enable/disable the acceptable master table feature on a port;
k) The External Port Configuration Port Parameter Data Set (externalPortConfigurationPortDS, see Table 14-14), which is used with the external port configuration option to indicate the desired state of a gPTP port.
l) The Asymmetry Measurement Mode Parameter Data Set (asymmetryMeasurementModeDS, see Table 14-15), which represents the capability to enable/disable the Asymmetry Compensation Measurement Procedure on a port (see Annex G), and is used instead of the cmldsAsymmetryMeasurementModeDS when CMLDS is not used and there is a single gPTP domain;
m) The Common Services Port Parameter Data Set (commonServicesPortDS, see Table 14-16), which enables a gPTP Port of a PTP Instance to determine which port of the respective common service corresponds to that gPTP Port.
n) The Common Mean Link Delay Service Default Parameter Data Set (cmldsDefaultDS in the above hierarchy, see Table 14-18), which describes the per-time-aware-system attributes of the Common Mean Link Delay Service;
o) The Common Mean Link Delay Service Link Port Parameter Data Set (cmldsLinkPortDS, see Table 14-18), which represents time-aware Link Port capabilities for the Common Mean Link Delay Service of a time-aware system;
p) The Common Mean Link Delay Service Link Port Parameter Statistics Data Set (cmldsLinkPortStatisticsDS, see Table 14-19), which represents statistics and counters associated with Link Port capabilities at a given time-aware system; and
q) The Common Mean Link Delay Service Asymmetry Measurement Mode Parameter Data Set (cmldsAsymmetryMeasurementModeDS, see Table 14-20), which represents the capability to enable/disable the Asymmetry Compensation Measurement Procedure on a Link Port (see Annex G).

NOTE—portDS, descriptionPortDS, portStatisticsDS, and acceptableMasterPortDS correspond to a logical PTP Port of a PTP Instance; a PTP Relay Instance or PTP End Instance physical port can contain one or more logical ports (see 8.5.1). For example, a PTP Relay Instance physical port can be connected to a full-duplex, point-to-point link that contains one logical port. As another example, a PTP Relay Instance physical port can be connected to a CSN link that contains more than one logical port.

14.2 Default Parameter Data Set (defaultDS)

The defaultDS represents the native capabilities of a PTP Instance, i.e., a PTP Relay Instance or a PTP End Instance.

14.2.1 clockIdentity

The value is the clockIdentity (see 8.5.2.2) of the local clock.

14.2.2 numberPorts

The value is the number of ports of the PTP Instance (see 8.6.2.8). For an end station the value is 1.

14.2.3 clockQuality

This is a structure whose data type is ClockQuality (see 6.4.3.8).

14.2.3.1 clockQuality.clockClass

The value is the clockClass of the PTP Instance, which implements the clockClass specifications of 8.6.2.2.

14.2.3.2 clockQuality.clockAccuracy

The value is the clockAccuracy of the PTP Instance, which implements the clockAccuracy specifications of 8.6.2.3.

14.2.3.3 clockQuality.offsetScaledLogVariance

The value is the offsetScaledLogVariance of the PTP Instance, which implements the offsetScaledLogVariance specifications of 8.6.2.4.

14.2.4 priority1

The value is the priority1 attribute of the PTP Instance (see 8.6.2.1).

14.2.5 priority2

The value is the priority2 attribute of the PTP Instance (see 8.6.2.5).

14.2.6 gmCapable

The value is TRUE if the PTP Instance is capable of being a grandmaster, and FALSE if the PTP Instance is not capable of being a grandmaster.

14.2.7 currentUtcOffset

The value is the offset between TAI and UTC, relative to the ClockMaster entity of this PTP Instance. It is equal to the global variable sysCurrentUtcOffset (see 10.3.9.18). The value is in units of seconds.

The default value is selected as follows:

a) The value is the value obtained from a primary reference if the value is known at the time of initialization, else
b) The value is the current IERS defined value of TAI - UTC (see IERS Bulletin C) when the PTP Instance is designed.

14.2.8 currentUtcOffsetValid

The value is TRUE if the currentUtcOffset, relative to the ClockMaster entity of this PTP Instance, is known to be correct. It is equal to the global variable sysCurrentUtcOffsetValid (see 10.3.9.14).

The default value is TRUE if the value of currentUtcOffset is known to be correct, otherwise it is set to FALSE.

14.2.9 leap59

A TRUE value indicates that the last minute of the current UTC day, relative to the ClockMaster entity of this PTP Instance, will contain 59 s. It is equal to the global variable sysLeap59 (see 10.3.9.13).

The value is selected as follows:

a) The value is obtained from a primary reference if known at the time of initialization, else
b) The value is set to FALSE.

14.2.10 leap61

A TRUE value indicates that the last minute of the current UTC day, relative to the ClockMaster entity of this PTP Instance, will contain 61 s. It is equal to the global variable sysLeap59 (see 10.3.9.12).

The value is selected as follows:

a) The value is obtained from a primary reference if known at the time of initialization, else
b) The value is set to FALSE.

14.2.11 timeTraceable

The value is set to TRUE if the timescale and the value of currentUtcOffset, relative to the ClockMaster entity of this PTP Instance, are traceable to a primary reference standard; otherwise the value is set to FALSE. It is equal to the global variable sysTimeTraceable (see 10.3.9.16).

The value is selected as follows:

a) If the time and the value of currentUtcOffset are traceable to a primary reference standard at the time of initialization, the value is set to TRUE, else
b) The value is set to FALSE.

14.2.12 frequencyTraceable

The value is set to TRUE if the frequency determining the timescale of the ClockMaster Entity of this PTP Instance is traceable to a primary standard; otherwise the value is set to FALSE. It is equal to the global variable sysFrequencyTraceable (see 10.3.9.17).

The value is selected as follows:

a) If the frequency is traceable to a primary reference standard at the time of initialization the value is set to TRUE, else
b) The value is set to FALSE.

14.2.13 ptpTimescale

The value is set to TRUE if the clock timescale of the ClockMaster Entity of this PTP Instance is PTP (see 8.2) and FALSE otherwise.

14.2.14 timeSource

The value is the source of time used by the grandmaster clock (see 8.6.2.7).

14.2.15 domainNumber

The value is the domain number of the gPTP domain for this instance of gPTP supported by the time-aware system (see 8.1).

14.2.16 sdoId

The value is the sdoId of the gPTP domain for this instance of gPTP supported by the time-aware system (see 8.1).

NOTE—The attribute sdoId is specified as a 12-bit unsigned integer in 8.1. The data type for the managed object sdoId is UInteger16 in Table 14-1, for compatibility with IEEE Std 1588. The range of the managed object is limited to 12 bits; in addition, only the single value 0x100 is specified in this standard for the gPTP domain of a PTP Instance.

14.2.17 externalPortConfigurationEnabled

The value is the externalPortConfigurationEnabled attribute of the PTP Instance (see 10.3.9.24).

14.2.18 instanceEnable

The value is the instanceEnable attribute of the PTP Instance (see 10.2.4.24).

14.2.19 defaultDS Table

There is one Default Parameter Table per PTP Instance of a time-aware system, as detailed in Table 14-1.

Table 14-1—defaultDS Table

Name	Data type	Operations supported[a]	References
clockIdentity	ClockIdentity	R	14.2.1
numberPorts	UInteger16	R	14.2.2
clockQualtiy.clockClass	UInteger8	R	14.2.3.1 IEEE Std 1588-2019, 7.6.2.5
clockQuality.clockAccuracy	Enumeration8	R	14.2.3.2 IEEE Std 1588-2019, 7.6.2.6
clockQuality.offsetScaledLogVariance	UInteger16	R	14.2.3.3
priority1	UInteger8	RW	14.2.4
priority2	UInteger8	RW	14.2.5
gmCapable	Boolean	R	14.2.6
currentUtcOffset	Integer16	RW	14.2.7
currentUtcOffsetValid	Boolean	RW	14.2.8

Table 14-1—defaultDS Table

Name	Data type	Operations supported[a]	References
leap59	Boolean	RW	14.2.9
leap61	Boolean	RW	14.2.10
timeTraceable	Boolean	R	14.2.11
frequencyTraceable	Boolean	R	14.2.12
ptpTimescale	Boolean	R	14.2.13
timeSource	TimeSource	R	14.2.14 and Table 8-2
domainNumber	UInteger8	RW	14.2.15
sdoId	UInteger16	R	14.2.16
externalPortConfigurationEnabled	Boolean	RW	14.2.17
instanceEnable	Boolean	RW	14.2.18

[a]R = Read only access; RW = Read/write access.

14.3 Current Parameter Data Set (currentDS)

The currentDS represents the position of a local system and other information, relative to the grandmaster.

14.3.1 stepsRemoved

The value is the number of gPTP communication paths traversed between the local clock and the grandmaster clock, as specified in 10.3.3.

The default value is 0.

NOTE—For example, stepsRemoved for a slave clock on the same gPTP communication path as the grandmaster clock will have a value of 1, indicating that a single path was traversed.

14.3.2 offsetFromMaster

The value is an implementation-specific representation of the current value of the time difference between a slave and the grandmaster, as computed by the slave, and as specified in 10.2.10. It is recommended that the data type be scaledNs. The default value is implementation specific.

14.3.3 lastGmPhaseChange

The value (see 10.2.4.16) is the phase change that occurred on the most recent change in either grandmaster or gmTimeBaseIndicator (see 9.2.2.3).

14.3.4 lastGmFreqChange

The value (see 10.2.4.17) is the frequency change that occurred on the most recent change in either grandmaster or gmTimeBaseIndicator (see 9.2.2.3).

14.3.5 gmTimebaseIndicator

The value is the value of timeBaseIndicator of the current grandmaster (see 9.2.2.3 and 9.6.2.3).

14.3.6 gmChangeCount

This statistics counter tracks the number of times the grandmaster has changed in a gPTP domain. This counter increments when the PortAnnounceInformation state machine enters the SUPERIOR_MASTER_PORT state or the INFERIOR_MASTER_OR_OTHER_PORT state (see 10.3.12 and Figure 10-14).

14.3.7 timeOfLastGmChangeEvent

This timestamp takes the value of sysUpTime (see RFC3418) when the most recent grandmaster change occurred in a gPTP domain. This timestamp is updated when the PortAnnounceInformation state machine enters the SUPERIOR_MASTER_PORT state or the INFERIOR_MASTER_OR_OTHER_PORT state (see 10.3.12 and Figure 10-14).

14.3.8 timeOfLastGmPhaseChangeEvent

This timestamp takes the value of sysUpTime (see RFC3418) when the most recent change in grandmaster phase occurred, due to a change of either the grandmaster or the grandmaster time base. This timestamp is updated when one of the following occurs:

a) The PortAnnounceInformation state machine enters the SUPERIOR_MASTER_PORT state or the INFERIOR_MASTER_OR_OTHER_PORT state (see 10.3.12 and Figure 10-14), or
b) The gmTimebaseIndicator managed object (see 14.3.5) changes and the lastGmPhaseChange field of the most recently received Follow_Up information TLV is nonzero.

14.3.9 timeOfLastGmFreqChangeEvent

This timestamp takes the value of sysUpTime (see RFC3418) when the most recent change in grandmaster frequency occurred, due to a change of either the grandmaster or the grandmaster time base. This timestamp is updated when one of the following occurs:

a) The PortAnnounceInformation state machine enters the SUPERIOR_MASTER_PORT state or the INFERIOR_MASTER_OR_OTHER_PORT state (see 10.3.12 and Figure 10-14), or
b) The gmTimebaseIndicator managed object (see 14.3.5) changes and the lastGmFreqChange field of the most recently received Follow_Up information TLV is nonzero.

14.3.10 currentDS Table

There is one currentDS Table per PTP Instance of a time-aware system, as detailed in Table 14-2.

14.4 Parent Parameter Data Set (parentDS)

The parentDS represents capabilities of the upstream system, toward the grandmaster, as measured at a local system.

14.4.1 parentPortIdentity

If this PTP Instance is the grandmaster, the value is a portIdentity whose clockIdentity is the clockIdentity of this PTP Instance, and whose portNumber is 0.

If this PTP Instance is not the grandmaster, the value is the portIdentity of the MasterPort (see Table 10-7) of the gPTP communication path attached to the single slave port of this PTP Instance.

Table 14-2—currentDS Table

Name	Data type	Operations supported[a]	References
stepsRemoved	UInteger16	R	14.3.1
offsetFromMaster	TimeInterval	R	14.3.2
lastGmPhaseChange	ScaledNs	R	14.3.3
lastGmFreqChange	Float64	R	14.3.4
gmTimebaseIndicator	UInteger16	R	14.3.5
gmChangeCount	UInteger32	R	14.3.6
timeOfLastGmChangeEvent	UInteger32 (sysUp Time, RFC 3418)	R	14.3.7
timeOfLastGmPhaseChangeEvent	UInteger32 (sysUp Time, RFC 3418)	R	14.3.8
timeOfLastGmFreqChangeEvent	UInteger32 (sysUp Time, RFC 3418)	R	14.3.9

[a]R = Read only access; RW = Read/write access.

The default value is a portIdentity for which:

a) The clockIdentity member is the value of the clockIdentity member of the default data set, and
b) The portNumber member is 0.

14.4.2 cumulativeRateRatio

The value is an estimate of the ratio of the frequency of the grandmaster to the frequency of the LocalClock entity of this PTP Instance. cumulativeRateRatio is expressed as the fractional frequency offset multiplied by 2^{41}, i.e., the quantity (rateRatio – 1.0)(2^{41}), where rateRatio is computed by the PortSyncSyncReceive state machine (see 10.2.8.1.4).

14.4.3 grandmasterIdentity

The value is the clockIdentity attribute (see 8.5.2.2) of the grandmaster clock.

The default value is the value of defaultDS.clockIdentity (14.2.1).

14.4.4 grandmasterClockQuality

This is a structure whose data type is ClockQuality (see 6.4.3.8).

14.4.4.1 grandmasterClockQuality.clockClass

The value is the clockClass (see 8.6.2.2) of the grandmaster clock.

The default value is the clockClass member of the default data set.

14.4.4.2 grandmasterClockQuality.clockAccuracy

The value is the clockAccuracy (see 8.6.2.3) of the grandmaster clock.

The default value is the clock accuracy member of the default data set.

14.4.4.3 grandmasterClockQuality.offsetScaledLogVariance

The value is the offsetScaledLogVariance (see 8.6.2.4) of the grandmaster clock.

The default value is the offsetScaledLogVariance member of the default data set.

14.4.5 grandmasterPriority1

The value is the priority1 attribute (see 8.6.2.1) of the grandmaster clock.

The default value is the priority1 value of the default data set.

14.4.6 grandmasterPriority2

The value is the priority2 attribute (see 8.6.2.5) of the grandmaster clock.

The default value is the priority2 value of the default data set.

14.4.7 ParentParameter Data Set Table

There is one parentDS Table per PTP Instance of a time-aware system, as detailed in Table 14-3.

Table 14-3—parentDS Table

Name	Data type	Operations supported[a]	References
parentPortIdentity	PortIdentity (see 6.4.3.7)	R	14.4.1
cumulativeRateRatio	Integer32	R	14.4.2
grandMasterIdentity	ClockIdentity	R	14.4.3
grandmasterClockQuality.clockClass	UInteger8	R	14.4.4.1 IEEE Std 1588-2019, 7.6.2.5
grandmasterClockQuality.clockAccuracy	Enumeration8	R	14.4.4.2 IEEE Std 1588-2019, 7.6.2.6
grandmasterClockQuality.offsetScaledLogVariance	UInteger16	R	14.4.4.3
grandmasterPriority1	UInteger8	R	14.4.5
grandmasterPriority2	UInteger8	R	14.4.6

[a]R = Read only access; RW = Read/write access.

14.5 Time Properties Parameter Data Set (timePropertiesDS)

The timePropertiesDS represents capabilities of the grandmaster, as measured at a local system.

14.5.1 currentUtcOffset

The value is currentUtcOffset for the current grandmaster (see 14.2.7). It is equal to the value of the global variable currentUtcOffset (see 10.3.9.10). The value is in units of seconds.

14.5.2 currentUtcOffsetValid

The value is currentUtcOffsetValid for the current grandmaster (see 14.2.8). It is equal to the global variable currentUtcOffsetValid (see 10.3.9.6).

14.5.3 leap59

The value is leap59 for the current grandmaster (see 14.2.9). It is equal to the global variable leap59 (see 10.3.9.5).

14.5.4 leap61

The value is leap59 for the current grandmaster (see 14.2.10). It is equal to the global variable leap61 (see 10.3.9.4).

14.5.5 timeTraceable

The value is timeTraceable for the current grandmaster (see 14.2.11). It is equal to the global variable timeTraceable (see 10.3.9.8).

14.5.6 frequencyTraceable

The value is frequencyTraceable for the current grandmaster (see 14.2.12). It is equal to the global variable frequencyTraceable (see 10.3.9.9).

14.5.7 ptpTimescale

The value is ptpTimescale for the current grandmaster (see 14.2.13).

14.5.8 timeSource

The value is timeSource for the current grandmaster (see 14.2.14). It is equal to the global variable timeSource (see 10.3.9.11).

14.5.9 timePropertiesDS Table

There is one timePropertiesDS Table per PTP Instance of a time-aware system, as detailed in Table 14-4.

Table 14-4—timePropertiesDS Table

Name	Data type	Operations supported[a]	References
currentUtcOffset	Integer16	R	14.5.1
currentUtcOffsetValid	Boolean	R	14.5.2
leap59	Boolean	R	14.5.3
leap61	Boolean	R	14.5.4
timeTraceable	Boolean	R	14.5.5
frequencyTraceable	Boolean	R	14.5.6
ptpTimescale	Boolean	R	14.5.7
timeSource	TimeSource	R	14.5.8 and Table 8-2

[a]R = Read only access; RW = Read/write access.

14.6 Path Trace Parameter Data Set (pathTraceDS)

The pathTraceDS represents the current path trace information available at the PTP Instance.

14.6.1 list

The value is the array of ClockIdentity values contained in the pathTrace array (see 10.3.9.23), which represents the current path trace information, and which is carried in the path trace TLV (see 10.6.3.3).

The initialization value shall be the empty list (i.e., an array of length 0).

The pathTraceDS.list shall be initialized to the empty list whenever the PTP Instance updates data sets based on decision code M1 or M2 (see 9.3.5).

14.6.2 enable

The value is TRUE.

NOTE—This member is included for compatibility with IEEE Std 1588. In IEEE Std 1588, the path trace mechanism is optional, and the pathTraceDS.enable member is configurable (its value in IEEE Std 1588 is TRUE or FALSE, depending on whether the path trace mechanism is operational or not operational, respectively.. However, the pathTrace mechanism is mandatory in this standard, and the value of enable is always TRUE.

14.6.3 pathTraceDS Table

There is one Path Trace Parameter Table per PTP Instance, as detailed in Table 14-5.

Table 14-5—pathTraceDS Table

Name	Data type	Operations supported[a]	References
list	ClockIdentity[N], where N is defined in 10.3.9.23	R	14.6.1
enable	Boolean	R	14.6.2

[a]R = Read only access; RW = Read/write access.

14.7 Acceptable Master Table Parameter Data Set (acceptableMasterTableDS)

The acceptableMasterTableDS represents the acceptable master table used when an EPON port is used by a PTP Instance of a time-aware system.

14.7.1 maxTableSize

The value is the maximum size of the AcceptableMasterTable. It is equal to the maxTableSize member of the AcceptableMasterTable structure (see 13.1.3.2).

14.7.2 actualTableSize

The value is the actual size of the AcceptableMasterTable. It is equal to the actualTableSize member of the AcceptableMasterTable structure (see 13.1.3.2 and 13.1.3.5), i.e., the current number of elements in the acceptable master array. The actual table size is less than or equal to the max table size.

14.7.3 acceptableMasterArray

Each element of this array is an AcceptableMaster structure (see 13.1.3.3 and 13.1.3.5).

14.7.4 acceptableMasterTableDS Table

There is one acceptableMasterTableDS Table per PTP Instance of a time-aware system, as detailed in Table 14-6.

Table 14-6—acceptableMasterTableDS Table

Name	Data type	Operations supported[a]	References
maxTableSize	UInteger16	R	14.7.1
actualTableSize	UInteger16	RW	14.7.2
acceptableMasterArray	AcceptableMaster[actualTableSize] (see 13.1.3.3)	RW	14.7.3

[a]R = Read only access; RW = Read/write access.

14.8 Port Parameter Data Set (portDS)

The portDS represents time-aware port capabilities for a PTP Instance of a time-aware system.

14.8.1 General

For the single port of a PTP End Instance and for each port of a PTP Relay Instance , the following portDS is maintained as the basis for protocol decisions and providing values for message fields. The number of such data sets is the same as the value of defaultDS.numberPorts.

14.8.2 portIdentity

The value is the portIdentity attribute of the local port (see 8.5.2).

14.8.3 portState

The value is the value of the port state of this port (see Table 10-2) and is taken from the enumeration in Table 14-7. It is equal to the value of the global variable selectedState (see 10.2.4.20) for this port.

The default value is 3 (DisabledPort).

14.8.4 ptpPortEnabled

The value is equal to the value of the Boolean ptpPortEnabled (see 10.2.5.13).

Table 14-7—portState enumeration

State	Value
DisabledPort	3
MasterPort	6
PassivePort	7
SlavePort	9
	All other values reserved
NOTE—The enumeration values are consistent with IEEE Std 1588-2019, Table 20.	

14.8.5 delayMechanism

The value indicates the mechanism for measuring mean propagation delay and neighbor rate ratio on the link attached to this port, and is taken from the enumeration in Table 14-8. If the domain number is not 0,

Table 14-8—delayMechanism enumeration

Delay Mechanism	Value	Specification
P2P	02	The gPTP Port uses the peer-to-peer delay mechanism
COMMON_P2P	03	The gPTP Port uses the CMLDS
SPECIAL	04	The gPTP Port uses a transport that has a native time transfer mechanism and, therefore, does not use the peer-to-peer delay mechanism (e.g., IEEE Std 802.11, IEEE Std 802.3 EPON)
	All other values reserved	
NOTE—The enumeration values are consistent with P1588/D1.2v14, Table 18.		

portDS.delay mechanism must not be P2P (see 11.2.17).

14.8.6 isMeasuringDelay

The value is equal to the value of the Boolean isMeasuringDelay (see 11.2.13.6 and 16.4.3.2).

14.8.7 asCapable

The value is equal to the value of the Boolean asCapable (see 10.2.5.1).

14.8.8 meanLinkDelay

The value is equal to the value of the per-port global variable meanLinkDelay (see 10.2.5.8). It is an estimate of the current one-way propagation time on the link attached to this port, measured as specified for the respective medium (see 11.2.17, 12.5.2, and 16.4). The value is zero for ports attached to IEEE 802.3 EPON links and for the master port of an IEEE 802.11 link, because one-way propagation delay is not measured on the latter and not directly measured on the former. It is recommended that the data type be scaledNs. The default value is zero.

14.8.9 meanLinkDelayThresh

The value is equal to the value of the per-port global variable meanLinkDelayThresh (see 11.2.13.7). It is the propagation time threshold, above which a port is not considered capable of participating in the IEEE 802.1AS protocol.

14.8.10 delayAsymmetry

The value is the asymmetry in the propagation delay on the link attached to this port relative to the grandmaster time base, as defined in 10.2.5.9 and 8.3. If propagation delay asymmetry is not modeled, then delayAsymmetry is 0.

14.8.11 neighborRateRatio

The value is an estimate of the ratio of the frequency of the LocalClock entity of the PTP Instance at the other end of the link attached to this port, to the frequency of the LocalClock entity of this PTP Instance (see 10.2.5.7). neighborRateRatio is expressed as the fractional frequency offset multiplied by 2^{41}, i.e., the quantity (neighborRateRatio – 1.0)(2^{41}).

14.8.12 initialLogAnnounceInterval

If useMgtSettableLogAnnounceInterval is FALSE, the value is the logarithm to base 2 of the announce interval used when (a) the port is initialized, or (b) a message interval request TLV is received with the logAnnounceInterval field set to 126 (see 10.7.2.2 and the AnnounceIntervalSetting state machine, 10.3.17).

14.8.13 currentLogAnnounceInterval

The value is the logarithm to the base 2 of the current announce interval (see 10.7.2.2).

14.8.14 useMgtSettableLogAnnounceInterval

The managed object is a Boolean that determines the source of the announce interval. If the value is TRUE, the value of currentLogAnnounceInterval is set equal to the value of mgtSettableLogAnnounceInterval (see 14.8.15). If the value is FALSE, the value of currentLogAnnounceInterval is determined by the AnnounceIntervalSetting state machine (see 10.3.17). The default value of useMgtSettableLogAnnounceInterval is FALSE for domain 0 and TRUE for domains other than domain 0.

14.8.15 mgtSettableLogAnnounceInterval

The value is the logarithm to base 2 of the announce interval used if useMgtSettableLogAnnounceInterval is TRUE. The value is not used if useMgtSettableLogAnnounceInterval is FALSE.

14.8.16 announceReceiptTimeout

The value is the number of Announce message transmission intervals that a slave port waits without receiving an Announce message, before assuming that the master is no longer transmitting Announce messages and the BMCA needs to be run, if appropriate (see 10.7.3.2).

14.8.17 initialLogSyncInterval

If useMgtSettableLogSyncInterval is FALSE, the value is the logarithm to base 2 of the sync interval used when (a) the port is initialized, or (b) a message interval request TLV is received with the logTimeSyncInterval field set to 126 (see 10.7.2.3, 11.5.2.3, 12.8.2, 13.8.2, and the SyncIntervalSetting state machine, 10.3.18).

14.8.18 currentLogSyncInterval

The value is the logarithm to the base 2 of the current time-synchronization transmission interval (see 10.7.2.3).

14.8.19 useMgtSettableLogSyncInterval

The managed object is a Boolean that determines the source of the sync interval. If the value is TRUE, the value of currentLogSyncInterval is set equal to the value of mgtSettableLogSyncInterval (see 14.8.20). If the value of the managed object is FALSE, the value of currentLogSyncInterval is determined by the SyncIntervalSetting state machine (see 10.3.18). The default value of useMgtSettableLogSyncInterval is FALSE for domain 0 and TRUE for domains other than domain 0.

14.8.20 mgtSettableLogSyncInterval

The value is the logarithm to base 2 of the sync interval if useMgtSettableLogSyncInterval is TRUE. The value is not used if useMgtSettableLogSyncInterval is FALSE.

14.8.21 syncReceiptTimeout

The value is the number of time-synchronization transmission intervals that a slave port waits without receiving synchronization information, before assuming that the master is no longer transmitting synchronization information and that the BMCA needs to be run, if appropriate (see 10.7.3.1).

14.8.22 syncReceiptTimeoutTimeInterval

The value is equal to the value of the per-port global variable syncReceiptTimeoutTimeInterval (see 10.2.5.3). It is the time interval after which sync receipt timeout occurs if time-synchronization information has not been received during the interval.

14.8.23 initialLogPdelayReqInterval

For full-duplex, IEEE 802.3 media and CSN media that use the peer-to-peer delay mechanism to measure path delay (see 16.4.3.1), the value is the logarithm to base 2 of the Pdelay_Req message transmission interval used when (a) the port is initialized, or (b) a message interval request TLV is received with the logLinkDelayInterval field set to 126 (see 11.5.2.2 and the LinkDelaySyncIntervalSetting state machine, 11.2.21).

For all other media, the value is 127.

14.8.24 currentLogPdelayReqInterval

For full-duplex, IEEE 802.3 media and CSN media that use the peer-to-peer delay mechanism to measure path delay (see 16.4.3.1), the value is the logarithm to the base 2 of the current Pdelay_Req message transmission interval (see 11.5.2.2).

For all other media, the value is 127.

14.8.25 useMgtSettableLogPdelayReqInterval

The managed object is a Boolean that determines the source of the mean time interval between successive Pdelay_Req messages. If the value is TRUE, the value of currentLogPdelayReqInterval is set equal to the value of mgtSettableLogPdelayReqInterval (see 14.8.26). If the value of the managed object is FALSE, the value of currentLogPdelayReqInterval is determined by the LinkDelayIntervalSetting state machine (see 11.2.2111.2.21). The default value of useMgtSettableLogPdelayReqInterval is FALSE.

14.8.26 mgtSettableLogPdelayReqInterval

The value is the logarithm to base 2 of the mean time interval between successive Pdelay_Req messages if useMgtSettableLogPdelayReqInterval is TRUE. The value is not used if useMgtSettableLogPdelayReqInterval is FALSE.

14.8.27 initialLogGptpCapableMessageInterval

The value is the logarithm to base 2 of the gPTP capable message interval used when (a) the port is initialized, or (b) a gPtpCapableMessage interval request TLV is received with the logGptpCapableMessageInterval field set to 126 (see 10.6.4.5 and the GptpCapableIntervalSetting state machine, 10.4.3).

14.8.28 currentLogGptpCapableMessageInterval

The value is the logarithm to the base 2 of the current gPTP capable message interval (see 10.7.2.5).

14.8.29 useMgtSettableLogGptpCapableMessageInterval

The managed object is a Boolean that determines the source of the gPTP capable message interval. If the value is TRUE, the value of currentLogGptpCapableMessageInterval is set equal to the value of mgtSettableLogGptpCapableMessageInterval (see 14.8.30). If the value of the managed object is FALSE, the value of currentLogGptpCapableMessageInterval is determined by the GptpCapableMessageIntervalSetting state machine (see 10.4.311.2.21). The default value of useMgtSettableLogGptpCapableMessageInterval is FALSE.

14.8.30 mgtSettableLogGptpCapableMessageInterval

The value is the logarithm to base 2 of the gPtpCapableMessageInterval if useMgtSettableLogGptpCapableMessageInterval is TRUE. The value is not used if useMgtSettableLogGptpCapableMessageInterval is FALSE.

14.8.31 initialComputeNeighborRateRatio

If useMgtSettableComputeNeighborRateRatio is FALSE then, for full-duplex, IEEE 802.3 media and CSN media that use the peer-to-peer delay mechanism to measure path delay (see 16.4.3.1), the value is the initial value of computeNeighborRateRatio (see 10.2.5.10).

For all other media, the value is TRUE.

14.8.32 currentComputeNeighborRateRatio

For full-duplex, IEEE 802.3 media and CSN media that use the peer-to-peer delay mechanism to measure path delay (see 16.4.3.1), the value is the current value of computeNeighborRateRatio.

For all other media, the value is TRUE.

14.8.33 useMgtSettableComputeNeighborRateRatio

The managed object is a Boolean that determines the source of the value of computeNeighborRateRatio. If the value is TRUE, the value of computeNeighborRateRatio is set equal to the value of mgtSettableComputeNeighborRateRatio (see 14.16.17). If the value of the managed object is FALSE, the value of currentComputeNeighborRateRatio is determined by the LinkDelayIntervalSetting state machine (see 11.2.2111.2.21). The default value of useMgtSettableLogPdelayReqInterval is FALSE.

14.8.34 mgtSettableComputeNeighborRateRatio

computeNeighborRateRatio is configured to this value if useMgtSettableComputeNeighborRateRatio is TRUE. The value is not used if useMgtSettableComputeNeighborRateRatio is FALSE.

14.8.35 initialComputeMeanLinkDelay

If useMgtSettableComputeMeanLinkDelay is FALSE then, for full-duplex, IEEE 802.3 media and CSN media that use the peer-to-peer delay mechanism to measure path delay (see 16.4.3.1), the value is the initial value of computeMeanLinkDelay (see 10.2.5.10).

For all other media, the value is TRUE.

14.8.36 currentComputeMeanLinkDelay

For full-duplex, IEEE 802.3 media and CSN media that use the peer-to-peer delay mechanism to measure path delay (see 16.4.3.1), the value is the current value of computeMeanLinkDelay.

For all other media, the value is TRUE.

14.8.37 useMgtSettableComputeMeanLinkDelay

The managed object is a Boolean that determines the source of the value of computeMeanLinkDelay. If the value is TRUE, the value of computeMeanLinkDelay is set equal to the value of mgtSettableComputeMeanLinkDelay (see 14.8.38). If the value of the managed object is FALSE, the value of currentComputeMeanLinkDelay is determined by the LinkDelayIntervalSetting state machine (see 11.2.2111.2.21). The default value of useMgtSettableComputeMeanLinkDelay is FALSE.

14.8.38 mgtSettableComputeMeanLinkDelay

computeMeanLinkDelay is configured to this value if useMgtSettableComputeMeanLinkDelay is TRUE. The value is not used if useMgtSettableComputeMeanLinkDelay is FALSE.

14.8.39 allowedLostResponses

The value is equal to the value of the per-port global variable allowedLostResponses (see 11.5.3 and 11.2.13.4). It is the number of Pdelay_Req messages for which a valid response is not received, above which a port is considered to not be exchanging peer delay messages with its neighbor.

14.8.40 allowedFaults

The value is equal to the value of the per-Link-Port global variable allowedFaults (see 11.5.4 and 11.2.13.5). It is the number of faults (see 11.5.4), above which asCapableAcrossDomains is set to FALSE, i.e., a Link Port is considered to not be capable of interoperating with its neighbor via the IEEE 802.1AS protocol (see 10.2.5.1).

14.8.41 logGptpCapableMessageInterval

The value is the logarithm to the base 2 of the transmission interval between successive Signaling messages that contain the gPTP capable TLV (see 10.7.2.1 and 10.7.2.5).

14.8.42 gPtpCapableReceiptTimeout

The value is the number of transmission intervals that a port waits without receiving the gPTP capable TLV, before assuming that the neighbor port is no longer invoking the gPTP protocol (see 10.7.3.3).

14.8.43 versionNumber

This value is set to versionPTP as specified in 10.6.2.2.4.

14.8.44 nup

For an OLT port of an IEEE 802.3 EPON link, the value is the effective index of refraction for the EPON upstream wavelength light of the optical path (see 13.1.4 and 13.7.1.2.2). The default value is 1.46770 for 1 Gb/s upstream links, and 1.46773 for 10 Gb/s upstream links.

For all other ports, the value is 0.

14.8.45 ndown

For an OLT port of an IEEE 802.3 EPON link, the value is the effective index of refraction for the EPON downstream wavelength light of the optical path (see 13.1.4 and 13.7.1.2.1). The default value is 1.46805 for 1 Gb/s downstream links, and 1.46851 for 10 Gb/s downstream links.For all other ports, the value is 0.

14.8.46 oneStepTxOper

The value is equal to the value of the per-port global variable oneStepTxOper (see 11.2.13.11). Its value is TRUE if the port is sending one-step Sync messages, and FALSE if the port is sending two-step Sync and Follow-Up messages.

14.8.47 oneStepReceive

The value is equal to the value of the per-port global variable oneStepReceive (see 11.2.13.9). Its value is TRUE if the port is capable of receiving and processing one-step Sync messages.

14.8.48 oneStepTransmit

The value is equal to the value of the per-port global variable oneStepTransmit (see 11.2.13.10). Its value is TRUE if the port is capable of transmitting one-step Sync messages.

14.8.49 initialOneStepTxOper

If useMgtSettableOneStepTxOper is FALSE, the value is used to initialize currentOneStepTxOper when the port is initialized. If useMgtSettableOneStepTxOper is TRUE, the value of initialOneStepTxOper is not used.

14.8.50 currentOneStepTxOper

The value is TRUE if it is desired, either via management or via a received Signaling message, that the port transmit one-step Sync messages. The value is FALSE if it is not desired, either via management or via a received Signaling message, that the port transmit one-step Sync messages.

NOTE—The port will send one-step Sync messages only if currentOneStepTxOper and oneStepTransmit (see 14.8.48) are both TRUE (see 11.2.16 and Figure 11-8).

14.8.51 useMgtSettableOneStepTxOper

The managed object is a Boolean that determines the source of currentOneStepTxOper. If the value is TRUE, the value of currentOneStepTxOper is set equal to the value of mgtSettableOneStepTxOper (see 14.8.52). If the value is FALSE, the value of currentOneStepTxOper is determined by the OneStepTxOperSetting state machine (see 11.2.16 and Figure 11-8). The default value of useMgtSettableOneStepTxOper is TRUE.

14.8.52 mgtSettableOneStepTxOper

If useMgtSettableOneStepTxOper is TRUE, currentOneStepTxOper is set equal to the value of mgtSettableOneStepTxOper. The value of mgtSettableOneStepTxOper is not used if useMgtSettableOneStepTxOper is FALSE. The default value of mgtSettableOneStepTxOper is FALSE for domains other than domain 0.

14.8.53 syncLocked

The value is equal to the value of the per-port global variable syncLocked (see 10.2.5.15). Its value is TRUE if the port will transmit a Sync as soon as possible after the slave port receives a Sync.

14.8.54 pdelayTruncatedTimestampsArray

For full-duplex IEEE 802.3 media, and CSN media that use the peer-to-peer delay mechanism to measure path delay (see 16.4.3.1), the values of the four elements of this array are as described in Table 14-9. For all other media, the values are zero. Array elements 0, 1, 2, and 3 correspond to the timestamps t1, t2, t3, and t4, modulo 2^{32}, respectively, in Figure 11-1, and are expressed in units of 2^{-16} ns (i.e., the value of each array element is equal to the remainder obtained upon dividing the respective timestamp, expressed in units of 2^{-16} ns, by 2^{48}). At any given time, the timestamp values stored in the array are for the same, and most recently completed, peer delay message exchange.

14.8.55 minorVersionNumber

This value is set to minorVersionPTP as specified in 10.6.2.2.3.

14.8.56 portDS Table

There is one portDS Table per port, per PTP Instance of a time-aware system. Each portDS Table contains a set of parameters for each port that supports the time-synchronization capability, as detailed in Table 14-10. Each table can be created or removed dynamically in implementations that support dynamic configuration of ports and components.

14.9 Description Port Parameter Data Set (descriptionPortDS)

The descriptionPortDS contains the profileIdentifier for this PTP profile, as specified in F.1.

Table 14-9—Description of pdelayTruncatedTimestampsArray

Array element	Timestamp description	Corresponding timestamp of Figure 11-1	Units
0	pdelayReqEventEgressTimestamp for Pdelay_Req message, of most recently-completed peer delay message exchange, transmitted by this PTP Instance (NOTE 1)	t1	2^{-16} ns
1	pdelayReqEventIngressTimestamp for Pdelay_Req message received at peer delay responder that this Link Port sends Pdelay_Req to, of most recently-completed peer delay message exchange; it is equal to the sum of the following: a) the ns field of the requestReceiptTimestamp (see Table 11-13), multiplied by 2^{16}, and b) the correctionField (see Table 11-6) (NOTE 2)	t2	2^{-16} ns
2	pdelayRespEventEgressTimestamp for Pdelay_Resp message, of most recently-completed peer delay message exchange, transmitted by peer delay responder that this Link Port sends Pdelay_Req to; it is equal to the sum of the following: a) the ns field of the responseOriginTimestamp (see Table 11-14), multiplied by 2^{16}, b) the correctionField (see Table 11-6). (NOTE 3)	t3	2^{-16} ns
3	pdelayRespEventIngressTimestamp for Pdelay_Resp message, of most recently-completed peer delay message exchange, received by this PTP Instance (NOTE 4)	t4	2^{-16} ns

Table 14-9—Description of pdelayTruncatedTimestampsArray

Array element	Timestamp description	Corresponding timestamp of Figure 11-1	Units

NOTE 1—This quantity is not simply the nanoseconds plus fractional nanoseconds portion of the pdelayReqEventEgressTimestamp. Rather, it is equal to [pdelayReqEventEgressTimestamp.seconds × $(10^9)(2^{16})$ + pdelayReqEventEgressTimestamp.fractionalNanoseconds] mod 2^{48}, where pdelayReqEventEgressTimestamp is expressed as an ExtendedTimestamp (see 6.4.3.5). Its units are 2^{-16} ns.

NOTE 2—This quantity is not simply the nanoseconds plus fractional nanoseconds portion of the pdelayReqEventIngressTimestamp. Rather, it is equal to [pdelayReqEventIngressTimestamp>.seconds × $(10^9)(2^{16})$ + pdelayReqEventIngressTimestamp.fractionalNanoseconds] mod 2^{48}, where pdelayReqEventIngressTimestamp is expressed as an ExtendedTimestamp (see 6.4.3.5). Its units are 2^{-16} ns.

NOTE 3—This quantity is not simply the nanoseconds plus fractional nanoseconds portion of the pdelayRespEventEgressTimestamp. Rather, it is equal to [pdelayRespEventEgressTimestamp.seconds × $(10^9)(2^{16})$ + pdelayRespEventEgressTimestamp.fractionalNanoseconds] mod 2^{48}, where pdelayRespEventEgressTimestamp is expressed as an ExtendedTimestamp (see 6.4.3.5). Its units are 2^{-16} ns.

NOTE 4—This quantity is not simply the nanoseconds plus fractional nanoseconds portion of the pdelayRespEventIngressTimestamp. Rather, it is equal to [pdelayRespEventIngressTimestamp.seconds × $(10^9)(2^{16})$ + pdelayRespEventIngressTimestamp.fractionalNanoseconds] mod 2^{48}, where pdelayRespEventIngressTimestamp is expressed as an ExtendedTimestamp (see 6.4.3.5). Its units are 2^{-16} ns.

Table 14-10—portDS Table

Name	Data type	Operations supported[a]	References
portIdentity	PortIdentity (see 6.4.3.7)	R	14.8.2
portState	Enumeration8	R	14.8.3, Table 14-7
ptpPortEnabled	Boolean	RW	14.8.4
delayMechanism	Enumeration8	RW	14.8.5
isMeasuringDelay	Boolean	R	14.8.6
asCapable	Boolean	R	14.8.7
meanLinkDelay	UScaledNs (recommended)	R	14.8.8
meanLinkDelayThresh	UScaledNs (recommended)	RW	14.8.9
delayAsymmetry	ScaledNs (recommended)	RW	14.8.10
neighborRateRatio	Integer32	R	14.8.11
initialLogAnnounceInterval	Integer8	RW	14.8.12
currentLogAnnounceInterval	Integer8	R	14.8.13
useMgtSettableLogAnnounceInterval	Boolean	RW	14.8.14
mgtSettableLogAnnounceInterval	Integer8	RW	14.8.15
announceReceiptTimeout	UInteger8	RW	14.8.16

Table 14-10—portDS Table *(continued)*

Name	Data type	Operations supported[a]	References
initialLogSyncInterval	Integer8	RW	14.8.17
currentLogSyncInterval	Integer8	R	14.8.18
useMgtSettableLogSyncInterval	Boolean	RW	14.8.19
mgtSettableLogSyncInterval	Integer8	RW	14.8.20
syncReceiptTimeout	UInteger8	RW	14.8.21
syncReceiptTimeoutTimeInterval	UScaledNs	R	14.8.22
initialLogPdelayReqInterval	Integer8	RW	14.8.23
currentLogPdelayReqInterval	Integer8	R	14.8.24
useMgtSettableLogPdelayReqInterval	Boolean	RW	14.8.25
mgtSettableLogPdelayReqInterval	Integer8	RW	14.8.26
initialLogGptpCapableMessageInterval	Integer8	RW	14.8.27
currentLogGptpCapableMessageInterval	Integer8	R	14.8.28
useMgtSettableLogGptpCapableMessageInterval	Integer8	RW	14.8.29
mgtSettableLogGptpCapableMessageInterval	Integer8	RW	14.8.30
initialComputeNeighborRateRatio	Integer8	RW	14.8.31
currentComputeNeighborRateRatio	Integer8	R	14.8.32
useMgtSettableComputeNeighborRateRatio	Boolean	RW	14.8.33
mgtSettableComputeNeighborRateRatio	Integer8	RW	14.8.34
initialComputeMeanLinkDelay	Integer8	RW	14.8.35
currentComputeMeanLinkDelay	Integer8	R	14.8.36
useMgtSettableComputeMeanLinkDelay	Boolean	RW	14.8.37
mgtSettableComputeMeanLinkDelay	Integer8	RW	14.8.38
allowedLostResponses	UInteger8	RW	14.8.39
allowedFaults	UInteger8	RW	14.8.40
logGptpCapableMessageInterval	Integer8	RW	14.8.41
gPtpCapableReceiptTimeout	UInteger8	RW	14.8.42
versionNumber	UInteger4	R	14.8.43
nup (NOTE)	Float64	RW	14.8.44
ndown (NOTE)	Float64	RW	14.8.45
oneStepTxOper	Boolean	R	14.8.46
oneStepReceive	Boolean	R	14.8.47

Table 14-10—portDS Table *(continued)*

Name	Data type	Operations supported[a]	References
oneStepTransmit	Boolean	R	14.8.48
initialOneStepTxOper	Boolean	RW	14.8.49
currentOneStepTxOper	Boolean	R	14.8.50
useMgtSettableOneStepTxOper	Boolean	RW	14.8.51
mgtSettableOneStepTxOper	Boolean	RW	14.8.52
syncLocked	Boolean	R	14.8.53
pdelayTruncatedTimestamp-sArray	UInteger48[4]	R	14.8.54
~~asymmetryMeasurementMode~~	~~Boolean~~	~~RW~~	~~14.8.55~~
minorVersionNumber	UInteger4	R	14.8.55
NOTE—The values of nup and ndown in Table 14-10 depend on the particular PHY used.			

[a]R = Read only access; RW = Read/write access.

14.9.1 profileIdentifier

The value is the profileIdentifier for this PTP profile (see Annex F, F.1).

14.9.2 descriptionPortDS Table

There is one descriptionPortDS Table per port of a PTP Instance, as detailed in Table 14-11.

Table 14-11—descriptionPortDS Table

Name	Data type	Operations supported[a]	References
profileIdentifier	Octet6	R	14.9.1

[a]R= Read only access.

14.10 Port Parameter Statistics Data Set (portStatisticsDS)

The portStatisticsDS provides counters associated with port capabilities at a given PTP Instance.

14.10.1 General

For the single port of a PTP End Instance and for each port of a PTP Relay Instance , the following portStatisticsDS provides counters. The number of such statistics sets is the same as the value of defaultDS.numberPorts.

14.10.2 rxSyncCount

A counter that increments every time synchronization information is received, denoted by a transition to TRUE from FALSE of the rcvdSync variable of the MDSyncReceiveSM state machine (see 11.2.14.1.2 and

Figure 11-6), when in the DISCARD, WAITING_FOR_SYNC, or WAITING_FOR_FOLLOW_UP states; or rcvdIndication transitions to TRUE (see Figure 12-7).

14.10.3 rxOneStepSyncCount

A counter that increments every time a one-step Sync message is received, denoted by a transition to TRUE from FALSE of the rcvdSync variable of the MDSyncReceiveSM state machine (see 11.2.14.1.2 and Figure 11-6) with the variable rcvdSyncPtr->twoStepFlag FALSE, when in the DISCARD or WAITING_FOR_SYNC states.

14.10.4 rxFollowUpCount

A counter that increments every time a Follow_Up message is received, denoted by a transition to TRUE from FALSE of the rcvdFollowUp variable of the MDSyncReceiveSM state machine (see 11.2.14.1.3 and Figure 11-6) when in the WAITING_FOR_FOLLOW_UP state.

14.10.5 rxPdelayRequestCount

A counter that increments every time a Pdelay_Req message is received, denoted by a transition to TRUE from FALSE of the rcvdPdelayReq variable of the MDPdelayResp state machine (see 11.2.20.2.1 and Figure 11-10) when in the WAITING_FOR_PDELAY_REQ or INITIAL_WAITING_FOR_PDELAY_REQ states.

14.10.6 rxPdelayResponseCount

A counter that increments every time a Pdelay_Resp message is received, denoted by a transition to TRUE from FALSE of the rcvdPdelayResp variable of the MDPdelayReq state machine (see 11.2.19.2.2 and Figure 11-9) when in the WAITING_FOR_PDELAY_RESP state.

14.10.7 rxPdelayResponseFollowUpCount

A counter that increments every time a Pdelay_Resp_Follow_Up message is received, denoted by a transition to TRUE from FALSE of the rcvdPdelayRespFollowUp variable of the MDPdelayReq state machine (see 11.2.19.2.4 and Figure 11-9) when in the WAITING_FOR_PDELAY_RESP_FOLLOW_UP state.

14.10.8 rxAnnounceCount

A counter that increments every time an Announce message is received, denoted by a transition to TRUE from FALSE of the rcvdAnnounce variable of the PortAnnounceReceive state machine (see 10.3.11 and Figure 10-13) when in the DISCARD or RECEIVE states.

14.10.9 rxPTPPacketDiscardCount

A counter that increments every time a PTP message of the respective PTP Instance is discarded, caused by the occurrence of any of the following conditions:

a) A received Announce message is not qualified, denoted by the function qualifyAnnounce (see 10.3.11.2.1 and 13.1.3.4) of the PortAnnounceReceive state machine (see 10.3.11 and Figure 10-13) returning FALSE;
b) A Follow_Up message corresponding to a received Sync message is not received, denoted by a transition of the condition (currentTime >= followUpReceiptTimeoutTime) to TRUE from FALSE when in the WAITING_FOR_FOLLOW_UP state of the MDSyncReceiveSM state machine (see 11.2.14 and Figure 11-6);

c) A Pdelay_Resp message corresponding to a transmitted Pdelay_Req message is not received, denoted by a transition from the WAITING_FOR_PDELAY_RESP state to the RESET state of the MDPdelayReq state machine (see 11.2.19 and Figure 11-9);
d) A Pdelay_Resp_Follow_Up message corresponding to a transmitted Pdelay_Req message is not received, denoted by a transition from the WAITING_FOR_PDELAY_RESP_FOLLOW_UP state to the RESET state of the MDPdelayReq state machine (see 11.2.19 and Figure 11-9).

14.10.10 syncReceiptTimeoutCount

A counter that increments every time sync receipt timeout occurs, denoted by entering the AGED state of the PortAnnounceInformation state machine (see 10.3.12 and Figure 10-14), with the condition (currentTime>= syncReceiptTimeoutTime && gmPresent) evaluating to TRUE.

14.10.11 announceReceiptTimeoutCount

A counter that increments every time announce receipt timeout occurs, denoted by entering the AGED state of the PortAnnounceInformation state machinefrom the CURRENT state of the PortAnnounceInformation state machine (see 10.3.12 and Figure 10-14), with the condition (currentTime >= announceReceiptTimeoutTime) evaluating to TRUE.

14.10.12 pdelayAllowedLostResponsesExceededCount

A counter that increments every time the value of the variable lostResponses (see, 11.2.19.2.9) exceeds the value of the variable allowedLostResponses (see 11.2.13.4), in the RESET state of the MDPdelayReq state machine (see 11.2.19 and Figure 11-9).

14.10.13 txSyncCount

A counter that increments every time synchronization information is transmitted, denoted by a transition to TRUE from FALSE of the rcvdMDSyncMDSS variable of the MDSyncSendSM state machine (see 11.2.15.1.1 and Figure 11-7), when in the INITIALIZING, SEND_FOLLOW_UP, or SET_CORRECTION_FIELD states; or the INITIATE_REQUEST_WAIT_CONFIRM_OR_SAVE_INFO state is entered in Figure 12-5 and TM is being used [i.e., (tmFtmSupport == 0x01 ||((tmFtmSupport & 0x01 == 0x01)&& ftmReqGrantedMaster)) in master state machine A of Figure 12-5]; or the INITIATE_REQUEST_WAIT_CONFIRM state is entered in Figure 12-6 (in this case FTM is being used).

14.10.14 txOneStepSyncCount

A counter that increments every time a one-step Sync message is transmitted, denoted by a transition to TRUE from FALSE of the rcvdMDSyncMDSS variable of the MDSyncSendSM state machine (see 11.2.15.1.1 and Figure 11-7) with the variable oneStepTxOper TRUE, when in the INITIALIZING, SEND_SYNC_ONE_STEP, or SET_CORRECTION_FIELD states.

14.10.15 txFollowUpCount

A counter that increments every time a Follow_Up message is transmitted, denoted by a transition to TRUE from FALSE of the rcvdMDTimestampReceive variable of the MDSyncSendSM state machine (see 11.2.15.1.3 and Figure 11-7), when in the SEND_SYNC state.

14.10.16 txPdelayRequestCount

A counter that increments every time a Pdelay_Req message is transmitted, denoted by entering the INITIAL_SEND_PDELAY_REQ or SEND_PDELAY_REQ states of the MDPdelayReq state machine (see 11.2.19 and Figure 11-9).

14.10.17 txPdelayResponseCount

A counter that increments every time a Pdelay_Resp message is transmitted, denoted by a transition to TRUE from FALSE of the rcvdPdelayReq variable of the MDPdelayResp state machine (see 11.2.20.2.1 and Figure 11-10) when in the WAITING_FOR_PDELAY_REQ or INITIAL_WAITING_FOR_PDELAY_REQ states, and resulting entry to the SENT_PDELAY_RESP_WAITING_FOR_TIMESTAMP state.

14.10.18 txPdelayResponseFollowUpCount

A counter that increments every time a Pdelay_Resp_Follow_Up message is transmitted, denoted by a transition to TRUE from FALSE of the rcvdMDTimestampReceive variable of the MDPdelayResp state machine (see 11.2.20.2.2 and Figure 11-10) when in the SENT_PDELAY_RESP_WAITING_FOR_TIMESTAMP state, and resulting entry to the WAITING_FOR_PDELAY_REQ state.

14.10.19 txAnnounceCount

A counter that increments every time an Announce message is transmitted, denoted by entering the TRANSMIT_ANNOUNCE state of the PortAnnounceReceive state machine (see 10.3.16 and Figure 10-18).

14.10.20 portStatisticsDS Table

There is one portStatisticsDS Table per port, per PTP Instance of a time-aware system. Each portStatisticsDS Table contains a set of counters for each port that supports the time-synchronization capability, as detailed in Table 14-12. Each table can be created or removed dynamically in implementations that support dynamic configuration of ports and components.

Table 14-12—portStatisticsDS Table

Name	Data type	Operations supported[a]	References
rxSyncCount	UInteger32	R	14.10.2
rxOneStepSyncCount	UInteger32	R	14.10.3
rxFollowUpCount	UInteger32	R	14.10.4
rxPdelayRequestCount	UInteger32	R	14.10.5
rxPdelayResponseCount	UInteger32	R	14.10.6
rxPdelayResponseFollowUp-Count	UInteger32	R	14.10.7
rxAnnounceCount	UInteger32	R	14.10.8
rxPTPPacketDiscardCount	UInteger32	R	14.10.9
syncReceiptTimeoutCount	UInteger32	R	14.10.10
announceReceiptTimeout-Count	UInteger32	R	14.10.11
pdelayAllowedLostResponses-ExceededCount	UInteger32	R	14.10.12
txSyncCount	UInteger32	R	14.10.13
txOneStepSyncCount	UInteger32	R	14.10.14
txFollowUpCount	UInteger32	R	14.10.15
txPdelayRequestCount	UInteger32	R	14.10.16
txPdelayResponseCount	UInteger32	R	14.10.17
txPdelayResponseFollowUp-Count	UInteger32	R	14.10.18
txAnnounceCount	UInteger32	R	14.10.19

[a]R= Read only access.

14.11 Acceptable Master Port Parameter Data Set (acceptableMasterPortDS)

The acceptableMasterPortDS represents the capability to enable/disable the acceptable master table feature on a port.

14.11.1 General

For the single port of a PTP End Instance and for each port of a PTP Relay Instance, the acceptableMasterPortDS contains the single member acceptableMasterTableEnabled, which is used to enable/disable the Acceptable Master Table Feature. The number of such data sets is the same as the value of defaultDS.numberPorts.

14.11.2 acceptableMasterTableEnabled

The value is equal to the value of the Boolean acceptableMasterTableEnabled (see 13.1.3.2 and 13.1.3.5).

14.11.3 acceptableMasterPortDS Table

There is one acceptableMasterPortDS Table per port, per PTP Instance of a time-aware system as detailed in Table 14-13.

Table 14-13—acceptableMasterPortDS Table

Name	Data type	Operations supported[a]	References
acceptableMasterT-ableEnabled	Boolean	RW	14.11.2

[a]R = Read only access; RW = Read/write access.

14.12 External Port Configuration Port Parameter Data Set (externalPortConfigurationPortDS)

The externalPortConfigurationPortDS is used with the external port configuration option to indicate the desired state for the gPTP Port.

14.12.1 General

The externalPortConfigurationPortDS contains the single member desiredState, which indicates the desired state for the gPTP Port. The number of such data sets is the same as the value of defaultDS.numberPorts.

14.12.2 desiredState

When the value of defaultDS.externalPortConfigurationEnabled is TRUE, the value of externalPortConfigurationPortDS.desiredState is the desired state of the gPTP Port. This member sets the value of the variable portStateInd (see 10.3.15.1.5). When a new value is written to the member by management, the variable rcvdPortStateInd (see 10.3.15.1.4) is set to TRUE.

14.12.3 externalPortConfigurationPortDS Table

There is one externalPortConfigurationPortDS Table per gPTPport, per PTP Instance of a time-aware system as detailed in Table 14-14.

Table 14-14—externalPortConfigurationPortDS Table

Name	Data type	Operations supported[a]	References
desiredState	Enumeration8 (see 6.4.2)	RW	14.12.2, Table 14-7

[a]R = Read only access; RW = Read/write access.

14.13 Asymmetry Measurement Mode Parameter Data Set (asymmetryMeasurementModeDS)

The asymmetryMeasurementModeDS represents the capability to enable/disable the Asymmetry Compensation Measurement Procedure on a Link Port (see Annex G). This data set is used instead of the cmldsAsymmetryMeasurementModeDS, when only domain 0 is present and CMLDS is not used.

14.13.1 General

For the single Port of a PTP End Instance and for each Port of a PTP Relay Instance, the asymmetryMeasurementModeDS contains the single member asymmetryMeasurementMode, which is used to enable/disable the Asymmetry Compensation Measurement Procedure. The number of such data sets is the same as the value of defaultDS.numberPorts.

14.13.2 asymmetryMeasurementMode

The value is equal to the value of the Boolean asymmetryMeasurementMode (see G.3). For full-duplex IEEE 802.3 media, the value is TRUE if an asymmetry measurement is being performed for the link attached to this Link Port, and FALSE otherwise. For all other media, the value shall be FALSE (see 10.2.5.2).

NOTE—If an asymmetry measurement is being performed for a link, asymmetryMeasurementMode must be TRUE for the Link Ports at each end of the link.

14.13.3 asymmetryMeasurementModeDS Table

There is one asymmetryMeasurementModeDS Table for the single PTP Instance whose domain number is 0, per Link Port, as detailed in Table 14-15. This data set is used only when there is a single gPTP domain and CMLDS is not used.

Table 14-15—asymmetryMeasurementModeDS Table

Name	Data type	Operations supported[a]	References
asymmetryMeasurementMode	Boolean	RW	14.13.2

[a]R = Read only access; RW = Read/write access.

14.14 Common Services Port Parameter Data Set (commonServicesPortDS)

The commonServicesPortDS enables a gPTP Port of a PTP Instance to determine which port of the respective common service corresponds to that gPTP Port.

14.14.1 General

At present, the only common service specified is the CMLDS, and the only member of the commonServicesPortDS is the cmldsLinkPortPortNumber. This member contains the port number of the CMLDS Link Port that corresponds to this gPTP Port.

14.14.2 cmldsLinkPortPortNumber

The value is the portNumber attribute of the cmldsLinkPortDS.portIdentity (see 14.16.2) of the Link Port that corresponds to this gPTP Port.

14.14.3 commonServicesPortDS Table

There is one acceptableMasterPortDS Table per port, per PTP Instance of a time-aware system as detailed in Table 14-16.

Table 14-16—commonServicesPortDS Table

Name	Data type	Operations supported[a]	References
cmldsLinkPortPort-Number	UInteger16	R	14.14.2

[a]R = Read only access.

14.15 Common Mean Link Delay Service Default Parameter Data Set (cmldsDefaultDS)

The cmldsDefaultDS describes the per-time-aware-system attributes of the Common Mean Link Delay Service.

14.15.1 clockIdentity

The value is the clockIdentity (see 8.5.2.2) that will be used to identify the Common Mean Link Delay Service.

14.15.2 numberLinkPorts

The value is the number of Link Ports of the time-aware system on which the Common Mean Link Delay Service is implemented. For an end station the value is 1.

14.15.3 sdoId

The value is 0x200. This is the sdoId for the Common Mean Link Delay Service (see 11.2.17).

NOTE—The attribute sdoId is specified as a 12-bit unsigned integer in 8.1. The data type for the managed object sdoId is UInteger16 in Table 14-16, for compatibility with IEEE Std 1588. The range of the managed object is limited to 12 bits

14.15.4 cmldsDefaultDS Table

There is one cmldsDefaultDS Table per time-aware system, as detailed in Table 14-17.

Table 14-17—cmldsDefaultDS Table

Name	Data type	Operations supported[a]	References
clockIdentity	ClockIdentity	R	14.15.1
numberLinkPorts	UInteger16	R	14.15.2
sdoId	UInteger16	R	14.15.3

[a]R = Read only access; RW = Read/write access.

14.16 Common Mean Link Delay Service Link Port Parameter Data Set (cmldsLinkPortDS)

The cmldsLinkPortDS represents time-aware Link Port capabilities for the Common Mean Link Delay Service of a Link Port of a time-aware system.

14.16.1 General

For every Link Port of the Common Mean Link Delay Service of a time-aware system, the following Link Port Parameter Data Set is maintained as the basis for protocol decisions and providing values for message fields. The number of such data sets is the same as the value of cmldsDefaultDS.numberLinkPorts.

14.16.2 portIdentity

The value is the portIdentity attribute of the local port (see 8.5.2).

14.16.3 cmldsLinkPortEnabled

The value is equal to the value of the Boolean cmldsLinkPortEnabled (see 11.2.18.1).

14.16.4 isMeasuringDelay

The value is equal to the value of the Boolean isMeasuringDelay (see 11.2.13.6 and 16.4.3.2).

14.16.5 asCapableAcrossDomains

The value is equal to the value of the Boolean asCapableAcrossDomains (see 11.2.2 and 11.2.13.12).

14.16.6 meanLinkDelay

The value is equal to the value of the per-port global variable meanLinkDelay (see 10.2.5.8). It is an estimate of the current one-way propagation time on the link attached to this Link Port, measured as specified for the respective medium (see 11.2.17, 12.5.2, and 16.4). The value is zero for Link Ports attached to IEEE 802.3 EPON links and for the master port of an IEEE 802.11 link, because one-way propagation delay is not measured on the latter and not directly measured on the former. It is recommended that the data type be scaledNs. The default value is zero.

14.16.7 meanLinkDelayThresh

The value is equal to the value of the per-Link-port global variable meanLinkDelayThresh (see 11.2.13.7). It is the propagation time threshold, above which a Link Port (and therefore any PTP Ports that use the CMLDS on this Link Port) is not considered capable of participating in the IEEE 802.1AS protocol.

14.16.8 delayAsymmetry

The value is the asymmetry in the propagation delay on the link attached to this Link Port relative to the local clock, as defined in 10.2.5.9 and 8.3. If propagation delay asymmetry is not modeled, then delayAsymmetry is 0.

14.16.9 neighborRateRatio

The value is an estimate of the ratio of the frequency of the LocalClock entity of the time-aware system at the other end of the link attached to this Link Port, to the frequency of the LocalClock entity of this time-aware system (see 10.2.5.7). neighborRateRatio is expressed as the fractional frequency offset multiplied by 2^{41}, i.e., the quantity (neighborRateRatio – 1.0)(2^{41}).

14.16.10 initialLogPdelayReqInterval

If useMgtSettableLogPdelayReqInterval is FALSE then, for full-duplex, IEEE 802.3 media and CSN media that use the peer-to-peer delay mechanism to measure path delay (see 16.4.3.1), the value is the logarithm to base 2 of the Pdelay_Req message transmission interval used when (a) the Link Port is initialized, or (b) a message interval request TLV is received with the logLinkDelayInterval field set to 126 (see 11.5.2.2 and the LinkDelayIntervalSetting state machine, 11.2.21).

For all other media, the value is 127.

14.16.11 currentLogPdelayReqInterval

For full-duplex, IEEE 802.3 media and CSN media that use the peer-to-peer delay mechanism to measure path delay (see 16.4.3.1), the value is the logarithm to the base 2 of the current Pdelay_Req message transmission interval (see 11.5.2.2).

For all other media, the value is 127.

14.16.12 useMgtSettableLogPdelayReqInterval

The managed object is a Boolean that determines the source of the sync interval and mean time interval between successive Pdelay_Req messages. If the value is TRUE, the value of currentLogPdelayReqInterval is set equal to the value of mgtSettableLogPdelayReqInterval (see 14.16.13). If the value of the managed object is FALSE, the value of currentLogPdelayReqInterval is determined by the LinkDelayIntervalSetting state machine (see 11.2.2111.2.21). The default value of useMgtSettableLogPdelayReqInterval is FALSE.

14.16.13 mgtSettableLogPdelayReqInterval

The value is the logarithm to base 2 of the mean time interval between successive Pdelay_Req messages if useMgtSettableLogPdelayReqInterval is TRUE. The value is not used if useMgtSettableLogPdelayReqInterval is FALSE.

14.16.14 initialComputeNeighborRateRatio

If useMgtSettableComputeNeighborRateRatio is FALSE then, for full-duplex, IEEE 802.3 media and CSN media that use the peer-to-peer delay mechanism to measure path delay (see 16.4.3.1), the value is the initial value of computeNeighborRateRatio (see 10.2.5.10).

For all other media, the value is TRUE.

14.16.15 currentComputeNeighborRateRatio

For full-duplex, IEEE 802.3 media and CSN media that use the peer-to-peer delay mechanism to measure path delay (see 16.4.3.1), the value is the current value of computeNeighborRateRatio.

For all other media, the value is TRUE.

14.16.16 useMgtSettableComputeNeighborRateRatio

The managed object is a Boolean that determines the source of the value of computeNeighborRateRatio. If the value is TRUE, the value of computeNeighborRateRatio is set equal to the value of mgtSettablecomputeNeighborRateRatio (see 14.16.17). If the value of the managed object is FALSE, the value of currentComputeNeighborRateRatio is determined by the LinkDelayIntervalSetting state machine (see 11.2.2111.2.21). The default value of useMgtSettableLogPdelayReqInterval is FALSE.

14.16.17 mgtSettableComputeNeighborRateRatio

The value is the logarithm to base 2 of computeNeighborRateRatio if useMgtSettableComputeNeighborRateRatio is TRUE. The value is not used if useMgtSettableComputeNeighborRateRatio is FALSE.

14.16.18 initialComputeMeanLinkDelay

If useMgtSettableComputeMeanLinkDelay is FALSE then, for full-duplex, IEEE 802.3 media and CSN media that use the peer-to-peer delay mechanism to measure path delay (see 16.4.3.1), the value is the initial value of computeMeanLinkDelay (see 10.2.5.10).

For all other media, the value is TRUE.

14.16.19 currentComputeMeanLinkDelay

For full-duplex, IEEE 802.3 media and CSN media that use the peer-to-peer delay mechanism to measure path delay (see 16.4.3.1), the value is the current value of computeMeanLinkDelay.

For all other media, the value is TRUE.

14.16.20 useMgtSettableComputeMeanLinkDelay

The managed object is a Boolean that determines the source of the value of computeMeanLinkDelay. If the value is TRUE, the value of computeMeanLinkDelay is set equal to the value of mgtSettableComputeMeanLinkDelay (see 14.16.17). If the value of the managed object is FALSE, the value of currentComputeMeanLinkDelay is determined by the LinkDelayIntervalSetting state machine (see 11.2.2111.2.21). The default value of useMgtSettableLogPdelayReqInterval is FALSE.

14.16.21 mgtSettableComputeMeanLinkDelay

The value is the logarithm to base 2 of computeMeanLinkDelay if useMgtSettableComputeMeanLinkDelay is TRUE. The value is not used if useMgtSettableComputeMeanLinkDelay is FALSE.

14.16.22 allowedLostResponses

The value is equal to the value of the per-Link-Port global variable allowedLostResponses (see 11.5.3 and 11.2.13.4). It is the number of Pdelay_Req messages for which a valid response is not received, above which a Link Port is considered to not be exchanging peer delay messages with its neighbor.

14.16.23 allowedFaults

The value is equal to the value of the per-Link-Port global variable allowedFaults (see 11.5.4 and 11.2.13.5). It is the number of faults (see 11.5.4), above which asCapableAcrossDomains is set to FALSE, i.e., a Link Port is considered to not be capable of interoperating with its neighbor via the IEEE 802.1AS protocol (see 10.2.5.1).

14.16.24 versionNumber

This value is set to versionPTP as specified in 10.6.2.2.4.

14.16.25 pdelayTruncatedTimestampsArray

For full-duplex IEEE 802.3 media, and CSN media that use the peer-to-peer delay mechanism to measure path delay (see 16.4.3.1), the values of the four elements of this array are as described in Table 14-9. For all other media, the values are zero. Array elements 0, 1, 2, and 3 correspond to the timestamps t1, t2, t3, and t4, modulo 2^{32}, respectively, in Figure 11-1, and expressed in units of 2^{-16} ns (i.e., the value of each array element is equal to the remainder obtained upon dividing the respective timestamp, expressed in units of 2^{-16} ns, by 2^{48}). At any given time, the timestamp values stored in the array are for the same, and most recently completed, peer delay message exchange.

NOTE—This managed object is used with the asymmetry measurement compensation procedure, which is based on line-swapping.

14.16.26 minorVersionNumber

This value is set to minorVersionPTP as specified in 10.6.2.2.3.

14.16.27 cmldsLinkPortDS Table

There is one cmldsLinkPortDS Table per Link Port, for the PTP Instance of a time-aware system. Each cmldsLinkPortDS Table contains a set of parameters for each Link Port that supports the time-synchronization capability, as detailed in Table 14-18. Each table can be created or removed dynamically in implementations that support dynamic configuration of Link Ports and components.

14.17 Common Mean Link Delay Service Link Port Parameter Statistics Data Set (cmldsLinkPortStatisticsDS)

The cmldsLinkPortStatisticsDS provides counters associated with Link Port capabilities at a given time-aware system.

Table 14-18—cmldsLinkPortDS Table

Name	Data type	Operations supported[a]	References
portIdentity	PortIdentity (see 6.4.3.7)	R	14.16.2
cmldsLinkPortEnabled	Boolean	R	14.16.3
isMeasuringDelay	Boolean	R	14.16.4
asCapableAcrossDomains	Boolean	R	14.16.5
meanLinkDelay	UScaledNs (recommended)	R	14.16.6
meanLinkDelayThresh	UScaledNs (recommended)	RW	14.16.7
delayAsymmetry	ScaledNs (recommended)	RW	14.16.8
neighborRateRatio	Integer32	R	14.16.9
initialLogPdelayReqInterval	Integer8	RW	14.16.10
currentLogPdelayReqInterval	Integer8	R	14.16.11
useMgtSettableLogPdelayReq-Interval	Boolean	RW	14.16.12
mgtSettableLogPdelayReqIn-terval	Integer8	RW	14.16.13
initialComputeNeighborRateR-atio	Integer8	RW	14.16.14
currentComputeNeighborRa-teRatio	Integer8	R	14.16.15
useMgtSettableComputeNeigh-borRateRatio	Boolean	RW	14.16.16
mgtSettableComputeNeighbor-RateRatio	Integer8	RW	14.16.17
initialComputeMeanLinkDelay	Integer8	RW	14.16.18
currentComputeMeanLinkDe-lay	Integer8	R	14.16.19
useMgtSettableComputeMean-LinkDelay	Boolean	RW	14.16.20
mgtSettableComputeMean-LinkDelay	Integer8	RW	14.16.21
allowedLostResponses	UInteger8	RW	14.16.22
allowedFaults	UInteger8	RW	14.16.23
versionNumber	UInteger4	R	14.16.24
pdelayTruncatedTimestamp-sArray	UInteger48[4]	R	14.16.25
minorVersionNumber	UInteger4	R	14.16.26

[a]R = Read only access; RW = Read/write access.

14.17.1 General

For every Link Port of the Common Mean Link Delay Service of a time-aware system, the following cmldsLinkPortStatisticsDS provides counters. The number of such statistics sets is the same as the value of cmldsDefaultDS.numberLinkPorts.

14.17.2 rxPdelayRequestCount

A counter that increments every time a Pdelay_Req message is received, denoted by a transition to TRUE from FALSE of the rcvdPdelayReq variable of the MDPdelayResp state machine (see 11.2.20.2.1 and Figure 11-10) when in the WAITING_FOR_PDELAY_REQ or INITIAL_WAITING_FOR_PDELAY_REQ states.

14.17.3 rxPdelayResponseCount

A counter that increments every time a Pdelay_Resp message is received, denoted by a transition to TRUE from FALSE of the rcvdPdelayResp variable of the MDPdelayReq state machine (see 11.2.19.2.2 and Figure 11-9) when in the WAITING_FOR_PDELAY_RESP state.

14.17.4 rxPdelayResponseFollowUpCount

A counter that increments every time a Pdelay_Resp_Follow_Up message is received, denoted by a transition to TRUE from FALSE of the rcvdPdelayRespFollowUp variable of the MDPdelayReq state machine (see 11.2.19.2.4 and Figure 11-9) when in the WAITING_FOR_PDELAY_RESP_FOLLOW_UP state.

14.17.5 rxPTPPacketDiscardCount

A counter that increments every time a PTP message of the Common Mean Link Delay Service is discarded, caused by the occurrence of any of the following conditions:

— A Pdelay_Resp message corresponding to a transmitted Pdelay_Req message is not received, denoted by a transition from the WAITING_FOR_PDELAY_RESP state to the RESET state of the MDPdelayReq state machine (see 11.2.19 and Figure 11-9);
— A Pdelay_Resp_Follow_Up message corresponding to a transmitted Pdelay_Req message is not received, denoted by a transition from the WAITING_FOR_PDELAY_RESP_FOLLOW_UP state to the RESET state of the MDPdelayReq state machine (see 11.2.19 and Figure 11-9).

14.17.6 pdelayAllowedLostResponsesExceededCount

A counter that increments every time the value of the variable lostResponses (see, 11.2.19.2.9) exceeds the value of the variable allowedLostResponses (see 11.2.13.4), in the RESET state of the MDPdelayReq state machine (see 11.2.19 and Figure 11-9).

14.17.7 txPdelayRequestCount

A counter that increments every time a Pdelay_Req message is transmitted, denoted by entering the INITIAL_SEND_PDELAY_REQ or SEND_PDELAY_REQ states of the MDPdelayReq state machine (see 11.2.19 and Figure 11-9).

14.17.8 txPdelayResponseCount

A counter that increments every time a Pdelay_Resp message is transmitted, denoted by a transition to TRUE from FALSE of the rcvdPdelayReq variable of the MDPdelayResp state machine (see 11.2.20.2.1 and Figure 11-10) when in the WAITING_FOR_PDELAY_REQ or INITIAL_WAITING_FOR_PDELAY_REQ states, and resulting entry to the SENT_PDELAY_RESP_WAITING_FOR_TIMESTAMP state.

14.17.9 txPdelayResponseFollowUpCount

A counter that increments every time a Pdelay_Resp_Follow_Up message is transmitted, denoted by a transition to TRUE from FALSE of the rcvdMDTimestampReceiveMDPResp variable of the

MDPdelayResp state machine (see 11.2.20.2.2 and Figure 11-10) when in the SENT_PDELAY_RESP_WAITING_FOR_TIMESTAMP state, and resulting entry to the WAITING_FOR_PDELAY_REQ state.

14.17.10 cmldsLinkPortStatisticsDS Table

There is one cmldsLinkPortStatisticsDS Table per Link Port of a time-aware system. The cmldsLinkPortStatisticsDS Table contains a set of counters for each Link Port that supports the time-synchronization capability, as detailed in Table 14-19. Each table can be created or removed dynamically in implementations that support dynamic configuration of Link Ports and components.

Table 14-19—cmldsLinkPortStatisticsDSTable

Name	Data type	Operations supported[a]	References
rxPdelayRequestCount	UInteger32	R	14.17.2
rxPdelayResponseCount	UInteger32	R	14.17.3
rxPdelayResponseFollowUp-Count	UInteger32	R	14.17.4
rxPTPPacketDiscardCount	UInteger32	R	14.17.5
pdelayAllowedLostResponses-ExceededCount	UInteger32	R	14.17.6
txPdelayRequestCount	UInteger32	R	14.17.7
txPdelayResponseCount	UInteger32	R	14.17.8
txPdelayResponseFollowUp-Count	UInteger32	R	14.17.9

[a]R= Read only access.

14.18 Common Mean Link Delay Service Asymmetry Measurement Mode Parameter Data Set (cmldsAsymmetryMeasurementModeDS)

The cmldsAsymmetryMeasurementModeDS represents the capability to enable/disable the Asymmetry Compensation Measurement Procedure on a Link Port (see Annex G).

14.18.1 General

For every Link Port of the Common Mean Link Delay Service of a time-aware system, the cmldsAsymmetryMeasurementModeDS contains the single member asymmetryMeasurementMode, which is used to enable/disable the Asymmetry Compensation Measurement Procedure. The number of such data sets is the same as the numberLinkPorts value of the cmldsDefaultDS.

14.18.2 asymmetryMeasurementMode

The value is equal to the value of the Boolean asymmetryMeasurementMode (see G.3). For full-duplex IEEE 802.3 media, the value is TRUE if an asymmetry measurement is being performed for the link attached to this Link Port, and FALSE otherwise. For all other media, the value shall be FALSE (see 10.2.5.2).

NOTE—If an asymmetry measurement is being performed for a link, asymmetryMeasurementMode must be TRUE for the Link Ports at each end of the link.

14.18.3 cmldsAsymmetryMeasurementModeDS Table

There is one cmldsAsymmetryMeasurementModeDS Table for all PTP Instances, per Link Port, as detailed in Table 14-20.

Table 14-20—cmldsAsymmetryMeasurementModeDS Table

Name	Data type	Operations supported[a]	References
asymmetryMeasure-mentMode	Boolean	RW	14.18.2

[a]R = Read only access; RW = Read/write access.

15. Managed object definitions

This clause contains a complete SMIv2 Management Information Base (MIB) set for all features of this standard.

15.1 Internet Standard Management Framework

For a detailed overview of the documents that describe the current Internet Standard Management Framework, please refer to section 7 of IETF RFC 3410 (2002).

Managed objects are accessed via a virtual information store, termed the Management Information Base or 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 STD 58, IETF RFC 2578, IETF RFC 2579, and IETF RFC 2580.

15.2 Structure of the MIB

The IEEE 802.1AS MIB provides objects to configure and manage the IEEE 802.1AS Timing and Synchronization for Time-Sensitive Applications.

The MIB contains a set of textual conventions and is additionally subdivided into sixteen subtrees. Each subtree is organized as a set of related objects. They are as follows:

1) The Default Parameter Data Set (defaultDS) represents the native capabilities of a PTP Instance;
2) The Current Parameter Data Set (currentDS) represents topological position of a local PTP Instance relative to the grandmaster;
3) The Parent Parameter Data Set (parentDS) represents capabilities of the upstream PTP Instance, toward the grandmaster as measured at a local PTP Instance;
4) The Time Properties Parameter Data Set (timePropertiesDS) represents capabilities of the grandmaster, as measured at a local PTP Instance;
5) The Path Trace Parameter Data Set (pathTraceDS) represents the current path trace information (see 10.3.9.23) available at the PTP Instance;
6) The Acceptable Master Table Parameter Data Set (acceptableMasterTableDS) represents the acceptable master table used when the media-dependent port type of EPON is present in a PTP Instance;
7) The Port Parameter Data Set represents time-aware capabilities at a given tgPTP Port, as a set of augmentation to the interface table entry (ifEntry);
8) The Description Port Parameter Data Set (descriptionPortDS) contains the profileIdentifier for this PTP profile as specified in F.1;
9) The Port Parameter Statistics Data Set (portStatisticsDS) represents statistics and counters associated with time-aware capabilities at a given PTP Relay Instance or PTP End Instance port;
10) The Acceptable Master Port Parameter Data Set (acceptableMasterPortDS) represents the capability to enable/disable the acceptable master table feature on a port;
11) The External Port Configuration Port Parameter Data Set (externalPortConfigurationPortDS) is used with the external port configuration option to indicate the desired state of a gPTP port;
12) The Asymmetry Measurement Mode Parameter Data Set (asymmetryMeasurementModeDS) represents the capability to enable/disable the Asymmetry Compensation Measurement Procedure on a port (see Annex G), and is used instead of the

cmldsAsymmetryMeasurementModeDS when CMLDS is not used and there is a single gPTP domain;

13) The Common Services Port Parameter Data Set (commonServicesPortDS) enables a gPTP Port of a PTP Instance to determine which port of the respective common service corresponds to that gPTP Port;
14) The Common Mean Link Delay Service Default Parameter Data Set (cmldsDefaultDS) describes the per-time-aware-system attributes of the Common Mean Link Delay Service;
15) The Common Mean Link Delay Service Link Port Parameter Data Set (cmldsLinkPortDS) represents time-aware Link Port capabilities for the Common Mean Link Delay Service of a time-aware system;
16) The Common Mean Link Delay Service Link Port Parameter Statistics Data Set (cmldsLinkPortStatisticsDS) represents statistics and counters associated with Link Port capabilities at a given time-aware system; and
17) The Common Mean Link Delay Service Asymmetry Measurement Mode Parameter Data Set (cmldsAsymmetryMeasurementModeDS) represents the capability to enable/disable the Asymmetry Compensation Measurement Procedure on a Link Port (see Annex G).

Table 15-1 show the structure of the MIB and the relationship of the MIB objects to the above seventeen data sets.

15.3 Relationship to IEEE Std 802.1AS-2011 MIB

The IEEE Std 802.1AS MIB module contained in 15.5 is a new MIB relative to the IEEE Std 802.1AS-2011 MIB. If this new MIB has been implemented and is accessed via the IEEE 802.1AS-2011 MIB, the index 0 is returned. If the IEEE Std 802.1AS-2011 MIB is used, none of the new features of IEEE Std 802.1AS-20xx will be accessible through that MIB.

15.4 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 IETF RFC 3410, section 8), 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 to ensure that the SNMP entity giving access to an instance of these MIB modules is properly configured to give access to the objects only to those principals (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

There are a number of management objects defined in the IEEE8021-AS MIB module that have a MAX-ACCESS clause of read-write and/or read-create. Such objects may 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 notaccessible) may 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.

The following objects in the IEEE8021-AS MIB can be manipulated to interfere with the operation of timing synchronization. This could, for example, be used to force a reinitialization of state machines, thus causing timing synchronization and network instability. Another possibility would be for an attacker to override grandmaster status, thus giving a user (or an attacker) unauthorized control over the network time.

Improper manipulation of the following writable objects could result in an unintended grandmaster to be elected, when a system is grandmaster capable in a gPTP domain. It could also be used maliciously to cause frequent grandmaster changes, thereby affecting network stability.

ieee8021AsV2DefaultDSPriority1
ieee8021AsV2DefaultDSPriority2

Improper manipulation of the following writable objects could result in a segmented time-aware network, could compromise the expected accuracy, and could interrupt paths of the gPTP domain.

ieee8021AsV2PortDSPtpPortEnabled
ieee8021AsV2PortDSDelayAsymmetry

Unintended access to any of the readable tables or variables in the IEEE8021-AS MIB alerts the reader that timing synchronization in gPTP domain is configured, and on which values timing parameters are configured, and which system is current grandmaster. This information can suggest to an attacker what applications are being run, and thus suggest application-specific attacks, or to enable the attacker to detect whether their attacks are being successful or not. 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.

Table 15-1—IEEE8021-AS MIB structure and object cross reference

MIB table	MIB object	Reference
ieee8021AsV2DefaultDS		defaultDS Table 14-1
	ieee8021AsV2DefaultDSClockIdentity	14.2.1
	ieee8021AsV2DefaultDSNumberPorts	14.2.2
	ieee8021AsV2DefaultDSClockQualityClockClass	14.2.3.1
	ieee8021AsV2DefaultDSClockQualityClockAccuracy	14.2.3.2
	ieee8021AsV2DefaultDSClockQualityOffsetScaledLogVariance	14.2.3.3
	ieee8021AsV2DefaultDSPriority1	14.2.4
	ieee8021AsV2DefaultDSPriority2	14.2.5
	ieee8021AsV2DefaultDSGmCapable	14.2.6
	ieee8021AsV2DefaultDSCurrentUtcOffset	14.2.7
	ieee8021AsV2DefaultDSCurrentUtcOffsetValid	14.2.8
	ieee8021AsV2DefaultDSLeap59	14.2.9
	ieee8021AsV2DefaultDSLeap61	14.2.10
	ieee8021AsV2DefaultDSTimeTraceable	14.2.11
	ieee8021AsV2DefaultDSFrequencyTraceable	14.2.12
	ieee8021AsV2DefaultDSPtpTimescale	14.2.13
	ieee8021AsV2DefaultDSTimeSource	14.2.14
	ieee8021AsV2DefaultDSDomainNumber	14.2.15
	ieee8021AsV2DefaultDSSdoId	14.2.16
	ieee8021AsV2DefaultDSExternalPortConfigurationEnabled	14.2.17
	ieee8021AsV2DefaultDSInstanceEnable	14.2.18
ieee8021AsV2CurrentDS		currentDS Table 14-2
	ieee8021AsV2CurrentDSStepsRemoved	14.3.1
	ieee8021AsV2CurrentDSOffsetFromMaster	14.3.2
	ieee8021AsV2CurrentDSLastGmPhaseChange	14.3.3
	ieee8021AsV2CurrentDSLastGmFreqChange	14.3.4
	ieee8021AsV2CurrentDSGmTimebaseIndicator	14.3.5
	ieee8021AsV2CurrentDSGmChangeCount	14.3.6
	ieee8021AsV2CurrentDSTimeOfLastGmChangeEvent	14.3.7
	ieee8021AsV2CurrentDSTimeOfLastGmPhaseChangeEvent	14.3.8
	ieee8021AsV2CurrentDSTimeOfLastGmFreqChangeEvent	14.3.9
ieee8021AsV2ParentDS		parentDS Table 14-3
	ieee8021AsV2ParentDSParentClockIdentity	14.4.1
	ieee8021AsV2ParentDSParentPortNumber	14.4.1
	ieee8021AsV2ParentDSCumulativeRateRatio	14.4.2
	ieee8021AsV2ParentDSGrandmasterIdentity	14.4.3
	ieee8021AsV2ParentDSGrandmasterClockQualityclockClass	14.4.4.1
	ieee8021AsV2ParentDSGrandmasterClockQualityclockAccuracy	14.4.4.2

Table 15-1—IEEE8021-AS MIB structure and object cross reference *(continued)*

MIB table	MIB object	Reference
	ieee8021AsV2ParentDSGrandmasterClockQualityoffsetScaledLogVar	14.4.4.3
	ieee8021AsV2ParentDSGrandmasterPriority1	14.4.5
	ieee8021AsV2ParentDSGrandmasterPriority2	14.4.6
ieee8021AsV2TimePropertiesDS		timePropertiesDS Table 14-4
	ieee8021AsV2TimePropertiesDSCurrentUtcOffset	14.5.1
	ieee8021AsV2TimePropertiesDSCurrentUtcOffsetValid	14.5.2
	ieee8021AsV2TimePropertiesDSLeap59	14.5.3
	ieee8021AsV2TimePropertiesDSLeap61	14.5.4
	ieee8021AsV2TimePropertiesDSTimeTraceable	14.5.5
	ieee8021AsV2TimePropertiesDSFrequencyTraceable	14.5.6
	ieee8021AsV2TimePropertiesDSPtpTimescale	14.5.7
	ieee8021AsV2TimePropertiesDSTimeSource	14.5.8
ieee8021AsV2PathTraceDS		pathTraceDS Table 14-5
	ieee8021AsV2PathTraceDSEnable	14.6.2
ieee8021AsV2PathTraceDSArray		pathTraceDS Table 14-5
	ieee8021AsV2PathTraceDSArrayList	14.6.1
ieee8021AsV2AcceptableMasterTableDS		acceptableMasterTableDSTable 14-6
	ieee8021AsV2AcceptableMasterTableDSMaxTableSize	14.7.1
	ieee8021AsV2AcceptableMasterTableDSActualTableSize	14.7.2
ieee8021AsV2AcceptableMasterTableDSArray		acceptableMasterTableDSTable 14-6
	ieee8021AsV2AcceptableMasterTableDSArrayPortIdentity	14.7.3
	ieee8021AsV2AcceptableMasterTableDSArrayAlternatePriority1	14.7.3
ieee8021AsV2PortDS		portDSTable 14-10
	ieee8021AsV2PortDSClockIdentity	14.8.2
	ieee8021AsV2PortDSPortNumber	14.8.2
	ieee8021AsV2PortDSPortState	14.8.3
	ieee8021AsV2PortDSPtpPortEnabled	14.8.4
	ieee8021AsV2PortDSdelayMechanism	14.8.5
	ieee8021AsV2PortDSIsMeasuringDelay	14.8.6
	ieee8021AsV2PortDSAsCapable	14.8.7
	ieee8021AsV2PortDSMeanLinkDelay	14.8.8
	ieee8021AsV2PortDSMeanLinkDelayThresh	14.8.9
	ieee8021AsV2PortDSDelayAsym	14.8.10
	ieee8021AsV2PortDSNbrRateRatio	14.8.11
	ieee8021AsV2PortDSInitialLogAnnounceInterval	14.8.12
	ieee8021AsV2PortDSCurrentLogAnnounceInterval	14.8.13
	ieee8021AsV2PortDSUseMgtSettableLogAnnounceInterval	14.8.14
	ieee8021AsV2PortDSMgtSettableLogAnnounceInterval	14.8.15
	ieee8021AsV2PortDSAnnounceReceiptTimeout	14.8.16
	ieee8021AsV2PortDSInitialLogSyncInterval	14.8.17
	ieee8021AsV2PortDSCurrentLogSyncInterval	14.8.18
	ieee8021AsV2PortDSUseMgtSettableLogSyncInterval	14.8.19
	ieee8021AsV2PortDSMgtSettableLogSyncInterval	14.8.20

Table 15-1—IEEE8021-AS MIB structure and object cross reference *(continued)*

MIB table	MIB object	Reference
	ieee8021AsV2PortDSSyncReceiptTimeout	14.8.21
	ieee8021AsV2PortDSSyncReceiptTimeoutTimeInterval	14.8.22
	ieee8021AsV2PortDSInitialLogPdelayReqInterval	14.8.23
	ieee8021AsV2PortDSCurrentLogPdelayReqInterval	14.8.24
	ieee8021AsV2PortDSUseMgtSettableLogPdelayReqInterval	14.8.25
	ieee8021AsV2PortDSMgtSettableLogPdelayReqInterval	14.8.26
	ieee8021AsV2PortDSInitialLogGptpCapableMessageInterval	14.8.27
	ieee8021AsV2PortDSCurrentLogGptpCapableMessageInterval	14.8.28
	ieee8021AsV2PortDSUseMgtSettableLogGptpCapableMessageInterval	14.8.29
	ieee8021AsV2PortDSMgtSettableLogGptpCapableMessageInterval	14.8.30
	ieee8021AsV2PortDSInitialComputeNbrRateRatio	14.8.31
	ieee8021AsV2PortDSCurrentComputeNbrRateRatio	14.8.32
	ieee8021AsV2PortDSUseMgtSettableComputeNbrRateRatio	14.8.33
	ieee8021AsV2PortDSMgtSettableComputeNbrRateRatio	14.8.34
	ieee8021AsV2PortDSInitialComputeMeanLinkDelay	14.8.35
	ieee8021AsV2PortDSCurrentComputeMeanLinkDelay	14.8.36
	ieee8021AsV2PortDSUseMgtSettableComputeMeanLinkDelay	14.8.37
	ieee8021AsV2PortDSMgtSettableComputeMeanLinkDelay	14.8.38
	ieee8021AsV2PortDSAllowedLostRsp	14.8.39
	ieee8021AsV2AllowedFaults	14.8.40
	ieee8021AsV2LogGptpCapableMessageInterval	14.8.41
	ieee8021AsV2GPtpCapableReceiptTimeout	14.8.42
	ieee8021AsV2PortDSVersionNumber	14.8.43
	ieee8021AsV2PortDSNup	14.8.44
	ieee8021AsV2PortDSNdown	14.8.45
	ieee8021AsV2PortDSOneStepTxOper	14.8.46
	ieee8021AsV2PortDSOneStepReceive	14.8.47
	ieee8021AsV2PortDSOneStepTransmit	14.8.48
	ieee8021AsV2PortDSInitialOneStepTxOper	14.8.49
	ieee8021AsV2PortDSCurrentOneStepTxOper	14.8.50
	ieee8021AsV2PortDSUseMgtSettableOneStepTxOper	14.8.51
	ieee8021AsV2PortDSMgtSettableOneStepTxOper	14.8.52
	ieee8021AsV2PortDSSyncLocked	14.8.53
	ieee8021AsV2PortDSPdelayTruncTST1	14.8.54
	ieee8021AsV2PortDSPdelayTruncTST2	14.8.54
	ieee8021AsV2PortDSPdelayTruncTST3	14.8.54
	ieee8021AsV2PortDSPdelayTruncTST4	14.8.54
	ieee8021AsV2PortDSMinorVersionNumber	14.8.55
ieee8021AsV2DescriptionPortDS		descriptionPortDS Table 14-11

Table 15-1—IEEE8021-AS MIB structure and object cross reference *(continued)*

MIB table	MIB object	Reference
	ieee8021AsV2DescriptionPortDSProfileIdentifier	14.9.1
ieee8021AsV2PortStatDS		portStatisticsDS Table 14-12
	ieee8021AsV2PortStatRxSyncCount	14.10.2
	ieee8021AsV2PortStatRxOneStepSyncCount	14.10.3
	ieee8021AsV2PortStatRxFollowUpCount	14.10.4
	ieee8021AsV2PortStatRxPdelayRequestCount	14.10.5
	ieee8021AsV2PortStatRxPdelayRspCount	14.10.6
	ieee8021AsV2PortStatRxPdelayRspFollowUpCount	14.10.7
	ieee8021AsV2PortStatRxAnnounceCount	14.10.8
	ieee8021AsV2PortStatRxPtpPacketDiscardCount	14.10.9
	ieee8021AsV2PortStatSyncReceiptTimeoutCount	14.10.10
	ieee8021AsV2PortStatAnnounceReceiptTimeoutCount	14.10.11
	ieee8021AsV2PortStatPdelayAllowedLostRspExceededC ount	14.10.12
	ieee8021AsV2PortStatTxSyncCount	14.10.13
	ieee8021AsV2PortStatTxOneStepSyncCount	14.10.14
	ieee8021AsV2PortStatTxFollowUpCount	14.10.15
	ieee8021AsV2PortStatTxPdelayRequestCount	14.10.16
	ieee8021AsV2PortStatTxPdelayRspCount	14.10.17
	ieee8021AsV2PortStatTxPdelayRspFollowUpCount	14.10.18
	ieee8021AsV2PortStatTxAnnounceCount	14.10.19
ieee8021AsV2AcceptableMasterPortDS		acceptableMasterTableDS Table 14-13
	ieee8021AsV2AcceptableMasterPortDSAcceptableMaster TableEnabled	14.11.2
ieee8021AsV2ExternalPortConfigurationPortDS		externalPortConfigurationPortDS Table 14-14
	ieee8021AsV2ExternalPortConfigurationPortDSDesiredSt ate	14.12.2
ieee8021AsV2AsymMeasurementModeDS		asymmetryMeasurementModeDS Table 14-15
	ieee8021AsV2AsymMeasurementModeDSAsymMeasure mentMode	14.13.2
ieee8021AsV2CommonServicesPortDS		commonServicesPortDS Table 14-16
	ieee8021AsV2CommonServicesPortDSCmldsLinkPortPor tNumber	14.14.2
ieee8021AsV2CommonMeanLinkDelayServiceDefaultDS		cmldsDefaultDS Table 14-17
	ieee8021AsV2CmldsDefaultDSClockIdentity	14.15.1
	ieee8021AsV2CmldsDefaultDSNumberLinkPorts	14.15.2
	ieee8021AsV2CmldsDefaultDSSdoId	14.15.3
ieee8021AsV2CommonMeanLinkDelayServiceLinkPort DS		cmldsLinkPortDS Table 14-18
	ieee8021AsV2CmldsLinkPortDSClockIdentity	14.16.2
	ieee8021AsV2CmldsLinkPortDSPortNumber	14.16.2
	ieee8021AsV2CmldsLinkPortDSCmldsLinkPortEnabled	14.16.3
	ieee8021AsV2CmldsLinkPortDSIsMeasuringDelay	14.16.4
	ieee8021AsV2CmldsLinkPortDSAsCapableAcrossDomai ns	14.16.5
	ieee8021AsV2CmldsLinkPortDSMeanLinkDelay	14.16.6

Table 15-1—IEEE8021-AS MIB structure and object cross reference *(continued)*

MIB table	MIB object	Reference
	ieee8021AsV2CmldsLinkPortDSMeanLinkDelayThresh	14.16.7
	ieee8021AsV2CmldsLinkPortDSDelayAsym	14.16.8
	ieee8021AsV2CmldsLinkPortDSNbrRateRatio	14.16.9
	ieee8021AsV2CmldsLinkPortDSInitialLogPdelayReqInterval	14.16.10
	ieee8021AsV2CmldsLinkPortDSCurrentLogPdelayReqInterval	14.16.11
	ieee8021AsV2CmldsLinkPortDSUseMgtSettableLogPdelayReqInterval	14.16.12
	ieee8021AsV2CmldsLinkPortDSMgtSettableLogPdelayReqInterval	14.16.13
	ieee8021AsV2CmldsLinkPortDSInitialComputeNbrRateRatio	14.16.14
	ieee8021AsV2CmldsLinkPortDSCurrentComputeNbrRateRatio	14.16.15
	ieee8021AsV2CmldsLinkPortDSUseMgtSettableComputeNbrRateRatio	14.16.16
	ieee8021AsV2CmldsLinkPortDSMgtSettableComputeNbrRateRatio	14.16.17
	ieee8021AsV2CmldsLinkPortDSInitialComputeMeanLinkDelay	14.16.18
	ieee8021AsV2CmldsLinkPortDSCurrentComputeMeanLinkDelay	14.16.19
	ieee8021AsV2CmldsLinkPortDSUseMgtSettableComputeMeanLinkDelay	14.16.20
	ieee8021AsV2CmldsLinkPortDSMgtSettableComputeMeanLinkDelay	14.16.21
	ieee8021AsV2CmldsLinkPortDSAllowedLostRsp	14.16.22
	ieee8021AsV2CmldsLinkPortDSAllowedFaults	14.16.23
	ieee8021AsV2CmldsLinkPortDSVersionNumber	14.16.24
	ieee8021AsV2CmldsLinkPortDSPdelayTruncTST1	14.16.25
	ieee8021AsV2CmldsLinkPortDSPdelayTruncTST2	14.16.25
	ieee8021AsV2CmldsLinkPortDSPdelayTruncTST3	14.16.25
	ieee8021AsV2CmldsLinkPortDSPdelayTruncTST4	14.16.25
	ieee8021AsV2CmldsLinkPortDSMinorVersionNumber	14.16.26
ieee8021AsV2CommonMeanLinkDelayServiceLinkPort StatDS		cmldsLinkPortStatisticsDS Table 14-19
	ieee8021AsV2CmldsLinkPortStatDSRxPdelayRequestCount	14.17.2
	ieee8021AsV2CmldsLinkPortStatDSRxPdelayRspCount	14.17.3
	ieee8021AsV2CmldsLinkPortStatDSRxPdelayRspFollowUpCount	14.17.4
	ieee8021AsV2CmldsLinkPortStatDSRxPtpPacketDiscardCount	14.17.5
	ieee8021AsV2CmldsLinkPortStatDSPdelayAllowedLostRspExceededCount	14.17.6
	ieee8021AsV2CmldsLinkPortStatDSTxPdelayRequestCount	14.17.7

Table 15-1—IEEE8021-AS MIB structure and object cross reference *(continued)*

MIB table	MIB object	Reference
	ieee8021AsV2CmldsLinkPortStatDSTxPdelayRspCount	14.17.8
	ieee8021AsV2CmldsLinkPortStatDSTxPdelayRspFollowUpCount	14.17.9
ieee8021AsV2CommonMeanLinkDelayServiceAsymMeasurementModeDS		cmldsAsymmetryMeasurementModeDS Table 14-20
	ieee8021AsV2CmldsAsymMeasurementModeDSAsymMeasurementMode	14.18.2

15.5 Textual conventions defined in this MIB

The following textual conventions are defined in this MIB:

a) Ieee8021AsV2ClockIdentity. IEEE 802 MAC address represented in “canonical” order defined by IEEE Std 802® [B3], EUI-64 [B2],
b) Ieee8021AsV2GPtpProfileIdentifier. Profile identifier (see 14.9.1),
c) Ieee8021AsV2ClockClassValue. Clock class value (see 8.6.2.2),
d) Ieee8021AsV2ClockAccuracyValue. Clock accuracy value (see 8.6.2.3),
e) Ieee8021AsV2TimeSourceValue. Source of time used by grandmaster (see 8.6.2.7),
f) Ieee8021ASV2PtpTimeInterval. Time intervals in units of 2^-16 nanoseconds. (see 6.4.3.3),
g) Ieee8021ASV2PtpPortIdentity. Identifies a port of a PTP Instance (see 6.4.3.7),
h) Ieee8021ASV2ScaledNs. Represents signed values of time and time interval in units of 2^-16 ns (see 6.4.3.1),
i) Ieee8021ASV2UScaledNs. Represents unsigned values of time and time interval in units of 2^-16 ns (see 6.4.3.2),
j) Ieee8021ASV2PTPInstanceIdentifier. Entity of a single time-aware system that executes gPTP in one gPTP domain (see 7.2.1 and 8.1),
k) Ieee8021ASV2Timestamp. Value of Ieee8021ASV2Timestamp is equal to the remainder obtained upon dividing the respective timestamp, expressed in units of 2^-16 ns, by 2^48) (see 14.8.54).

15.6 IEEE 802.1AS MIB module

In the following MIB module, should any discrepancy between the DESCRIPTION text and the corresponding definition in Clause 14 occur, the definition in Clause 14 shall take precedence.

```
IEEE8021-AS-V2-MIB DEFINITIONS ::= BEGIN
-- ================================================================
-- MIB for support of 802.1AS Timing and Synchronization in
-- IEEE 802.1Q Bridged Local Area Networks
-- ================================================================

IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE, Unsigned32, Integer32, Counter32
        FROM SNMPv2-SMI              -- [RFC2578]
    TEXTUAL-CONVENTION, TruthValue, RowStatus, TimeStamp
        FROM SNMPv2-TC               -- [RFC2579]
    MODULE-COMPLIANCE, OBJECT-GROUP  -- [RFC2580]
        FROM SNMPv2-CONF
      SnmpAdminString
            FROM SNMP-FRAMEWORK-MIB      -- [RFC3411]
    InterfaceIndexOrZero
        FROM IF-MIB                  -- [RFC2863]
    Float64TC
        FROM FLOAT-TC-MIB            -- [RFC6340]
    IEEE8021BridgePortNumber
        FROM IEEE8021-TC-MIB
    ;

ieee8021AsV2TimeSyncMib MODULE-IDENTITY
    LAST-UPDATED "201811280000Z" -- November 28, 2018
    ORGANIZATION "IEEE 802.1 Working Group"
    CONTACT-INFO
            "WG-URL: http://grouper.ieee.org/groups/802/1/index.html
            WG-EMail: STDS-802-1@IEEE.ORG
            Contact: IEEE 802.1 Working Group Chair
             Postal: C/O IEEE 802.1 Working Group
                     IEEE Standards Association
                     445 Hoes Lane
                     P.O. Box 1331
                     Piscataway
                     NJ 08855-1331
                     USA
             E-mail: STDS-802-1-L@LISTSERV.IEEE.ORG"

    DESCRIPTION
            "The Management Information Base module for
            IEEE 802.1AS time-synchronization protocol."

    REVISION    "201811280000Z" -- November 28, 2018
    DESCRIPTION "Published as part of IEEE Std 802.1AS-2019, a revision.
                 This MIB module 1) adds support for multiple domains
through
                 hierarchical instances of datasets, and 2) adds common
                 service datasets that are common to all PTP Instances.

                 Unless otherwise indicated, the references in this MIB
                 module are to IEEE Std 802.1AS-2019.

                 This MIB Structure is comprised of (See 14.1):
```

```
                A) instanceList[], per PTP Instance in a system
                    a) defaultDS
                    b) currentDS
                    c) parentDS
                    d) timePropertiesDS
                    e) pathTraceDS
                    f) acceptableMasterTableDS
                    g) portList[], per PTP port (per PTP Instance)
                           1) portDS
                           2) descriptionPortDS
                           3) portStatisticsDS
                           4) acceptableMasterPortDS
                           5) externalPortConfigurationPortDS
                           6) asymmetryMeasurementModeDS
                           7) commonServicesPortDS
                B) commonServices, per PTP port.
                    a) commonMeanLinkDelayService
                           1) cmldsDefaultDS
                           2) cmldsLinkPortList[] per PTP port.
                                i) cmldsLinkPortDS
                                ii) cmldsLinkPortStatisticsDS
                                iii) cmldsAsymmetryMeasurementModeDS
                    b) <future common services can follow>

                Published as part of IEEE Std 802.1AS-REV
                Copyright (C) IEEE (2017, 2018).
                This version of this MIB module is part of IEEE Std
                802.1AS-2019; see the draft itself for full legal
                notices."

   ::= { iso(1) org(3) ieee(111)
          standards-association-numbers-series-standards (2)
          lan-man-stds (802) ieee802dot1 (1) ieee802dot1mibs (1) 33 }

ieee8021AsV2MIBObjects    OBJECT IDENTIFIER ::= {ieee8021AsV2TimeSyncMib
1}
ieee8021AsV2Conformance   OBJECT IDENTIFIER ::= {ieee8021AsV2TimeSyncMib
2}

-- ================================================================
-- Textual Conventions
-- ================================================================

Ieee8021AsV2ClockIdentity ::= TEXTUAL-CONVENTION
    DISPLAY-HINT
        "1x:"
    STATUS current
    DESCRIPTION
        "The Ieee8021AsV2ClockIdentity type identifies a PTP Instance.
         The clockIdentity attribute shall be as specified in
         IEEE Std 1588-2019."
    REFERENCE    "6.4.3.6, 8.5.2.2 and IEEE Std 1588-2019 7.5.2.2"
    SYNTAX OCTET STRING (SIZE (8))
```

```
Ieee8021AsV2GPtpProfileIdentifier ::= TEXTUAL-CONVENTION
    DISPLAY-HINT
        "1x:"
    STATUS current
    DESCRIPTION
        "The Ieee8021AsV2GPtpProfileIdentifier attribute is the
         profileIdentifier for this PTP profile."
    REFERENCE    "14.9.1, Annex F, F.1 "
    SYNTAX OCTET STRING (SIZE (6))

Ieee8021AsV2ClockClassValue ::= TEXTUAL-CONVENTION
    STATUS       current
    DESCRIPTION
        "The Ieee8021AsV2ClockClassValue attribute denotes the
traceability
         of the synchronized time distributed by a ClockMaster when it is
         the grandmaster.
         A more detailed description of clockClass can be found in
         IEEE Std 1588-2019."
    REFERENCE    "8.6.2.2 and IEEE Std 1588-2019 7.6.2.5"
    SYNTAX     INTEGER {
        primarySync(6),
        primarySyncLost(7),
        applicationSpecificSync(13),
        applicationSpecficSyncLost(14),
        primarySyncAlternativeA(52),
        applicationSpecificAlternativeA(58),
        primarySyncAlternativeB(187),
        applicationSpecficAlternativeB(193),
        defaultClock(248),
        slaveOnlyClock(255)
        }

Ieee8021AsV2ClockAccuracyValue ::= TEXTUAL-CONVENTION
    STATUS       current
    DESCRIPTION
        "The Ieee8021AsV2ClockAccuracyValue attribute indicates the
         expected time accuracy of a ClockMaster.
         A more detailed description of clockAccuracy can be found in
         IEEE Std 1588-2019."
    REFERENCE    "8.6.2.3 and IEEE Std 1588-2019 7.6.2.6"
    SYNTAX     INTEGER {
        timeAccurateTo25ns(32),
        timeAccurateTo100ns(33),
        timeAccurateTo250ns(34),
        timeAccurateTo1us(35),
        timeAccurateTo2dot5us(36),
        timeAccurateTo10us(37),
        timeAccurateTo25us(38),
        timeAccurateTo100us(39),
        timeAccurateTo250us(40),
        timeAccurateTo1ms(41),
        timeAccurateTo2dot5ms(42),
        timeAccurateTo10ms(43),
```

```
        timeAccurateTo25ms(44),
        timeAccurateTo100ms(45),
        timeAccurateTo250ms(46),
        timeAccurateTo1s(47),
        timeAccurateTo10s(48),
        timeAccurateToGT10s(49),
        timeAccurateToUnknown(254)
        }

Ieee8021AsV2TimeSourceValue ::= TEXTUAL-CONVENTION
    STATUS       current
    DESCRIPTION
        "The Ieee8021AsV2TimeSourceValue is an information only
         attribute indicating the type of source of time used by a
         ClockMaster. The value is not used in the selection of the
         grandmaster. The values of TimeSource are given below and
         are specified in Table 8-2. These represent categories. For
         example, the GPS entry includes not only the GPS system of
         the U.S. Department of Defense but the European Galileo system
         and other present and future GNSSs.

        In the absence of a default value set by a user of this standard,
         the default value of timeSource shall be INTERNAL_OSCILLATOR.

         A more detailed description of timeSource can be found in
         IEEE Std 1588-2019.

         The following interpretation is placed on the value:
            0x10:  Atomic Clock,
            0x20:  GPS,
            0x30:  Terrestrial Radio,
            0x40:  PTP,
            0x50:  NTP,
            0x60:  Hand Set,
            0x90:  Other,
            0xA0:  Internal Oscillator "
    REFERENCE  "8.6.2.7, 8-2 and IEEE Std 1588-2019 7.6.2.8"
    SYNTAX      INTEGER {
            atomicClock(16),
            gps(32),
            terrestrialRadio(48),
            ptp(64),
            ntp(80),
            handSet(96),
            other(144),
            internalOscillator(160)
           }

Ieee8021ASV2PtpTimeInterval ::= TEXTUAL-CONVENTION
    STATUS current
    DESCRIPTION
            "The Ieee8021ASV2PtpTimeInterval type represents time
intervals
             in units of 2^-16 nanoseconds. Positive or negative time
```

```
                 intervals outside the maximum range of this data type shall
                 be encoded as the largest positive and negative values of
                 the data type respectively.
                For example: 2.5 ns is expressed as: (hex) 0x0000 0000 0002
8000"
    REFERENCE "6.4.3.3"
SYNTAX OCTET STRING (SIZE (8))

Ieee8021ASV2PtpPortIdentity ::= TEXTUAL-CONVENTION
STATUS current
DESCRIPTION
                "The Ieee8021ASV2PtpPortIdentity type identifies a port of a
                 PTP Instance.
                 The first 8 octets within this value specifies the
ClockIdentity.
                 The last 2 octets within this value specifies the port
number."
    REFERENCE   "6.4.3.7"
SYNTAX OCTET STRING (SIZE (10))

Ieee8021ASV2ScaledNs ::= TEXTUAL-CONVENTION
STATUS current
DESCRIPTION
                "The Ieee8021ASV2ScaledNs type represents signed values of
                time and time interval in units of 2^-16 ns.
               Positive or negative values of time or time interval outside
the
                maximum range of this data type are encoded as the largest
                positive or negative value of the data type, respectively.
                For example: -2.5 ns is expressed as:
                (hex) 0xFFFF FFFF FFFF FFFF FFFD 8000"
    REFERENCE   "6.4.3.1"
SYNTAX OCTET STRING (SIZE (12))

Ieee8021ASV2UScaledNs ::= TEXTUAL-CONVENTION
STATUS current
DESCRIPTION
               "The Ieee8021ASV2UScaledNs type represents unsigned values of
                time and time interval in units of 2^-16 ns.
               Positive or negative values of time or time interval outside
the
                maximum range of this data type are encoded as the largest
                positive or negative value of the data type, respectively.
                For example: 2.5 ns is expressed as:
                (hex) 0x0000 0000 0000 0000 0002 8000"
    REFERENCE   "6.4.3.2"
SYNTAX OCTET STRING (SIZE (12))

Ieee8021ASV2PTPInstanceIdentifier ::= TEXTUAL-CONVENTION
DISPLAY-HINT "d"
STATUS current
DESCRIPTION
               "The entity of a single time-aware system that executes gPTP
in
```

```
            one gPTP domain is called a PTP Instance. A time-aware system
can
             contain multiple PTP Instances, which are each associated
with
             a different gPTP domain. There are two types of PTP
Instances,
             a PTP End Instance and a PTP Relay Instance."
    REFERENCE   "7.2.1"
SYNTAX Unsigned32

Ieee8021ASV2Timestamp ::= TEXTUAL-CONVENTION
STATUS current
DESCRIPTION
           "The value of Ieee8021ASV2Timestamp is equal to the remainder
obtained upon dividing the respective timestamp, expressed in
            units of 2^-16 ns, by 2^48)."
    REFERENCE   "14.8.53, 14.16.25 and Table 14-9"
SYNTAX OCTET STRING (SIZE (6))

-- ===================================================================
-- subtrees in the IEEE8021-AS-MIB
--
-- System Time-Aware Parameters/Capabilities for each instance of
-- gPTP domain. ieee8021AsV2InstanceListIndex that is of
-- ieee8021AsV2DomainIdentificationNumber object-type is used as Index.
--
-- ===================================================================

-- ===================================================================
-- The PTP Instance set is used to allow for dynamic creation and
-- deletion of PTP Instances and logical ports implementations that
-- support dynamic create/delete of devices.
-- ===================================================================

ieee8021AsV2PtpInstanceTable   OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021AsV2PtpInstanceEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "This table is used to allow for dynamic creation and deletion
        of PTP Instances and logical ports implementations that support
        dynamic create/delete of devices."
    REFERENCE   "14.1"
    ::= { ieee8021AsV2MIBObjects 1 }

ieee8021AsV2PtpInstanceEntry   OBJECT-TYPE
    SYNTAX      Ieee8021AsV2PtpInstanceEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "An entry that specifies a PTP Instance."
    INDEX { ieee8021AsV2PtpInstance }
    ::= { ieee8021AsV2PtpInstanceTable 1 }
```

```
Ieee8021AsV2PtpInstanceEntry ::=
  SEQUENCE {
    ieee8021AsV2PtpInstance           Ieee8021ASV2PTPInstanceIdentifier,
    ieee8021AsV2PtpInstanceName        SnmpAdminString,
    ieee8021AsV2PtpInstanceRowStatus   RowStatus
  }

ieee8021AsV2PtpInstance OBJECT-TYPE
    SYNTAX Ieee8021ASV2PTPInstanceIdentifier
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
    "The entity of a single time-aware system that executes gPTP in
     one gPTP domain is called a PTP Instance. A time-aware system can
     contain multiple PTP Instances, which are each associated with
     a different gPTP domain. There are two types of PTP Instances,
     a PTP End Instance and a PTP Relay Instance."
    REFERENCE "7.2.1"
    ::= { ieee8021AsV2PtpInstanceEntry 1 }

ieee8021AsV2PtpInstanceName OBJECT-TYPE
    SYNTAX SnmpAdminString
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
    "Name for identification of a PTP Instance."
    DEFVAL { "" }
    ::= { ieee8021AsV2PtpInstanceEntry 2 }

ieee8021AsV2PtpInstanceRowStatus OBJECT-TYPE
    SYNTAX RowStatus
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
    "This attribute is used to create and delete PTP Instances."
    REFERENCE "14.1"
    ::= { ieee8021AsV2PtpInstanceEntry 3 }

-- ==================================================================
-- The Default data set represent native time capability of a time-
-- aware system and is consistent with respective IEEE 1588 data set.
-- ==================================================================

ieee8021AsV2DefaultDSTable   OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021AsV2DefaultDSEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "The Default Parameter Data Set represents the native capabilities
         of a PTP Instance, i.e., a PTP Relay Instance or a
         PTP End Instance."
    REFERENCE   "14.2"
    ::= { ieee8021AsV2MIBObjects 2 }
```

```
ieee8021AsV2DefaultDSEntry   OBJECT-TYPE
    SYNTAX      Ieee8021AsV2DefaultDSEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "Default Data Set contains the profile Identifier for
        this instance of gPTP domain."
    INDEX { ieee8021AsV2PtpInstance }
    ::= { ieee8021AsV2DefaultDSTable 1 }

Ieee8021AsV2DefaultDSEntry ::=
  SEQUENCE {
    ieee8021AsV2DefaultDSClockIdentity
Ieee8021AsV2ClockIdentity,
    ieee8021AsV2DefaultDSNumberPorts            Unsigned32,
    ieee8021AsV2DefaultDSClockQualityClockClass
Ieee8021AsV2ClockClassValue,
    ieee8021AsV2DefaultDSClockQualityClockAccuracy
Ieee8021AsV2ClockAccuracyValue,
    ieee8021AsV2DefaultDSClockQualityOffsetScaledLogVariance Unsigned32,
    ieee8021AsV2DefaultDSPriority1              Unsigned32,
    ieee8021AsV2DefaultDSPriority2              Unsigned32,
    ieee8021AsV2DefaultDSGmCapable              TruthValue,
    ieee8021AsV2DefaultDSCurrentUtcOffset       Integer32,
    ieee8021AsV2DefaultDSCurrentUtcOffsetValid  TruthValue,
    ieee8021AsV2DefaultDSLeap59                 TruthValue,
    ieee8021AsV2DefaultDSLeap61                 TruthValue,
    ieee8021AsV2DefaultDSTimeTraceable          TruthValue,
    ieee8021AsV2DefaultDSFrequencyTraceable     TruthValue,
    ieee8021AsV2DefaultDSPtpTimescale           TruthValue,
    ieee8021AsV2DefaultDSTimeSource
Ieee8021AsV2TimeSourceValue,
    ieee8021AsV2DefaultDSDomainNumber           Unsigned32,
    ieee8021AsV2DefaultDSSdoId                  Unsigned32,
    ieee8021AsV2DefaultDSExternalPortConfigurationEnabled TruthValue,
    ieee8021AsV2DefaultDSInstanceEnable         TruthValue
    }

ieee8021AsV2DefaultDSClockIdentity OBJECT-TYPE
    SYNTAX      Ieee8021AsV2ClockIdentity
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
              "The value is the clockIdentity of the local clock.
               The clockIdentity attribute shall be as specified in
               IEEE Std 1588-2019."
    REFERENCE   "14.2.1 and IEEE Std 1588-2019 7.5.2.2"
    ::= { ieee8021AsV2DefaultDSEntry 1 }


ieee8021AsV2DefaultDSNumberPorts OBJECT-TYPE
    SYNTAX      Unsigned32(1..65535)
    MAX-ACCESS  read-only
    STATUS      current
```

```
    DESCRIPTION
                "The number of ports of the PTP Instance. For an end
                station the value is 1."
    REFERENCE   "14.2.2"
    ::= { ieee8021AsV2DefaultDSEntry 2 }


ieee8021AsV2DefaultDSClockQualityClockClass OBJECT-TYPE
    SYNTAX      Ieee8021AsV2ClockClassValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                "The value is the clockClass of the PTP Instance, which
                 implements the clockClass specifications of 8.6.2.2."
    REFERENCE   "14.2.3.1"
    ::= { ieee8021AsV2DefaultDSEntry 3 }


ieee8021AsV2DefaultDSClockQualityClockAccuracy OBJECT-TYPE
    SYNTAX      Ieee8021AsV2ClockAccuracyValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
              "The value is the clockAccuracy of the PTP Instance, which
implements the clockAccuracy specifications of 8.6.2.3."
    REFERENCE   "14.2.3.2"
    ::= { ieee8021AsV2DefaultDSEntry 4 }


ieee8021AsV2DefaultDSClockQualityOffsetScaledLogVariance OBJECT-TYPE
    SYNTAX      Unsigned32(0..65535)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
             "The value is the offsetScaledLogVariance of the PTP
             Instance, which implements the offsetScaledLogVariance
             specifications of 8.6.2.4."
    REFERENCE   "14.2.3.3"
    ::= { ieee8021AsV2DefaultDSEntry 5 }

ieee8021AsV2DefaultDSPriority1 OBJECT-TYPE
    SYNTAX      Unsigned32(0..255)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
                "The value is the priority1 attribute of the PTP
Instance."
    REFERENCE   "14.2.4"
    ::= { ieee8021AsV2DefaultDSEntry 6 }


ieee8021AsV2DefaultDSPriority2 OBJECT-TYPE
    SYNTAX      Unsigned32(0..255)
    MAX-ACCESS  read-write
```

```
    STATUS       current
    DESCRIPTION
                 "The value is the priority2 attribute of the PTP
Instance."
    REFERENCE    "14.2.5"
    DEFVAL { 248 }
    ::= { ieee8021AsV2DefaultDSEntry 7 }


ieee8021AsV2DefaultDSGmCapable OBJECT-TYPE
    SYNTAX       TruthValue
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
                 "The value is TRUE (1) if the PTP Instance is capable of
                  being a grandmaster, and FALSE (2) if the PTP Instance
                  is not capable of being a grandmaster."
    REFERENCE    "14.2.6"
    ::= { ieee8021AsV2DefaultDSEntry 8 }


ieee8021AsV2DefaultDSCurrentUtcOffset OBJECT-TYPE
    SYNTAX       Integer32(-32768..32767)
    UNITS        "seconds"
    MAX-ACCESS   read-write
    STATUS       current
    DESCRIPTION
                "The value is the offset between TAI and UTC, relative to
                 the ClockMaster entity of this PTP Instance. It is equal
                  to the global variable sysCurrentUtcOffset.
                  The value is in units of seconds.

                  The default value is selected as follows:
                     a)The value is the value obtained from a primary
                      reference if the value is known at the time of
                      initialization, else
                      b)The value is the current IERS defined value of
                     TAI - UTC (see IERS Bulletin C) when the PTP Instance
                       is designed.currentUtcOffsetValid"
    REFERENCE    "14.2.7"
    ::= { ieee8021AsV2DefaultDSEntry 9 }


ieee8021AsV2DefaultDSCurrentUtcOffsetValid OBJECT-TYPE
    SYNTAX       TruthValue
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
                 "The default value is TRUE (1) if the value of
                  ieee8021AsV2DefaultDSCurrentUtcOffset is known to be
                  correct, otherwise it is set to FALSE (2)."
    REFERENCE    "14.2.7 b)"
    ::= { ieee8021AsV2DefaultDSEntry 10 }
```

```
ieee8021AsV2DefaultDSLeap59 OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
                "A TRUE (1) value indicates that the last minute of the
                 current UTC day, relative to the ClockMaster entity of
                this PTP Instance, will contain 59 s. It is equal to the
                 global variable sysLeap59.

                 The value is selected as follows:
                    a)The value is obtained from a primary reference if
                     known at the time of initialization, else
                    b)The value is set to FALSE (2)."
    REFERENCE   "14.2.8"
    ::= { ieee8021AsV2DefaultDSEntry 11 }


ieee8021AsV2DefaultDSLeap61 OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
                "A TRUE (1) value indicates that the last minute of the
                 current UTC day, relative to the ClockMaster entity of
                this PTP Instance, will contain 61 s. It is equal to the
global variable sysLeap61.

                 The value is selected as follows:
                     a)The value is obtained from a primary reference if
                      known at the time of initialization, else
                     b)The value is set to FALSE (2)."
    REFERENCE   "14.2.9"
    ::= { ieee8021AsV2DefaultDSEntry 12 }


ieee8021AsV2DefaultDSTimeTraceable OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                "The value is set to TRUE (1) if the timescale and the
value
                of currentUtcOffset, relative to the ClockMaster entity
of
                this PTP Instance, are traceable to a primary reference
standard; otherwise the value is set to FALSE (2).
                 It is equal to the global variable sysTimeTraceable.

                 The value is selected as follows:
                    a)If the time and the value of currentUtcOffset
                     are traceable to a primary reference standard at the
                    time of initialization, the value is set to TRUE (1),
```

```
                      else
                     b)The value is set to FALSE (2)."
        REFERENCE   "14.2.10"
        ::= { ieee8021AsV2DefaultDSEntry 13 }


    ieee8021AsV2DefaultDSFrequencyTraceable OBJECT-TYPE
        SYNTAX      TruthValue
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
                   "The value is set to TRUE (1) if the frequency determining
    the
                    timescale of the ClockMaster Entity of this PTP Instance
    is
                    traceable to a primary standard; otherwise the value is
    set
                     to FALSE (2). It is equal to the global variable
                     sysFrequencyTraceable.

                     The value is selected as follows:
                       a)If the frequency is traceable to a primary reference
                         standard at the time of initialization the value is
    set
                          to TRUE (1), else
                         b)The value is set to FALSE (2)."
        REFERENCE   "14.2.11"
        ::= { ieee8021AsV2DefaultDSEntry 14 }

    ieee8021AsV2DefaultDSPtpTimescale OBJECT-TYPE
        SYNTAX      TruthValue
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
                    "The value is set to TRUE (1) if the clock timescale of
    the
                     ClockMaster Entity of this PTP Instance is PTP and
                     FALSE (2) otherwise."
        REFERENCE   "14.2.12"
        ::= { ieee8021AsV2DefaultDSEntry 15 }

    ieee8021AsV2DefaultDSTimeSource OBJECT-TYPE
        SYNTAX      Ieee8021AsV2TimeSourceValue
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
                    "The value is the source of time used by the grandmaster
                     clock."
        REFERENCE   "14.2.13"
        ::= { ieee8021AsV2DefaultDSEntry 16 }

    ieee8021AsV2DefaultDSDomainNumber OBJECT-TYPE
        SYNTAX      Unsigned32(0..127)
        MAX-ACCESS  read-write
```

```
    STATUS      current
    DESCRIPTION
                "The value is the domain number of the gPTP domain for
this
                 instance of gPTP supported by the time-aware system."
    REFERENCE   "14.2.14"
    ::= { ieee8021AsV2DefaultDSEntry 17 }

ieee8021AsV2DefaultDSSdoId OBJECT-TYPE
    SYNTAX      Unsigned32(0..4095)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                "The value is the sdoId of the gPTP domain for this
instance
                 of gPTP supported by the time-aware system.
                 For compatibility with IEEE Std 1588, the range of the
                managed object is limited to 12 bits; in addition, only
the
                single value 0x100 is specified in this standard for the
                 gPTP domain of a PTP Instance."
    REFERENCE   "14.2.15"
    ::= { ieee8021AsV2DefaultDSEntry 18 }

ieee8021AsV2DefaultDSExternalPortConfigurationEnabled OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
                "The value is the externalPortConfigurationEnabled
attribute
                 of the PTP Instance."
    REFERENCE   "14.2.16"
    ::= { ieee8021AsV2DefaultDSEntry 19 }

ieee8021AsV2DefaultDSInstanceEnable OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
                "The value is the instanceEnable attribute of the PTP
Instance."
    REFERENCE   "14.2.17"
    ::= { ieee8021AsV2DefaultDSEntry 20 }

-- ===================================================================
-- The Current data set represent this system's topological location
-- relative to the known grandmaster system.
-- This data set is consistent with respective IEEE 1588 data set.
-- ===================================================================

ieee8021AsV2CurrentDSTable      OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021AsV2CurrentDSEntry
    MAX-ACCESS  not-accessible
```

```
    STATUS      current
    DESCRIPTION
       "The Current Parameter Data Set represents the position of a local
        system and other information, relative to the grandmaster.
       "
    REFERENCE   "14.3"
    ::= { ieee8021AsV2MIBObjects 3 }

ieee8021AsV2CurrentDSEntry   OBJECT-TYPE
    SYNTAX      Ieee8021AsV2CurrentDSEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "Current Data Set for a specific PTP Instance."
    INDEX { ieee8021AsV2PtpInstance }
    ::= { ieee8021AsV2CurrentDSTable 1 }

Ieee8021AsV2CurrentDSEntry ::=
  SEQUENCE {
    ieee8021AsV2CurrentDSStepsRemoved                     Unsigned32,
    ieee8021AsV2CurrentDSOffsetFromMaster
Ieee8021ASV2PtpTimeInterval,
    ieee8021AsV2CurrentDSLastGmPhaseChange
Ieee8021ASV2ScaledNs,
    ieee8021AsV2CurrentDSLastGmFreqChange                 Float64TC,
    ieee8021AsV2CurrentDSGmTimebaseIndicator              Unsigned32,
    ieee8021AsV2CurrentDSGmChangeCount                    Counter32,
    ieee8021AsV2CurrentDSTimeOfLastGmChangeEvent          TimeStamp,
    ieee8021AsV2CurrentDSTimeOfLastGmPhaseChangeEvent     TimeStamp,
    ieee8021AsV2CurrentDSTimeOfLastGmFreqChangeEvent      TimeStamp
    }

ieee8021AsV2CurrentDSStepsRemoved OBJECT-TYPE
    SYNTAX      Unsigned32(0..65535)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
              "The value is the number of gPTP communication paths
               traversed between the local clock and the grandmaster
               clock, as specified in 10.3.3.

               For example, stepsRemoved for a slave clock on the same
              gPTP communication path as the grandmaster clock will have
              a value of 1, indicating that a single path was traversed."
    REFERENCE   "14.3.1"
    ::= { ieee8021AsV2CurrentDSEntry 1 }


ieee8021AsV2CurrentDSOffsetFromMaster OBJECT-TYPE
    SYNTAX      Ieee8021ASV2PtpTimeInterval
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
              "The value is an implementation-specific representation of
```

```
                    the current value of the time difference between a slave
                    and the grandmaster, as computed by the slave, and as
specified in 10.2.10."
    REFERENCE   "14.3.2"
    ::= { ieee8021AsV2CurrentDSEntry 2 }

ieee8021AsV2CurrentDSLastGmPhaseChange OBJECT-TYPE
    SYNTAX      Ieee8021ASV2ScaledNs
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
               "The value is the phase change that occurred on the most
               recent change in either grandmaster or
gmTimeBaseIndicator."
    REFERENCE   "14.3.3"
    ::= { ieee8021AsV2CurrentDSEntry 3 }

ieee8021AsV2CurrentDSLastGmFreqChange OBJECT-TYPE
    SYNTAX      Float64TC
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
               "The value is the frequency change that occurred on the
most
                recent change in either grandmaster or
gmTimeBaseIndicator."
    REFERENCE   "14.3.4"
    ::= { ieee8021AsV2CurrentDSEntry 4 }

ieee8021AsV2CurrentDSGmTimebaseIndicator OBJECT-TYPE
    SYNTAX      Unsigned32(0..65535)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
               "The value is the value of timeBaseIndicator of the
                current grandmaster."
    REFERENCE   "14.3.5"
    ::= { ieee8021AsV2CurrentDSEntry 5 }

ieee8021AsV2CurrentDSGmChangeCount OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                "This statistics counter tracks the number of times the
                 grandmaster has changed in a gPTP domain. This counter
               increments when the PortAnnounceInformation state machine
                 enters the SUPERIOR_MASTER_PORT state or the
                 INFERIOR_MASTER_OR_OTHER_PORT state."
    REFERENCE   "14.3.6"
    ::= { ieee8021AsV2CurrentDSEntry 6 }

ieee8021AsV2CurrentDSTimeOfLastGmChangeEvent OBJECT-TYPE
    SYNTAX      TimeStamp
```

```
    UNITS       "0.01 seconds"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
              "This timestamp takes the value of sysUpTime (see RFC3418)
when
                the most recent grandmaster change occurred in a gPTP
domain.
               This timestamp is updated when the PortAnnounceInformation
               state machine enters the SUPERIOR_MASTER_PORT state or the
                INFERIOR_MASTER_OR_OTHER_PORT state."
    REFERENCE   "14.3.7"
    ::= { ieee8021AsV2CurrentDSEntry 7 }


ieee8021AsV2CurrentDSTimeOfLastGmPhaseChangeEvent OBJECT-TYPE
    SYNTAX      TimeStamp
    UNITS       "0.01 seconds"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
               "This timestamp takes the value of sysUpTime (see RFC3418)
                 when the most recent change in grandmaster phase
occurred,
                 due to a change of either the grandmaster or the
grandmaster
                 time base. This timestamp is updated when one of the
                 following occurs:
                   a)The PortAnnounceInformation state machine enters the
                      SUPERIOR_MASTER_PORT state or the
                      INFERIOR_MASTER_OR_OTHER_PORT state, or
                    b)The gmTimebaseIndicator managed object changes and
the
                    lastGmPhaseChange field of the most recently received
                      Follow_Up information TLV is nonzero."
    REFERENCE   "14.3.8"
    ::= { ieee8021AsV2CurrentDSEntry 8 }

ieee8021AsV2CurrentDSTimeOfLastGmFreqChangeEvent OBJECT-TYPE
    SYNTAX      TimeStamp
    UNITS       "0.01 seconds"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
               "This timestamp takes the value of sysUpTime (see RFC3418)
                 when the most recent change in grandmaster frequency
occurred,
                 due to a change of either the grandmaster or the
grandmaster
                 time base. This timestamp is updated when one of the
                 following occurs:
                   a)The PortAnnounceInformation state machine enters the
                      SUPERIOR_MASTER_PORT state or the
                      INFERIOR_MASTER_OR_OTHER_PORT state, or
```

```
                      b)The gmTimebaseIndicator managed object changes and
the
                       lastGmFreqChange field of the most recently received
                        Follow_Up information TLV is nonzero."
    REFERENCE   "14.3.9"
    ::= { ieee8021AsV2CurrentDSEntry 9 }

-- ===================================================================
-- The Parent data set represent timing upstream (toward grandmaster)
-- system's parameters as measured at this system.
-- This data set is consistent with respective IEEE 1588 data set.
-- ===================================================================
ieee8021AsV2ParentDSTable     OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021AsV2ParentDSEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "The Parent Parameter Data Set represents capabilities of the
        upstream system, toward the grandmaster, as measured at a local
        system."
    REFERENCE   "14.4"
    ::= { ieee8021AsV2MIBObjects 4 }

ieee8021AsV2ParentDSEntry   OBJECT-TYPE
    SYNTAX      Ieee8021AsV2ParentDSEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "Parent Data Set for a specific PTP Instance."
    INDEX { ieee8021AsV2PtpInstance }
    ::= { ieee8021AsV2ParentDSTable 1 }

Ieee8021AsV2ParentDSEntry ::=
  SEQUENCE {
    ieee8021AsV2ParentDSParentClockIdentity
Ieee8021AsV2ClockIdentity,
    ieee8021AsV2ParentDSParentPortNumber        Unsigned32,
    ieee8021AsV2ParentDSCumlativeRateRatio      Integer32,
    ieee8021AsV2ParentDSGrandmasterIdentity
Ieee8021AsV2ClockIdentity,
    ieee8021AsV2ParentDSGrandmasterClockQualityclockClass
                                             Ieee8021AsV2ClockClassValue,
    ieee8021AsV2ParentDSGrandmasterClockQualityclockAccuracy

Ieee8021AsV2ClockAccuracyValue,
    ieee8021AsV2ParentDSGrandmasterClockQualityoffsetScaledLogVar
                                                Unsigned32,
    ieee8021AsV2ParentDSGrandmasterPriority1    Unsigned32,
    ieee8021AsV2ParentDSGrandmasterPriority2    Unsigned32
    }

ieee8021AsV2ParentDSParentClockIdentity OBJECT-TYPE
    SYNTAX      Ieee8021AsV2ClockIdentity
    MAX-ACCESS  read-only
```

```
    STATUS      current
    DESCRIPTION
                "The value is the first of the parentPortIdentity
attribute
                 for this instance of gPTP domain, which is a set made of
                  Ieee8021AsV2ClockIdentity and portNumber."
    REFERENCE   "14.4.1"
    ::= { ieee8021AsV2ParentDSEntry 1 }

ieee8021AsV2ParentDSParentPortNumber OBJECT-TYPE
    SYNTAX      Unsigned32(0..65535)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                "The value is the second of the parentPortIdentity
attribute
                 for this instance of gPTP domain, which is a set made of
                  Ieee8021AsV2ClockIdentity and portNumber."
    REFERENCE   "14.4.1"
    ::= { ieee8021AsV2ParentDSEntry 2 }

ieee8021AsV2ParentDSCumlativeRateRatio OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
               "The value is an estimate of the ratio of the frequency of
                the grandmaster to the frequency of the LocalClock entity
                  of this PTP Instance.
                  CumulativeRateRatio is expressed as the fractional
                  frequency offset multiplied by 2^41, i.e., the quantity
                  (rateRatio - 1.0)(2^41), where rateRatio is computed by
                  the PortSyncSyncReceive state machine."
    REFERENCE   "14.4.2"
   ::= { ieee8021AsV2ParentDSEntry 3 }

ieee8021AsV2ParentDSGrandmasterIdentity OBJECT-TYPE
    SYNTAX      Ieee8021AsV2ClockIdentity
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                "The value is the clockIdentity attribute of the
                 grandmaster clock."
    REFERENCE   "14.4.3"
    ::= { ieee8021AsV2ParentDSEntry 4 }

ieee8021AsV2ParentDSGrandmasterClockQualityclockClass OBJECT-TYPE
    SYNTAX      Ieee8021AsV2ClockClassValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                "The value is the clockClass of the grandmaster clock."
    REFERENCE   "14.4.4.1"
    ::= { ieee8021AsV2ParentDSEntry 5 }
```

```
ieee8021AsV2ParentDSGrandmasterClockQualityclockAccuracy OBJECT-TYPE
    SYNTAX      Ieee8021AsV2ClockAccuracyValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
               "The value is the clockAccuracy of the grandmaster clock."
    REFERENCE   "14.4.4.2"
    ::= { ieee8021AsV2ParentDSEntry 6 }

ieee8021AsV2ParentDSGrandmasterClockQualityoffsetScaledLogVar
                OBJECT-TYPE
    SYNTAX      Unsigned32(0..65535)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                "The value is the offsetScaledLogVariance of the
                 grandmaster clock."
    REFERENCE   "14.4.4.3"
    ::= { ieee8021AsV2ParentDSEntry 7 }

ieee8021AsV2ParentDSGrandmasterPriority1 OBJECT-TYPE
    SYNTAX      Unsigned32(0..255)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                "The value is the priority1 attribute of the
                 grandmaster clock."
    REFERENCE   "14.4.5"
    ::= { ieee8021AsV2ParentDSEntry 8 }

ieee8021AsV2ParentDSGrandmasterPriority2 OBJECT-TYPE
    SYNTAX      Unsigned32(0..255)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
               "The value is the priority2 attribute of the
                grandmaster clock."
    REFERENCE   "14.4.6"
    ::= { ieee8021AsV2ParentDSEntry 9 }

-- ===================================================================
-- TimePropertiesDS represents the grandmaster's parameters, as
-- measured at this system and are derived from IEEE 802.1AS protocol.
-- This data set is consistent with respective IEEE 1588 data set.
-- ===================================================================
ieee8021AsV2TimePropertiesDSTable    OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021AsV2TimePropertiesDSEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "The Time Properties Parameter Data Set represents capabilities of
         the grandmaster, as measured at a local system"
    REFERENCE   "14.5"
```

```
        ::= { ieee8021AsV2MIBObjects 5 }

ieee8021AsV2TimePropertiesDSEntry   OBJECT-TYPE
    SYNTAX      Ieee8021AsV2TimePropertiesDSEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "Time Properties Data Set contains the profile Identifier for
        this instance of gPTP domain."
    INDEX { ieee8021AsV2PtpInstance }
    ::= { ieee8021AsV2TimePropertiesDSTable 1 }

Ieee8021AsV2TimePropertiesDSEntry ::=
  SEQUENCE {
    ieee8021AsV2TimePropertiesDSCurrentUtcOffset      Integer32,
    ieee8021AsV2TimePropertiesDSCurrentUtcOffsetValid TruthValue,
    ieee8021AsV2TimePropertiesDSLeap59                TruthValue,
    ieee8021AsV2TimePropertiesDSLeap61                TruthValue,
    ieee8021AsV2TimePropertiesDSTimeTraceable         TruthValue,
    ieee8021AsV2TimePropertiesDSFrequencyTraceable    TruthValue,
    ieee8021AsV2TimePropertiesDSPtpTimescale          TruthValue,
    ieee8021AsV2TimePropertiesDSTimeSource
Ieee8021AsV2TimeSourceValue
    }

ieee8021AsV2TimePropertiesDSCurrentUtcOffset OBJECT-TYPE
    SYNTAX      Integer32(-32768..32767)
    UNITS       "seconds"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                "The value is currentUtcOffset for the current
grandmaster.
                 It is equal to the value of the global variable
                 currentUtcOffset. The value is in units of seconds."
    REFERENCE   "14.5.1"
    ::= { ieee8021AsV2TimePropertiesDSEntry 1 }

ieee8021AsV2TimePropertiesDSCurrentUtcOffsetValid OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The value is currentUtcOffsetValid for the current grandmaster.
         It is equal to the global variable currentUtcOffsetValid."
    REFERENCE   "14.5.2"
    ::= { ieee8021AsV2TimePropertiesDSEntry 2 }

ieee8021AsV2TimePropertiesDSLeap59 OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                "The value is leap59 for the current grandmaster. It is
```

```
                  equal to the global variable leap59."
    REFERENCE   "14.5.3"
    ::= { ieee8021AsV2TimePropertiesDSEntry 3 }

ieee8021AsV2TimePropertiesDSLeap61 OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                "The value is leap61 for the current grandmaster. It is
                 equal to the global variable leap61."
    REFERENCE   "14.5.4"
    ::= { ieee8021AsV2TimePropertiesDSEntry 4 }

ieee8021AsV2TimePropertiesDSTimeTraceable OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
               "The value is timeTraceable for the current grandmaster.
It
                 is equal to the global variable timeTraceable."
    REFERENCE   "14.5.5"
    ::= { ieee8021AsV2TimePropertiesDSEntry 5 }

ieee8021AsV2TimePropertiesDSFrequencyTraceable OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                "The value is frequencyTraceable for the current
grandmaster.
                 It is equal to the global variable frequencyTraceable."
    REFERENCE   "14.5.6"
    ::= { ieee8021AsV2TimePropertiesDSEntry 6 }

ieee8021AsV2TimePropertiesDSPtpTimescale OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                "The value is ptpTimescale for the current grandmaster."
    REFERENCE   "14.5.7"
    ::= { ieee8021AsV2TimePropertiesDSEntry 7 }

ieee8021AsV2TimePropertiesDSTimeSource OBJECT-TYPE
    SYNTAX      Ieee8021AsV2TimeSourceValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                "The value is timeSource for the current grandmaster. It
                 is equal to the global variable timeSource"
    REFERENCE   "14.5.8"
    ::= { ieee8021AsV2TimePropertiesDSEntry 8 }
```

```
-- ===============================================================
-- The Path Trace Parameter Data set represents the current path
-- trace information available at the PTP Instance.
-- ===============================================================

ieee8021AsV2PathTraceDSTable    OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021AsV2PathTraceDSEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "The pathTraceDS represents the current path trace information
        available at the PTP Instance."
    REFERENCE   "14.6"
::= { ieee8021AsV2MIBObjects 6 }

ieee8021AsV2PathTraceDSEntry   OBJECT-TYPE
    SYNTAX      Ieee8021AsV2PathTraceDSEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "Path Trace Data Set for a specific PTP Instance."
    INDEX { ieee8021AsV2PtpInstance }
    ::= { ieee8021AsV2PathTraceDSTable 1 }

Ieee8021AsV2PathTraceDSEntry ::=
SEQUENCE
    {
                 ieee8021AsV2PathTraceDSEnable
TruthValue
    }

ieee8021AsV2PathTraceDSEnable OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                "The value is TRUE.
                 NOTE: This member is included for compatibility with
                IEEE Std 1588. In IEEE Std 1588, the path trace mechanism
                 is optional, and the pathTraceDS.enable member is
                 configurable (its value in IEEE Std 1588 is TRUE (1) or
                FALSE (2), depending on whether the path trace mechanism
is
                 operational or not operational, respectively. However,
the
                 pathTrace mechanism is mandatory in this standard, and
the
                 value of enable is always TRUE (1)."
    REFERENCE   "14.6.2"
    ::= { ieee8021AsV2PathTraceDSEntry 2 }

ieee8021AsV2PathTraceDSArrayTable    OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021AsV2PathTraceDSArrayEntry
```

```
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
           "This object contains an array of ClockIdentity values contained
            in the pathTrace array, which represents the current path trace
            information, and which is carried in the path trace TLV per
            PTP Instance."
        REFERENCE   "14.6.1"
    ::= { ieee8021AsV2MIBObjects 7 }

    ieee8021AsV2PathTraceDSArrayEntry   OBJECT-TYPE
        SYNTAX      Ieee8021AsV2PathTraceDSArrayEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
           "Path Trace Data Set Table Array for a specific PTP Instance."
        INDEX { ieee8021AsV2PtpInstance, ieee8021AsV2PathTraceDSArrayIndex }
        ::= { ieee8021AsV2PathTraceDSArrayTable 1 }

    Ieee8021AsV2PathTraceDSArrayEntry ::=
      SEQUENCE {
        ieee8021AsV2PathTraceDSArrayIndex  Unsigned32,
        ieee8021AsV2PathTraceDSArrayList      Ieee8021AsV2ClockIdentity
        }

    ieee8021AsV2PathTraceDSArrayIndex OBJECT-TYPE
        SYNTAX      Unsigned32 (1..179)
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
                    "Index of the Path Trace Data Set Array."
        REFERENCE   "10.3.9.23"
        ::= { ieee8021AsV2PathTraceDSArrayEntry 1 }

    ieee8021AsV2PathTraceDSArrayList OBJECT-TYPE
        SYNTAX      Ieee8021AsV2ClockIdentity
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
                   "The value is the array of ClockIdentity values contained
                     in the pathTrace array, which represents the current
                    path trace information, and which is carried in the path
                     trace TLV."
        REFERENCE   "14.6.1"
        ::= { ieee8021AsV2PathTraceDSArrayEntry 2 }


    -- ******************************************************************
    -- The Acceptable Master Table Parameter Data Set represents the
    -- acceptable master table used when an EPON port is used by a PTP
    -- Instance of a time-aware system.
    -- ******************************************************************

    ieee8021AsV2AcceptableMasterTableDSTable    OBJECT-TYPE
```

```
    SYNTAX      SEQUENCE OF Ieee8021AsV2AcceptableMasterTableDSEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "The acceptableMasterTableDS represents the acceptable master
        table used when an EPON port is used by a PTP Instance of a
        time-aware system."
    REFERENCE   "14.7"
::= { ieee8021AsV2MIBObjects 8 }

ieee8021AsV2AcceptableMasterTableDSEntry   OBJECT-TYPE
    SYNTAX      Ieee8021AsV2AcceptableMasterTableDSEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "Acceptable Master Table Data Set represents the acceptable master
        table used when an EPON port is used by a PTP Instance of a
        time-aware system."
    INDEX { ieee8021AsV2PtpInstance }
    ::= { ieee8021AsV2AcceptableMasterTableDSTable 1 }

Ieee8021AsV2AcceptableMasterTableDSEntry ::=
  SEQUENCE {
                ieee8021AsV2AcceptableMasterTableDSMaxTableSize
Unsigned32,
                ieee8021AsV2AcceptableMasterTableDSActualTableSize
Unsigned32
    }

ieee8021AsV2AcceptableMasterTableDSMaxTableSize  OBJECT-TYPE
    SYNTAX      Unsigned32(0..65535)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                "The value is the maximum size of the
AcceptableMasterTable.
                 It is equal to the maxTableSize member of the
                 AcceptableMasterTable structure."
    REFERENCE   "14.7.1"
    ::= { ieee8021AsV2AcceptableMasterTableDSEntry 1 }

ieee8021AsV2AcceptableMasterTableDSActualTableSize  OBJECT-TYPE
    SYNTAX      Unsigned32(0..65535)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
                "The value is the actual size of the
AcceptableMasterTable.
                 It is equal to the actualTableSize member of the
               AcceptableMasterTable structure, i.e., the current number
                 of elements in the acceptable master array. The actual
                table size is less than or equal to the max table size."
    REFERENCE   "14.7.2"
    ::= { ieee8021AsV2AcceptableMasterTableDSEntry 2 }
```

```
ieee8021AsV2AcceptableMasterTableDSArrayTable    OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021AsV2AcceptableMasterTableDSArrayEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "The acceptableMasterTableDS represents the acceptable master
table
        used when an EPON port is used by a PTP Instance of a time-aware
        system."
    REFERENCE   "14.7"
::= { ieee8021AsV2MIBObjects 9 }

ieee8021AsV2AcceptableMasterTableDSArrayEntry   OBJECT-TYPE
    SYNTAX      Ieee8021AsV2AcceptableMasterTableDSArrayEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "Each element of this array is an AcceptableMaster structure per
PTP Instance."
    INDEX { ieee8021AsV2PtpInstance,
ieee8021AsV2AcceptableMasterTableDSArrayIndex }
    ::= { ieee8021AsV2AcceptableMasterTableDSArrayTable 1 }

Ieee8021AsV2AcceptableMasterTableDSArrayEntry ::=
  SEQUENCE {
              ieee8021AsV2AcceptableMasterTableDSArrayIndex
Unsigned32,
                ieee8021AsV2AcceptableMasterTableDSArrayPortIdentity
Ieee8021ASV2PtpPortIdentity,

ieee8021AsV2AcceptableMasterTableDSArrayAlternatePriority1
Unsigned32
    }

ieee8021AsV2AcceptableMasterTableDSArrayIndex  OBJECT-TYPE
    SYNTAX      Unsigned32(0..65535)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
                "Index of the Acceptable Master Table Data Set Array."
    REFERENCE   "14.7.3"
    ::= { ieee8021AsV2AcceptableMasterTableDSArrayEntry 1 }

ieee8021AsV2AcceptableMasterTableDSArrayPortIdentity  OBJECT-TYPE
    SYNTAX      Ieee8021ASV2PtpPortIdentity
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
               "The acceptablePortIdentity member is the PortIdentity of
                an acceptable master port."
    REFERENCE   "14.7.3"
    ::= { ieee8021AsV2AcceptableMasterTableDSArrayEntry 2 }
```

```
ieee8021AsV2AcceptableMasterTableDSArrayAlternatePriority1  OBJECT-TYPE
    SYNTAX      Unsigned32(0..255)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
               "The alternatePriority1 member contains an alternate value
                 for the priority1 attribute of the acceptable master
port."
    REFERENCE   "14.7.3"
    ::= { ieee8021AsV2AcceptableMasterTableDSArrayEntry 3 }

-- ==============================================================
-- The Port Parameter Data Set (portDS) represents time-aware
-- port capabilities for a PTP Instance of a time-aware system.
-- ==============================================================
ieee8021AsV2PortDSTable   OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021AsV2PortDSEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "For the single port of a PTP End Instance and for each port of a
        PTP Relay Instance , the following portDS is maintained as the
        basis for protocol decisions and providing values for message
fields.
        The number of such data sets is the same as the value of
        defaultDS.numberPorts."
    REFERENCE   "14.8"
    ::= { ieee8021AsV2MIBObjects 10 }

ieee8021AsV2PortDSEntry   OBJECT-TYPE
    SYNTAX      Ieee8021AsV2PortDSEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "A list of objects pertaining to a gPTP Port of a PTP Instance."
    INDEX { ieee8021AsV2PtpInstance,
            ieee8021AsV2PortDSIndex }
    ::= { ieee8021AsV2PortDSTable 1 }

Ieee8021AsV2PortDSEntry ::=
     SEQUENCE {
         ieee8021AsV2BridgeBasePort          IEEE8021BridgePortNumber,
         ieee8021AsV2PortDSIndex         InterfaceIndexOrZero,
         ieee8021AsV2PortDSClockIdentity
Ieee8021AsV2ClockIdentity,
         ieee8021AsV2PortDSPortNumber                  Unsigned32,
         ieee8021AsV2PortDSPortState                    INTEGER,
         ieee8021AsV2PortDSPtpPortEnabled              TruthValue,
         ieee8021AsV2PortDSDelayMechanism              INTEGER,
         ieee8021AsV2PortDSIsMeasuringDelay             TruthValue,
         ieee8021AsV2PortDSAsCapable                    TruthValue,
         ieee8021AsV2PortDSMeanLinkDelay
Ieee8021ASV2UScaledNs,
```

```
         ieee8021AsV2PortDSMeanLinkDelayThresh
Ieee8021ASV2UScaledNs,
         ieee8021AsV2PortDSDelayAsym               Ieee8021ASV2ScaledNs,
         ieee8021AsV2PortDSNbrRateRatio            Integer32,
         ieee8021AsV2PortDSInitialLogAnnounceInterval   Integer32,
         ieee8021AsV2PortDSCurrentLogAnnounceInterval   Integer32,
         ieee8021AsV2PortDSUseMgtSettableLogAnnounceInterval
TruthValue,
         ieee8021AsV2PortDSMgtSettableLogAnnounceInterval
Integer32,
         ieee8021AsV2PortDSAnnounceReceiptTimeout       Unsigned32,
         ieee8021AsV2PortDSInitialLogSyncInterval       Integer32,
         ieee8021AsV2PortDSCurrentLogSyncInterval       Integer32,
         ieee8021AsV2PortDSUseMgtSettableLogSyncInterval
TruthValue,
        ieee8021AsV2PortDSMgtSettableLogSyncInterval          Integer32,
         ieee8021AsV2PortDSSyncReceiptTimeout           Unsigned32,
         ieee8021AsV2PortDSSyncReceiptTimeoutTimeInterval
Ieee8021ASV2UScaledNs,
         ieee8021AsV2PortDSInitialLogPdelayReqInterval Integer32,
         ieee8021AsV2PortDSCurrentLogPdelayReqInterval Integer32,
         ieee8021AsV2PortDSUseMgtSettableLogPdelayReqInterval
TruthValue,
         ieee8021AsV2PortDSMgtSettableLogPdelayReqInterval
Integer32,
         ieee8021AsV2PortDSInitialLogGptpCapableMessageInterval
Integer32,
         ieee8021AsV2PortDSCurrentLogGptpCapableMessageInterval
Integer32,
         ieee8021AsV2PortDSUseMgtSettableLogGptpCapableMessageInterval
TruthValue,
         ieee8021AsV2PortDSMgtSettableLogGptpCapableMessageInterval
Integer32,
         ieee8021AsV2PortDSInitialComputeNbrRateRatio     Integer32,
         ieee8021AsV2PortDSCurrentComputeNbrRateRatio     Integer32,
         ieee8021AsV2PortDSUseMgtSettableComputeNbrRateRatio
TruthValue,
         ieee8021AsV2PortDSMgtSettableComputeNbrRateRatio     Integer32,
         ieee8021AsV2PortDSInitialComputeMeanLinkDelay     Integer32,
         ieee8021AsV2PortDSCurrentComputeMeanLinkDelay     Integer32,
         ieee8021AsV2PortDSUseMgtSettableComputeMeanLinkDelay
TruthValue,
        ieee8021AsV2PortDSMgtSettableComputeMeanLinkDelay     Integer32,
         ieee8021AsV2PortDSAllowedLostRsp        Unsigned32,
         ieee8021AsV2PortDSAllowedFaults               Unsigned32,
         ieee8021AsV2PortDSLogGptpCapableMessageInterval Integer32,
         ieee8021AsV2PortDSGPtpCapableReceiptTimeout     Unsigned32,
         ieee8021AsV2PortDSVersionNumber               Unsigned32,
         ieee8021AsV2PortDSNup                       Float64TC,
         ieee8021AsV2PortDSNdown                     Float64TC,
         ieee8021AsV2PortDSOneStepTxOper               TruthValue,
         ieee8021AsV2PortDSOneStepReceive              TruthValue,
         ieee8021AsV2PortDSOneStepTransmit             TruthValue,
         ieee8021AsV2PortDSInitialOneStepTxOper        TruthValue,
```

```
          ieee8021AsV2PortDSCurrentOneStepTxOper        TruthValue,
          ieee8021AsV2PortDSUseMgtSettableOneStepTxOper
TruthValue,
         ieee8021AsV2PortDSMgtSettableOneStepTxOper           TruthValue,
          ieee8021AsV2PortDSSyncLocked                  TruthValue,
          ieee8021AsV2PortDSPdelayTruncTST1
Ieee8021ASV2Timestamp,
          ieee8021AsV2PortDSPdelayTruncTST2
Ieee8021ASV2Timestamp,
          ieee8021AsV2PortDSPdelayTruncTST3
Ieee8021ASV2Timestamp,
          ieee8021AsV2PortDSPdelayTruncTST4
Ieee8021ASV2Timestamp,
          ieee8021AsV2PortDSMinorVersionNumber          Unsigned32
       }

ieee8021AsV2BridgeBasePort OBJECT-TYPE
    SYNTAX      IEEE8021BridgePortNumber
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
               "This object identifies the bridge port number of
                the port for which this entry contains bridge management
                information.  For end stations, this port number shall
                be (1)."
    REFERENCE   "IEEE Std 802.1AS PortDS Group gPTP Port Index"
    ::= { ieee8021AsV2PortDSEntry 1 }

ieee8021AsV2PortDSIndex OBJECT-TYPE
    SYNTAX      InterfaceIndexOrZero
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
               "This object identifies the gPTP interface group within
                the system for which this entry contains information. It
                is the value of the instance of the IfIndex object,
                defined in the IF-MIB, for the gPTP interface group
                corresponding to this port, or the value 0 if the port
                has not been bound to an underlying frame source and
                sink.

                For a given media port of a Bridge or an end station,
                there can be one or more gPTP Port, and depends whether
                a media port supports point to point link (e.g. IEEE
                802.3 Ethernet) or point to multi-point (e.g. CSN, IEEE
                802.3 EPON, etc) links on the media port."
    REFERENCE   "IEEE Std 802.1AS PortDS Group gPTP Port Index"
    ::= { ieee8021AsV2PortDSEntry 2 }

ieee8021AsV2PortDSClockIdentity OBJECT-TYPE
    SYNTAX      Ieee8021AsV2ClockIdentity
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
```

```
                   "The value is the first of the portIdentity attribute
                    of the local port, which is a set made of
                    Ieee8021AsV2ClockIdentity and portNumber."
        REFERENCE   "14.8.2"
        ::= { ieee8021AsV2PortDSEntry 3 }


    ieee8021AsV2PortDSPortNumber OBJECT-TYPE
        SYNTAX      Unsigned32(0..65535)
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
                   "The value is the second of the portIdentity attribute
                    of the local port, which is a set made of
                    Ieee8021AsV2ClockIdentity and portNumber."
        REFERENCE   "14.8.2"
        ::= { ieee8021AsV2PortDSEntry 4 }

    ieee8021AsV2PortDSPortState OBJECT-TYPE
        SYNTAX      INTEGER {
                disabledPort(3),
                masterPort(6),
                passivePort(7),
                slavePort(9)
               }
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
                   "The value is the value of the port state of this port
                   (see Table 10-2) and is taken from the enumeration in
                   Table 14-7. It is equal to the value of the global
    variable selectedState."
        REFERENCE   "14.8.3"
        ::= { ieee8021AsV2PortDSEntry 5 }

    ieee8021AsV2PortDSPtpPortEnabled OBJECT-TYPE
        SYNTAX      TruthValue
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
                   "The value is equal to the value of the Boolean
    ptpPortEnabled."
        REFERENCE   "14.8.4"
        ::= { ieee8021AsV2PortDSEntry 6 }

    ieee8021AsV2PortDSDelayMechanism OBJECT-TYPE
        SYNTAX      INTEGER {
                p2p(2),
                commonp2p(3),
                special(4)
               }
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
```

```
                "The value indicates the mechanism for measuring mean
                 propagation delay and neighbor rate ratio on the link
                attached to this port, and is taken from the enumeration
                 in Table 14-8. If the domain number is not 0,
portDS.delay mechanism must not be P2P."
    REFERENCE   "14.8.5"
    ::= { ieee8021AsV2PortDSEntry 7 }

ieee8021AsV2PortDSIsMeasuringDelay OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                "The value is equal to the value of the Boolean
                 isMeasuringDelay."
    REFERENCE   "14.8.6"
    ::= { ieee8021AsV2PortDSEntry 8 }


ieee8021AsV2PortDSAsCapable OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                "The value is equal to the value of the Boolean
asCapable."
    REFERENCE   "14.8.7"
    ::= { ieee8021AsV2PortDSEntry 9 }

ieee8021AsV2PortDSMeanLinkDelay OBJECT-TYPE
    SYNTAX      Ieee8021ASV2UScaledNs
    UNITS       "2**-16 ns * 2**64"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
               "The value is equal to the value of the per-port global
                variable meanLinkDelay. It is an estimate of the current
                one-way propagation time on the link attached to this
port,
              measured as specified for the respective medium. The value
is
               zero for ports attached to IEEE 802.3 EPON links and for
the
                master port of an IEEE 802.11 link, because one-way
propagation
                delay is not measured on the latter and not directly
measured
                on the former."
    REFERENCE   "14.8.8"
    ::= { ieee8021AsV2PortDSEntry 10 }


ieee8021AsV2PortDSMeanLinkDelayThresh OBJECT-TYPE
    SYNTAX      Ieee8021ASV2UScaledNs
```

```
    UNITS       "2**-16 ns * 2 ** 64"
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
               "The value is equal to the value of the per-port global
                variable meanLinkDelayThresh. It is the propagation time
               threshold, above which a port is not considered capable of
                participating in the IEEE 802.1AS protocol."
    REFERENCE   "14.8.9"
    ::= { ieee8021AsV2PortDSEntry 11 }

ieee8021AsV2PortDSDelayAsym OBJECT-TYPE
    SYNTAX      Ieee8021ASV2ScaledNs
    UNITS       "2**-16 ns * 2**64"
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
                "The value is the asymmetry in the propagation delay on
                 the link attached to this port relative to the
grandmaster
                time base, as defined in 10.2.5.9 and 8.3. If propagation
                 delay asymmetry is not modeled, then delayAsymmetry is
0."
    REFERENCE   "14.8.10 and 8.3"
    ::= { ieee8021AsV2PortDSEntry 12 }

ieee8021AsV2PortDSNbrRateRatio OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
               "The value is an estimate of the ratio of the frequency of
                 the LocalClock entity of the PTP Instance at the other
end
                 of the link attached to this port, to the frequency of
the
                LocalClock entity of this PTP Instance. neighborRateRatio
                 is expressed as the fractional frequency offset
multiplied
                 by 241, i.e., the quantity (neighborRateRatio -
1.0)(2^41)."
    REFERENCE   "14.8.11"
    ::= { ieee8021AsV2PortDSEntry 13 }


ieee8021AsV2PortDSInitialLogAnnounceInterval OBJECT-TYPE
    SYNTAX      Integer32(-128..127)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
                "If useMgtSettableLogAnnounceInterval is FALSE (2), the
                value is the logarithm to base 2 of the announce interval
                 used when (a) the port is initialized, or (b) a message
```

```
                 interval request TLV is received with the
logAnnounceInterval
                 field set to 126."
    REFERENCE   "14.8.12"
    DEFVAL { 0 }
    ::= { ieee8021AsV2PortDSEntry 14 }

ieee8021AsV2PortDSCurrentLogAnnounceInterval OBJECT-TYPE
    SYNTAX      Integer32(-128..127)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                "The value is the logarithm to the base 2 of the
                 current announce interval."
    REFERENCE   "14.8.13"
    ::= { ieee8021AsV2PortDSEntry 15 }

ieee8021AsV2PortDSUseMgtSettableLogAnnounceInterval OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
                "The managed object is a Boolean that determines the
                 source of the announce interval. If the value is TRUE
(1),
                the value of currentLogAnnounceInterval is set equal to
the
                value of mgtSettableLogAnnounceInterval. If the value is
                 FALSE (2), the value of currentLogAnnounceInterval is
               determined by the AnnounceIntervalSetting state machine.
The
                 default value of useMgtSettableLogAnnounceInterval is
                 FALSE (2) for domain 0 and TRUE (1) for domains other
than
                 domain 0."
    REFERENCE   "14.8.14"
    ::= { ieee8021AsV2PortDSEntry 16 }

ieee8021AsV2PortDSMgtSettableLogAnnounceInterval OBJECT-TYPE
    SYNTAX      Integer32(-128..127)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
                "The value is the logarithm to base 2 of the announce
interval
                 used if useMgtSettableLogAnnounceInterval is TRUE (1).
                 The value is not used if
useMgtSettableLogAnnounceInterval is
                 FALSE (2)."
    REFERENCE   "14.8.15"
    ::= { ieee8021AsV2PortDSEntry 17 }

ieee8021AsV2PortDSAnnounceReceiptTimeout OBJECT-TYPE
    SYNTAX      Unsigned32(0..255)
```

```
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
               "The value is the number of Announce message transmission
                intervals that a slave port waits without receiving an
                Announce message, before assuming that the master is no
                longer transmitting Announce messages and the BMCA needs
                to be run, if appropriate."
    REFERENCE   "14.8.16"
    DEFVAL { 3 }
    ::= { ieee8021AsV2PortDSEntry 18 }


ieee8021AsV2PortDSInitialLogSyncInterval OBJECT-TYPE
    SYNTAX      Integer32(-128..127)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
                "If useMgtSettableLogSyncInterval is FALSE (2), the
                 value is the logarithm to base 2 of the sync interval
used
                 when (a) the port is initialized, or (b) a message
interval
                 request TLV is received with the logTimeSyncInterval
field set
                 to 126."
    REFERENCE   "14.8.17"
    ::= { ieee8021AsV2PortDSEntry 19 }

ieee8021AsV2PortDSCurrentLogSyncInterval OBJECT-TYPE
    SYNTAX      Integer32(-128..127)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                "The value is the logarithm to the base 2 of the current
                 time-synchronization transmission interval."
    REFERENCE   "14.8.18"
    ::= { ieee8021AsV2PortDSEntry 20 }

ieee8021AsV2PortDSUseMgtSettableLogSyncInterval OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
                "The managed object is a Boolean that determines the
source
                of the sync interval. If the value is TRUE (1), the value
                 of currentLogSyncInterval is set equal to the value of
                mgtSettableLogSyncInterval. If the value of the managed
object
                 is FALSE (2), the value of currentLogSyncInterval is
                determined by the SyncIntervalSetting state machine. The
                default value of useMgtSettableLogSyncInterval is FALSE
(2)
```

```
                 for domain 0 and TRUE (1) for domains other than domain
0."
    REFERENCE   "14.8.19"
    ::= { ieee8021AsV2PortDSEntry 21 }

ieee8021AsV2PortDSMgtSettableLogSyncInterval OBJECT-TYPE
    SYNTAX      Integer32(-128..127)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
               "The value is the logarithm to base 2 of the sync interval
                if useMgtSettableLogSyncInterval is TRUE (1). The value
is
                not used if useMgtSettableLogSyncInterval is FALSE (2)."
    REFERENCE   "14.8.20"
    ::= { ieee8021AsV2PortDSEntry 22 }

ieee8021AsV2PortDSSyncReceiptTimeout OBJECT-TYPE
    SYNTAX      Unsigned32(0..255)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
                "The value is the number of time-synchronization
transmission
                 intervals that a slave port waits without receiving
                 synchronization information, before assuming that the
master
               is no longer transmitting synchronization information and
that
                 the BMCA needs to be run, if appropriate."
    REFERENCE   "14.8.21"
    DEFVAL { 3 }
    ::= { ieee8021AsV2PortDSEntry 23 }


ieee8021AsV2PortDSSyncReceiptTimeoutTimeInterval OBJECT-TYPE
    SYNTAX      Ieee8021ASV2UScaledNs
    UNITS       "2**-16 ns"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
               "The value is equal to the value of the per-port global
                variable syncReceiptTimeoutTimeInterval. It is the time
                interval after which sync receipt timeout occurs if
                time-synchronization information has not been received
during
                the interval."
    REFERENCE   "14.8.22"
    ::= { ieee8021AsV2PortDSEntry 24 }

ieee8021AsV2PortDSInitialLogPdelayReqInterval OBJECT-TYPE
    SYNTAX      Integer32(-128..127)
    MAX-ACCESS  read-write
    STATUS      current
```

```
    DESCRIPTION
                "For full-duplex, IEEE 802.3 media and CSN media that use
                 the peer-to-peer delay mechanism to measure path delay,
                 the value is the logarithm to base 2 of the Pdelay_Req
                 message transmission interval used when (a) the port is
                 initialized, or (b) a message interval request TLV is
                received with the logLinkDelayInterval field set to 126.
                 For all other media, the value is 127."
    REFERENCE   "14.8.23"
    ::= { ieee8021AsV2PortDSEntry 25 }

ieee8021AsV2PortDSCurrentLogPdelayReqInterval OBJECT-TYPE
    SYNTAX      Integer32(-128..127)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                "For full-duplex, IEEE 802.3 media and CSN media that use
                 the peer-to-peer delay mechanism to measure path delay,
                 the value is the logarithm to the base 2 of the current
                 Pdelay_Req message transmission interval.
                 For all other media, the value is 127."
    REFERENCE   "14.8.24"
    ::= { ieee8021AsV2PortDSEntry 26 }

ieee8021AsV2PortDSUseMgtSettableLogPdelayReqInterval OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
                "The managed object is a Boolean that determines the
source
                 of the mean time interval between successive Pdelay_Req
                 messages. If the value is TRUE (1), the value of
                currentLogPdelayReqInterval is set equal to the value of
                 mgtSettableLogPdelayReqInterval. If the value of the
managed
                 object is FALSE (2), the value of
currentLogPdelayReqInterval
                 is determined by the LinkDelayIntervalSetting state
machine.
                The default value of useMgtSettableLogPdelayReqInterval
is FALSE (2)."
    REFERENCE   "14.8.25"
    DEFVAL { false }
    ::= { ieee8021AsV2PortDSEntry 27 }

ieee8021AsV2PortDSMgtSettableLogPdelayReqInterval OBJECT-TYPE
    SYNTAX      Integer32 (-128..127)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
                "The value is the logarithm to base 2 of the mean time
                 interval between successive Pdelay_Req messages if
                 useMgtSettableLogPdelayReqInterval is TRUE (1). The
```

```
                    value is not used if useMgtSettableLogPdelayReqInterval
                    is FALSE (2)."
    REFERENCE   "14.8.26"
    ::= { ieee8021AsV2PortDSEntry 28 }

ieee8021AsV2PortDSInitialLogGptpCapableMessageInterval OBJECT-TYPE
    SYNTAX      Integer32 (-128..127)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
                "The value is the logarithm to base 2 of the gPTP capable
               message interval used when (a) the port is initialized, or
               (b) a gPtpCapableMessage interval request TLV is received
with
                the logGptpCapableMessageInterval field set to 126."
    REFERENCE   "14.8.27"
    ::= { ieee8021AsV2PortDSEntry 29 }

ieee8021AsV2PortDSCurrentLogGptpCapableMessageInterval OBJECT-TYPE
    SYNTAX      Integer32 (-128..127)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                "The value is the logarithm to the base 2 of the current
                 gPTP capable message interval."
    REFERENCE   "14.8.28"
    ::= { ieee8021AsV2PortDSEntry 30 }

ieee8021AsV2PortDSUseMgtSettableLogGptpCapableMessageInterval OBJECT-
TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
                "The managed object is a Boolean that determines the
source
                 of the gPTP capable message interval. If the value is
                 TRUE (1), the value of
currentLogGptpCapableMessageInterval
                 is set equal to the value of
               mgtSettableLogGptpCapableMessageInterval. If the value of
                 the managed object is FALSE (2), the value of
               currentLogGptpCapableMessageInterval is determined by the
                 GptpCapableMessageIntervalSetting state machine.
                 The default value of
useMgtSettableLogGptpCapableMessageInterval is FALSE (2)."
    REFERENCE   "14.8.29"
    DEFVAL { false }
    ::= { ieee8021AsV2PortDSEntry 31 }

ieee8021AsV2PortDSMgtSettableLogGptpCapableMessageInterval OBJECT-TYPE
    SYNTAX      Integer32 (-128..127)
    MAX-ACCESS  read-write
    STATUS      current
```

```
    DESCRIPTION
                "The value is the logarithm to base 2 of the
                gPtpCapableMessageInterval if
                useMgtSettableLogGptpCapableMessageInterval is TRUE (1).
                The value is not used if
                useMgtSettableLogGptpCapableMessageInterval is FALSE
(2)."
    REFERENCE   "14.8.30"
    ::= { ieee8021AsV2PortDSEntry 32 }

ieee8021AsV2PortDSInitialComputeNbrRateRatio OBJECT-TYPE
    SYNTAX      Integer32 (-128..127)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
                "If useMgtSettableComputeNeighborRateRatio is FALSE (2)
                 then, for full-duplex, IEEE 802.3 media and CSN media
that
                 use the peer-to-peer delay mechanism to measure path
delay,
                 the value is the initial value of
computeNeighborRateRatio."
    REFERENCE   "14.8.31"
    ::= { ieee8021AsV2PortDSEntry 33 }

ieee8021AsV2PortDSCurrentComputeNbrRateRatio OBJECT-TYPE
    SYNTAX      Integer32 (-128..127)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
               "For full-duplex, IEEE 802.3 media and CSN media that use
                 the peer-to-peer delay mechanism to measure path delay,
                 the value is the current value of
computeNeighborRateRatio."
    REFERENCE   "14.8.32"
    ::= { ieee8021AsV2PortDSEntry 34 }

ieee8021AsV2PortDSUseMgtSettableComputeNbrRateRatio OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
                "The managed object is a Boolean that determines the
source
               of the value of computeNeighborRateRatio. If the value is
                 TRUE (1), the value of computeNeighborRateRatio is set
equal
                to the value of mgtSettableComputeNeighborRateRatio. If
the
                 value of the managed object is FALSE (2), the value of
                 currentComputeNeighborRateRatio is determined by the
                 LinkDelayIntervalSetting state machine.
                The default value of useMgtSettableLogPdelayReqInterval
is
```

```
                     FALSE (2)."
        REFERENCE   "14.8.33"
        DEFVAL { false }
        ::= { ieee8021AsV2PortDSEntry 35 }

    ieee8021AsV2PortDSMgtSettableComputeNbrRateRatio OBJECT-TYPE
        SYNTAX      Integer32 (-128..127)
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
                    "ComputeNeighborRateRatio is configured to this value if
                     useMgtSettableComputeNeighborRateRatio is TRUE (1). The
                     value is not used if
    useMgtSettableComputeNeighborRateRatio
                     is FALSE (2)."
        REFERENCE   "14.8.34"
        ::= { ieee8021AsV2PortDSEntry 36 }

    ieee8021AsV2PortDSInitialComputeMeanLinkDelay OBJECT-TYPE
        SYNTAX      Integer32 (-128..127)
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
                   "If useMgtSettableComputeMeanLinkDelay is FALSE (2) then,
                    for full-duplex, IEEE 802.3 media and CSN media that use
                     the peer-to-peer delay mechanism to measure path delay,
                    the value is the initial value of computeMeanLinkDelay."
        REFERENCE   "14.8.35"
        ::= { ieee8021AsV2PortDSEntry 37 }

    ieee8021AsV2PortDSCurrentComputeMeanLinkDelay OBJECT-TYPE
        SYNTAX      Integer32 (-128..127)
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
                   "For full-duplex, IEEE 802.3 media and CSN media that use
                     the peer-to-peer delay mechanism to measure path delay,
                    the value is the current value of computeMeanLinkDelay."
        REFERENCE   "14.8.36"
        ::= { ieee8021AsV2PortDSEntry 38 }

    ieee8021AsV2PortDSUseMgtSettableComputeMeanLinkDelay OBJECT-TYPE
        SYNTAX      TruthValue
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
                    "The managed object is a Boolean that determines the
    source
                     of the value of computeMeanLinkDelay. If the value is
                   TRUE (1), the value of computeMeanLinkDelay is set equal
    to
                     the value of mgtSettableComputeMeanLinkDelay. If the
    value
                     of the managed object is FALSE (2), the value of
```

```
                  currentComputeMeanLinkDelay is determined by the
                  LinkDelayIntervalSetting state machine.
                  The default value of useMgtSettableComputeMeanLinkDelay
                  is FALSE (2)."
    REFERENCE   "14.8.37"
    DEFVAL { false }
    ::= { ieee8021AsV2PortDSEntry 39 }

ieee8021AsV2PortDSMgtSettableComputeMeanLinkDelay OBJECT-TYPE
    SYNTAX      Integer32 (-128..127)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
                "ComputeMeanLinkDelay is configured to this value if
                useMgtSettableComputeMeanLinkDelay is TRUE (1). The
                value is not used if useMgtSettableComputeMeanLinkDelay
                is FALSE (2)."
    REFERENCE   "14.8.38"
    ::= { ieee8021AsV2PortDSEntry 40 }

ieee8021AsV2PortDSAllowedLostRsp OBJECT-TYPE
    SYNTAX      Unsigned32(1..255)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
                "The value is equal to the value of the per-port global
                variable allowedLostResponses. It is the number of
Pdelay_Req
               messages for which a valid response is not received, above
               which a port is considered to not be exchanging peer delay
                messages with its neighbor."
    REFERENCE   "14.8.39 and 11.5.3"
    DEFVAL { 9 }
    ::= { ieee8021AsV2PortDSEntry 41 }

ieee8021AsV2PortDSAllowedFaults OBJECT-TYPE
    SYNTAX      Unsigned32(1..255)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
                "The value is equal to the value of the per-Link-Port
global
               variable allowedFaults. It is the number of faults, above
                which asCapableAcrossDomains is set to FALSE (1), i.e.,
                a Link Port is considered to not be capable of
interoperating
                with its neighbor via the IEEE 802.1AS protocol."
    REFERENCE   "14.8.40"
    DEFVAL { 9 }
    ::= { ieee8021AsV2PortDSEntry 42 }

ieee8021AsV2PortDSLogGptpCapableMessageInterval OBJECT-TYPE
    SYNTAX      Integer32(-128..127)
    MAX-ACCESS  read-write
```

```
    STATUS      current
    DESCRIPTION
                "The value is the logarithm to the base 2 of the
transmission
                 interval between successive Signaling messages that
contain
                 the gPTP capable TLV."
    REFERENCE   "14.8.41"
    DEFVAL { 0 }
    ::= { ieee8021AsV2PortDSEntry 43 }

ieee8021AsV2PortDSGPtpCapableReceiptTimeout OBJECT-TYPE
    SYNTAX      Unsigned32(1..255)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
               "The value is the number of transmission intervals that a
               port waits without receiving the gPTP capable TLV, before
               assuming that the neighbor port is no longer invoking the
gPTP
                 protocol."
    REFERENCE   "14.8.42"
    DEFVAL { 9 }
    ::= { ieee8021AsV2PortDSEntry 44 }

ieee8021AsV2PortDSVersionNumber OBJECT-TYPE
    SYNTAX      Unsigned32(0..16)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                "This value is set to versionPTP as specified in
10.6.2.2.4."
    REFERENCE   "14.8.43"
    ::= { ieee8021AsV2PortDSEntry 45 }

ieee8021AsV2PortDSNup OBJECT-TYPE
    SYNTAX      Float64TC
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
               "For an OLT port of an IEEE 802.3 EPON link, the value is
                the effective index of refraction for the EPON upstream
              wavelength light of the optical path. The default value is
                1.46770 for 1 Gb/s upstream links, and 1.46773 for
                10 Gb/s upstream links.
                For all other ports, the value is 0."
    REFERENCE   "14.8.44"
    ::= { ieee8021AsV2PortDSEntry 46 }

ieee8021AsV2PortDSNdown OBJECT-TYPE
    SYNTAX      Float64TC
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
```

```
                "For an OLT port of an IEEE 802.3 EPON link, the value is
                the effective index of refraction for the EPON downstream
                 wavelength light of the optical path. The default value
is
                 1.46805 for 1 Gb/s downstream links, and 1.46851 for
                 10 Gb/s downstream links.
                 For all other ports, the value is 0."
    REFERENCE   "14.8.45"
    ::= { ieee8021AsV2PortDSEntry 47 }

ieee8021AsV2PortDSOneStepTxOper OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
               "The value is equal to the value of the per-port global
                variable oneStepTxOper. Its value is TRUE (1) if the
                port is sending one-step Sync messages, and FALSE (2)
                if the port is sending two-step Sync and Follow-Up
messages."
    REFERENCE   "14.8.46"
    ::= { ieee8021AsV2PortDSEntry 48 }

ieee8021AsV2PortDSOneStepReceive OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
               "The value is equal to the value of the per-port global
                variable oneStepReceive. Its value is TRUE (1) if the
                port is capable of receiving and processing one-step
                Sync messages."
    REFERENCE   "14.8.47"
    ::= { ieee8021AsV2PortDSEntry 49 }

ieee8021AsV2PortDSOneStepTransmit OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
               "The value is equal to the value of the per-port global
                variable oneStepTransmit. Its value is TRUE (1) if the
                port is capable of transmitting one-step Sync messages."
    REFERENCE   "14.8.48"
    ::= { ieee8021AsV2PortDSEntry 50 }

ieee8021AsV2PortDSInitialOneStepTxOper OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
              "If useMgtSettableOneStepTxOper is FALSE (2), the value is
                used to initialize currentOneStepTxOper when the port is
                initialized. If useMgtSettableOneStepTxOper is TRUE (1),
```

```
                the value of initialOneStepTxOper is not used."
    REFERENCE   "14.8.49"
    ::= { ieee8021AsV2PortDSEntry 51 }

ieee8021AsV2PortDSCurrentOneStepTxOper OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
               "The value is TRUE (1) if it is desired, either via
                management or via a received Signaling message, that the
port
               transmit one-step Sync messages. The value is FALSE (2) if
               it is not desired, either via management or via a received
                Signaling message, that the port transmit one-step Sync
                messages.
                NOTE: The port will send one-step Sync messages only if
                currentOneStepTxOper and oneStepTransmit are both TRUE
(1)."
    REFERENCE   "14.8.50"
    ::= { ieee8021AsV2PortDSEntry 52 }

ieee8021AsV2PortDSUseMgtSettableOneStepTxOper OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
              "The managed object is a Boolean that determines the source
                of currentOneStepTxOper. If the value is TRUE (1), the
               value of currentOneStepTxOper is set equal to the value of
                mgtSettableOneStepTxOper. If the value is FALSE (2), the
                value of currentOneStepTxOper is determined by the
                OneStepTxOperSetting state machine.
                The default value of useMgtSettableOneStepTxOper is TRUE
(1)."
    REFERENCE   "14.8.51"
    DEFVAL     { true }
    ::= { ieee8021AsV2PortDSEntry 53 }

ieee8021AsV2PortDSMgtSettableOneStepTxOper OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
               "If useMgtSettableOneStepTxOper is TRUE (1),
                currentOneStepTxOper is set equal to the value of
                mgtSettableOneStepTxOper. The value of
mgtSettableOneStepTxOper
                is not used if useMgtSettableOneStepTxOper is FALSE (2).
               The default value of mgtSettableOneStepTxOper is FALSE (2)
                for domains other than domain 0."
    REFERENCE   "14.8.52"
    ::= { ieee8021AsV2PortDSEntry 54 }
```

```
ieee8021AsV2PortDSSyncLocked OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
               "The value is equal to the value of the per-port global
                variable syncLocked. Its value is TRUE (1) if the port
will
                transmit a Sync as soon as possible after the slave port
                receives a Sync."
    REFERENCE   "14.8.53"
    ::= { ieee8021AsV2PortDSEntry 55 }

ieee8021AsV2PortDSPdelayTruncTST1 OBJECT-TYPE
    SYNTAX      Ieee8021ASV2Timestamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
               "For full-duplex IEEE 802.3 media, and CSN media that use
the
                peer-to-peer delay mechanism to measure path delay, the
first
                value, T1, of the four elements of this array is as
described
                in Table 14-9. For all other media, the values are zero.
               This object corresponds to the timestamp t1 modulo 2^32 in
               Figure 11-1, and expressed in units of 2^-16 ns (i.e., the
                value of this array element is equal to the remainder
obtained
                upon dividing the respective timestamp , expressed in
units
                of 2^-16 ns, by 2^48).
               At any given time, the timestamp values stored in the T1,
T2,
                T3, T4 PdelayTruncTS are for the same, and most recently
                completed, peer delay message exchange."
    REFERENCE   "14.8.54"
    ::= { ieee8021AsV2PortDSEntry 56 }

ieee8021AsV2PortDSPdelayTruncTST2 OBJECT-TYPE
    SYNTAX      Ieee8021ASV2Timestamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
               "For full-duplex IEEE 802.3 media, and CSN media that use
the
                peer-to-peer delay mechanism to measure path delay, the
second
                value, T2, of the four elements of this array is as
described
                in Table 14-9. For all other media, the values are zero.
               This object corresponds to the timestamp t1 modulo 2^32 in
               Figure 11-1, and expressed in units of 2^-16 ns (i.e., the
```

```
                 value of this array element is equal to the remainder
obtained
                 upon dividing the respective timestamp , expressed in
units
                 of 2^-16 ns, by 2^48).
                At any given time, the timestamp values stored in the T1,
T2,
                 T3, T4 PdelayTruncTS are for the same, and most recently
                 completed, peer delay message exchange."
    REFERENCE   "14.8.54"
    ::= { ieee8021AsV2PortDSEntry 57 }

ieee8021AsV2PortDSPdelayTruncTST3 OBJECT-TYPE
    SYNTAX      Ieee8021ASV2Timestamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                "For full-duplex IEEE 802.3 media, and CSN media that use
the
                 peer-to-peer delay mechanism to measure path delay, the
third
                 value, T3, of the four elements of this array is as
described
                 in Table 14-9. For all other media, the values are zero.
                This object corresponds to the timestamp t1 modulo 2^32 in
                Figure 11-1, and expressed in units of 2^-16 ns (i.e., the
                 value of this array element is equal to the remainder
obtained
                 upon dividing the respective timestamp , expressed in
units
                 of 2^-16 ns, by 2^48).
                At any given time, the timestamp values stored in the T1,
T2,
                 T3, T4 PdelayTruncTS are for the same, and most recently
                 completed, peer delay message exchange."
    REFERENCE   "14.8.54"
    ::= { ieee8021AsV2PortDSEntry 58 }

ieee8021AsV2PortDSPdelayTruncTST4 OBJECT-TYPE
    SYNTAX      Ieee8021ASV2Timestamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                "For full-duplex IEEE 802.3 media, and CSN media that use
the
                 peer-to-peer delay mechanism to measure path delay, the
fourth
                 value, T4, of the four elements of this array is as
described
                 in Table 14-9. For all other media, the values are zero.
                This object corresponds to the timestamp t1 modulo 2^32 in
                Figure 11-1, and expressed in units of 2^-16 ns (i.e., the
                 value of this array element is equal to the remainder
obtained
```

```
                upon dividing the respective timestamp , expressed in
units
                of 2^-16 ns, by 2^48).
               At any given time, the timestamp values stored in the T1,
T2,
                T3, T4 PdelayTruncTS are for the same, and most recently
                completed, peer delay message exchange."
    REFERENCE   "14.8.54"
    ::= { ieee8021AsV2PortDSEntry 59 }

ieee8021AsV2PortDSMinorVersionNumber OBJECT-TYPE
    SYNTAX      Unsigned32 (0..15)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
               "This value is set to minorVersionPTP as specified
                in 10.6.2.2.3."
    REFERENCE   "14.8.55"
    ::= { ieee8021AsV2PortDSEntry 60 }

-- =============================================================
-- The Description Port Parameter Data Set contains the
-- profileIdentifier for this PTP profile, as specified in
-- Annex F.1.
-- =============================================================

ieee8021AsV2DescriptionPortDSTable   OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021AsV2DescriptionPortDSEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "The descriptionPortDS contains the profileIdentifier for
        this PTP profile, as specified in Annex F.1."
    REFERENCE   "14.9"
    ::= { ieee8021AsV2MIBObjects 11 }

ieee8021AsV2DescriptionPortDSEntry   OBJECT-TYPE
    SYNTAX      Ieee8021AsV2DescriptionPortDSEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "The descriptionPortDS contains the profileIdentifier for
        this PTP profile"
    INDEX { ieee8021AsV2PtpInstance,
            ieee8021AsV2DescriptionPortDSAsIndex }
    ::= { ieee8021AsV2DescriptionPortDSTable 1 }

Ieee8021AsV2DescriptionPortDSEntry ::=
     SEQUENCE {
         ieee8021AsV2DescriptionPortDSAsIndex
                                           InterfaceIndexOrZero,
         ieee8021AsV2DescriptionPortDSProfileIdentifier

Ieee8021AsV2GPtpProfileIdentifier }
```

```
ieee8021AsV2DescriptionPortDSAsIndex OBJECT-TYPE
    SYNTAX      InterfaceIndexOrZero
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
               "This object identifies the gPTP interface group within
                the system for which this entry contains information. It
                is the value of the instance of the IfIndex object,
                defined in the IF-MIB, for the gPTP interface group
                corresponding to this port, or the value 0 if the port
                has not been bound to an underlying frame source and
                sink.

                For a given media port of a Bridge or an end station,
                there can be one or more gPTP Port, and depends whether
                a media port supports point to point link (e.g. IEEE
                802.3 Ethernet) or point to multi-point (e.g. CSN, IEEE
                802.3 EPON, etc) links on the media port."
    REFERENCE   "IEEE Std 802.1AS Description Port Parameter DS Group
                 gPTP Port Index"
    ::= { ieee8021AsV2DescriptionPortDSEntry 1 }

ieee8021AsV2DescriptionPortDSProfileIdentifier OBJECT-TYPE
    SYNTAX      Ieee8021AsV2GPtpProfileIdentifier
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
               "The value is the profileIdentifier for this PTP profile."
    REFERENCE   "14.9.1 and Annex F, F.1"
    ::= { ieee8021AsV2DescriptionPortDSEntry 2 }

-- ==============================================================
-- The Port Parameter Statistics Data Set provides counters
-- associated with port capabilities at a given PTP Instance.
-- ==============================================================

ieee8021AsV2PortStatDSTable   OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021AsV2PortStatDSEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "The portStatisticsDS provides counters associated with port
        capabilities at a given PTP Instance."
    REFERENCE   "14.10"
    ::= { ieee8021AsV2MIBObjects 12 }

ieee8021AsV2PortStatDSEntry   OBJECT-TYPE
    SYNTAX      Ieee8021AsV2PortStatDSEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "Port Statistics Data Set provides counters associated with
        port capabilities at a given PTP Instance."
```

```
        INDEX { ieee8021AsV2PtpInstance,
                ieee8021AsV2PortDSIndex }
        ::= { ieee8021AsV2PortStatDSTable 1 }

    Ieee8021AsV2PortStatDSEntry ::=
         SEQUENCE {
             ieee8021AsV2PortStatRxSyncCount               Counter32,
             ieee8021AsV2PortStatRxOneStepSyncCount        Counter32,
             ieee8021AsV2PortStatRxFollowUpCount           Counter32,
             ieee8021AsV2PortStatRxPdelayRequestCount      Counter32,
             ieee8021AsV2PortStatRxPdelayRspCount     Counter32,
             ieee8021AsV2PortStatRxPdelayRspFollowUpCount  Counter32,
             ieee8021AsV2PortStatRxAnnounceCount           Counter32,
             ieee8021AsV2PortStatRxPtpPacketDiscardCount   Counter32,
             ieee8021AsV2PortStatSyncReceiptTimeoutCount    Counter32,
             ieee8021AsV2PortStatAnnounceReceiptTimeoutCount   Counter32,
            ieee8021AsV2PortStatPdelayAllowedLostRspExceededCount Counter32,
             ieee8021AsV2PortStatTxSyncCount               Counter32,
             ieee8021AsV2PortStatTxOneStepSyncCount        Counter32,
             ieee8021AsV2PortStatTxFollowUpCount           Counter32,
             ieee8021AsV2PortStatTxPdelayRequestCount      Counter32,
             ieee8021AsV2PortStatTxPdelayRspCount     Counter32,
             ieee8021AsV2PortStatTxPdelayRspFollowUpCount Counter32,
             ieee8021AsV2PortStatTxAnnounceCount           Counter32
         }

    ieee8021AsV2PortStatRxSyncCount OBJECT-TYPE
        SYNTAX      Counter32
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
                    "A counter that increments every time synchronization
                     information is received."
        REFERENCE   "14.10.2"
        ::= { ieee8021AsV2PortStatDSEntry 1 }

    ieee8021AsV2PortStatRxOneStepSyncCount OBJECT-TYPE
        SYNTAX      Counter32
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
                    "A counter that increments every time a one-step Sync
                     message is received."
        REFERENCE   "14.10.3"
        ::= { ieee8021AsV2PortStatDSEntry 2 }

    ieee8021AsV2PortStatRxFollowUpCount OBJECT-TYPE
        SYNTAX      Counter32
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
                   "A counter that increments every time a Follow_Up message
                     is received."
        REFERENCE   "14.10.4"
```

```
    ::= { ieee8021AsV2PortStatDSEntry 3 }

ieee8021AsV2PortStatRxPdelayRequestCount OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
               "A counter that increments every time a Pdelay_Req message
                is received."
    REFERENCE   "14.10.5"
    ::= { ieee8021AsV2PortStatDSEntry 4 }

ieee8021AsV2PortStatRxPdelayRspCount OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                "A counter that increments every time a Pdelay_Resp
message
                 is received."
    REFERENCE   "14.10.6"
    ::= { ieee8021AsV2PortStatDSEntry 5 }

ieee8021AsV2PortStatRxPdelayRspFollowUpCount OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                "A counter that increments every time a
Pdelay_Resp_Follow_Up
                 message is received."
    REFERENCE   "14.10.7"
    ::= { ieee8021AsV2PortStatDSEntry 6 }

ieee8021AsV2PortStatRxAnnounceCount OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
               "A counter that increments every time an Announce message
                is received."
    REFERENCE   "14.10.8"
    ::= { ieee8021AsV2PortStatDSEntry 7 }

ieee8021AsV2PortStatRxPtpPacketDiscardCount OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
               "A counter that increments every time a PTP message of the
                respective PTP Instance is discarded."
    REFERENCE   "14.10.9"
    ::= { ieee8021AsV2PortStatDSEntry 8 }
```

```
ieee8021AsV2PortStatSyncReceiptTimeoutCount OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
               "A counter that increments every time sync receipt timeout
                occurs."
    REFERENCE   "14.10.10"
    ::= { ieee8021AsV2PortStatDSEntry 9 }

ieee8021AsV2PortStatAnnounceReceiptTimeoutCount OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                "A counter that increments every time announce receipt
timeout
                 occurs."
    REFERENCE   "14.10.11"
    ::= { ieee8021AsV2PortStatDSEntry 10 }

ieee8021AsV2PortStatPdelayAllowedLostRspExceededCount OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                "A counter that increments every time the value of the
                variable lostResponses exceeds the value of the variable
                 allowedLostResponses."
    REFERENCE   "14.10.12"
    ::= { ieee8021AsV2PortStatDSEntry 11 }

ieee8021AsV2PortStatTxSyncCount OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                "A counter that increments every time synchronization
                 information is transmitted."
    REFERENCE   "14.10.13"
    ::= { ieee8021AsV2PortStatDSEntry 12 }

ieee8021AsV2PortStatTxOneStepSyncCount OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                "A counter that increments every time a one-step Sync
                 message is transmitted."
    REFERENCE   "14.10.14"
    ::= { ieee8021AsV2PortStatDSEntry 13 }

ieee8021AsV2PortStatTxFollowUpCount OBJECT-TYPE
    SYNTAX      Counter32
```

```
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
               "A counter that increments every time a Follow_Up message
                is transmitted."
    REFERENCE   "14.10.15"
    ::= { ieee8021AsV2PortStatDSEntry 14 }

ieee8021AsV2PortStatTxPdelayRequestCount OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
               "A counter that increments every time a Pdelay_Req message
                is transmitted."
    REFERENCE   "14.10.16"
    ::= { ieee8021AsV2PortStatDSEntry 15 }

ieee8021AsV2PortStatTxPdelayRspCount OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                "A counter that increments every time a Pdelay_Resp
message
                is transmitted."
    REFERENCE   "14.10.17"
    ::= { ieee8021AsV2PortStatDSEntry 16 }

ieee8021AsV2PortStatTxPdelayRspFollowUpCount OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                "A counter that increments every time a
                 Pdelay_Resp_Follow_Up message is transmitted."
    REFERENCE   "14.10.18"
    ::= { ieee8021AsV2PortStatDSEntry 17 }

ieee8021AsV2PortStatTxAnnounceCount OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
               "A counter that increments every time an Announce message
is transmitted."
    REFERENCE   "14.10.19"
    ::= { ieee8021AsV2PortStatDSEntry 18 }

-- ================================================================
-- The Acceptable Master Port Parameter Data Ser represents the
-- capability to enable/disable the acceptable master table
-- feature on a port.
-- ================================================================
```

```
ieee8021AsV2AcceptableMasterPortDSTable   OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021AsV2AcceptableMasterPortDSEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "For the single port of a PTP End Instance and for each port
        of a PTP Relay Instance, the acceptableMasterPortDS contains
        the single member acceptableMasterTableEnabled, which is used
        to enable/disable the Acceptable Master Table Feature. The
        number of such data sets is the same as the value of
        defaultDS.numberPorts."
    REFERENCE   "14.11"
    ::= { ieee8021AsV2MIBObjects 13 }

ieee8021AsV2AcceptableMasterPortDSEntry   OBJECT-TYPE
    SYNTAX      Ieee8021AsV2AcceptableMasterPortDSEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "The Acceptable Master Port Data Set represents the capability
        to enable/disable the acceptable master table feature on a
        port.
        For the single port of a PTP End Instance and for each port of
        a PTP Relay Instance, the acceptableMasterPortDS contains the
single
        member acceptableMasterTableEnabled, which is used to enable/
disable
       the Acceptable Master Table Feature. The number of such data sets
is
        the same as the value of defaultDS.numberPorts."
    INDEX { ieee8021AsV2PtpInstance,
            ieee8021AsV2AcceptableMasterPortDSAsIndex }
    ::= { ieee8021AsV2AcceptableMasterPortDSTable 1 }

Ieee8021AsV2AcceptableMasterPortDSEntry ::=
  SEQUENCE {
      ieee8021AsV2AcceptableMasterPortDSAsIndex    InterfaceIndexOrZero,
    ieee8021AsV2AcceptableMasterPortDSAcceptableMasterTableEnabled
TruthValue
      }

ieee8021AsV2AcceptableMasterPortDSAsIndex OBJECT-TYPE
    SYNTAX      InterfaceIndexOrZero
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
                "An index to identify an entry in the Acceptable Master
                Port Table Data Set."
    REFERENCE   "14.11"
    ::= { ieee8021AsV2AcceptableMasterPortDSEntry 1 }

ieee8021AsV2AcceptableMasterPortDSAcceptableMasterTableEnabled OBJECT-
TYPE
```

```
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
                "The value is equal to the value of the Boolean
                 acceptableMasterTableEnabled."
    REFERENCE   "14.11.2"
    ::= { ieee8021AsV2AcceptableMasterPortDSEntry 2 }

-- ================================================================
-- The External Port Configuration Port Data Set is used with
-- the external port configuration option to indicate the
-- desired state for the gPTP Port.
-- ================================================================
ieee8021AsV2ExternalPortConfigurationPortDSTable   OBJECT-TYPE
    SYNTAX      SEQUENCE OF
Ieee8021AsV2ExternalPortConfigurationPortDSEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "The externalPortConfigurationPortDS contains the single member
        desiredState, which indicates the desired state for the gPTP
Port.
        The number of such data sets is the same as the value of
        defaultDS.numberPorts."
    REFERENCE   "14.12"
    ::= { ieee8021AsV2MIBObjects 14 }

ieee8021AsV2ExternalPortConfigurationPortDSEntry   OBJECT-TYPE
    SYNTAX      Ieee8021AsV2ExternalPortConfigurationPortDSEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "The externalPortConfigurationPortDS contains the single member
        desiredState, which indicates the desired state for the gPTP
Port.
        The number of such data sets is the same as the value of
        defaultDS.numberPorts."
    INDEX { ieee8021AsV2PtpInstance,
            ieee8021AsV2ExternalPortConfigurationPortDSAsIndex }
    ::= { ieee8021AsV2ExternalPortConfigurationPortDSTable 1 }

Ieee8021AsV2ExternalPortConfigurationPortDSEntry ::=
  SEQUENCE {
      ieee8021AsV2ExternalPortConfigurationPortDSAsIndex
InterfaceIndexOrZero,
      ieee8021AsV2ExternalPortConfigurationPortDSDesiredState
INTEGER
      }

ieee8021AsV2ExternalPortConfigurationPortDSAsIndex OBJECT-TYPE
    SYNTAX      InterfaceIndexOrZero
    MAX-ACCESS  not-accessible
    STATUS      current
```

```
    DESCRIPTION
               "An index to identify an entry in the External Port
                Configuration Port Table Data Set."
    REFERENCE   "14.12"
    ::= { ieee8021AsV2ExternalPortConfigurationPortDSEntry 1 }

ieee8021AsV2ExternalPortConfigurationPortDSDesiredState OBJECT-TYPE
    SYNTAX           INTEGER {
            disabledPort(3),
            masterPort(6),
            passivePort(7),
            slavePort(9)
           }
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
                "When the value of
defaultDS.externalPortConfigurationEnabled
                 is TRUE (1), the value of
                 externalPortConfigurationPortDS.desiredState is the
desired
                state of the gPTP Port. This member sets the value of the
                variable portStateInd. When a new value is written to the
                 member by management, the variable rcvdPortStateInd is
set to TRUE (1)."
    REFERENCE   "14.12.2"
    ::= { ieee8021AsV2ExternalPortConfigurationPortDSEntry 2 }

-- =============================================================
-- Asymmetry Measurement Mode Parameter Data Set
-- to enable/disable the feature on a port.
-- =============================================================

ieee8021AsV2AsymMeasurementModeDSTable   OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021AsV2AsymMeasurementModeDSEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "The asymmetryMeasurementModeDS represents the capability to
        enable/disable the Asymmetry Compensation Measurement Procedure
        on a Link Port (see Annex G). This data set is used instead of
        the cmldsAsymmetryMeasurementModeDS, when only domain 0 is
        present and CMLDS is not used."
    REFERENCE   "14.13"
    ::= { ieee8021AsV2MIBObjects 15 }

ieee8021AsV2AsymMeasurementModeDSEntry   OBJECT-TYPE
    SYNTAX      Ieee8021AsV2AsymMeasurementModeDSEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "The asymmetryMeasurementModeDS represents the capability to
        enable/disable the Asymmetry Compensation Measurement Procedure
        on a Link Port (see Annex G). This data set is used instead of
```

```
            the cmldsAsymmetryMeasurementModeDS, when only domain 0 is
            present and CMLDS is not used. "
        INDEX { ieee8021AsV2PtpInstance,
                ieee8021AsV2AsymMeasurementModeDSAsIndex }
        ::= { ieee8021AsV2AsymMeasurementModeDSTable 1 }

Ieee8021AsV2AsymMeasurementModeDSEntry ::=
     SEQUENCE {
          ieee8021AsV2AsymMeasurementModeDSAsIndex
InterfaceIndexOrZero,
          ieee8021AsV2AsymMeasurementModeDSAsymMeasurementMode
TruthValue
      }

ieee8021AsV2AsymMeasurementModeDSAsIndex OBJECT-TYPE
    SYNTAX      InterfaceIndexOrZero
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
              "An index to identify an entry in the Asymmetry Measurement
                Mode Data Set."
    REFERENCE   "14.13"
    ::= { ieee8021AsV2AsymMeasurementModeDSEntry 1 }

ieee8021AsV2AsymMeasurementModeDSAsymMeasurementMode OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
                "The value is equal to the value of the Boolean
                 asymmetryMeasurementMode. For full-duplex IEEE 802.3
                media, the value is TRUE (1) if an asymmetry measurement
                 is being performed for the link attached to this Link
Port,
                 and FALSE (2) otherwise. For all other media, the value
                 shall be FALSE (2).
               NOTE: If an asymmetry measurement is being performed for
a
                 link, asymmetryMeasurementMode must be TRUE (1) for the
                 Link Ports at each end of the link."
    REFERENCE   "14.13.2"
    ::= { ieee8021AsV2AsymMeasurementModeDSEntry 2 }

-- ================================================================
-- The Common Services Port Parameter Data Set enables a gPTP
-- Port of a PTP Instance to determine which port of the
-- respective common service corresponds to that gPTP Port.
-- ================================================================
ieee8021AsV2CommonServicesPortDSTable   OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021AsV2CommonServicesPortDSEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
```

```
       "At present, the only common service specified is the CMLDS, and
the
        only member of the commonServicesPortDS is the
cmldsLinkPortPortNumber.
        This member contains the port number of the CMLDS Link Port that
        corresponds to this gPTP Port."
    REFERENCE   "14.14"
    ::= { ieee8021AsV2MIBObjects 16 }

ieee8021AsV2CommonServicesPortDSEntry   OBJECT-TYPE
    SYNTAX      Ieee8021AsV2CommonServicesPortDSEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "At present, the only common service specified is the CMLDS, and
the
        only member of the commonServicesPortDS is the
cmldsLinkPortPortNumber.
        This member contains the port number of the CMLDS Link Port that
        corresponds to this gPTP Port."
    INDEX { ieee8021AsV2PtpInstance,
            ieee8021AsV2CommonServicesPortDSAsIndex }
    ::= { ieee8021AsV2CommonServicesPortDSTable 1 }

Ieee8021AsV2CommonServicesPortDSEntry ::=
  SEQUENCE {
      ieee8021AsV2CommonServicesPortDSAsIndex    InterfaceIndexOrZero,
      ieee8021AsV2CommonServicesPortDSCmldsLinkPortPortNumber
Unsigned32
      }

ieee8021AsV2CommonServicesPortDSAsIndex OBJECT-TYPE
    SYNTAX      InterfaceIndexOrZero
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
              "An index to identify an entry in the Common Services Port
               Data Set."
    REFERENCE   "14.14"
    ::= { ieee8021AsV2CommonServicesPortDSEntry 1 }

ieee8021AsV2CommonServicesPortDSCmldsLinkPortPortNumber OBJECT-TYPE
    SYNTAX           Unsigned32 (0..65535)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
               "The value is the portNumber attribute of the
                cmldsLinkPortDS.portIdentity of the Link Port that
                corresponds to this gPTP Port."
    REFERENCE   "14.14.2"
    ::= { ieee8021AsV2CommonServicesPortDSEntry 2 }

-- =============================================================
-- The Common Mean Link Delay Service Default Parameter Data Set
```

```
-- describes the per-time-aware-system attributes of the Common
-- Mean Link Delay Service.
-- ==============================================================

ieee8021AsV2CommonMeanLinkDelayServiceDefaultDSTable   OBJECT-TYPE
    SYNTAX      SEQUENCE OF
Ieee8021AsV2CommonMeanLinkDelayServiceDefaultDSEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
      "The cmldsDefaultDS describes the per-time-aware-system attributes
        of the Common Mean Link Delay Service."
    REFERENCE   "14.15"
    ::= { ieee8021AsV2MIBObjects 17 }

ieee8021AsV2CommonMeanLinkDelayServiceDefaultDSEntry   OBJECT-TYPE
    SYNTAX      Ieee8021AsV2CommonMeanLinkDelayServiceDefaultDSEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
      "The cmldsDefaultDS describes the per-time-aware-system attributes
        of the Common Mean Link Delay Service."
    INDEX { ieee8021AsV2CmldsDefaultDSAsIndex }
    ::= { ieee8021AsV2CommonMeanLinkDelayServiceDefaultDSTable 1 }

Ieee8021AsV2CommonMeanLinkDelayServiceDefaultDSEntry ::=
   SEQUENCE {
      ieee8021AsV2CmldsDefaultDSAsIndex  InterfaceIndexOrZero,
      ieee8021AsV2CmldsDefaultDSClockIdentity
Ieee8021AsV2ClockIdentity,
      ieee8021AsV2CmldsDefaultDSNumberLinkPorts Unsigned32,
      ieee8021AsV2CmldsDefaultDSSdoID           Unsigned32
      }


ieee8021AsV2CmldsDefaultDSAsIndex OBJECT-TYPE
    SYNTAX      InterfaceIndexOrZero
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
                "An index to identify an entry in the Common Mean Link
                Delay Default Data Set."
    REFERENCE   "14.15"
    ::= { ieee8021AsV2CommonMeanLinkDelayServiceDefaultDSEntry 1 }

ieee8021AsV2CmldsDefaultDSClockIdentity OBJECT-TYPE
    SYNTAX      Ieee8021AsV2ClockIdentity
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                 "The value is the clockIdentity that will be used to
                  identify the Common Mean Link Delay Service."
    REFERENCE   "14.15.1"
    ::= { ieee8021AsV2CommonMeanLinkDelayServiceDefaultDSEntry 2 }
```

```
ieee8021AsV2CmldsDefaultDSNumberLinkPorts OBJECT-TYPE
    SYNTAX      Unsigned32 (0..65535)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                "The value is the number of Link Ports of the time-aware
                 system on which the Common Mean Link Delay Service is
                 implemented. For an end station the value is 1."
    REFERENCE   "14.15.2"
    ::= { ieee8021AsV2CommonMeanLinkDelayServiceDefaultDSEntry 3 }

ieee8021AsV2CmldsDefaultDSSdoID OBJECT-TYPE
    SYNTAX      Unsigned32 (0..4095)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                "The value is 0x200. This is the sdoId for the
                 Common Mean Link Delay Service.
                 NOTE: The attribute sdoId is specified as a 12-bit
                 unsigned integer in 8.1. The data type for the managed
                 object sdoId is UInteger16 in Table 14-17, for
                 compatibility with IEEE Std 1588. The range of the
                 managed object is limited to 12 bits."
    REFERENCE   "14.15.3"
    ::= { ieee8021AsV2CommonMeanLinkDelayServiceDefaultDSEntry 4 }

-- ================================================================
-- The Common Mean Link Delay Service Link Port Parameter Data Set
-- represents time-aware Link Port capabilities for the Common Mean
-- Link Delay Service of a Link Port of a time-aware system.
-- ================================================================

ieee8021AsV2CommonMeanLinkDelayServiceLinkPortDSTable   OBJECT-TYPE
    SYNTAX      SEQUENCE OF
Ieee8021AsV2CommonMeanLinkDelayServiceLinkPortDSEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "For every Link Port of the Common Mean Link Delay Service of a
        time-aware system, the following Link Port Parameter Data Set
        is maintained as the basis for protocol decisions and providing
        values for message fields. The number of such data sets is the
same
        as the value of cmldsDefaultDS.numberLinkPorts."
    REFERENCE   "14.16"
    ::= { ieee8021AsV2MIBObjects 18 }

ieee8021AsV2CommonMeanLinkDelayServiceLinkPortDSEntry   OBJECT-TYPE
    SYNTAX      Ieee8021AsV2CommonMeanLinkDelayServiceLinkPortDSEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "For every Link Port of the Common Mean Link Delay Service of a
```

```
            time-aware system, the following Link Port Parameter Data Set
            is maintained as the basis for protocol decisions and providing
            values for message fields. The number of such data sets is the
same
            as the value of cmldsDefaultDS.numberLinkPorts."
      INDEX { ieee8021AsV2BridgeBasePort,
              ieee8021AsV2CmldsLinkPortDSAsIndex }
      ::= { ieee8021AsV2CommonMeanLinkDelayServiceLinkPortDSTable 1 }

Ieee8021AsV2CommonMeanLinkDelayServiceLinkPortDSEntry ::=
   SEQUENCE {
     ieee8021AsV2CmldsLinkPortDSAsIndex     InterfaceIndexOrZero,
     ieee8021AsV2CmldsLinkPortDSClockIdentity
Ieee8021AsV2ClockIdentity,
     ieee8021AsV2CmldsLinkPortDSPortNumber             Unsigned32,
     ieee8021AsV2CmldsLinkPortDSCmldsLinkPortEnabled   TruthValue,
     ieee8021AsV2CmldsLinkPortDSIsMeasuringDelay       TruthValue,
     ieee8021AsV2CmldsLinkPortDSAsCapableAcrossDomains TruthValue,
     ieee8021AsV2CmldsLinkPortDSMeanLinkDelay    Ieee8021ASV2UScaledNs,
     ieee8021AsV2CmldsLinkPortDSMeanLinkDelayThresh
Ieee8021ASV2UScaledNs,
     ieee8021AsV2CmldsLinkPortDSDelayAsym       Ieee8021ASV2ScaledNs,
     ieee8021AsV2CmldsLinkPortDSNbrRateRatio      Integer32,
     ieee8021AsV2CmldsLinkPortDSInitialLogPdelayReqInterval Integer32,
     ieee8021AsV2CmldsLinkPortDSCurrentLogPdelayReqInterval Integer32,
     ieee8021AsV2CmldsLinkPortDSUseMgtSettableLogPdelayReqInterval
TruthValue,
     ieee8021AsV2CmldsLinkPortDSMgtSettableLogPdelayReqInterval
Integer32,
     ieee8021AsV2CmldsLinkPortDSInitialComputeNbrRateRatio     Integer32,
     ieee8021AsV2CmldsLinkPortDSCurrentComputeNbrRateRatio     Integer32,
     ieee8021AsV2CmldsLinkPortDSUseMgtSettableComputeNbrRateRatio
TruthValue,
     ieee8021AsV2CmldsLinkPortDSMgtSettableComputeNbrRateRatio
Integer32,
     ieee8021AsV2CmldsLinkPortDSInitialComputeMeanLinkDelay
Integer32,
     ieee8021AsV2CmldsLinkPortDSCurrentComputeMeanLinkDelay
Integer32,
     ieee8021AsV2CmldsLinkPortDSUseMgtSettableComputeMeanLinkDelay
TruthValue,
     ieee8021AsV2CmldsLinkPortDSMgtSettableComputeMeanLinkDelay
Integer32,
     ieee8021AsV2CmldsLinkPortDSAllowedLostRsp   Unsigned32,
     ieee8021AsV2CmldsLinkPortDSAllowedFaults          Unsigned32,
     ieee8021AsV2CmldsLinkPortDSVersionNumber          Unsigned32,
     ieee8021AsV2CmldsLinkPortDSPdelayTruncTST1 Ieee8021ASV2Timestamp,
     ieee8021AsV2CmldsLinkPortDSPdelayTruncTST2 Ieee8021ASV2Timestamp,
     ieee8021AsV2CmldsLinkPortDSPdelayTruncTST3 Ieee8021ASV2Timestamp,
     ieee8021AsV2CmldsLinkPortDSPdelayTruncTST4 Ieee8021ASV2Timestamp,
     ieee8021AsV2CmldsLinkPortDSMinorVersionNumber     Unsigned32
      }

ieee8021AsV2CmldsLinkPortDSAsIndex OBJECT-TYPE
```

```
    SYNTAX      InterfaceIndexOrZero
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
               "An index to identify an entry in the Comon Mean Link
               Delay Link Port Data Set."
    REFERENCE   "14.16"
    ::= { ieee8021AsV2CommonMeanLinkDelayServiceLinkPortDSEntry 1 }

ieee8021AsV2CmldsLinkPortDSClockIdentity OBJECT-TYPE
    SYNTAX      Ieee8021AsV2ClockIdentity
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                "The value is the first of the portIdentity attribute
                 of the local port, which is a set made of
                 Ieee8021AsV2ClockIdentity and portNumber."
    REFERENCE   "14.16.2"
    ::= { ieee8021AsV2CommonMeanLinkDelayServiceLinkPortDSEntry 2 }

ieee8021AsV2CmldsLinkPortDSPortNumber OBJECT-TYPE
    SYNTAX      Unsigned32(0..65535)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                "The value is the second of the portIdentity attribute
                 of the local port, which is a set made of
                 Ieee8021AsV2ClockIdentity and portNumber."
    REFERENCE   "14.16.2"
    ::= { ieee8021AsV2CommonMeanLinkDelayServiceLinkPortDSEntry 3 }

ieee8021AsV2CmldsLinkPortDSCmldsLinkPortEnabled
 OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                "The value is equal to the value of the Boolean
                 cmldsLinkPortEnabled."
    REFERENCE   "14.16.3"
    ::= { ieee8021AsV2CommonMeanLinkDelayServiceLinkPortDSEntry 4 }

ieee8021AsV2CmldsLinkPortDSIsMeasuringDelay
 OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                "The value is equal to the value of the Boolean
                 isMeasuringDelay."
    REFERENCE   "14.16.4"
    ::= { ieee8021AsV2CommonMeanLinkDelayServiceLinkPortDSEntry 5 }

ieee8021AsV2CmldsLinkPortDSAsCapableAcrossDomains
```

```
 OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                "The value is equal to the value of the Boolean
                 asCapableAcrossDomains."
    REFERENCE   "14.16.5"
    ::= { ieee8021AsV2CommonMeanLinkDelayServiceLinkPortDSEntry 6 }

ieee8021AsV2CmldsLinkPortDSMeanLinkDelay
 OBJECT-TYPE
    SYNTAX      Ieee8021ASV2UScaledNs
    UNITS       "2**-16 ns * 2**64"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
               "The value is equal to the value of the per-port global
                variable meanLinkDelay. It is an estimate of the current
               one-way propagation time on the link attached to this Link
               Port, measured as specified for the respective medium. The
                value is zero for Link Ports attached to IEEE 802.3 EPON
links
                and for the master port of an IEEE 802.11 link, because
one-way
                propagation delay is not measured on the latter and not
                directly measured on the former."
    REFERENCE   "14.16.6"
    ::= { ieee8021AsV2CommonMeanLinkDelayServiceLinkPortDSEntry 7 }

ieee8021AsV2CmldsLinkPortDSMeanLinkDelayThresh
 OBJECT-TYPE
    SYNTAX      Ieee8021ASV2UScaledNs
    UNITS       "2**-16 ns * 2 ** 64"
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
               "The value is equal to the value of the per-Link-port
global
                variable meanLinkDelayThresh. It is the propagation time
               threshold, above which a Link Port (and therefore any PTP
Ports
                that use the CMLDS on this Link Port) is not considered
capable
                of participating in the IEEE 802.1AS protocol."
    REFERENCE   "14.16.7"
    ::= { ieee8021AsV2CommonMeanLinkDelayServiceLinkPortDSEntry 8 }

ieee8021AsV2CmldsLinkPortDSDelayAsym
 OBJECT-TYPE
    SYNTAX      Ieee8021ASV2ScaledNs
    UNITS       "2**-16 ns * 2**64"
    MAX-ACCESS  read-write
    STATUS      current
```

```
    DESCRIPTION
                "The value is the asymmetry in the propagation delay on
the
                 link attached to this Link Port relative to the local
clock.
                 If propagation delay asymmetry is not modeled, then
                 delayAsymmetry is 0."
    REFERENCE   "14.16.8"
    ::= { ieee8021AsV2CommonMeanLinkDelayServiceLinkPortDSEntry 9 }

ieee8021AsV2CmldsLinkPortDSNbrRateRatio
 OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                "The value is an estimate of the ratio of the frequency
of the
                LocalClock entity of the time-aware system at the other
end of
                the link attached to this Link Port, to the frequency of
the
                 LocalClock entity of this time-aware system.
neighborRateRatio
                 is expressed as the fractional frequency offset
multiplied
                 by 2^41, i.e., the quantity (neighborRateRatio -
1.0)(2^41)."
    REFERENCE   "14.16.9"
    ::= { ieee8021AsV2CommonMeanLinkDelayServiceLinkPortDSEntry 10 }

ieee8021AsV2CmldsLinkPortDSInitialLogPdelayReqInterval OBJECT-TYPE
    SYNTAX      Integer32(-128..127)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
               "If useMgtSettableLogPdelayReqInterval is FALSE (2) then,
                for full-duplex, IEEE 802.3 media and CSN media that use
                the peer-to-peer delay mechanism to measure path delay,
the
                 value is the logarithm to base 2 of the Pdelay_Req
message
                 transmission interval used when (a) the Link Port is
                 initialized, or (b) a message interval request TLV is
received
                 with the logLinkDelayInterval field set to 126.
                 For all other media, the value is 127."
    REFERENCE   "14.16.10"
    ::= { ieee8021AsV2CommonMeanLinkDelayServiceLinkPortDSEntry 11 }

ieee8021AsV2CmldsLinkPortDSCurrentLogPdelayReqInterval OBJECT-TYPE
    SYNTAX      Integer32(-128..127)
    MAX-ACCESS  read-only
    STATUS      current
```

```
    DESCRIPTION
                "For full-duplex, IEEE 802.3 media and CSN media that use
                 the peer-to-peer delay mechanism to measure path delay,
                 the value is the logarithm to the base 2 of the current
                 Pdelay_Req message transmission interval.
                 For all other media, the value is 127."
    REFERENCE   "14.16.11"
    ::= { ieee8021AsV2CommonMeanLinkDelayServiceLinkPortDSEntry 12 }

ieee8021AsV2CmldsLinkPortDSUseMgtSettableLogPdelayReqInterval OBJECT-
TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
                "The managed object is a Boolean that determines the
source
                 of the sync interval and mean time interval between
                successive Pdelay_Req messages. If the value is TRUE (1),
                the value of currentLogPdelayReqInterval is set equal to
                 the value of mgtSettableLogPdelayReqInterval. If the
value
                 of the managed object is FALSE (2), the value of
                 currentLogPdelayReqInterval is determined by the
                 LinkDelayIntervalSetting state machine."
    REFERENCE   "14.16.12"
    DEFVAL { false }
    ::= { ieee8021AsV2CommonMeanLinkDelayServiceLinkPortDSEntry 13 }

ieee8021AsV2CmldsLinkPortDSMgtSettableLogPdelayReqInterval OBJECT-TYPE
    SYNTAX      Integer32 (-128..127)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
                "The value is the logarithm to base 2 of the mean time
                 interval between successive Pdelay_Req messages if
                 useMgtSettableLogPdelayReqInterval is TRUE (1). The
                 value is not used if useMgtSettableLogPdelayReqInterval
                 is FALSE (2)."
    REFERENCE   "14.16.13"
    ::= { ieee8021AsV2CommonMeanLinkDelayServiceLinkPortDSEntry 14 }

ieee8021AsV2CmldsLinkPortDSInitialComputeNbrRateRatio OBJECT-TYPE
    SYNTAX      Integer32 (-128..127)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
                "If useMgtSettableComputeNeighborRateRatio is FALSE (2)
                 then, for full-duplex, IEEE 802.3 media and CSN media
                that use the peer-to-peer delay mechanism to measure path
                 delay, the value is the initial value of
                 computeNeighborRateRatio.
                 For all other media, the value is TRUE."
    REFERENCE   "14.16.14"
```

```
    ::= { ieee8021AsV2CommonMeanLinkDelayServiceLinkPortDSEntry 15 }

ieee8021AsV2CmldsLinkPortDSCurrentComputeNbrRateRatio OBJECT-TYPE
    SYNTAX      Integer32 (-128..127)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
               "For full-duplex, IEEE 802.3 media and CSN media that use
                the peer-to-peer delay mechanism to measure path delay,
                the value is the current value of
computeNeighborRateRatio.
                For all other media, the value is TRUE (1)."
    REFERENCE   "14.16.15"
    ::= { ieee8021AsV2CommonMeanLinkDelayServiceLinkPortDSEntry 16 }

ieee8021AsV2CmldsLinkPortDSUseMgtSettableComputeNbrRateRatio OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
               "The managed object is a Boolean that determines the
source
               of the value of computeNeighborRateRatio. If the value is
                TRUE (1), the value of computeNeighborRateRatio is set
equal
                to the value of mgtSettablecomputeNeighborRateRatio. If
                the value of the managed object is FALSE (2), the
               value of currentComputeNeighborRateRatio is determined by
                the LinkDelayIntervalSetting state machine."
    REFERENCE   "14.16.16"
    DEFVAL { false }
    ::= { ieee8021AsV2CommonMeanLinkDelayServiceLinkPortDSEntry 17 }

ieee8021AsV2CmldsLinkPortDSMgtSettableComputeNbrRateRatio OBJECT-TYPE
    SYNTAX      Integer32 (-128..127)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
               "The value is the logarithm to base 2 of
                computeNeighborRateRatio if
                useMgtSettableComputeNeighborRateRatio is TRUE (1). The
                value is not used if
useMgtSettableComputeNeighborRateRatio
                is FALSE (2)."
    REFERENCE   "14.16.17"
    ::= { ieee8021AsV2CommonMeanLinkDelayServiceLinkPortDSEntry 18 }

ieee8021AsV2CmldsLinkPortDSInitialComputeMeanLinkDelay OBJECT-TYPE
    SYNTAX      Integer32 (-128..127)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
               "If useMgtSettableComputeMeanLinkDelay is FALSE (2) then,
                for full-duplex, IEEE 802.3 media and CSN media that use
```

```
                    the peer-to-peer delay mechanism to measure path delay,
                    the value is the initial value of computeMeanLinkDelay.
                    For all other media, the value is TRUE (1)."
    REFERENCE   "14.16.18"
    ::= { ieee8021AsV2CommonMeanLinkDelayServiceLinkPortDSEntry 19 }

ieee8021AsV2CmldsLinkPortDSCurrentComputeMeanLinkDelay OBJECT-TYPE
    SYNTAX      Integer32 (-128..127)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                "For full-duplex, IEEE 802.3 media and CSN media that use
                    the peer-to-peer delay mechanism to measure path delay,
                    the value is the current value of computeMeanLinkDelay.
                    For all other media, the value is TRUE."
    REFERENCE   "14.16.19"
    ::= { ieee8021AsV2CommonMeanLinkDelayServiceLinkPortDSEntry 20 }

ieee8021AsV2CmldsLinkPortDSUseMgtSettableComputeMeanLinkDelay OBJECT-
TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
                "The managed object is a Boolean that determines the
source
                    of the value of computeMeanLinkDelay. If the value is
                   TRUE (1), the value of computeMeanLinkDelay is set equal
                    to the value of mgtSettableComputeMeanLinkDelay. If the
                    value of the managed object is FALSE (2), the value of
                    currentComputeMeanLinkDelay is determined by the
                    LinkDelayIntervalSetting state machine."
    REFERENCE   "14.16.20"
    DEFVAL { false }
    ::= { ieee8021AsV2CommonMeanLinkDelayServiceLinkPortDSEntry 21 }

ieee8021AsV2CmldsLinkPortDSMgtSettableComputeMeanLinkDelay OBJECT-TYPE
    SYNTAX      Integer32 (-128..127)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
                "The value is the logarithm to base 2 of
computeMeanLinkDelay
                    if useMgtSettableComputeMeanLinkDelay is TRUE (1). The
value
                    is not used if useMgtSettableComputeMeanLinkDelay is
                    FALSE (2)."
    REFERENCE   "14.16.21"
    ::= { ieee8021AsV2CommonMeanLinkDelayServiceLinkPortDSEntry 22 }

ieee8021AsV2CmldsLinkPortDSAllowedLostRsp
 OBJECT-TYPE
    SYNTAX      Unsigned32(1..255)
    MAX-ACCESS  read-write
```

```
    STATUS      current
    DESCRIPTION
                "The value is equal to the value of the per-Link-Port
                 global variable allowedLostResponses. It is the number
                 of Pdelay_Req messages for which a valid response is not
                 received, above which a Link Port is considered to not
                 be exchanging peer delay messages with its neighbor."
    REFERENCE   "14.16.22"
    DEFVAL { 9 }
    ::= { ieee8021AsV2CommonMeanLinkDelayServiceLinkPortDSEntry 23 }


ieee8021AsV2CmldsLinkPortDSAllowedFaults OBJECT-TYPE
    SYNTAX      Unsigned32(1..255)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
                "The value is equal to the value of the per-Link-Port
global
               variable allowedFaults. It is the number of faults, above
               which asCapableAcrossDomains is set to FALSE (2), i.e., a
                 Link Port is considered to not be capable of
interoperating
                 with its neighbor via the IEEE 802.1AS protocol"
    REFERENCE   "14.16.23"
    DEFVAL { 9 }
    ::= { ieee8021AsV2CommonMeanLinkDelayServiceLinkPortDSEntry 24 }

ieee8021AsV2CmldsLinkPortDSVersionNumber OBJECT-TYPE
    SYNTAX      Unsigned32(0..15)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                "This value is set to versionPTP as specified
                 in 10.6.2.2.4."
    REFERENCE   "14.16.24"
    ::= { ieee8021AsV2CommonMeanLinkDelayServiceLinkPortDSEntry 25 }

ieee8021AsV2CmldsLinkPortDSPdelayTruncTST1
 OBJECT-TYPE
    SYNTAX      Ieee8021ASV2Timestamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
              "For full-duplex IEEE 802.3 media, and CSN media that use
the
                peer-to-peer delay mechanism to measure path delay, the
first
                value, T1, of the four elements of this array is as
described
                in Table 14-9. For all other media, the values are zero.
               This object corresponds to the timestamp t1 modulo 2^32 in
               Figure 11-1, and expressed in units of 2^-16 ns (i.e., the
```

```
                    value of this array element is equal to the remainder
obtained
                    upon dividing the respective timestamp , expressed in
units
                    of 2^-16 ns, by 2^48).
                   At any given time, the timestamp values stored in the T1,
T2,
                    T3, T4 PdelayTruncTS are for the same, and most recently
                    completed, peer delay message exchange.
                    NOTE: This managed object is used with the asymmetry
                    measurement compensation procedure, which is based on
                    line-swapping."
    REFERENCE   "14.16.25"
    ::= { ieee8021AsV2CommonMeanLinkDelayServiceLinkPortDSEntry 26 }

ieee8021AsV2CmldsLinkPortDSPdelayTruncTST2
 OBJECT-TYPE
    SYNTAX      Ieee8021ASV2Timestamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                   "For full-duplex IEEE 802.3 media, and CSN media that use
the
                    peer-to-peer delay mechanism to measure path delay, the
second
                    value, T2, of the four elements of this array is as
described
                    in Table 14-9. For all other media, the values are zero.
                   This object corresponds to the timestamp t1 modulo 2^32 in
                   Figure 11-1, and expressed in units of 2^-16 ns (i.e., the
                    value of this array element is equal to the remainder
obtained
                   upon dividing the respective timestamp, expressed in units
                    of 2^-16 ns, by 2^48).
                   At any given time, the timestamp values stored in the T1,
T2,
                    T3, T4 PdelayTruncTS are for the same, and most recently
                    completed, peer delay message exchange.
                    NOTE: This managed object is used with the asymmetry
                    measurement compensation procedure, which is based on
                    line-swapping."
    REFERENCE   "14.16.25"
    ::= { ieee8021AsV2CommonMeanLinkDelayServiceLinkPortDSEntry 27 }

ieee8021AsV2CmldsLinkPortDSPdelayTruncTST3
 OBJECT-TYPE
    SYNTAX      Ieee8021ASV2Timestamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                   "For full-duplex IEEE 802.3 media, and CSN media that use
the
                    peer-to-peer delay mechanism to measure path delay, the
third
```

```
                    value, T3, of the four elements of this array is as
described
                    in Table 14-9. For all other media, the values are zero.
                   This object corresponds to the timestamp t1 modulo 2^32 in
                   Figure 11-1, and expressed in units of 2^-16 ns (i.e., the
                    value of this array element is equal to the remainder
obtained
                   upon dividing the respective timestamp, expressed in units
                    of 2^-16 ns, by 2^48).
                   At any given time, the timestamp values stored in the T1,
T2,
                    T3, T4 PdelayTruncTS are for the same, and most recently
                    completed, peer delay message exchange.
                    NOTE: This managed object is used with the asymmetry
                    measurement compensation procedure, which is based on
                    line-swapping."
    REFERENCE   "14.16.25"
    ::= { ieee8021AsV2CommonMeanLinkDelayServiceLinkPortDSEntry 28 }

ieee8021AsV2CmldsLinkPortDSPdelayTruncTST4
 OBJECT-TYPE
    SYNTAX      Ieee8021ASV2Timestamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                   "For full-duplex IEEE 802.3 media, and CSN media that use
the
                    peer-to-peer delay mechanism to measure path delay, the
fourth
                    value, T4, of the four elements of this array is as
described
                    in Table 14-9. For all other media, the values are zero.
                   This object corresponds to the timestamp t1 modulo 2^32 in
                   Figure 11-1, and expressed in units of 2^-16 ns (i.e., the
                    value of this array element is equal to the remainder
obtained
                   upon dividing the respective timestamp, expressed in units
                    of 2^-16 ns, by 2^48).
                   At any given time, the timestamp values stored in the T1,
T2,
                    T3, T4 PdelayTruncTS are for the same, and most recently
                    completed, peer delay message exchange.
                    NOTE: This managed object is used with the asymmetry
                    measurement compensation procedure, which is based on
                    line-swapping."
    REFERENCE   "14.16.25"
    ::= { ieee8021AsV2CommonMeanLinkDelayServiceLinkPortDSEntry 29 }

ieee8021AsV2CmldsLinkPortDSMinorVersionNumber OBJECT-TYPE
    SYNTAX      Unsigned32 (0..15)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                    "This value is set to minorVersionPTP as specified in
```

```
                  10.6.2.2.3."
        REFERENCE   "14.16.26"
        ::= { ieee8021AsV2CommonMeanLinkDelayServiceLinkPortDSEntry 30 }

-- ================================================================
-- The Common Mean Link Delay Service Link Port Parameter
-- Statistics Data Set provides counters associated with Link
-- Port capabilities at a given time-aware system.
-- ================================================================

ieee8021AsV2CommonMeanLinkDelayServiceLinkPortStatDSTable  OBJECT-TYPE
        SYNTAX      SEQUENCE OF
Ieee8021AsV2CommonMeanLinkDelayServiceLinkPortStatDSEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
           "For every Link Port of the Common Mean Link Delay Service of a
            time-aware system, the following cmldsLinkPortStatisticsDS
provides
            counters. The number of such statistics sets is the same as the
value
            of cmldsDefaultDS.numberLinkPorts."
        REFERENCE   "14.17"
        ::= { ieee8021AsV2MIBObjects 19 }

ieee8021AsV2CommonMeanLinkDelayServiceLinkPortStatDSEntry   OBJECT-TYPE
       SYNTAX      Ieee8021AsV2CommonMeanLinkDelayServiceLinkPortStatDSEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
           "For every Link Port of the Common Mean Link Delay Service of a
            time-aware system, the following cmldsLinkPortStatisticsDS
provides
            counters. The number of such statistics sets is the same as the
value
            of cmldsDefaultDS.numberLinkPorts."
        INDEX { ieee8021AsV2BridgeBasePort,
                ieee8021AsV2CmldsLinkPortStatDSIndex }
        ::= { ieee8021AsV2CommonMeanLinkDelayServiceLinkPortStatDSTable 1 }

Ieee8021AsV2CommonMeanLinkDelayServiceLinkPortStatDSEntry ::=
    SEQUENCE {
        ieee8021AsV2CmldsLinkPortStatDSIndex           InterfaceIndexOrZero,
        ieee8021AsV2CmldsLinkPortStatDSRxPdelayRequestCount
Counter32,
        ieee8021AsV2CmldsLinkPortStatDSRxPdelayRspCount           Counter32,
        ieee8021AsV2CmldsLinkPortStatDSRxPdelayRspFollowUpCount   Counter32,
        ieee8021AsV2CmldsLinkPortStatDSRxPtpPacketDiscardCount
Counter32,
        ieee8021AsV2CmldsLinkPortStatDSPdelayAllowedLostRspExceededCount
Counter32,
        ieee8021AsV2CmldsLinkPortStatDSTxPdelayRequestCount
Counter32,
        ieee8021AsV2CmldsLinkPortStatDSTxPdelayRspCount           Counter32,
```

```
        ieee8021AsV2CmldsLinkPortStatDSTxPdelayRspFollowUpCount   Counter32
          }

    ieee8021AsV2CmldsLinkPortStatDSIndex OBJECT-TYPE
        SYNTAX      InterfaceIndexOrZero
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
                   "An index to identify an entry in the Common Mean Link
                   Port Statistics Data Set."
        REFERENCE   "14.17"
        ::= { ieee8021AsV2CommonMeanLinkDelayServiceLinkPortStatDSEntry 1 }

    ieee8021AsV2CmldsLinkPortStatDSRxPdelayRequestCount OBJECT-TYPE
        SYNTAX      Counter32
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
                  "A counter that increments every time a Pdelay_Req message
    is
                     received."
        REFERENCE   "14.17.2"
        ::= { ieee8021AsV2CommonMeanLinkDelayServiceLinkPortStatDSEntry 2 }

    ieee8021AsV2CmldsLinkPortStatDSRxPdelayRspCount OBJECT-TYPE
        SYNTAX      Counter32
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
                    "A counter that increments every time a Pdelay_Resp
    message is
                     received."
        REFERENCE   "14.17.3"
        ::= { ieee8021AsV2CommonMeanLinkDelayServiceLinkPortStatDSEntry 3 }

    ieee8021AsV2CmldsLinkPortStatDSRxPdelayRspFollowUpCount OBJECT-TYPE
        SYNTAX      Counter32
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
                    "A counter that increments every time a
    Pdelay_Resp_Follow_Up
                     message is received."
        REFERENCE   "14.17.4"
        ::= { ieee8021AsV2CommonMeanLinkDelayServiceLinkPortStatDSEntry 4 }

    ieee8021AsV2CmldsLinkPortStatDSRxPtpPacketDiscardCount
     OBJECT-TYPE
        SYNTAX      Counter32
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
                   "A counter that increments every time a PTP message of the
```

```
                  Common Mean Link Delay Service is discarded, caused by
the
                  occurrence of any of the conditions given in 14.17.5."
    REFERENCE   "14.17.5"
    ::= { ieee8021AsV2CommonMeanLinkDelayServiceLinkPortStatDSEntry 5 }

ieee8021AsV2CmldsLinkPortStatDSPdelayAllowedLostRspExceededCount
 OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                 "A counter that increments every time the value of the
variable
                  lostResponses exceeds the value of the variable
                  allowedLostResponses, in the RESET state of the
                  MDPdelayReq state machine."
    REFERENCE   "14.17.6"
    ::= { ieee8021AsV2CommonMeanLinkDelayServiceLinkPortStatDSEntry 6 }

ieee8021AsV2CmldsLinkPortStatDSTxPdelayRequestCount
 OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
               "A counter that increments every time a Pdelay_Req message
is
                  transmitted."
    REFERENCE   "14.17.7"
    ::= { ieee8021AsV2CommonMeanLinkDelayServiceLinkPortStatDSEntry 7 }

ieee8021AsV2CmldsLinkPortStatDSTxPdelayRspCount
 OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                 "A counter that increments every time a Pdelay_Resp
message is
                  transmitted."
    REFERENCE   "14.17.8"
    ::= { ieee8021AsV2CommonMeanLinkDelayServiceLinkPortStatDSEntry 8 }

ieee8021AsV2CmldsLinkPortStatDSTxPdelayRspFollowUpCount
 OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
                 "A counter that increments every time a
Pdelay_Resp_Follow_Up
                  message is transmitted."
    REFERENCE   "14.17.9"
```

```
    ::= { ieee8021AsV2CommonMeanLinkDelayServiceLinkPortStatDSEntry 9 }

-- =============================================================
-- The Common Mean Link Delay Service Asymmetry Measurement Mode
-- Parameter Data Set represents the capability to enable/disable
-- the Asymmetry Compensation Measurement Procedure on a Link Port
-- (see Annex G).
-- =============================================================

ieee8021AsV2CommonMeanLinkDelayServiceAsymMeasurementModeDSTable
OBJECT-TYPE
    SYNTAX      SEQUENCE OF
Ieee8021AsV2CommonMeanLinkDelayServiceAsymMeasurementModeDSEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "The Common Mean Link Delay Service Asymmetry Measurement Mode
        Parameter Data Set represents the capability to enable/disable
        the Asymmetry Compensation Measurement Procedure on a Link Port
        (see Annex G)."
    REFERENCE   "14.18"
    ::= { ieee8021AsV2MIBObjects 20 }

ieee8021AsV2CommonMeanLinkDelayServiceAsymMeasurementModeDSEntry
OBJECT-TYPE
    SYNTAX
Ieee8021AsV2CommonMeanLinkDelayServiceAsymMeasurementModeDSEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "This table uses
        ieee8021AsV2CmldsAsymmetryMeasurementModeDSAsIndex, and
        corresponds to
ieee8021AsV2CommonMeanLinkDelayServiceAsymmetryMeasurementModeDSTable
        entry.
       "
    INDEX { ieee8021AsV2BridgeBasePort,
            ieee8021AsV2CmldsAsymMeasurementModeDSAsIndex }
    ::= {
ieee8021AsV2CommonMeanLinkDelayServiceAsymMeasurementModeDSTable 1 }

Ieee8021AsV2CommonMeanLinkDelayServiceAsymMeasurementModeDSEntry ::=
     SEQUENCE {
 ieee8021AsV2CmldsAsymMeasurementModeDSAsIndex InterfaceIndexOrZero,
 ieee8021AsV2CmldsAsymMeasurementModeDSAsymMeasurementMode    TruthValue
      }

ieee8021AsV2CmldsAsymMeasurementModeDSAsIndex OBJECT-TYPE
    SYNTAX      InterfaceIndexOrZero
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
               "This object identifies the gPTP interface group within
                the system for which this entry contains information. It
```

```
                    is the value of the instance of the IfIndex object,
                    defined in the IF-MIB, for the gPTP interface group
                    corresponding to this port, or the value 0 if the port
                    has not been bound to an underlying frame source and
                    sink.

                    For a given media port of a Bridge or an end station,
                    there can be one or more gPTP Port, and depends whether
                    a media port supports point to point link (e.g. IEEE
                    802.3 Ethernet) or point to multi-point (e.g. CSN, IEEE
                    802.3 EPON, etc) links on the media port."
        REFERENCE   "IEEE Std 802.1AS
CommonMeanLinkDelaySvcAsymMeasurementModeParamDS Group gPTP Port Index"
  ::= { ieee8021AsV2CommonMeanLinkDelayServiceAsymMeasurementModeDSEntry
1 }

ieee8021AsV2CmldsAsymMeasurementModeDSAsymMeasurementMode
        OBJECT-TYPE
        SYNTAX      TruthValue
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
                    "The value is equal to the value of the Boolean
                     asymmetryMeasurementMode(see G.3). For full-duplex
                     IEEE 802.3 media, the value is TRUE (1) if an asymmetry
                     measurement is being performed for the link attached to
                     this Link Port, and FALSE (2) otherwise. For all other
                     media, the value shall be FALSE (2) (see 10.2.4.2).
                     NOTE: If an asymmetry measurement is being performed
                     for a link, asymmetryMeasurementMode must be TRUE (1)
                     for the Link Ports at each end of the link.
                     There is one Common Mean Link Delay Service Asymmetry
                     Measurement Mode Parameter Data Set Table for all PTP
                     Instances, per Link Port.
                "
        REFERENCE   "14.18.2"
  ::= { ieee8021AsV2CommonMeanLinkDelayServiceAsymMeasurementModeDSEntry
2 }

-- *****************************************************************
-- IEEE 802.1ASV2 MIB - Conformance Information
-- *****************************************************************
ieee8021AsV2Groups          OBJECT IDENTIFIER ::= {
ieee8021AsV2Conformance 1 }
ieee8021AsV2Compliances     OBJECT IDENTIFIER ::= {
ieee8021AsV2Conformance 2 }

-- ============================================================
-- units of conformance
-- ============================================================

ieee8021AsV2PtpInstanceGroup OBJECT-GROUP
        OBJECTS {
            ieee8021AsV2PtpInstanceName,
```

```
            ieee8021AsV2PtpInstanceRowStatus
        }
        STATUS      current
        DESCRIPTION
            "A collection of objects providing information for dynamic
creation and deletion
             of PTP Instances and logical ports."
        ::= { ieee8021AsV2Groups 1 }

ieee8021AsV2DefaultDSGroup OBJECT-GROUP
        OBJECTS {
            ieee8021AsV2DefaultDSClockIdentity,
            ieee8021AsV2DefaultDSNumberPorts,
            ieee8021AsV2DefaultDSClockQualityClockClass,
            ieee8021AsV2DefaultDSClockQualityClockAccuracy,
            ieee8021AsV2DefaultDSClockQualityOffsetScaledLogVariance,
            ieee8021AsV2DefaultDSPriority1,
            ieee8021AsV2DefaultDSPriority2,
            ieee8021AsV2DefaultDSGmCapable,
            ieee8021AsV2DefaultDSCurrentUtcOffset,
            ieee8021AsV2DefaultDSCurrentUtcOffsetValid,
            ieee8021AsV2DefaultDSLeap59,
            ieee8021AsV2DefaultDSLeap61,
            ieee8021AsV2DefaultDSTimeTraceable,
            ieee8021AsV2DefaultDSFrequencyTraceable,
            ieee8021AsV2DefaultDSPtpTimescale,
            ieee8021AsV2DefaultDSTimeSource,
            ieee8021AsV2DefaultDSDomainNumber,
            ieee8021AsV2DefaultDSSdoId,
            ieee8021AsV2DefaultDSExternalPortConfigurationEnabled,
            ieee8021AsV2DefaultDSInstanceEnable
        }
        STATUS      current
        DESCRIPTION
            "A collection of objects providing information on the Default
Parameter Data Set representing the native capabilities
             of a PTP Instance, i.e., a PTP Relay Instance or a PTP End
Instance."
        ::= { ieee8021AsV2Groups 2 }

ieee8021AsV2CurrentDSGroup OBJECT-GROUP
        OBJECTS {
            ieee8021AsV2CurrentDSStepsRemoved,
            ieee8021AsV2CurrentDSOffsetFromMaster,
            ieee8021AsV2CurrentDSLastGmPhaseChange,
            ieee8021AsV2CurrentDSLastGmFreqChange,
            ieee8021AsV2CurrentDSGmTimebaseIndicator,
            ieee8021AsV2CurrentDSGmChangeCount,
            ieee8021AsV2CurrentDSTimeOfLastGmChangeEvent,
            ieee8021AsV2CurrentDSTimeOfLastGmPhaseChangeEvent,
            ieee8021AsV2CurrentDSTimeOfLastGmFreqChangeEvent
        }
        STATUS      current
        DESCRIPTION
```

```
        "A collection of objects providing information on the Current
Parameter Data Set representing the position of a local
         system and other information, relative to the grandmaster."
    ::= { ieee8021AsV2Groups 3 }

ieee8021AsV2ParentDSGroup OBJECT-GROUP
    OBJECTS {
        ieee8021AsV2ParentDSParentClockIdentity,
        ieee8021AsV2ParentDSParentPortNumber,
        ieee8021AsV2ParentDSCumlativeRateRatio,
        ieee8021AsV2ParentDSGrandmasterIdentity,
        ieee8021AsV2ParentDSGrandmasterClockQualityclockClass,
        ieee8021AsV2ParentDSGrandmasterClockQualityclockAccuracy,
        ieee8021AsV2ParentDSGrandmasterClockQualityoffsetScaledLogVar,
        ieee8021AsV2ParentDSGrandmasterPriority1,
        ieee8021AsV2ParentDSGrandmasterPriority2
    }
    STATUS      current
    DESCRIPTION
        "A collection of objects providing information on the Parent
Parameter Data Set representing capabilities of the
         upstream system, toward the grandmaster, as measured at a local
system."
    ::= { ieee8021AsV2Groups 4 }

ieee8021AsV2TimePropertiesDSGroup OBJECT-GROUP
    OBJECTS {
        ieee8021AsV2TimePropertiesDSCurrentUtcOffset,
        ieee8021AsV2TimePropertiesDSCurrentUtcOffsetValid,
        ieee8021AsV2TimePropertiesDSLeap59,
        ieee8021AsV2TimePropertiesDSLeap61,
        ieee8021AsV2TimePropertiesDSTimeTraceable,
        ieee8021AsV2TimePropertiesDSFrequencyTraceable,
        ieee8021AsV2TimePropertiesDSPtpTimescale,
        ieee8021AsV2TimePropertiesDSTimeSource
    }
    STATUS      current
    DESCRIPTION
        "A collection of objects providing information on the Time
Properties Parameter Data Set representing capabilities of
         the grandmaster, as measured at a local system."
    ::= { ieee8021AsV2Groups 5 }

ieee8021AsV2PathTraceDSGroup OBJECT-GROUP
    OBJECTS {
        ieee8021AsV2PathTraceDSEnable
    }
    STATUS      current
    DESCRIPTION
        "A collection of objects providing information on the Path Trace
Data Set representing the current path trace information
         available at the PTP Instance."
    ::= { ieee8021AsV2Groups 6 }
```

```
ieee8021AsV2PathTraceDSArrayTableGroup OBJECT-GROUP
    OBJECTS {
        ieee8021AsV2PathTraceDSArrayList
    }
    STATUS      current
    DESCRIPTION
        "A collection of objects providing information of an array of
ClockIdentity values contained
         in the pathTrace array, representing the current path trace
information, and which is
         carried in the path trace TLV per PTP Instance."
    ::= { ieee8021AsV2Groups 7 }

ieee8021AsV2AcceptableMasterTableDSGroup OBJECT-GROUP
    OBJECTS {
        ieee8021AsV2AcceptableMasterTableDSMaxTableSize,
        ieee8021AsV2AcceptableMasterTableDSActualTableSize
    }
    STATUS      current
    DESCRIPTION
       "A collection of objects providing information on the Acceptable
Master Table Data Set representing the acceptable master
        table used when an EPON port is used by a PTP Instance of a time-
aware system."
    ::= { ieee8021AsV2Groups 8 }

ieee8021AsV2AcceptableMasterTableDSArrayGroup OBJECT-GROUP
    OBJECTS {
        ieee8021AsV2AcceptableMasterTableDSArrayPortIdentity,
        ieee8021AsV2AcceptableMasterTableDSArrayAlternatePriority1
    }
    STATUS      current
    DESCRIPTION
       "A collection of objects providing information on the Acceptable
Master Table Array Data Set representing the acceptable master
       table used when an EPON port is used by a PTP Instance of a time-
aware system."
    ::= { ieee8021AsV2Groups 9 }

ieee8021AsV2PortDSGroup OBJECT-GROUP
    OBJECTS {
        ieee8021AsV2PortDSClockIdentity,
        ieee8021AsV2PortDSPortNumber,
        ieee8021AsV2PortDSPortState,
        ieee8021AsV2PortDSPtpPortEnabled,
        ieee8021AsV2PortDSDelayMechanism,
        ieee8021AsV2PortDSIsMeasuringDelay,
        ieee8021AsV2PortDSAsCapable,
        ieee8021AsV2PortDSMeanLinkDelay,
        ieee8021AsV2PortDSMeanLinkDelayThresh,
        ieee8021AsV2PortDSDelayAsym,
        ieee8021AsV2PortDSNbrRateRatio,
        ieee8021AsV2PortDSInitialLogAnnounceInterval,
        ieee8021AsV2PortDSCurrentLogAnnounceInterval,
```

```
            ieee8021AsV2PortDSUseMgtSettableLogAnnounceInterval,
            ieee8021AsV2PortDSMgtSettableLogAnnounceInterval,
            ieee8021AsV2PortDSAnnounceReceiptTimeout,
            ieee8021AsV2PortDSInitialLogSyncInterval,
            ieee8021AsV2PortDSCurrentLogSyncInterval,
            ieee8021AsV2PortDSUseMgtSettableLogSyncInterval,
            ieee8021AsV2PortDSMgtSettableLogSyncInterval,
            ieee8021AsV2PortDSSyncReceiptTimeout,
            ieee8021AsV2PortDSSyncReceiptTimeoutTimeInterval,
            ieee8021AsV2PortDSInitialLogPdelayReqInterval,
            ieee8021AsV2PortDSCurrentLogPdelayReqInterval,
            ieee8021AsV2PortDSUseMgtSettableLogPdelayReqInterval,
            ieee8021AsV2PortDSMgtSettableLogPdelayReqInterval,
            ieee8021AsV2PortDSInitialLogGptpCapableMessageInterval,
            ieee8021AsV2PortDSCurrentLogGptpCapableMessageInterval,
            ieee8021AsV2PortDSUseMgtSettableLogGptpCapableMessageInterval,
            ieee8021AsV2PortDSMgtSettableLogGptpCapableMessageInterval,
            ieee8021AsV2PortDSInitialComputeNbrRateRatio,
            ieee8021AsV2PortDSCurrentComputeNbrRateRatio,
            ieee8021AsV2PortDSUseMgtSettableComputeNbrRateRatio,
            ieee8021AsV2PortDSMgtSettableComputeNbrRateRatio,
            ieee8021AsV2PortDSInitialComputeMeanLinkDelay,
            ieee8021AsV2PortDSCurrentComputeMeanLinkDelay,
            ieee8021AsV2PortDSUseMgtSettableComputeMeanLinkDelay,
            ieee8021AsV2PortDSMgtSettableComputeMeanLinkDelay,
            ieee8021AsV2PortDSAllowedLostRsp,
            ieee8021AsV2PortDSAllowedFaults,
            ieee8021AsV2PortDSLogGptpCapableMessageInterval,
            ieee8021AsV2PortDSGPtpCapableReceiptTimeout,
            ieee8021AsV2PortDSVersionNumber,
            ieee8021AsV2PortDSNup,
            ieee8021AsV2PortDSNdown,
            ieee8021AsV2PortDSOneStepTxOper,
            ieee8021AsV2PortDSOneStepReceive,
            ieee8021AsV2PortDSOneStepTransmit,
            ieee8021AsV2PortDSInitialOneStepTxOper,
            ieee8021AsV2PortDSCurrentOneStepTxOper,
            ieee8021AsV2PortDSUseMgtSettableOneStepTxOper,
            ieee8021AsV2PortDSMgtSettableOneStepTxOper,
            ieee8021AsV2PortDSSyncLocked,
            ieee8021AsV2PortDSPdelayTruncTST1,
            ieee8021AsV2PortDSPdelayTruncTST2,
            ieee8021AsV2PortDSPdelayTruncTST3,
            ieee8021AsV2PortDSPdelayTruncTST4,
            ieee8021AsV2PortDSMinorVersionNumber
        }
        STATUS      current
        DESCRIPTION
            "A collection of objects providing information on gPTP Port
    related variables in a time-aware Bridge or for a time-aware end
    station."
        ::= { ieee8021AsV2Groups 10 }

    ieee8021AsV2DescriptionPortDSGroup OBJECT-GROUP
```

```
    OBJECTS {
        ieee8021AsV2DescriptionPortDSProfileIdentifier
    }
    STATUS      current
    DESCRIPTION
       "A collection of objects providing information on the Description
Port Data Set containing the profileIdentifier for
         this PTP profile, as specified in Annex F.1."
    ::= { ieee8021AsV2Groups 11 }

ieee8021AsV2PortStatIfGroup OBJECT-GROUP
    OBJECTS {
        ieee8021AsV2PortStatRxSyncCount,
        ieee8021AsV2PortStatRxOneStepSyncCount,
        ieee8021AsV2PortStatRxFollowUpCount,
        ieee8021AsV2PortStatRxPdelayRequestCount,
        ieee8021AsV2PortStatRxPdelayRspCount,
        ieee8021AsV2PortStatRxPdelayRspFollowUpCount,
        ieee8021AsV2PortStatRxAnnounceCount,
        ieee8021AsV2PortStatRxPtpPacketDiscardCount,
        ieee8021AsV2PortStatSyncReceiptTimeoutCount,
        ieee8021AsV2PortStatAnnounceReceiptTimeoutCount,
        ieee8021AsV2PortStatPdelayAllowedLostRspExceededCount,
        ieee8021AsV2PortStatTxSyncCount,
        ieee8021AsV2PortStatTxOneStepSyncCount,
        ieee8021AsV2PortStatTxFollowUpCount,
        ieee8021AsV2PortStatTxPdelayRequestCount,
        ieee8021AsV2PortStatTxPdelayRspCount,
        ieee8021AsV2PortStatTxPdelayRspFollowUpCount,
        ieee8021AsV2PortStatTxAnnounceCount
    }
    STATUS      current
    DESCRIPTION
        "A collection of objects providing information on the Port
Statistics Data Set provideing counters associated with port
         capabilities at a given PTP Instance."
    ::= { ieee8021AsV2Groups 12 }

ieee8021AsV2AcceptableMasterPortDSGroup OBJECT-GROUP
    OBJECTS {
        ieee8021AsV2AcceptableMasterPortDSAcceptableMasterTableEnabled
    }
    STATUS      current
    DESCRIPTION
        "A collection of objects providing information for the single
port of a PTP End Instance and for each port
         of a PTP Relay Instance."
    ::= { ieee8021AsV2Groups 13 }

ieee8021AsV2ExternalPortConfigurationPortDSGroup OBJECT-GROUP
    OBJECTS {
        ieee8021AsV2ExternalPortConfigurationPortDSDesiredState
    }
    STATUS      current
```

```
    DESCRIPTION
        "A collection of objects providing information on the External
Port Configuration Port Data Set containing the single member
         desiredState, which indicates the desired state for the gPTP
Port."
    ::= { ieee8021AsV2Groups 14 }

ieee8021AsV2AsymMeasurementModeDSGroup OBJECT-GROUP
    OBJECTS {
        ieee8021AsV2AsymMeasurementModeDSAsymMeasurementMode
    }
    STATUS      current
    DESCRIPTION
        "A collection of objects providing information on the Asymmetry
Measurement Mode Data Set representing the capability to
        enable/disable the Asymmetry Compensation Measurement Procedure
on a Link Port (see Annex G)."
    ::= { ieee8021AsV2Groups 15 }

ieee8021AsV2CommonServicesPortDSGroup OBJECT-GROUP
    OBJECTS {
        ieee8021AsV2CommonServicesPortDSCmldsLinkPortPortNumber
    }
    STATUS      current
    DESCRIPTION
        "A collection of objects providing information on the Common
Services Port Data Set."
    ::= { ieee8021AsV2Groups 16 }

ieee8021AsV2CommonMeanLinkDelayServiceDefaultDSGroup OBJECT-GROUP
    OBJECTS {
        ieee8021AsV2CmldsDefaultDSClockIdentity,
        ieee8021AsV2CmldsDefaultDSNumberLinkPorts,
        ieee8021AsV2CmldsDefaultDSSdoID
    }
    STATUS      current
    DESCRIPTION
        "A collection of objects providing information on the CMLDs
Default Data Set describing the per-time-aware-system attributes
         of the Common Mean Link Delay Service."
    ::= { ieee8021AsV2Groups 17 }

ieee8021AsV2CommonMeanLinkDelayServiceLinkPortDSGroup OBJECT-GROUP
    OBJECTS {
        ieee8021AsV2CmldsLinkPortDSClockIdentity,
        ieee8021AsV2CmldsLinkPortDSPortNumber,
        ieee8021AsV2CmldsLinkPortDSCmldsLinkPortEnabled,
        ieee8021AsV2CmldsLinkPortDSIsMeasuringDelay,
        ieee8021AsV2CmldsLinkPortDSAsCapableAcrossDomains,
        ieee8021AsV2CmldsLinkPortDSMeanLinkDelay,
        ieee8021AsV2CmldsLinkPortDSMeanLinkDelayThresh,
        ieee8021AsV2CmldsLinkPortDSDelayAsym,
        ieee8021AsV2CmldsLinkPortDSNbrRateRatio,
        ieee8021AsV2CmldsLinkPortDSInitialLogPdelayReqInterval,
```

```
            ieee8021AsV2CmldsLinkPortDSCurrentLogPdelayReqInterval,
            ieee8021AsV2CmldsLinkPortDSUseMgtSettableLogPdelayReqInterval,
            ieee8021AsV2CmldsLinkPortDSMgtSettableLogPdelayReqInterval,
            ieee8021AsV2CmldsLinkPortDSInitialComputeNbrRateRatio,
            ieee8021AsV2CmldsLinkPortDSCurrentComputeNbrRateRatio,
            ieee8021AsV2CmldsLinkPortDSUseMgtSettableComputeNbrRateRatio,
            ieee8021AsV2CmldsLinkPortDSMgtSettableComputeNbrRateRatio,
            ieee8021AsV2CmldsLinkPortDSInitialComputeMeanLinkDelay,
            ieee8021AsV2CmldsLinkPortDSCurrentComputeMeanLinkDelay,
            ieee8021AsV2CmldsLinkPortDSUseMgtSettableComputeMeanLinkDelay,
            ieee8021AsV2CmldsLinkPortDSMgtSettableComputeMeanLinkDelay,
            ieee8021AsV2CmldsLinkPortDSAllowedLostRsp,
            ieee8021AsV2CmldsLinkPortDSAllowedFaults,
            ieee8021AsV2CmldsLinkPortDSVersionNumber,
            ieee8021AsV2CmldsLinkPortDSPdelayTruncTST1,
            ieee8021AsV2CmldsLinkPortDSPdelayTruncTST2,
            ieee8021AsV2CmldsLinkPortDSPdelayTruncTST3,
            ieee8021AsV2CmldsLinkPortDSPdelayTruncTST4,
            ieee8021AsV2CmldsLinkPortDSMinorVersionNumber
        }
        STATUS      current
        DESCRIPTION
            "A collection of objects providing information for every Link
    Port of the Common Mean Link Delay Service of a
             time-aware system."
        ::= { ieee8021AsV2Groups 18 }

    ieee8021AsV2CommonMeanLinkDelayServiceLinkPortStatDSGroup OBJECT-GROUP
        OBJECTS {
            ieee8021AsV2CmldsLinkPortStatDSRxPdelayRequestCount,
            ieee8021AsV2CmldsLinkPortStatDSRxPdelayRspCount,
            ieee8021AsV2CmldsLinkPortStatDSRxPdelayRspFollowUpCount,
            ieee8021AsV2CmldsLinkPortStatDSRxPtpPacketDiscardCount,

    ieee8021AsV2CmldsLinkPortStatDSPdelayAllowedLostRspExceededCount,
            ieee8021AsV2CmldsLinkPortStatDSTxPdelayRequestCount,
            ieee8021AsV2CmldsLinkPortStatDSTxPdelayRspCount,
            ieee8021AsV2CmldsLinkPortStatDSTxPdelayRspFollowUpCount
        }
        STATUS      current
        DESCRIPTION
            "A collection of objects providing information for every Link
    Port Statistics of the Common Mean Link Delay Service of a
             time-aware system."
        ::= { ieee8021AsV2Groups 19 }

    ieee8021AsV2CommonMeanLinkDelayServiceAsymMeasurementModeDSGroup OBJECT-
    GROUP
        OBJECTS {
            ieee8021AsV2CmldsAsymMeasurementModeDSAsymMeasurementMode
        }
        STATUS      current
        DESCRIPTION
```

```
        "A collection of objects providing information on the Common Mean
Link Delay Service Asymmetry Measurement Mode
         Parameter Data Set representing the capability to enable/disable
the Asymmetry Compensation Measurement Procedure on a Link Port (see
Annex G)."
    ::= { ieee8021AsV2Groups 20 }

-- ================================================================
-- compliance statements
-- ================================================================

ieee8021AsV2Compliance MODULE-COMPLIANCE
    STATUS      current
    DESCRIPTION
        "The compliance statement for devices supporting
         IEEE Std 802.1AS-2019."

    MODULE -- this module

    GROUP ieee8021AsV2PtpInstanceGroup
    DESCRIPTION
        "Implementation of this group is optional."

    GROUP ieee8021AsV2DefaultDSGroup
    DESCRIPTION
        "Implementation of this group is optional."

    GROUP ieee8021AsV2CurrentDSGroup
    DESCRIPTION
        "Implementation of this group is optional."

    GROUP ieee8021AsV2ParentDSGroup
    DESCRIPTION
        "Implementation of this group is optional."

    GROUP ieee8021AsV2TimePropertiesDSGroup
    DESCRIPTION
        "Implementation of this group is optional."

    GROUP ieee8021AsV2PathTraceDSGroup
    DESCRIPTION
        "Implementation of this group is optional."

    GROUP ieee8021AsV2PathTraceDSArrayTableGroup
    DESCRIPTION
        "Implementation of this group is optional."

    GROUP ieee8021AsV2AcceptableMasterTableDSGroup
    DESCRIPTION
        "Implementation of this group is optional."

    GROUP ieee8021AsV2AcceptableMasterTableDSArrayGroup
    DESCRIPTION
       "Implementation of this group is optional."
```

```
    GROUP ieee8021AsV2PortDSGroup
    DESCRIPTION
        "Implementation of this group is optional."

    GROUP ieee8021AsV2DescriptionPortDSGroup
    DESCRIPTION
        "Implementation of this group is optional."

    GROUP ieee8021AsV2PortStatIfGroup
    DESCRIPTION
        "Implementation of this group is optional."

    GROUP ieee8021AsV2AcceptableMasterPortDSGroup
    DESCRIPTION
        "Implementation of this group is optional."

    GROUP ieee8021AsV2ExternalPortConfigurationPortDSGroup
    DESCRIPTION
        "Implementation of this group is optional."

    GROUP ieee8021AsV2AsymMeasurementModeDSGroup
    DESCRIPTION
        "Implementation of this group is optional."

    GROUP ieee8021AsV2CommonServicesPortDSGroup
    DESCRIPTION
        "Implementation of this group is optional."

    GROUP ieee8021AsV2CommonMeanLinkDelayServiceDefaultDSGroup
    DESCRIPTION
        "Implementation of this group is optional."

    GROUP ieee8021AsV2CommonMeanLinkDelayServiceLinkPortDSGroup
    DESCRIPTION
        "Implementation of this group is optional."

    GROUP ieee8021AsV2CommonMeanLinkDelayServiceLinkPortStatDSGroup
    DESCRIPTION
        "Implementation of this group is optional."

    GROUP
ieee8021AsV2CommonMeanLinkDelayServiceAsymMeasurementModeDSGroup
    DESCRIPTION
        "Implementation of this group is optional."

    ::= { ieee8021AsV2Compliances 1 }

END
```

16. Media-dependent layer specification for CSN Network

16.1 Overview

Accurate synchronized time is distributed throughout a gPTP domain through time measurements between adjacent PTP Relay Instances or PTP End Instrances in a packet network. Time is communicated from the root of the clock spanning tree (i.e., the grandmaster) toward the leaves of the tree (i.e., from leaf-facing "master" ports to root-facing "slave" ports) through measurements made across the links connecting the PTP Instances. While the semantics of time transfer are consistent across the time-aware packet network, the method for communicating synchronized time from a master station to its immediate downstream link partner varies depending on the type of link interconnecting the two PTP Instances.

This appendix specifies the protocol that provides accurate synchronized time across links of a *coordinated shared network* (CSN) as part of a packet network.

16.2 Coordinated Shared Network characteristics

A CSN is a contention-free, time-division, multiplexed-access network of devices sharing a common medium and supporting reserved bandwidth based on priority or flow (QoS). One of the nodes of the CSN acts as the network coordinator, granting transmission opportunities to the otherCSN nodes of the network. A CSN network physically is a shared medium, in that a CSN node has a single physical port connected to the half-duplex medium, but is logically a fully connected one-hop mesh network, in that everyCSN node can transmit frames to every other CSN node over the shared medium.

A CSN supports two types of transmission: unicast transmission for point-to-point (CSN node-to-node) transmission and multicast/broadcast transmission for point-to-multipoint (CSN node-to-other/all-nodes) transmission. Figure 16-1 illustrates a CSN network acting as a backbone for PTP Instances.

NOTE—In this clause, the term *node* is used to refer to a CSN node (i.e., it does not refer to a PTP Relay Instance or PTP End Instance). A CSN node is a 2-port PTP Relay Instance that forwards data packets between a segment external to the CSN (which can connect to an upstream or downstream PTP Instance) and the CSN network, all at the data link layer. Nonetheless, to avoid confusion the term *node* is usually preceded by *CSN*, except in cases where it is obvious that CSN nodes are being referred to.

16.3 Layering for CSN links

One PortSync entity and one MD entity are together associated with each CSN logical port (CSN node-to-node link) as illustrated in Figure 16-2. The PortSync entities is described in 10.1.1. The MD entity translates media-independent primitives to MD primitives as necessary for communicating synchronized time over the CSN links. The CSN MD entity shall implement the MDSyncSendSM and MDSyncReceiveSM states machines of 11.2.14 and 11.2.15.

The CSN MD entity either implements the MDPdelayReq and MDPdelayResp state machines of 11.2.19 and 11.2.20 to measure the propagation delay on a CSN link, or measures it through a CSN-native method and populates the variables described in 16.4.3.2.

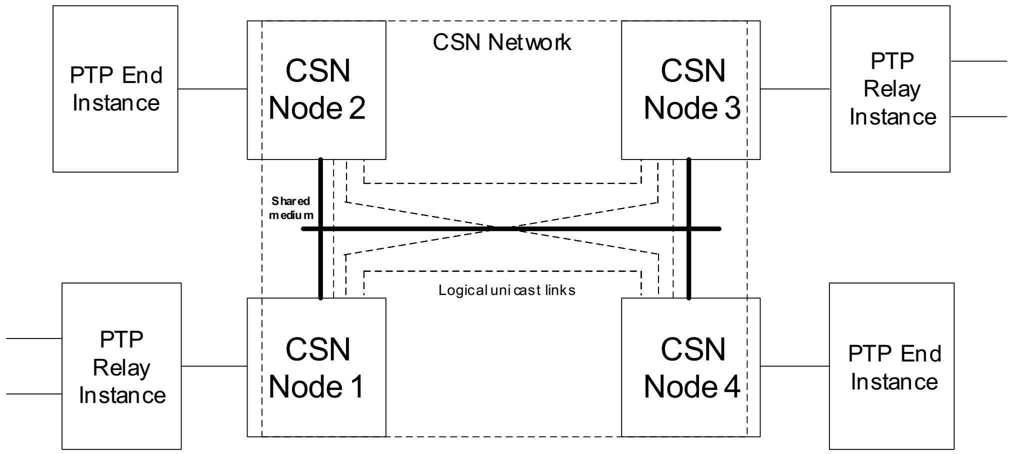


Figure 16-1—Example of CSN backbone in a TSN LAN

16.4 Path delay measurement over a CSN backbone

16.4.1 General

The Path Delay over a CSN backbone is calculated for the following path types: (1) between the upstream PTP Relay Instance and the ingress CSN node, (2) between the ingress and egress CSN nodes, and (3) between the egress CSN node and the downstream PTP Instance (PTP Relay Instance or PTP End Instance).

To maintain the synchronization, residence time on each PTP Instance and the propagation delay between PTP Instances is measured, requiring precise timestamping on both CSN node ingress and egress ports as illustrated in Figure 16-4 ("Path i" in the figure refers to the paths enumerated in Figure 16-3). In Figure 16-4, ti_1 is the syncEventEgressTimestamp for the Sync message at the upstream PTP Relay Instance, ti_2 is the syncEventIngressTimestamp for the Sync message at the ingress CSN time-aware node, te_1 is the syncEventEgressTimestamp for the Sync message at the egress CSN time-aware node, and te_2 is the syncEventIngressTimestamp for the Sync message at the downstream PTP Relay Instance or PTP End Instance.

16.4.2 Path delay measurement between CSN node and neighbor PTP Instance

The path delay measurement between a CSN node and a neighbor PTP Instance is made as specified for the respective medium. This path delay measurement is made for the link between the CSN node and the neighbor PTP Instance.

16.4.3 Path delay measurement between CSN nodes

The path delay between the two nodes of a CSN is the propagation delay for the logical link that connects those two nodes. The method of measuring the path delay between two CSN nodes has two variations which are described in 16.4.3.1 and 16.4.3.2, respectively. The specific method to be used for a specific link technology is specified in 16.6.

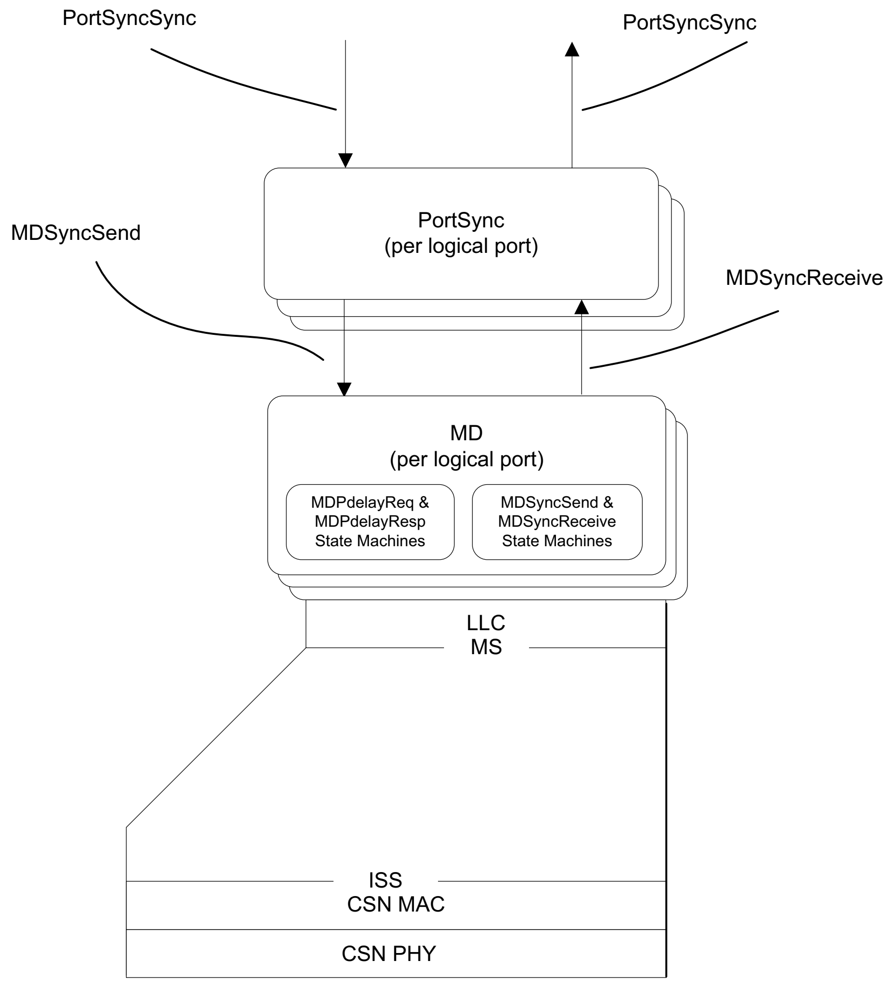


Figure 16-2—Media-dependent and lower entities in CSN nodes

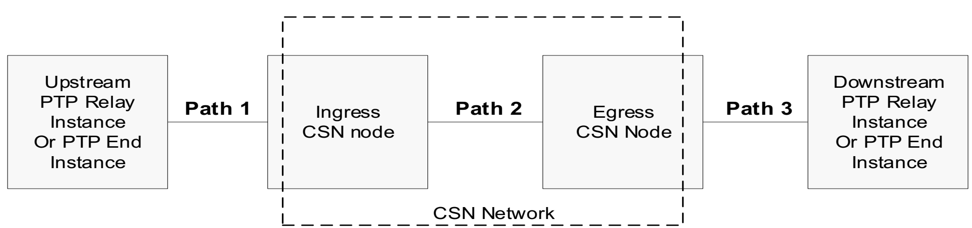


Figure 16-3—Path types over CSN as IEEE 802.1AS backbone

16.4.3.1 Path delay measurement without network clock reference

Each CSN node has a free-running local clock. The path delay measurement uses the peer-to-peer delay mechanism protocol, messages, and state machines described in Clause 11 for full-duplex, point-to-point

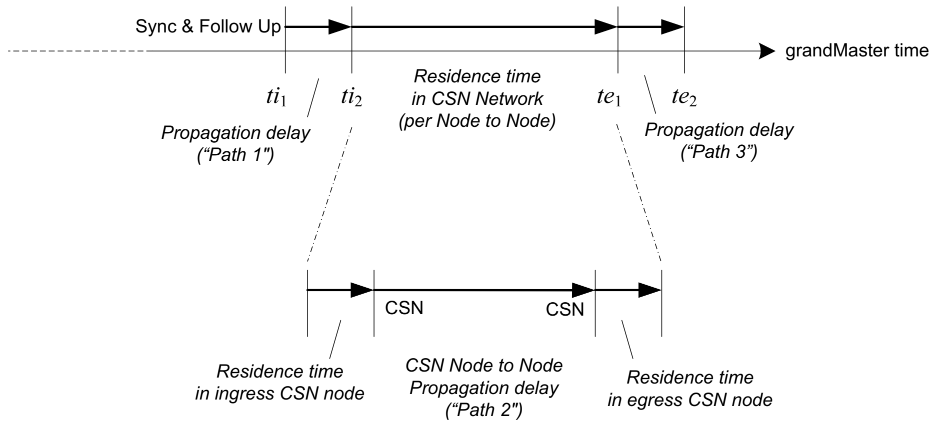


Figure 16-4—Propagation delay and residence time over a CSN backbone

links, as illustrated by Figure 16-5. The criteria of 11.2.17 for determining whether the peer-to-peer delay mechanism is instance-specific or is provided by CMLDS apply here.

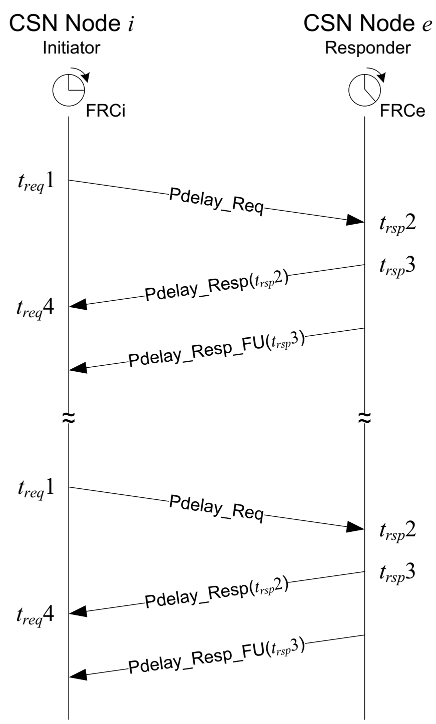


Figure 16-5—CSN node-to-node path delay measurement

The computation of the neighborRateRatio and meanLinkDelay between two CSN nodes is done using the timestamps at the initiator and information conveyed in the successive Pdelay_Resp and Pdelay_Resp_Follow_Up messages. Any scheme that uses this information is acceptable, as long as the performance requirements of B.2.4 are met. As one example, the neighborRateRatio is computed as the ratio between a time interval measured by the local clock of the responder and its associated time interval measured by the local clock of the initiator, using a set of received Pdelay_Resp and

Pdelay_Resp_Follow_Up messages and a second set of received Pdelay_Resp and Pdelay_Resp_Follow_Up messages some number of Pdelay_Req message transmission intervals later, i.e., as show in Equation (16.1):

$$\frac{(t_{rsp}3)_N - (t_{rsp}3)_0}{(t_{req}4)_N - (t_{req}4)_0} \quad (16.1)$$

where $(t_{rsp}3)_k$ is the time relative to the local clock of the responder that the k^{th} Pdelay_Resp message is sent, $(t_{req}4)_k$ is the time relative to the local clock of the initiator that the k^{th} Pdelay_Resp message is received, N is the number of Pdelay_Req message transmission intervals separating the first set of received Pdelay_Resp and Pdelay_Resp_Follow_Up messages and the second set, and the successive sets of received Pdelay_Resp and Pdelay_Resp_Follow_Up messages are indexed from 0 to N with the first set indexed 0. The meanLinkDelay between the PTP Instance and the CSN node is computed as shown in Equation (16.2):

$$\frac{(t_{req}4 - t_{req}1)r - (t_{rsp}3 - t_{rsp}2)}{2} \quad (16.2)$$

where r is equal to neighborRateRatio, $t_{req}1$ is the time relative to the local clock of the initiator that the Pdelay_Req message for this message exchange is sent, $t_{rsp}2$ is the time relative to the local clock of the responder that the Pdelay_Req message for this message exchange is received, $t_{rsp}3$ is the time relative to the local clock of the responder that the Pdelay_Resp message for this message exchange is sent, and $t_{req}4$ is the time relative to the local clock of the initiator that the Pdelay_Resp message for this message exchange is received.

NOTE—The difference between mean propagation delay relative to the grandmaster time base and relative to the time base of the CSN node at the other end of the attached link (i.e., the responder CSN node) is usually negligible. To see this, note that the former can be obtained from the latter by multiplying the latter by the ratio of the grandmaster frequency to the frequency of the LocalClock entity of the CSN node at the other end of the link. This ratio differs from 1 by 200 ppm or less. For example, for a worst-case frequency offset of the LocalClock entity of the CSN node at the other end of the link, relative to the grandmaster, of 200 ppm, and a measured propagation time of 100 ns, the difference in mean propagation delay relative to the two time bases is 20 ps.

Although the propagation delay between two CSN nodes is constant, a Pdelay_Req message is still sent periodically by each CSN node to each other active CSN node of the network to measure the neighborRateRatio between the node and each other node. Each CSN node shall implement the state machines described in 11.2.19 and 11.2.20.

16.4.3.2 Native CSN path delay measurement

Some CSN technologies feature a native mechanism that provides a path delay measurement with accuracy similar to the accuracy the peer delay protocol provides. For these CSNs, the path delay can be provided using the native measurement method rather than using the Pdelay protocol defined in 11.2.19 and 11.2.20. Such a situation is described in more detail as follows. The CSN MD entity populates the following per-port and MD-entity global variables (described respectively in 10.2.5 and 11.2.13) as indicated:

— asCapable (10.2.5.1) is set to TRUE,
— neighborRateRatio (10.2.5.7) is set to the value provided by the native CSN measurement,
— meanLinkDelay (10.2.5.8) is set to the value provided by the native CSN measurement,
— computeNeighborRateRatio (10.2.5.10) is set to FALSE,
— computeMeanLinkDelay (10.2.5.11) is set to FALSE, and
— isMeasuringDelay (11.2.13.6) is set to TRUE to indicate that the CSN MD entity is measuring path delay (in this case, using its internal mechanism).
— domainNumber (8.1) is set to the domain number of this gPTP domain

16.4.3.3 Intrinsic CSN path delay measurement

If the CSN network features a native mechanism that causes each CSN node's local clock to be fully synchronized to the local clocks of other nodes of the CSN such that the synchronized CSN time complies with the requirements specified in B.1, the CSN nodes need not implement the path delay mechanism but rather treat the path delay as part of the residence time of the distributed system. The propagation of the Sync messages in this case is described in 16.5.2.

16.5 Synchronization messages

The CSN network shall propagate synchronized time over the CSN to CSN end stations and to downstream non-CSN links, using Sync (and, if the message is two-step, the associated Follow_Up) messages, as illustrated in Figure 16-6.

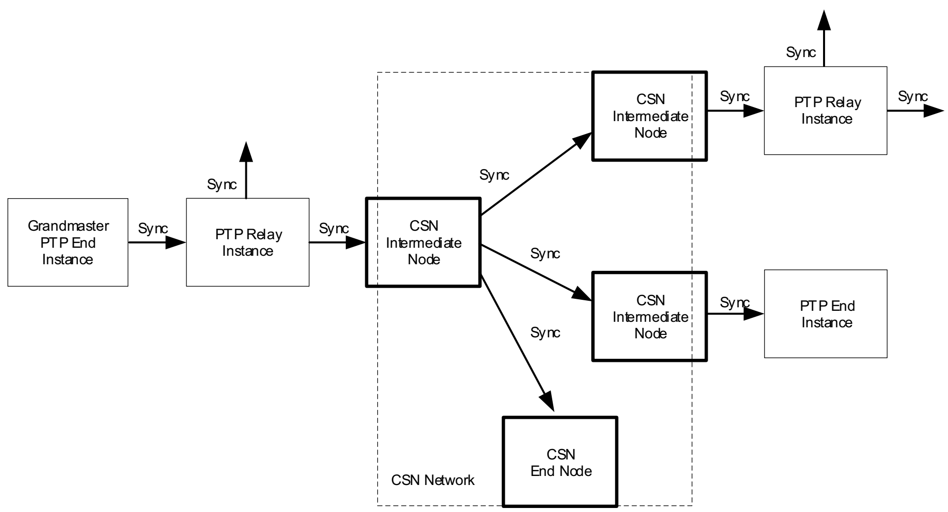


Figure 16-6—IEEE 802.1AS Sync Message Propagation over the CSN backbone

Once the path delays have been measured (a) between the upstream PTP Relay Instance or PTP End Instance and the ingress CSN node, (b) between the CSN nodes, and (c) between the egress CSN node and the downstream PTP Relay Instance or PTP End Instance, the CSN backbone can propagate the synchronization information received at its boundary nodes.

As with path delay measurements, various CSN technologies choose various methods for propagating time. These methods are described as follows.

16.5.1 Synchronization message propagation on CSN without network reference clock

If the CSN network does not feature a native mechanism that synchronizes the CSN node local clocks to each other or to a reference, such that the CSN synchronized time complies with the requirements specified in B.1, the CSN local clock at CSN ingress and CSN egress nodes are considered independent free running clocks.

In this case, synchronization over the CSN links uses the Sync and Follow_Up protocol (or only Sync if optional one-step processing is used), messages, and state machines specified for full-duplex point-to-point links in 10.2.8, 10.2.12, 11.2.14, and 11.2.15. Individual CSN link technologies may specify media-specific encapsulation of gPTP event messages. See Table 16-2 for selection of options per link technology.

One PortSync and one MD entity is instantiated per logical port (i.e., per CSN link). A CSN node behaves equivalently to a PTP Instance. Sync and, in the two-step case, Follow_Up messages are either transmitted using unicast on each link or broadcasted. However if Sync and Follow_Up messages are broadcasted:

- the Sync and Follow_Up messages are broadcast with the same port number used to broadcast Announce messages,
- all PortSync/MD entity pairs except one set their logSyncInterval attribute (see 10.7.2.3) to 127, causing them to not generate any Sync messages, and
- a dynamic selection of the MD entity that broadcasts the Sync message is needed (a CSN node can dynamically leave the CSN network). The dynamic selection algorithm is implementation specific and out of the scope of this standard.

16.5.2 Synchronization message propagation on a CSN with network reference clock

If the CSN network features a native mechanism that allows the CSN node's local clocks to be fully synchronized to each other in a way that complies with the requirements specified in B.1, it is possible to simplify the path delay mechanism, as described below. This method is an alternative to 16.5.1. Individual CSN link technologies may specify media-specific encapsulation of gPTP event messages. See Table 16-2 for selection of options per link technology.

Sync messages are timestamped (1) when received at the ingress CSN node's PTP Instance port (syncEventIngressTimestamp) and (2) when transmitted at the egress CSN node's PTP Instance port (syncEventEgressTimestamp). The elapsed time between the egress and ingress timestamps is computed as the CSN residence time. In this scheme, the Sync message handling is split between the MDSyncSendSM state machine in the CSN ingress node and the MDSyncReceiveSM state machine in the CSN egress node as described in 16.5.2.1 and 16.5.2.2.

The reference plane for a CSN port and the path delay measurement method are specific to the type of CSN technology, and are defined in Table 16-2.

16.5.2.1 CSN ingress node

The CSN ingress node timestamps Sync messages received from the upstream PTP Instance and compute the upstreamTxTime as described in 11.2.14.2.1, item f).

In addition, the setFollowUp function of the MDSyncSendSM state machine [see 11.2.15.2.3, item a)] ismodified as follows:

a) The quantity *rateRatio* × *(syncEventEgressTimestamp – upstreamTxTime)* is not added to the followUpCorrectionField of the Follow_Up message (or the Sync message in the one-step case) to be transmitted by the ingress to the egress CSN node
b) The CSN TLV (see 16.5.2.1.1) is appended to the Follow_Up message (or to the Sync message in the one-step case) transmitted by the ingress to the egress CSN node.

16.5.2.1.1 CSN TLV

16.5.2.1.1.1 General

The fields of the CSN TLV are specified in Table 16-1 and 16.5.2.1.1.3 through 16.5.2.1.1.7. This TLV is a standard organization extension TLV for the Follow_Up message (or the Sync message in the one-step case), as specified in 14.2 of IEEE Std 1588-2019. This TLV is not allowed to occur before the Follow_Up information TLV (see 11.4.4.3).

Table 16-1—CSN TLV

Bits ~~8~~7 ~~7~~6 ~~6~~5 ~~5~~4 43 ~~3~~2 ~~2~~1 ~~1~~0	Octets	Offset From Start of TLV
tlvType	2	0
lengthField	2	2
organizationId	3	4
organizationSubType	3	7
rxTime	12	10
neighborRateRatio	4	14
meanLinkDelay	12	26
delayAsymmetry	12	38
domainNumber	1	50

16.5.2.1.1.2 tlvType (Enumeration16)

The value of the tlvType field is 0x3.

NOTE—This is the value that indicates the TLV is a vendor and standard organization extension TLV, as specified in 14.2.2.1 and Table 50 of IEEE Std 1588-2019. The value is specified there as ORGANIZATION_EXTENSION, whose value is 0x3.

16.5.2.1.1.3 lengthField (UInteger16)

The value of the length is 46.

16.5.2.1.1.4 organizationId (Octet3)

The value of organizationId is 00-80-C2.

16.5.2.1.1.5 organizationSubType (Enumeration24)

The value of organizationSubType is 3.

16.5.2.1.1.6 upstreamTxTime (UScaledNs)

The computed upstreamTxTime value as described in 11.2.14.2.1 item f).

16.5.2.1.1.7 neighborRateRatio (Integer32)

The neighborRateRatio value described in 10.2.5.7.

16.5.2.1.1.8 meanLinkDelay (UScaledNs)

The meanLinkDelay value described in 10.2.5.8.

16.5.2.1.1.9 delayAsymmetry (ScaledNs)

The delayAsymmetry value described in 10.2.5.9.

16.5.2.1.1.10 domainNumber(UInteger8)

The domain number of this gPTP domain.

16.5.2.2 CSN egress node

The CSN egress port sets neighborRateRatio to 1 and meanLinkDelay to 0 for its CSN port.

The CSN egress port modifies the function setMDSyncReceive of the MDSyncReceiveSM state machine [11.2.14.2.1 item f)] for the port attached to the CSN link entity to extract upstreamTxTime, meanLinkDelay, and neighborRateRatio from the respective fields of the CSN TLV in the Follow_Up message (or Sync message in the one-step case) received from the CSN ingress node.

The CSN egress port also modifies the ClockSlaveSync state machine (see 10.2.13) to get the upstreamTxTime, meanLinkDelay, neighborRateRatio, and delayAsymmetry values from the respective fields of the CSN TLV in the Follow_Up message (or Sync message in the one-step case) received from the CSN ingress node.

The CSN egress node removes the CSN TLV from the Follow_Up message (or Sync message in the one-step case) received from the CSN ingress node and transmits the messageto the downstream PTP Instance.

16.6 Specific CSN requirements

The reference plane for a CSN port is specific to the type of CSN technology and is defined in Table 16-2.

16.6.1 MoCA-specific behavior

The MoCA network is a CSN. The non-MoCA port of a CSN time-aware node behaves as a PTP Instance port, which might or might not use the MoCA CTC clock for timestamping event messages. If the non-MoCA port of the CSN time-aware node uses a different clock than the MoCA CTC clock, then the CSN time-aware node reconciles the non-CTC timestamp with the CTC time.

Table 16-2—Definitions and option selections per link technology

CSN link technology	Reference plane	Path Delay measurement	gPTP event message encapsulation
Multimedia over Coax Alliance (MoCA) v2.0	The first bit of an Event message crossing to and from the MoCA CTC clock domain.	MoCA Ranging Protocol (method of either 16.4.3.2 or 16.4.3.3)	Encapsulated in control frames as described in the MoCA MAC/PHY Specification v2.0.
ITU-T G.hn (SG15)	The first bit of an Event message crossing the A-interface(see 16.6.2).	peer-to-peer mechanism as defined in 16.4.3.1.	None

Sync messages shall be timestamped using the CTC clock (1) when the Sync message crosses the MoCA ingress node's timestamp reference plane (syncEventIngressTimestamp) and (2) when it crosses the egress CSN node's reference plane (syncEventEgressTimestamp).

The elapsed time between the egress and ingress timestamps, syncEventIngressTimestamp – syncEventIngressTimestamp, is computed as the MoCA residence time.

The MoCA port whose port state is MasterPort propagates the Sync and Follow_Up messages (or only the Sync message in the one-step case) as described in 16.5.2. The CSN TLV values of the Follow_Up message sent over the MoCA network are computed using the LocalClock, i.e., the MoCA CTC clock.

IEEE 802.1 AS Frames shall be transmitted over the MoCA network as MoCA control frames as described in the MoCA MAC/PHY Specification v2.0.

NOTE—The Channel-Time Clock (CTC) is specified in the MoCA MAC/PHY Specification v2.0.

16.6.2 ITU-T G.hn-specific behavior

A port of a PTP Instance that includes one or more ITU-T G.hn ports shall behave as defined in 16.3, 16.4, and 16.5, but for aspects where more than one behavior or option is described, the system behaves as defined in Table 16-2.

ITU-T G.hn defines a 32-bit timestamp (which is placed in the TSMP field of a G.hn encapsulation header). This timestamp, as described in Table 16-2, is captured each time an Event message is transmitted or received by an ITU-T G.hn port and is used for all gPTP event messages, including SYNC and PDELAY messages.

16.7 Grandmaster capability

Each CSN Node may be grandmaster capable, allowing a CSN node to act as the grandmaster clock either for a homogeneous CSN network or for a heterogeneous network.

The Announce messages are either sent via unicast on each link or broadcasted. However if Announce messages are broadcasted, the Announce message shall use the same port number as used by the Sync and Follow_Up messages (or only the Sync message in the one-step case) by a single PortSync/MD entity pair on the port. This is accomplished by setting the logAnnounceInterval attribute (see 10.7.2.2) to 127 for all but one PortSync/MD entity pair, causing them to not generate any Announce messages.

16.8 CSN clock and node performance requirements

The CSN clock performance complies with the requirements specified in B.1. The CSN node performance should follow the recommendations of B.2.2 and B.2.3. The CSN node performance complies with the requirements specified in B.2.4.

Annex A

(normative)

Protocol Implementation Conformance Statement (PICS) proforma[15]

A.1 Introduction

The supplier of a protocol implementation that is claimed to conform to this standard shall complete the following PICS proforma.

A completed PICS proforma is the PICS for the implementation in question. The PICS is a statement of which capabilities and options of the protocol have been implemented. The PICS can have a number of uses, including the following:

a) By the protocol implementer, as a checklist to reduce the risk of failure to conform to the standard through oversight;
b) By the 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 PICS proforma;
c) By the user—or potential user—of the implementation, as a basis for initially checking the possibility of interworking with another implementation (note that, while interworking can never be guaranteed, failure to interwork can often be predicted from incompatible PICS);
d) By a protocol tester, as the basis for selecting appropriate tests against which to assess the claim for conformance of the implementation.

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

A.2.2 General abbreviations

N/A	not applicable
PICS	Protocol Implementation Conformance Statement

[15]*Copyright release for PICS proformas:* Users of this standard may freely reproduce the PICS proforma in this annex so that it can be used for its intended purpose and may further publish the completed PICS.

A.3 Instructions for completing the PICS proforma

A.3.1 General structure of the PICS proforma

The first part of the PICS 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 PICS 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. (Note that 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 standard, and the fifth column provides the space for the answers.

A supplier may also provide (or be required to provide) further information, categorized as either Additional Information or Exception Information. 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 (e.g., simply a numeral). There are no other restrictions on its format and presentation.

A completed PICS 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 PICS may be able to describe all such configurations. However, the supplier has the choice of providing more than one PICS, 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 PICS. It is not intended or expected that a large quantity will be supplied, and a PICS 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 standard but that have a bearing upon the answers to some items.

References to items of Additional Information may be entered next to any answer in the questionnaire and may be included in items of Exception Information.

A.3.3 Exception Information

It may 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; 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 standard.

NOTE—A possible reason for the situation described above is that a defect in this standard 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 PICS 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 upon whether or not 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 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 PICS proforma; the value of the predicate is true if the item is marked as supported, and is false otherwise;
b) 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;
c) 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.4 PICS proforma for IEEE Std 802.1AS-2019

NOTE 1—Only the first three items are required for all implementations; other information may be completed as appropriate in meeting the requirement for full identification.

NOTE 2—The terms Name and Version should be interpreted appropriately to correspond with a supplier's terminology (e.g., Type, Series, Model).

A.4.1 Implementation identification

Supplier	
Contact point for queries about the PICS	
Implementation Name(s) and Version(s)	
Other information necessary for full identification—e.g., name(s) and version(s) of machines and/or operating system names	

A.4.2 Protocol summary, IEEE Std 802.1AS

Identification of protocol specification	IEEE Std 802.1AS-2019, IEEE Standard for Local and metropolitan area networks—Timing and Synchronization for Time-Sensitive Applications
Identification of amendments and corrigenda to the PICS proforma which have been completed as part of the PICS	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 IEEE Std 802.1AS-2019)	No [ ] Yes [ ]

Date of Statement	

A.5 Major capabilities

Item	Feature	Status	References	Support
DOM0	Does the device support a PTP Instance with domain number 0, in accordance with the requirements of 8.1?	M	5.4(a), 8.1	Yes [ ]
DOMADD	Does the device support one or more PTP Instances with domain number in the range 1-127?	O	5.4.1(f), 8.1	Yes [ ] No [ ]
MINTA	Does the device support at least one port with minimal requirements?	M	10.2.13, 5.4(c), A.7	Yes [ ]
BMC	Does the device implement the best master clock algorithm?	M	10.2.13, 5.4(f), 10.3, A.9	Yes [ ]
SIG	Does the device transmit Signaling messages?	O	10.2.13, 5.4.1(e), 10.6.4, A.8	Yes [ ] No [ ]
GMCAP	Is the device capable of acting as a grandmaster?	O	10.2.13, 5.4.1(c), 10.1.2, A.10	Yes [ ] No [ ]
BRDG	Does the device act as a PTP Relay Instance on two or more ports?	O	5.4.1(d), 5.4.2	Yes [ ] No [ ]
MIMSTR	Does the device support media-independent master functionality on at least one port?	GMCAP or BRDG:M	5.4.1(b), A.11	Yes [ ] N/A [ ]
MIPERF	Does the device support the performance requirements?	M	10.2.13, 5.4(j), A.12	Yes [ ]
EXT	Does the device support external port configuration?	O	5.4.1(g), A.21	Yes [ ] No [ ]
MDFDPP	Does the device support media-dependent full-duplex, point-to-point functionality on one or more ports?	O.1	5.5, 11, A.6, A.13	Yes [ ] No [ ]
MDDOT11	Does the device support media-dependent IEEE 802.11 link functionality on one or more ports?	O.1	5.6, 12, A.6, A.14	Yes [ ] No [ ]
MDEPON	Does the device support IEEE 802.3 Passive Optical Networking (EPON)	O.1	5.7, 13, A.6, A.15	Yes [ ] No [ ]
MDGHN	Does the device support media-dependent G.hn functionality on one or more ports?	O.1	5.8(b), 16.6.2, A.18	Yes [ ] No [ ]
MDMOCA	Does the device support media-dependent MoCA functionality on one or more ports?	O.1	5.8(b), 16.6.1, A.17	Yes [ ] No [ ]
MDCSN	Does the device support media-dependent CSN functionality on one or more ports?	MDGHN or MDMOCA:M	5.8, 16, A.6, A.16	Yes [ ] No [ ]
MGT	Is management of the time-aware system supported?	O	5.4.1(j), 14	Yes [ ] No [ ]
RMGT	Is a remote management protocol supported?	MGT: O	5.4.1(k), A.19	Yes [ ] No [ ]
APPL	Does the device support one or more of the application interfaces?	O	5.4.1(i), 9, A.20	Yes [ ] No [ ]

A.6 Media access control methods

Item	Feature	Status	References	Support
MAC-IEEE-802.3 MAC-IEEE-802.11	Which MAC methods are implemented in conformance with the relevant MAC standards	O:2 O:2	11.1 12.1	Yes [ ] No [ ] Yes [ ] No [ ]
MAC-1	Has a PICS been completed for each of the MAC methods implemented as required by the relevant MAC Standards?	M		Yes [ ]
MAC-2	Do all the MAC methods implemented support the MAC Timing aware Service as specified?	M	11 12 13	Yes [ ]

A.7 Minimal PTP Instance

Item	Feature	Status	References	Support
MINTA-1	Does the device implement the functionality specified by the SiteSyncSync state machine in Figure 10-3 in compliance with the requirements of 10.2.7?	M	5.4(g), 10.2.7	Yes [ ]
MINTA-2	Does the device implement the functionality specified by the PortSyncSyncReceive state machine in Figure 10-4 on each port in compliance with the requirements of 10.2.8?	M	5.4(d)	Yes [ ]
MINTA-3	Does the device implement the functionality specified by the ClockSlaveSync state machine in Figure 10-9 in compliance with the requirements of 10.2.13?	M	10.2.13, 5.4(e)	Yes [ ]
MINTA-4	Does the device port sending a Signaling message that contains a message interval request TLV adjust its syncReceiptTimeoutTimeInterval in compliance with the requirements of 10.6.4.3.7 and Table 10-16?	SIG:M	10.6.4.3.7	Yes [ ] N/A [ ]
MINTA-5	Is the clockIdentity constructed in compliance with the requirements of 8.5.2.2?	M	8.5.2.2	Yes [ ]
MINTA-6	Is the domain number for all transmitted messages set to 0 in compliance with the requirements of 8.1?	M	8.1	Yes [ ]
MINTA-7	Is the IEEE 802.1AS time measured relative to the PTP epoch in compliance with the requirements of 8.2.2?	M	8.2.2	Yes [ ]
MINTA-8	If path delay asymmetry is modeled by this device does it comply with the requirements of 8.3?	O	8.3	Yes [ ] No [ ]
MINTA-9	Do all derived data types that are transmitted in IEEE 802.1AS messages and headers comply with 6.4.4?	M	6.4.4	Yes [ ]

Item	Feature	Status	References	Support
MINTA-10	Is the granularity of the local clock 40 ns or better in compliance with the requirements of B.1.2?	M	B.1.2	Yes [ ]
MINTA-11	Is the frequency of the local clock relative to TAI ±100 ppm in compliance with the requirements of B.1.1?	M	B.1.1	Yes [ ]
MINTA-12	Does the PTP Instance ignore TLVs, of Announce and Signaling messages, that it cannot parse and attempt to parse the next TLV, in compliance with the requirements of 10.6.1?	M	10.6.1	Yes [ ]
MINTA-13	Does the PTP Instance support the state machines related to signaling gPTP protocol capability?	M	5.4(h), 10.4	Yes [ ]
MINTA-14	For receive of all messages, and for transmit of all messages except Announce and Signaling, does the PTP Instance support the message requirements?	M	5.4(i), 10.5, 10.6, 10.7	Yes [ ]
MINTA-15	Does the PTP Instance support the gPTP requirements specified in clause 8, including the PTP Instance attributes?	M	5.4(a), 8, 8.6.2	Yes [ ]
MINTA-16	Does the PTP Instance support the requirements for time-synchronization state machines?	M	5.4(b)	Yes [ ]
MINTA-17	Does the PTP Instance implement the path trace TLV (i.e., process this TLV when received in an Announce message, and attach this TLV to a transmitted Announce message unless the TLV would cause the maximum frame size to be exceeded)?	M	10.3.11, 10.3.13, 10.3.14, 10.3.16	Yes [ ]
MINTA-18	Does the PTP Instance forward TLVs as required?	M	10.6.1	Yes [ ]

A.8 Signaling

Item	Feature	Status	References	Support
SIG-1	Do the sequence numbers of Signaling messages comply with the requirements of 10.5.7?	SIG:M	10.5.7	Yes [ ]
SIG-2	Does the Signaling message body comply with the requirements of 10.6.4.1 and Table 10-13?	SIG:M	10.6.4.1	Yes [ ]
SIG-3	Does the Signaling message header comply with the requirements of Table 10-7 and 10.6.2.1, including all of its subclauses (10.6.2.2.1–10.6.2.2.14)?	SIG:M	10.6.2.1	Yes [ ]
SIG-4	Are all Signaling message reserved fields equal to 0 in compliance with the requirements of 10.6.1?	SIG:M	10.6.1	Yes [ ]

Item	Feature	Status	References	Support
SIG-5	Is the destination MAC address for all Signaling messages equal to 01:80:C2:00:00:0E in compliance with the requirements of 10.5.3?	SIG:M	10.5.3	Yes [ ]
SIG-6	Is the EtherType for all Signaling messages equal to 88-F7 in compliance with the requirements of 10.5.4?	SIG:M	10.5.4	Yes [ ]
SIG-7	Does the message interval request TLV for signaling messages comply with the requirements in 10.6.4.3.2 through 10.6.4.3.9 and Table 10-14?	SIG:M	10.6.4.3.2	Yes [ ]

A.9 Best master clock

Item	Feature	Status	References	Support
BMC-1	Does the device implement the functionality specified by the PortAnnounceReceive state machine in Figure 10-13 on each port in compliance with the requirements of 10.3.11?	M	10.3.11	Yes [ ]
BMC-2	Does the device implement the functionality specified by the PortAnnounceInformation state machine in Figure 10-14 on each port in compliance with the requirements of 10.3.12?	M	10.3.12	Yes [ ]
BMC-3	Does the device implement the functionality specified by the PortStateSelection state machine in Figure 10-15 on each port in compliance with the requirements of 10.3.13? NOTE—There is one instance of the PortStateSelection state machine for the PTP Instance, for each gPTP domain. Some of the PortStateSelection state machine computations are performed for each port, and some of the computations are performed for the PTP Instance as a whole (and all the computations are performed for each gPTP domain).	M	10.3.13	Yes [ ]
BMC-4	If the value of clockA's SystemIdentity is less than that of clockB, is clockA selected as Grandmaster in compliance with the requirements of 10.3.2?	M	10.3.2	Yes [ ]
BMC-5	Does the value of priority1 comply with the requirements of 8.6.2.1?	M	8.6.2.1	Yes [ ]
BMC-6	Does the value of clockClass comply with the requirements of 8.6.2.2?	M	8.6.2.2	Yes [ ]
BMC-7	Does the value of priority2 comply with the requirements of 8.6.2.5?	M	8.6.2.5	Yes [ ]
BMC-8	Does the value of clockAccuracy comply with requirements of 8.6.2.3?	M	8.6.2.3	Yes [ ]
BMC-9	Does the value of offsetScaledVariance comply with the requirements of 8.6.2.4?	M	8.6.2.4	Yes [ ]
BMC-10	Does the value of timeSource comply with requirements of 8.6.2.7 and Table 8-2?	M	8.6.2.7	Yes [ ]
BMC-11	Is the port number equal to 1 in compliance with the requirements of 8.5.2.3?	~BRDG:M	8.5.2.3	Yes [ ] N/A [ ]
BMC-12	Are the ports numbered 1 through N for each of N ports in compliance with the requirements of 8.5.2.3?	M	8.5.2.3	Yes [ ]
BMC-13	Does the clockIdentity field comply with the requirements of 8.5.2.2?	M	8.5.2.2	Yes [ ]
BMC-14	When no grandmaster capable device is available does the behavior of the device comply with the requirements of 10.2.13.2, i.e., the clockSlaveTime should be provided by the local clock?	M	10.2.13.2	Yes [ ]

Item	Feature	Status	References	Support
BMC-15	Does the value of announceReceiptTimeout comply with the requirements of 10.7.3.2?	M	10.7.3.2	Yes [ ]
BMC-16	Does the SlavePort remove the port from the BMC selection after announceReceiptTimeout expires in compliance with the requirements of 10.7.3.2?	M	10.7.3.2	Yes [ ]
BMC-17	Does the value of syncReceiptTimeout comply with the requirements of 10.7.3.1?	M	10.7.3.1	Yes [ ]
BMC-18	Does the SlavePort remove the port from the BMC selection after syncReceiptTimeout expires in compliance with 10.7.3.1?	M	10.7.3.1	Yes [ ]
BMC-19	Does the device port sending a message interval request signaling message adjust its announceReceiptTimeoutTimeInterval in compliance with the requirements of 10.6.4.3.8 and Table 10-17?	SIG:M	10.6.4.3.8	Yes [ ]
BMC-20	If the device implements the ClockSourceTime interface, does the value of lastGmPhaseChange comply with the requirements of 9.2.2 and 6.4.3.3?	O	9.2.2	Yes [ ]
BMC-21	Does the transmitted timing information comply with the requirements of 10.3.1, including specifications for externalPortConfigurationEnabled value of false?	GMCAP:M	10.3.1	Yes [ ]
BMC-22	Does the device implement BMCA requirements that are not listed in the preceding BMC rows?	M	10.3.2, 10.3.3 10.3.4, 10.3.5, 10.3.6, 10.3.8, 10.3.10	Yes [ ]

A.10 Grandmaster-capable system

Item	Feature	Status	References	Support
GMCAP-1	Does the device implement the functionality specified by the ClockMasterSyncSend state machine in compliance with the requirements of 10.2.9 and Figure 10-5?	GMCAP:M	10.2.9	Yes [ ]
GMCAP-2	Does the device implement the functionality specified by the ClockMasterSyncOffset state machine in compliance with the requirements of 10.2.10 and Figure 10-6?	GMCAP:M	10.2.10	Yes [ ]
GMCAP-3	Does the device implement the functionality specified by the ClockMasterSyncReceive state machine in compliance with the requirements of 10.2.11 and Figure 10-7?	GMCAP:M	10.2.11	Yes [ ]

A.11 Media-independent master

Item	Feature	Status	References	Support
MIMSTR-1	Does the device implement the functionality of the AnnounceIntervalSetting state machine in compliance with the requirements of 10.3.17 and Figure 10-19 on each port?	MIMSTR:M	10.3.17	Yes [ ]
MIMSTR-2	Does the device implement the functionality of the PortSyncSyncSend state machine in compliance with the requirements of 10.2.9 and Figure 10-8 on each port?	MIMSTR:M	10.2.9	Yes [ ]
MIMSTR-3	Does the device implement the functionality of the PortAnnounceTransmit state machine in compliance with the requirements of 10.3.16 and Figure 10-18 on each port?	MIMSTR:M	10.3.16	Yes [ ]
MIMSTR-4	Does the destination MAC address of all Announce messages equal 01:80:C2:00:00:0E?	MIMSTR:M	10.5.3	Yes [ ]
MIMSTR-5	Does the EtherType of all Announce messages equal 88-F7?	MIMSTR:M	10.5.4	Yes [ ]
MIMSTR-6	Do the sequence numbers of Announce messages comply with the requirements of 10.5.7?	MIMSTR:M	10.5.7	Yes [ ]
MIMSTR-7	Does the Announce message header comply with Table 10-7 and 10.6.2.2, including all of its subclauses (10.6.2.2.1–10.6.2.2.14)?	MIMSTR:M	10.6.2.1	Yes [ ]
MIMSTR-8	Does the Announce message body comply with the requirements in 10.6.3.1 and Table 10-11?	MIMSTR:M	10.6.3.1	Yes [ ]
MIMSTR-9	Are all Announce message reserved fields equal to 0?	MIMSTR:M	10.6.1	Yes [ ]

Item	Feature	Status	References	Support
MIMSTR-10	If it is not otherwise specified is the logAnnounceInterval equal to zero or within the allowed range?	MIMSTR:M	10.7.2.1	Yes [ ]
MIMSTR-11	Does the value of currentUtcOffset comply with the requirements of 8.2.3?	MIMSTR:M	8.2.3	Yes [ ]
MIMSTR-12	Do the values of the leap59, leap61, and currentUtcOffsetValid flags comply with the requirements of 10.3.8?	MIMSTR:M	10.3.8	Yes [ ]
MIMSTR-13	Does this device ensure that messages that traverse it or originate from it are not transmitted with VLAN tags in complaince with the requirements of 11.3.3?	MIMSTR:M	11.3.3	Yes [ ]
MIMSTR-14	Is the computation of cumulative rateRatio in accordance with 10.2.8.3?	MIMSTR:M	10.2.8.3	Yes [ ] N/A [ ]
MIMSTR-15	For transmit of the Announce message, does the device support the message requirements?	MIMSTR:M	10.5, 10.6, 10.7	Yes [ ]

A.12 Media-independent performance requirements

Item	Feature	Status	References	Support
MIPERF-1	Does the device comply with the performance requirements of B.1?	M	B.1	Yes [ ]
MIPERF-2	Does the device comply with the performance requirements of B.2.4?	M	B.2.4	Yes [ ]
MIPERF-3	Does the device comply with the performance requirements of B.2.2?	O	B.2.2	Yes [ ] No [ ]
MIPERF-4	Does the device comply with the performance requirements of B.2.3?	O	B.2.3	Yes [ ] No [ ]

A.13 Media-dependent, full-duplex, point-to-point link

Item	Feature	Status	References	Support
MDFDPP-1	Does this port implement the functionality of the MDSyncReceiveSM state machine in compliance with the requirements of 11.2.14 and Figure 11-6?	MDFDPP:M	11.2.14	Yes [ ]
MDFDPP-2	Does this port implement the functionality of the MDSyncSendSM state machine in compliance with the requirements of 11.2.15 and Figure 11-7?	MIMSTR and MDFDPP:M	11.2.15	Yes [ ]
MDFDPP-3	Does this port implement the functionality of the MDPdelayRequest state machine in compliance with the requirements of 11.2.19 and Figure 11-9?	MDFDPP:M	11.2.19	Yes [ ]
MDFDPP-4	Does this port implement the functionality of the MDPdelayResponse state machine in compliance with the requirements of 11.2.20 and Figure 11-10?	MDFDPP:M	11.2.20	Yes [ ]
MDFDPP-5	Does this port implement the functionality of the SyncIntervalSetting state machine in compliance with the requirements of 10.3.18 and Figure 10-20?	MDFDPP:M	10.3.18 5.5(c) 10.3.18	Yes [ ]
MDFDPP-6	Does this port implement the functionality of the LinkDelayIntervalSetting state machine in compliance with the requirements of 11.2.21 and Figure 11-11?	MDFDPP:M	11.2.21	Yes [ ]
MDFDPP-7	Does this port timestamp Sync messages on ingress with respect to the LocalClock in compliance with 11.3.2.1 and 11.3.9?	MDFDPP:M	11.3.2.1	Yes [ ]
MDFDPP-8	Does this port timestamp Sync messages on egress with respect to the LocalClock in compliance with the requirements of 11.3.2.1 and 11.3.9?	MIMSTR and MDFDPP:M	11.3.2.1	Yes [ ]
MDFDPP-9	Does this port timestamp Pdelay_Req messages on ingress and egress with respect to the LocalClock in compliance with the requirements of 11.3.2.1 and 11.3.9?	MDFDPP:M	11.3.2.1	Yes [ ]
MDFDPP-10	Does this port timestamp Pdelay_Resp messages on ingress and egress with respect to the LocalClock in compliance with the requirements of 11.3.2.1 and 11.3.9?	MDFDPP:M	11.3.2.1	Yes [ ]
MDFDPP-11	Are all IEEE 802.1AS messages on this port sent without a Q-tag in compliance with the requirements of 11.3.3?	MDFDPP:M	11.3.3	Yes [ ]
MDFDPP-12	Do all media-dependent messages transmitted on this port use a destination MAC address taken from Table 11-3 in compliance with the requirements of 11.3.4 [01-80-C2-00-00-0E]?	MDFDPP:M	11.3.4	Yes [ ]
MDFDPP-13	Do all media-dependent messages transmitted on this port use a source MAC address that is assigned to that port in compliance with the requirements of 11.3.4?	MDFDPP:M	11.3.4	Yes [ ]

Item	Feature	Status	References	Support
MDFDPP-14	Do all media-dependent message tranmitted on this port us an EtherType specified in Table 11-4 [88-F7]?	MDFDPP:M	11.3.5	Yes [ ]
MDFDPP-15	Does the header of all the media-dependent messages on this port comply with the requirements of the subclauses of 11.4.2 and Table 10-7?	MDFDPP:M	11.4.2	Yes [ ] N/A [ ]
MDFDPP-16	Does the body of Sync messages sent on this port comply with the requirements of 11.4.3, Table 11-8, and Table 11-9?	MDFDPP:M	11.4.3	Yes [ ]
MDFDPP-17	Does the body of Follow_Up messages sent on this port comply with the requirements of 11.4.4, 6.4.3.3 (lastGmPhaseChange), and Table 11-10?	MDFDPP:M	11.4.4, 6.4.3.3	Yes [ ]
MDFDPP-18	Does the body of Pdelay_Req messages sent on this port comply with the requirements of 11.4.5 and Table 11-12?	MDFDPP:M	11.4.5	Yes [ ]
MDFDPP-19	Does the body of Pdelay_Resp messages sent on this port comply with the requirements of 11.4.6 and Table 11-13?	MDFDPP:M	11.4.6	Yes [ ]
MDFDPP-20	Does the body of Pdelay_Resp_Follow_Up messages sent on this port comply with the requirements of 11.4.7 and Table 11-14?	MDFDPP:M	11.4.7	Yes [ ]
MDFDPP-21	Are all reserved fields in media-dependent messages sent on this port set to 0 in compliance with the requirements of 11.4.1?	MDFDPP:M	11.4.1	Yes [ ]
MDFDPP-22	Do the Sync message sequence numbers comply with the requirements of 11.3.8?	MIMSTR and MDFDPP:M	11.3.8	Yes [ ] N/A [ ]
MDFDPP-23	Do the Pdelay_Req message sequence numbers comply with the requirements of 11.3.8?	MDFDPP:M	11.3.8	Yes [ ]
MDFDPP-24	Does the Pdelay mean request transmission interval comply with the requirements of 11.5.2.2?	MDFDPP:M	11.5.2.2	Yes [ ]
MDFDPP-25	Does the Sync mean transmission interval comply with the requirements of 11.5.2.3?	MDFDPP:M	11.5.2.3	Yes [ ]
MDFDPP-26	Does the full-duplex, point-to-point media-dependent layer set the asCapable global variable in the media-independent PortSync entity in compliance with the requirements of 11.2.2?	MDFDPP:M	11.2.2	Yes [ ]
MDFDPP-27	Does the device's use of flow control comply with the requirements of 11.2.3 and 11.2.4?	MDFDPP:M	11.2.3, 11.2.4	Yes [ ]
MDFDPP-28	Does the device consider the port to not be exchanging Pdelay messages when a valid response is not received in compliance with the requirements of 11.5.3?	MDFDPP:M	11.5.3	Yes [ ]
MDFDPP-29	Does the PTP Instance ignore TLVs, of PTP messages, that it cannot parse and attempt to parse the next TLV, in compliance with the requirements of 11.4.1?	MDFDPP:M	11.4.1	Yes [ ]
MDFDPP-30	Does the time-aware system initialize neighborPropDelayThresh as specified in 11.2.2?	MDFDPP:M	11.2.2	Yes [ ]

Item	Feature	Status	References	Support
MDFDPP-31	Does this port of the time-aware system support asymmetry measurement mode (see Annex G for informative description)?	MDFDPP:O	14.13, 14.18, 10.2.5, 10.2.8, 10.3.12, 10.3.13, 10.3.15, 10.3.16, 11.2.14, 11.2.15, 11.2.20	Yes [ ] No [ ]
MDFDPP-32	Does this port support one-step receive?	MDFDPP:O	11.2.14	Yes [ ] No [ ]
MDFDPP-33	Does this port support one-step transmit?	MDFDPP:O	11.2.15	Yes [ ] No [ ]
MDFDPP-34	Does this port implement the functionality of the OneStepTxOperSetting state machine in compliance with the requirements of 11.2.21, 11.3, 11.2.16, and Figure 11-8?	MDFDPP:O	11.3, 11.2.21	Yes [ ] No [ ]
MDFDPP-35	Does this port support propagation delay averaging?	MDFDPP:O	11.2.19.3.4	Yes [ ] No [ ]

A.14 Media-dependent IEEE 802.11 link

Item	Feature	Status	References	Support
MDDOT11-1	Does the IEEE 802.11 MAC implement the master port functionality in compliance with the requirements of 12.5.1?	MDDOT11 and MIMSTR:M	5.6(d), 12.5.1	Yes [ ]
MDDOT11-2	Does the IEEE 802.11 MAC implement the slave port functionality in compliance with the requirements of 12.5.2?	MDDOT11:M	5.6(a), 5.6(b), 5.6(d), 12.5.2	Yes [ ]
MDDOT11-3	Does the IEEE 802.11 MAC determine the value of asCapable in compliance with the requirements of 12.4?	MDDOT11:M	12.4	Yes [ ]
MDDOT11-4	Does the IEEE 802.11 MAC determine the value of mean time interval between synchronization messages in compliance with the requirements of 12.8?	MDDOT11 and MIMSTR:M	12.8	Yes [ ]
MDDOT11-5	Does the 802.11 MAC support the use of the VendorSpecific information element of 12.7 to carry end-to-end link-independent timing information?	MDDOT11:M	12.7	Yes [ ]
MDDOT11-6	Does the 802.11 MAC implement Fine Timing Measurement as a master port	MDDOT11-1:O	5.6(c), 5.6(e), 12.5.1	Yes [ ] No [ ]
MDDOT11-7	Does the 802.11 MAC implement Fine Timing Measurement as a slave port	MDDOT11-2:O	5.6(c), 5.6(e), 12.5.2	Yes [ ] No [ ]

A.15 Media-dependent IEEE 802.3 EPON link

Item	Feature	Status	References	Support
MDEPON-1	Does the TIMESYNC message format comply with the requirements of 13.3 and Table 13-1?	MDEPON:M	13.3	Yes [ ]
MDEPON-2	Does the device implement the functionality specified by the requester state machine in compliance with the requirements of 13.7.1 and Figure 13-3?	MDEPON and MIMSTR:M	13.7.1.4	Yes [ ]
MDEPON-3	Does the device implement the functionality specified by the responder state machine in compliance with the requirements of and Figure 13-4?	MDEPON:M	13.7.2.4	Yes [ ]
MDEPON-4	Does the TIMESYNC message transmission interval comply with the requirements of 13.8.1 and 13.8.2?	MDEPON:M	13.8.1, 13.8.2	Yes [ ]
MDEPON-5	Does the implementation of best master selection comply with the requirements of 13.1.3?	MDEPON:M	13.1.3	Yes [ ]
MDEPON-6	Does the determination of the value of asCapable comply with the requirements of 13.4?	MDEPON:M	13.4	Yes [ ]

A.16 Media-dependent CSN link

Item	Feature	Status	References	Support
MDCSN-1	Does the device implement the functionality of the MDSyncSendSM state machine in compliance with 11.2.15?	MDCSN and MIMSTR:M	11.2.15	Yes [ ]
MDCSN-2	Does the device implement the functionality of the MDSyncReceiveSM state machine in compliance with 11.2.14?	MDCSN:M	11.2.14	Yes [ ]
MDCSN-3	Does the device calculate path delay in compliance with the requirement of 16.4 and its subclauses?	MDCSN:M	16.4.1, 16.4.2, 16.4.3	Yes [ ]
MDCSN-4	Does the device propagate synchronized time in compliance with the requirements of 16.5 and its subclauses?	MDCSN:M	16.5.1, 16.5.2	Yes [ ]
MDCSN-5	Does the device act as grandmaster in compliance with the requirements of 16.7 and its subclauses?	GMCAP and MDCSN:M	16.7	Yes [ ]
MDCSN-6	Does the device comply with the performance requirements of 16.8?	GMCAP and MDCSN:M	16.8	Yes [ ]

A.17 Media-dependent MoCA link

Item	Feature	Status	References	Support
MDMOCA-1	Does the MoCA MD entity propagate Sync messages in compliance with the requirements of 16.6.1?	MDMOCA:M	16.6.1	Yes [ ]

A.18 Media-dependent ITU-T G.hn link

Item	Feature	Status	References	Support
MDGHN-1	Does the GHN MD entity propagate Sync messages in compliance with the requirements of 16.6.2?	MDGHN:M	16.6.2	Yes [ ]

A.19 Remote management

Item	Feature	Status	References	Support
	If item RMGT is not supported, mark N/A.			N/A[ ]
RMGT-1	What management protocol standard(s) or specification(s) are supported?	RMGT:M	5.4.1(k)(1)	

A.19 Remote management *(continued)*

Item	Feature	Status	References	Support
RMGT-2	What standard(s) or specification(s) for managed object definitions and encodings are supported for use by the remote management protocol?	RMGT:M	5.4.1(k)(2)	
RMGT-3	If the Simple Network Management Protocol (SNMP) is listed in RMGT-2, is the IEEE 8021-AS-MIB module fully supported (per its MODULE-COMPLIANCE)?	RMGT:O	5.4.1(k)(3), 15	Yes [ ] No [ ]

A.20 Application interfaces

Item	Feature	Status	References	Support
	If item APPL is not supported, mark N/A.			N/A[ ]
APPL-1	What application interfaces(s) are supported?	APPL:M	5.4.1(i)	

A.21 External port configuration

Item	Feature	Status	References	Support
	If item EXT is not supported, mark N/A.			N/A[ ]
EXT-1	Does the device support the specifications for externalPortConfigurationEnabled value of true?	EXT:M	10.3.1	Yes [ ]
EXT-2	Does the device support the PortAnnounceInformationExt state machine?	EXT:M	10.3.14	Yes [ ]
EXT-3	Does the device support the PortStateSettingExt state machine?	EXT:M	10.3.15	Yes [ ]

Annex B

(normative)

Performance requirements

B.1 LocalClock requirements

B.1.1 Frequency accuracy

The fractional frequency offset of the LocalClock relative to the TAI frequency (see ISO 80000-3:2006 and Annex C) shall be within ±100 ppm.

B.1.2 Time measurement granularity

The granularity with which the LocalClock measures time shall be less than or equal to 40/(1-0.0001) ns.

B.1.3 Noise generation

B.1.3.1 Jitter generation

The jitter generation of the free-running LocalClock shall not exceed 2 ns peak-to-peak, when measured over a 60 s measurement interval using a band-pass filter that consists of the following low-pass and high-pass filters:

a) High-pass filter: first-order characteristic (i.e., 0 dB gain peaking), 20 dB/decade roll-off, and 3 dB bandwidth (i.e., corner frequency) of 10 Hz
b) Low-pass filter: maximally-flat (i.e., Butterworth) characteristic, 60 dB/decade roll-off, and 3 dB bandwidth equal to the Nyquist rate of the LocalClock entity (i.e., one-half the nominal frequency of the LocalClock entity)

B.1.3.2 Wander generation

Wander generation is specified using the Time Deviation (TDEV) parameter. The corresponding values of the Allan Deviation (ADEV) and PTP Deviation (PTPDEV) are given for information; the former is also useful in describing the wander generation of clocks and oscillators, and the latter is related to the offsetScaledLogVariance attribute (see 8.6.2.4). Information on ADEV and TDEV are contained in ITU-T G.810 [B11] and IEEE Std 1139™-1999 [B4]. Information on ADEV and PTPDEV are contained in 7.6.3 of IEEE Std 1588-2019.

TDEV, denoted $\sigma_x(\tau)$, is estimated from a set of measurements, as shown in Equation (B.1):

$$\sigma_x(\tau) = \sqrt{\frac{1}{6n^2(N-3n+1)}\sum_{j=1}^{N-3n+1}\left[\sum_{i=j}^{n+j-1}(x_{i+2n}-2x_{i+n}+x_i)\right]^2}, n = 1, 2, \ldots\left\lfloor\frac{N}{3}\right\rfloor \tag{B.1}$$

where

$\tau = n\tau_0$ = observation interval
τ_0 = sampling interval
N = total number of samples [$(N–1)\tau_0$ = measurement interval]
$\lfloor y \rfloor$ denotes the floor function, i.e., the greatest integer less than or equal to y
x_i = measured phase (time) error at the i^{th} sampling time [the units of x_i and $\sigma_x(\tau)$ are the same]

ADEV, denoted $\sigma_y(\tau)$, is estimated from a set of measurements, as shown in Equation (B.2):

$$\sigma_y(\tau) = \sqrt{\frac{1}{2n^2\tau_0^2(N-2n)}\sum_{i=1}^{N-2n}(x_{i+2n}-2x_{i+n}+x_i)^2}, n = 1, 2, \ldots\left\lfloor\frac{N-1}{2}\right\rfloor \quad \text{(B.2)}$$

where the notation is the same as defined above for TDEV.

PTPDEV, denoted $\sigma_{PTP}(\tau)$, is estimated from a set of measurements, as shown in Equation (B.3):

$$\sigma_{PTP}(\tau) = \sqrt{\frac{1}{6(N-2n)}\sum_{i=1}^{N-2n}(x_{i+2n}-2x_{i+n}+x_i)^2}, n = 1, 2, \ldots\left\lfloor\frac{N-1}{2}\right\rfloor \quad \text{(B.3)}$$

where the notation is the same as defined above for TDEV.

TDEV, ADEV, and PTPDEV are second-order statistics on the phase error. All three statistics are functions of second differences of the phase error. This means that these statistics are not affected by a constant frequency offset. This behavior is desired, because these statistics are used here to constrain noise generation.

TDEV for the LocalClock entity shall not exceed the mask of Table B-1 and Figure B-1, when measured using

a) A measurement interval that is at least 120 s (i.e., at least 12 times the longest observation interval),
b) A low-pass filter with 3 dB bandwidth of 10 Hz, first-order characteristic, and 20 dB/decade roll-off, and
c) A sampling interval τ_0 that does not exceed 1/30 s.

Table B-1—Wander generation TDEV requirement for LocalClock entity

TDEV limit	Observation interval τ
No requirement	$\tau < 0.05$ s
5.0τ ns	$0.05 \le \tau \le 10$ s
No requirement	$\tau > 10$ s

The ADEV limit that corresponds to the TDEV requirement of Table B-1 and Figure B-1 is shown in Table B-2 and Figure B-2, respectively.

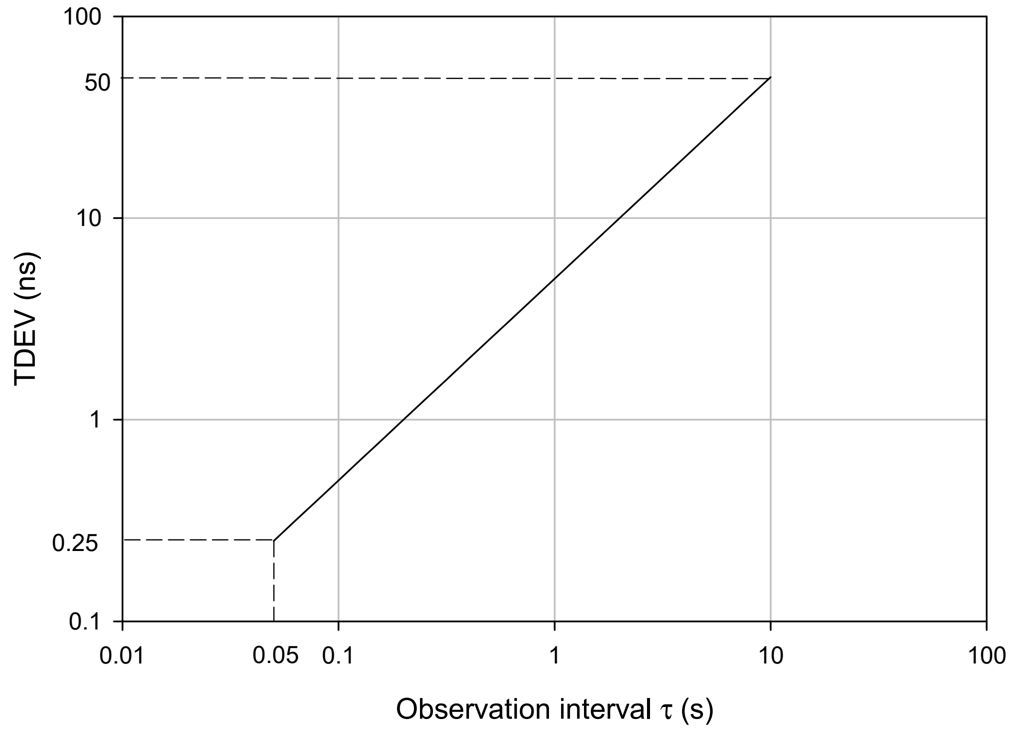


Figure B-1—Wander generation (TDEV) requirement for LocalClock entity

Table B-2—ADEV limit corresponding to wander generation requirement of Table B-1

ADEV limit	Observation interval τ
No requirement	$\tau < 0.05$ s
1.054×10^{-8}	$0.05 \le \tau \le 10$ s
No requirement	$\tau > 10$ s

The PTPDEV limit that corresponds to the TDEV requirement of Table B-1 and Figure B-1 is shown in Table B-3 and Figure B-3, respect, respectively.

B.2 PTP Instance requirements

B.2.1 General

In order to achieve the accuracy goals, certain constraints are placed on the responsiveness and accuracy of PTP Instances.

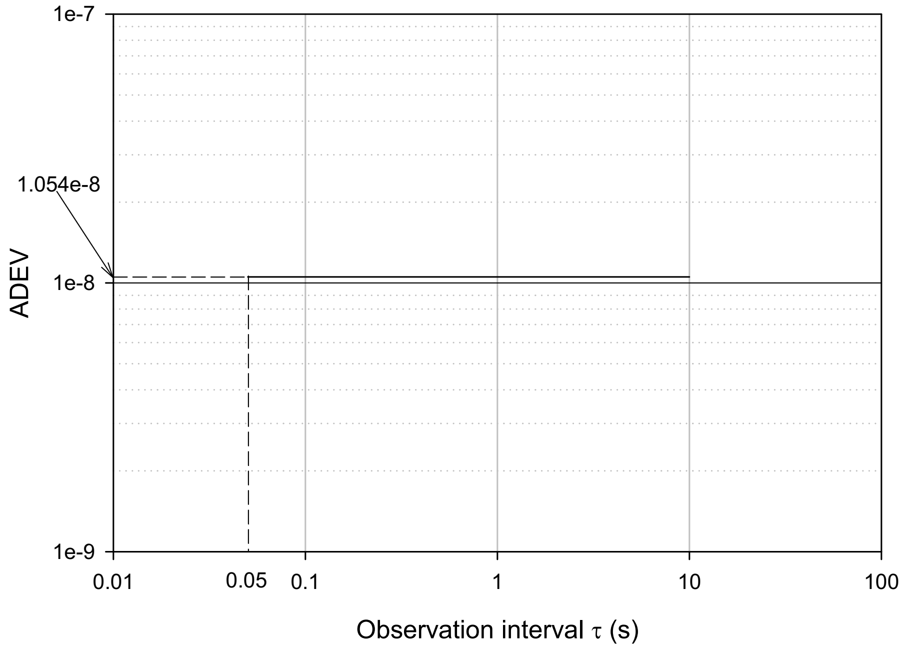


Figure B-2—ADEV limit corresponding to wander generation requirement of Figure B-1

Table B-3—PTPDEV limit corresponding to wander generation requirement of Table B-1

PTPDEV limit	Observation interval τ
No requirement	$\tau < 0.05$ s
6.08τ ns	$0.05 \le \tau \le 10$ s
No requirement	$\tau > 10$ s

B.2.2 Residence time

The residence time (see 3.22) of a PTP Instance, measured relative to the TAI second (see 8.2), should be less than or equal to 10 ms.

Any error in the measured frequency offset relative to the grandmaster (i.e., error in accumulated rateRatio) results in an error in the transported synchronized time that is equal to the frequency offset error multiplied by the residence time (see 7.3.3 and 11.1.3).

B.2.3 Pdelay turnaround time

The pdelay turnaround time is the duration of the interval between the receipt of a Pdelay_Req message by a port of a time-aware system, and the sending of the corresponding Pdelay_Resp message.

The pdelay turnaround time of a time-aware system, measured relative to the TAI second (see 8.2), should be less than or equal to 10 ms.

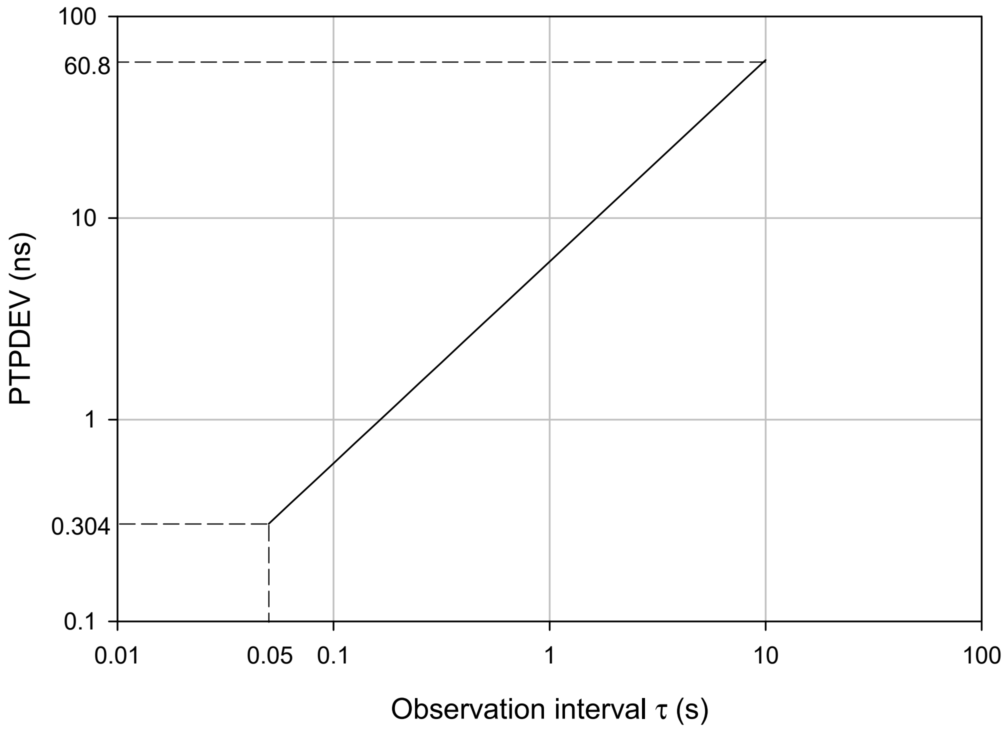


Figure B-3—PTPDEV limit corresponding to wander generation requirement of Figure B-1

A non-zero pdelay turnaround time and any error in the measured frequency offset between the peer delay initiator and peer delay responder (i.e., error in neighborRateRatio) results in an error in the measured mean propagation delay. This in turn results in an error in the transported synchronized time (see 11.1.2 for more details).

NOTE—While a larger value of pdelay turnaround time can result in worse time-synchronization performance, the peer delay protocol will still operate as long as the peer delay initiator receives Pdelay_Resp and Pdelay_Resp_Follow_Up within a time interval since sending Pdelay_Req that is less than the current Pdelay_Req message transmission interval (see 11.2.19.4 and 11.5.2.2).

B.2.4 Measurement of rate ratio

This standard requires the measurement of rate ratio or, equivalently, frequency offset, in several subclauses (see 10.2.10, 10.2.11, 11.2.19, 12.5.2, 16.4.2, and 16.4.3.1). The error inherent in any scheme used to measure rate ratio shall not exceed ±0.1 ppm.

NOTE—This requirement is consistent with a rate ratio measurement made by measuring the default Pdelay_Req message transmission interval (the nominal interval duration is 1 s, see 11.5.2.2) relative to the clocks whose rate ratio is desired, assuming the clocks meet the time measurement granularity requirement of B.1.2 (i.e., no worse than 40 ns).

B.3 End-to-end time-synchronization performance

The requirements of this standard and of standards referenced for each medium ensure that any two PTP Instances separated by six or fewer PTP Instances (i.e., seven or fewer hops) will be synchronized to within 1 µs peak-to-peak of each other during steady-state operation (i.e., each PTP Instance receives time-synchronization information every sync interval).

B.4 End-to-end jitter and wander performance

The requirements of this standard and standards referenced by this standard ensure that the synchronized time at a PTP Instance that is separated from the grandmaster by six or fewer PTP Instances (i.e., seven or fewer hops) will, when filtered by a reference endpoint filter with rolloff of 20 dB/decade, gain peaking that does not exceed 0.1 dB, and bandwidth that does not exceed the value given in each entry of Table B-4, have maximum time interval error (MTIE, see ITU-T Recommendation G.810 [B11]) that does not exceed the MTIE for that entry of Table B-4, and jitter that does not exceed the peak-to-peak jitter of Table B-4 when measured through the corresponding high-pass jitter measurement filter given in Table B-4.

NOTE—For example, the endpoint filter can be of the following form:

$$y_k = a_1 y_{k-1} + a_2 y_{k-2} + \dots + a_n y_{k-n} + b_0 x_k + b_1 x_{k-1} + \dots + b_n x_{k-n}$$

where the x_k are the unfiltered synchronized time values, the y_k are the filtered synchronized time values, and the a_k and b_k are filter coefficients. The a_k and b_k are chosen such that the filter has desired bandwidth and gain peaking that does not exceed 0.1 dB. The preceding equation is a general infinite impulse response (IIR) digital filter. Simplified forms, e.g., a second order IIR filter obtained by setting $n = 2$, or a finite impulse response (FIR) filter obtained by setting the a_k to zero are possible.

Table B-4—Maximum endpoint filter bandwidths needed to meet respective MTIE masks and peak-to-peak jitter limits

Endpoint filter maximum bandwidth (Hz)	Corresponding MTIE mask of Figure B-4 that is not exceeded	Corresponding jitter high-pass measurement filter (Hz)	Corresponding peak-to-peak jitter that is not exceeded (ns)
10	Mask 2 (Figure B-4,Table B-6)	8000	10.2
1	Mask 1 (Figure B-4,Table B-5)	200	11.1

Mask 1 of Table B-4 and Figure B-4 corresponds to consumer digital audio applications. Mask 2 of Table B-4 and Figure B-4 corresponds to professional digital audio applications. Mask 1 is derived from the requirements given in [B19]. Mask 2 is derived from the requirements given in [B20], [B21], and [B22]. The methodology for deriving MTIE from the various jitter and synchronization requirements is described in [B23], and the detailed derivations for masks 1 and 2 are given in [B23].

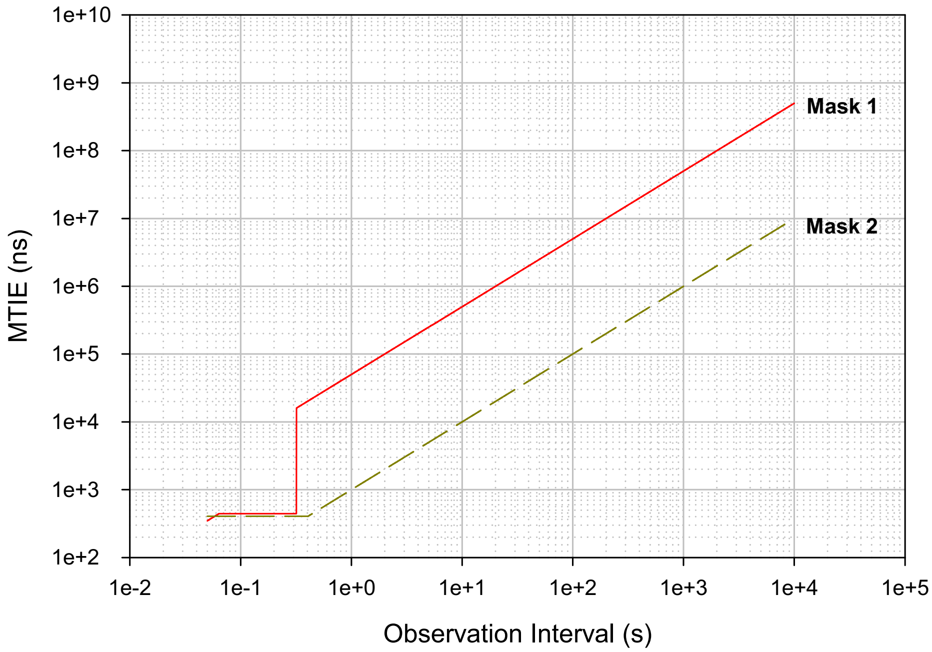


Figure B-4—MTIE masks met for maximum endpoint filter bandwidths of Table B-4

Table B-5—Breakpoints for Mask 1

Observation interval S (s)	MTIE (ns)
0.05 ≤ S < 0.0637	6954.8S
0.0637 ≤ S < 0.3183	443
0.3183 ≤ S ≤ 10000	50000S

Table B-6—Breakpoints for Mask 2

Observation interval S (s)	MTIE (ns)
0.05 ≤ S < 0.4069	407
0.4069 ≤ S < 10000	1000S

Annex C

(informative)

Timescales and epochs

C.1 Overview

A more detailed discussion of many of the topics in this annex can be found in Allan, Ashby, and Hodge [B1].

For historical reasons, time is specified in a variety of ways as listed in Table C-1. GPS, PTP, and TAI times are based on values yielded by atomic clocks and advance on each second. NTP and UTC times are similar, but are occasionally adjusted by one leap second, to account for differences between the atomic clocks and the rotation time of the earth.

Table C-1—Timescale parameters

	Timescale				
Parameter	**GPS**	**PTP**	**TAI**	**NTP**	**UTC**
approximate epoch†	1980-01-06 1999-08-22	1970-01-01	no epoch defined	1900-01-01	no epoch defined
representation	weeks.seconds	seconds	YYYY-MM-DD hh:mm:ss	seconds	YYYY-MM-DD hh:mm:ss
rollover (years)	19.7	≈8 900 000	10 000	≈136	10 000
leapSeconds	no			yes	

NOTE 1—After 1972-01-01 00:00:00 TAI, TAI and UTC differ by only integer seconds.
NOTE 2—The following represent the same instant in time:
1958-01-01 00:00:00 TAI and 1958-01-01 00:00:00 UTC

Legend:
† Each approximate epoch occurs at 00:00:00 on the respective date
GPS global positioning satellite
NTP Network Time Protocol
PTP Precision Time Protocol
TAI International Atomic Time (from the French term *Temps Atomique International*)
UTC Coordinated Universal Time (the acronym is a compromise between the English term *coordinated universal time* and the French term *temps universel coordonné*
English speakers wanted the initials of their language: CUT for *coordinated universal time*
French speakers wanted the initials of their language: TUC for *temps universel coordonné*

C.2 TAI and UTC

TAI and UTC are international standards for time based on the SI second (see [B1], [B24], and [B25]). The SI second is the duration of 9 192 631 770 periods of the radiation corresponding to the transition between the two hyperfine levels of the ground state of the cesium 133 atom (see [B10], [B12], [B15], [B16], [B24], and [B25] for more details on UTC, TAI, and the SI second). TAI is implemented by a suite of atomic clocks and forms the timekeeping basis for other timescales in common use. The rate at which UTC time advances is normally identical to the rate of TAI. An exception is an occasion when UTC is modified by adding or subtracting exactly one whole leap second. The TAI frequency is the frequency of a signal whose period is the TAI second.

The TAI and UTC timescales were introduced as of 1958-01-01, and the times 1958-01-01 00:00:00 TAI and 1958-01-01 00:00:00 UTC represent the same instant in time. However, this instant in time is not an epoch for TAI and UTC. An epoch, in the sense of the origin of a timescale (see 8.2.2), is not defined for either TAI or UTC. TAI and UTC are expressed in the form YYYY-MM-DD hh:mm:ss, rather than as elapsed time since an epoch; here, YYYY-MM-DD denotes the date and hh:mm:ss denotes the time in each day.

Prior to 1972-01-01, corrections to the offset between UTC and TAI were made by applying fractional-second corrections to UTC and corrections to the rate at which UTC advanced relative to the rate at which TAI advanced. After 1972-01-01, leap-second corrections are applied to UTC by inserting or deleting second(s) at the end of the last minute of preferably the last day of June or December. Also after 1972-01-01, UTC and TAI advance at the same rate. As of 2006-01-01, TAI and UTC times differed by +33 s (i.e., TAI time minus UTC time equals +33 s for 2006-01-01).

In computer networks, the common POSIX-based time conversion algorithms are typically used to produce the correct ISO 8601:2004 [B6] printed representations for both TAI and UTC.

The PTP epoch is set such that a direct application of the POSIX algorithm to a PTP timescale timestamp converts the PTP timestamp to the ISO 8601:2004 [B6] printed representation of TAI. PTP also distributes the current offset between TAI and UTC in the currentUtcOffset field of Announce messages. Except during leap seconds, subtracting currentUtcOffset from a PTP timestamp and then applying the POSIX algorithm results in the ISO 8601:2004 printed representation of UTC. Conversely, except during leap seconds, applying the inverse POSIX algorithm and adding currentUtcOffset converts from the ISO 8601:2004 printed form of UTC to the form required to generate a PTP timestamp.

For example, at 0h 2 Jan 1972 TAI, the value of PTP Instance Time was 63,158,400. At this time currentUtcOffset was 10. The POSIX algorithm applied to the value (63,158,400 - 10) gives a value of 23:59:50 1972-01-01 (10 s before 0h 2 January 1972 UTC). The value of PTP Instance Time on 0h 2 Jan 1972 TAI is computed by observing that PTP Instance Time = 0 on 0h 1 Jan 1970 TAI, i.e. Modified Julian Day (MJD, see [B25]) 40587. On 0h 2 Jan 1972 TAI, MJD = 41318. Thus, PTP Instance Time on 0h 2 Jan 1972 TAI is 0 + 86400 × (41318 - 40587). Note that if this calculation were done for a day in which a leap second occurred, a more complex algorithm would be required to ensure that, for the duration of the leap second, the ISO 8601:2004 print form seconds value was 60 for a positive leap second.

C.3 NTP and GPS

Two standard time sources of particular interest in implementing PTP Instances: NTP and GPS. Both NTP and GPS systems are expected to provide time references for calibration of the grand-master supplied PTP time.

NTP represents seconds as a 32-bit unsigned integer that rolls-over every 2^{32} s ≈ 136 year, with the first such rollover occurring in the year 2036. The precision of NTP systems is usually in the millisecond range.

NTP is a widely used protocol for synchronizing computer systems. NTP is based on sets of servers, to which NTP clients synchronize. These servers themselves are synchronized to time servers that are traceable to international standards.

NTP provides the current time in NTP version 4, the current *leapSeconds* value, and warning flags indicating that a *leapSecond* will be inserted at the end of the current UTC day. The NTP clock effectively stops for one second when the leap second is inserted.

GPS time comes from a global positioning satellite system, GPS, maintained by the U.S. Department of Defense. The precision of GPS system is usually in the 10 ns to 100 ns range. GPS system transmissions represent the time as {*weeks*, *secondsInWeek*}, the number of weeks since the GPS epoch and the number of seconds since the beginning of the current week.

GPS also provides the current *leapSeconds* value, and warning flags marking the introduction of a leap second correction. UTC and TAI times can be computed solely based the information contained in the GPS transmissions.

GPS timing receivers generally manage the epoch transitions (1024-week rollovers), providing the correct time (YYYY-MM-DD hh:mm:ss) in TAI and/or UTC timescales, and often also local time; in addition to providing the raw GPS week, second of week, and leap-second information.

C.4 Timescale conversions

Previously discussed representations of time can be readily converted to/from PTP *time* based on a constant offset and the distributed *leapSeconds* value, as specified in Table C-2. Within Table C-2, all variables represent integers; "/" and "%" represent a integer divide and remainder operation, respectively.

Table C-2—Timescale conversions

ta		**PTP value** tb
Name	**Format**	
GPS	weeks:seconds	tb = ta.seconds + 315 964 819 + (gpsRollovers * 1024 + ta.weeks) * (7 * DAYSECS);
		ta.weeks = (tb – 315 964 819) / (7 * DAYSECS) – gpsRollovers*1024; ta.seconds = (tb – 315 964 819) % (7 * DAYSECS);
TAI	date{YYYY,MM,DD}:time{hh,mm,ss}	tb = DateToDays("1970-01-01", ta.date) * DAYSECS + ((ta.time.hh * 24) + ta.time.mm) *60) + ta.time.ss;

		secs = tb % DAYSECS;
		ta.date = DaysToDate("1970-01-01", tb / DAYSECS);
		ta.time.hh = secs / 3600;
		ta.time.mm = (secs % 3600)/60;
		ta.time.ss = (secs % 60);

NTP	seconds	tb = (ta+leapSeconds)–2208988800;
		ta = (tb–leapSeconds)+2208988800;
UTC	date{YYYY,MM,DD}:time{hh,mm,ss}	tb = DateToDays("1970-01-01", ta.date) * DAYSECS + ((ta.time.hh * 24) + ta.time.mm) *60) + ta.time.ss + leapSeconds;

		tc = tb – leapSeconds;
		secs = tc % DAYSECS;
		ta.date = DaysToDate("1970-01-01", tc/DAYSECS);
		ta.time.hh = secs / 3600;
		ta.time.mm = (secs % 3600)/60;
		ta.time.ss = (secs % 60);

NOTE—*gpsRollovers* Currently equals 1; changed from 0 to 1 between 1999-08-15 and 1999-08-22.
DAYSECS The number of seconds within a day: (60*60*24).
leapSeconds Extra seconds to account for variations in the earth-rotation times: 33 on 2006-01-01.
DateToDays For arguments DateToDays(*past*, *present*), returns days between *past* and *present* dates.
DaysToDate For arguments DaysToDate(*past*, *days*), returns the current date, *days* after the *past* date.

C.5 Time zones and GMT

The term Greenwich Mean Time (GMT) once referred to mean solar time at the Royal Observatory in Greenwich, England. GMT now commonly refers to the timescale UTC; or the UK winter time zone (Western European Time, WET). Such GMT references are, strictly speaking, incorrect but nevertheless quite common. The following representations correspond to the same instant of time:

18:07:00 (GMT), commonplace usage	13:07:00 (Eastern Standard Time, EST)
18:07:00 (UTC)	01:07 PM (Eastern Standard Time, EST)
18:07:00 (Western European Time, WET)	10:07:00 (Pacific Standard Time, PST)
06:07 PM (Western European Time, WET)	10:07 AM (Pacific Standard Time, PST)

Annex D

(informative)

Annex D is currently unused.

Annex E

(informative)

Annex E is now Clause 16

Annex F

(informative)

PTP profile included in this standard

The specification in this standard of synchronized time transport over a full-duplex, point-to-point link includes a PTP profile. The information contained in a PTP profile is described in 20.3 of IEEE Std 1588-2019. This annex summarizes the PTP profile for transport of timing over full-duplex, point-to-point links. This PTP profile is also used in the transport of timing over CSN for the case where a CSN network clock reference is not present (see Clause 16). This PTP profile is not used in the transport of timing over IEEE 802.11 links and IEEE 802.3 EPON links; both these transports use native timing mechanisms to assist in the synchronized time transport.

F.1 Identification

The identification values for this PTP profile (see 20.3.3 of IEEE Std 11588-2019) are as follows:

PTP Profile:
IEEE 802.1AS PTP profile for transport of timing
Profile Name: IEEE 802.1AS PTP profile
profileNumber: 0
primaryVersion 2
revisionNumber: 0
profileIdentifier: 00-80-C2-00-02-00

NOTE—In the above profileIdentifier (see 20.3.3 of IEEE Std 1588-2019):

a) 00-80-C2 (first three octets) is the OUI owned by the IEEE 802.1 Working Group (it identifies the organization that specifies the PTP profile);
b) 00 (fourth octet) is a number that identifies the PTP profile, among all profiles specified by the organization that owns the OUI (or, in general, OUI or CID) of a);
c) 02 (fifth octet) is the primary version of this PTP profile; and
d) 00 (sixth octet) is the revisionNumber of this PTP profile.

This profile is specified by the IEEE 802.1 Working Group of the IEEE 802 LAN/MAN Standards Committee.

A copy may be obtained by ordering IEEE Std 802.1AS-2019 from the IEEE Standards Organization.[16]

This PTP profile is a revision of the PTP profile included in IEEE Std 802.1AS-2011, i.e., major changes have been made to the profile relative to Version 1.0. Therefore, the primaryVersion is changed to 2, with revisionNumber 0.

F.2 PTP attribute values

The ranges and default values for time-aware system and PTP Instance attributes covered by this profile are as follows:

[16]IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, Piscataway, NJ 08854, USA (http://standards.ieee.org).

a) A domain whose domain number is 0 is present. A domain whose domain number is in the range 1 - 127 may be present (see 8.1).
b) The default logAnnounceInterval (see 10.7.2.2) is 0. The value 127 is supported.
c) The default logSyncInterval (see 11.5.2.3) is –3. The value 127 is supported.
d) The default logPdelayReqInterval (see 11.5.2.2) is 0. The value 127 is supported.
e) The default announceReceiptTimeout (see 10.7.3.2) is 3.
f) The default values of priority1, for different media, are specified in 8.6.2.1, Table 8-1. The value of priority1 for a PTP Instance that is not grandmaster-capable is 255.
g) The default value of priority2 is 248 (see 8.6.2.5).
h) The default observation interval for offsetScaledLogVariance is equal to the default sync interval, i.e., 0.125 s (see 8.6.2.4).

F.3 PTP options

a) The best master clock algorithm is the alternate BMCA specified in 10.3.
b) The following options of 17.7 of IEEE Std 1588-2019 are invoked:
    1) The FAULTY state is not used.
    2) The UNCALIBRATED state is not used.
    3) The LISTENING state is not used
    4) The PRE_MASTER state, and PRE_MASTER qualification are not used.
    5) The foreign master feature is not used.
c) The management mechanism is the mechanism specified in Clause 14 and Clause 15.
d) The path delay mechanism is the peer-to-peer delay mechanism (see Clause 11).
e) The transport mechanism is full-duplex, point-to-point, and uses attribute values described in Annex F of IEEE Std 1588-2019 for IEEE 802.3 Ethernet. Specifically, the address, EtherType, and subtype are specified in 11.3.4, 11.3.5, and 11.3.6.
f) A PTP Instance that contains one PortSync and one MD entity is an ordinary clock. A PTP Instance that contains more than one PortSync and more than one MD entity is a boundary clock.
g) Each port of a time-aware system measures the frequency offset of its neighbor, at the other end of the attached link, relative to itself (see Clause 11). The frequency offset, relative to the grandmaster, is accumulated in a standard organization TLV that is attached to the Follow_Up message if the port is two-step and the Sync message if the port is one-step (see 11.4.4.3). The standard organization TLV also carries information on grandmaster traceability and phase and frequency change due to the most recent grandmaster change. The physical adjustment of the frequency of the LocalClock entity (i.e., physical syntonization) is allowed but not required.

NOTE—This feature is similar to the cumulative frequency transfer method specified in 16.10 of IEEE Std 1588-2019; however, it is not the same feature, and uses a different TLV from the one used in the IEEE Std1588-2019 feature. This feature existed in the 2011 edition of this standard, and is retained for backward compatibility, and also because the TLV of this standard carries additional information that is not carried in the IEEE Std 1588-2019 feature [see item g) above].

h) A standard organization TLV is defined to allow a PTP Port to signal to its neighbor PTP Port that it is capable of invoking the gPTP protocol (see and 10.6.4.4).
i) The path trace feature of 16.2 of IEEE Std 1588-2019 is used (see 10.6.3.2.8).
j) A standard organization TLV is defined that allows a port of a time-aware system to request that its neighbor slow down or speed up the rate at which it sends Sync/Follow_Up, peer delay, and/or Announce messages (see 10.6.4.3).
k) The acceptable master table feature of IEEE Std 1588-2019 is used with IEEE 802.3 EPON links to ensure that the OLT is master and ONUs are slaves.

NOTE—This feature is used with EPON links, and therefore could be considered to be outside the PTP profile (because EPON links are not part of the PTP profile). It is included here because it is one of the optional features described in IEEE Std 1588-2019.

l) The profile isolation feature of IEEE Std 1588-2019 is not explicitly used; however, the PTP profile specified in this standard is isolated from other PTP profiles because it uses sdoId 0x100 (see 8.1).
m) A time-aware system can have more than one gPTP domain (see 8.1)
n) The Common Mean Link Delay Service specified in 16.6 of IEEE Std 1588-2019 is required if more than one domain is implemented, and is optional if one domain (with domainNumber 0) is implemented (see 11.2.17).
o) The security mechanism of 16.6 of IEEE Std 1588-2019, and security annex (Annex S) of IEEE Std 1588-2019 are not used.
p) The external port configuration feature of 17.6 of IEEE Std 1588-2019 is optional in this standard.
q) Except for items g) through p) above, the optional features of Clause 16 and Clause 17 and the optional Annexes of IEEE Std 1588-2019 are not used.

F.4 LocalClock and PTP Instance performance requirements

The LocalClock performance requirements are as specified in B.1. The PTP Instance performance requirements are as specified in Annex B.2.

Annex G

(informative)

The Asymmetry Compensation Measurement Procedure based on line-swapping

G.1 Introduction

This annex describes the asymmetry compensation measurement procedure based on the line-swapping method. The entire procedure is controlled by the Network Management System (NMS), except that the line-swapping is manually operated. This Annex is intended to address the delay asymmetry due to length differences between transmit and receive fibers, for long fiber runs.

NOTE—When a port is put into asymmetry measurement mode, it is put into this mode on all domains. The per-port, global variable asymmetryMeasurement mode is common to and accessible by all domains (see 10.2.5.2).

G.2 Pre-conditions for measurement

The following pre-conditions should be met, both to improve the accuracy of the measurement and to make the measurement procedure more convenient:

a) The measurement environment, including the testing nodes (i.e., the time-aware systems at the endpoint of the link whose asymmetry is being compensated) and related nodes (i.e., nodes in the paths between the testing nodes and the grandmaster), should enable gPTP and BMCA, so that the test can be made for each link without changing the configuration.
b) The testing nodes should have redundant paths for synchronization, so that they remain synchronized to the grandmaster during the asymmetry compensation measurement.

G.3 Measurement procedure

The assumed measurement environment is shown in Figure G-1. Before the measurement starts, every node has enabled syntonization (i.e., by measuring neighborRateRatio (see 10.2.5.7) for each port of each time-aware system and accumulating the respective values over the synchronization spanning tree paths to obtain rateRatio (see 10.2.8.1.4) relative to the grandmaster[17]) and time synchronization. The testing link is between Port2 on node ACC1 and Port1 on node ACC2.

The measurement procedure is as follows:

a) The NMS puts Port1 of ACC2 and Port2 of ACC1 into asymmetry measurement mode through the MIB (i.e., by setting the managed object asymmetryMeasurementMode for each port to TRUE). These two ports will not affect the PTP calculations of either node when the ports are in asymmetry measurement mode. If synchronization flowed over the link connecting these ports prior to their being put into asymmetry measurement node, the BMCA will result in a reconfiguration of the synchronization spanning tree so that both ACC1 and ACC2 remain synchronized.

[17]This standard neither requires nor prohibits syntonization (see 3.26) at the physical layer, e.g,, using Synchronous Ethernet as the physical layer. If frequency is syntonized at the physical layer, the respective neighborRateRatio and rateRatio values are expected to be close to 1.

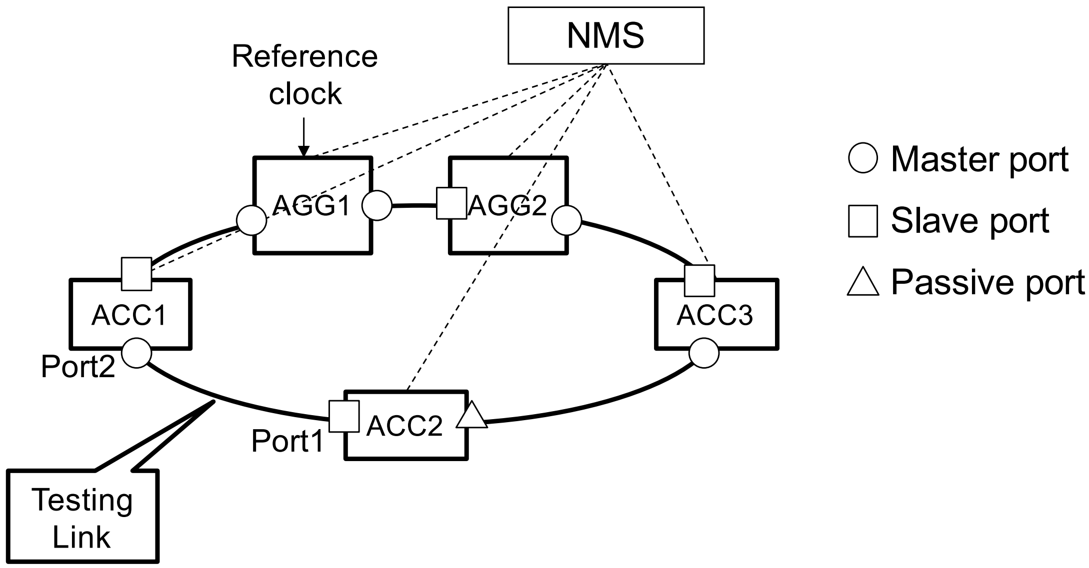


Figure G-1—Asymmetry compensation measurement procedure

b) Port1 of ACC2 will send Pdelay_Req messages periodically using the peer delay measurement mechanism and receive Pdelay_Resp and Pdelay_Resp_Follow_Up messages. For each set of messages, ACC2 should save t3 (the <pdelayRespEventEgressTimestamp>, see 11.3.2.1, carried in the Pdelay_Resp_Follow_Up message) and t4 (the <pdelayRespEventIngressTimestamp>, taken when the Pdelay_Resp message timestamp point crosses the reference plane at Port1 of ACC2 on reception).
c) The NMS reads and saves multiple sets of (t3, t4) from ACC2 through the MIB. It is necessary that each set of (t3, t4) is from the same measurement, i.e., from the same peer delay message exchange. The number of (t3, t4) sets can be decided as required, and is outside the scope of gPTP.
d) Manually exchange the transmit and receive fibers of Port2 of ACC1 and Port1 of ACC2. Wait until the port status and protocol status become stable again.
e) Port1 of ACC2 will again make periodic peer delay measurements, and save each set of measurement values (t3', t4') (the primes are used to de measurements that have occurred after the transmit and receive fibers have been exchanged).
f) The NMS reads and saves the multiple sets of (t3', t4') from ACC2 through the MIB. Then the NMS can compute the delay asymmetry, in units of time, as (t4' - t4)*neighborRateRatio - (t3' - t3). The NMS can use multiple sets of (t3, t4, t3', t4') to compute average values to get a more accurate result. The averaging method can be decided as required, and is outside the scope of gPTP. If ACC1 and ACC2 are frequency synchronized, then neighborRateRatio is 1 and the delay asymmetry is (t4' - t3') - (t4 - t3).
g) Based on the above result, the NMS sets the asymmetry value for Port1 of ACC2 and Port2 of ACC1 through the MIB.
h) After the NMS sets Port 1 of ACC2 and Port2 of ACC1 into normal mode (i.e., by setting the managed object asymmetryMeasurementMode for each port to FALSE), the delay asymmetry measurement of the testing link is completed. ACC1 and ACC2 will use the computed delay asymmetry as compensation in the PTP calculations.

It is also possible to compute the asymmetry ratio, i.e., the ratio of the delay on the receive fiber at ACC2 (Delay_rx_fiber, the delay of the Pdelay_Resp) to the delay on the transmit fiber at ACC2 (Delay_tx_fiber, the delay of the Pdelay_Req), both after line swapping. In this case, the NMS should collect multiple sets of (t1, t2, t3, t4) and (t1', t2', t3', t4') before and after line-swapping, respectively, where t1 is the

<pdelayReqEventEgressTimestamp> (taken when the Pdelay_Req message timestamp point crosses the reference plane at Port1 of ACC2 on transmission), t2 is the <pdelayReqEventIngressTimestamp> (carried in the requestReceiptTimestamp field of the Pdelay_Resp message), and t3 and t4 are as given above. The NMS can compute the delay on the receive fiber as [(t4' - t1)*neighborRateRatio - (t3' - t2)] /2 and the delay on the transmit fiber as [(t4 - t1')*neighborRateRatio - (t3 - t2')]/2. Then the asymmetry ratio is Delay_rx_fiber/Delay_tx_fiber = [(t4' - t1)*neighborRateRatio - (t3' - t2)] / [(t4 - t1')*neighborRateRatio - (t3 - t2')]

Annex H

(informative)

Bibliography

[B1] Allan, David W., Ashby, Neil, and Hodge, Clifford C., "The Science of Timekeeping," Hewlett Packard Application 1289, 1997.[18]

[B2] IEEE MA-L, MA-M, and MA-S assignments may be obtained from the IEEE Registration Authority. These assignements may be used to form EUI-48s and EUI-64s, as described in the following tutorial: (a) *Guidelines for Use of Extended Unique Identifier (EUI), Organizationally Unique Identifier (OUI), and Company ID (CID)*.[19]

[B3] IEEE Std 802®, IEEE Standard for Local and Metropolitan Area Networks—Overview and Architecture.

[B4] IEEE Std 1139™-1999, IEEE Standard Definitions of Physical Quantities for Fundamental Frequency and Time Metrology—Random Instabilities.

[B5] IEEE Std 1003.1™-2008, IEEE Standard for Information Technology—Portable Operating System Interface (POSIX®) Base Specifications, Issue 7.

[B6] ISO 8601:2004, Data elements and interchange formats—Information interchange—Representation of dates and times.[20]

[B7] ISO/IEC 8802-2, Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 2: Logical link control.

[B8] ISO/IEC 9945:2003, Information technology. Portable Operating System Interface (POSIX®).

[B9] ISO/IEC 14882:2003, Programming languages—C++.

[B10] ITU-R Recommendation TF.460-6, Standard-frequency and time-signal emissions, 2002.[21]

[B11] ITU-T Recommendation G.810, Definitions and Terminology for Synchronization Networks, ITU-T, Geneva, August, 1996, Corregendum 1, November, 2001.

[B12] Jekeli, Christopher, "Geometric Reference Systems in Geodesy," Division of Geodesy and Geospatial Science, School of Earth Sciences, Ohio State University, July 2006.[22]

[18]Available at http://www.allanstime.com/Publications/DWA/Science_Timekeeping/TheScienceOfTimekeeping.pdf.

[19]Available at https://standards.ieee.org/regauth/. Tutorials on these assigned numbers can be found on this Web site.

[20]ISO publications are available from the ISO Central Secretariat, 1, ch. de la Voie-Creuse, Case Postale 56, CH-1211, Geneva 20, Switzerland (http://www.iso.org/). IEC publications are available from the Central Office of the International Electrotechnical Commission, 3, rue de Varembé, P.O. Box 131, CH-1211, Geneva 20, Switzerland (http://www.iec.ch/). ISO/IEC publications are also available in the United States from the Sales Department, American National Standards Institute, 25 West 43rd Street, 4th Floor, New York, NY 10036, USA (http://www.ansi.org/).

[21]ITU publications are available from the International Telecommunications Union, Place des Nations, CH-1211, Geneva 20, Switzerland/Suisse (http://www.itu.int/).

[22]Available at https://kb.osu.edu/dspace/bitstream/1811/24301/1/Geom_Ref_Sys_Geodesy.pdf.

[B13] MoCA MAC/PHY Specification v1.0, MoCA-M/P-SPEC-V1.0-07122009, Multimedia over Coax Alliance (MoCA), July 12, 2009.[23]

[B14] MoCA MAC/PHY Specification Extensions v1.1, MoCA-M/P-SPEC-V1.1-06162009, Multimedia over Coax Alliance (MoCA), June 16, 2009.

[B15] Service de la Rotation Terrestre, Observatoire de Paris, 61, Av. de l'Observatoire 75014 Paris (France).

[B16] "The International System of Units (SI)," 8th edition, Bureau International des Poids et Mesures, 2006.[24]

[B17] U.S. Naval Observatory.[25]

[B18] Anthanasios Papoulis, "Probability, Random Variables, and Stochastic Processes (Third Edition)," Mc-Graw-Hill, 1991.

[B19] IEC 60958-3, Digital Audio Interface – Part3: Consumer Applications, International Electrotechnical Commission, Geneva, 2016.

[B20] IEC 60958-3, Digital Audio Interface – Part4: Professional Applications (TA4), International Electrotechnical Commission, Geneva, 2016.

[B21] AES3-2009 (reaffirmed in 2014), AES Recommended practice for digital audio engineering – Serial transmission format for two-channel linearly represented digital audio data, Audio Engineering Society, 2014.

[B22] AES11-2009 (reaffirmed in 2014), AES recommended practice for digital audio engineering – Synchronization of digital audio equipment in studio operations, Audio Engineering Society, 2014.

[B23] Geoffrey M. Garner, End-to-End Jitter and Wander Requirements for ResE Applications, presentation to IEEE 802.3 Residential Ethernet Study Group, May, 2005 (available at http://www.ieee802.org/3/re_study/public/200505/garner_3_0505.pdf).

[B24] Proc. 21st General Assembly of the IAU, IAU Trans., 1991, vol. XXIB, Kluwer.

[B25] Gerard Petit and Brian Luzum (eds.), IERS Conventions, IERS Technical Note No. 36, 2010 (available at ftp://tai.bipm.org/iers/conv2010/tn36.pdf).

[B26] Geoffrey M. Garner, Derivation of FTM Parameters in 12.6 of 802.1AS-Rev, presentation to IEEE 802.1 TSN TG, October 28, 2018 (available at http://www.ieee802.org/1/files/public/docs2018/as-garner-derivation-of-ftm-parameters-1118.pdf).

[23]MoCA specifications are available from the Multimedia over Coax Alliance at http://www.mocalliance.org/specs.

[24]Available at http://www.bipm.org/en/si/si_brochure/.

[25]Available at http://maia.usno.navy.mil/ser7/tai-utc.dat.