

INTERNATIONAL STANDARD

**IEC
61588**

First edition
2004-09

IEEE 1588™

Precision clock synchronization protocol for networked measurement and control systems



IEEE

Reference number
IEC 61588(E):2004
IEEE Std. 1588(E):2002
Documents
AVA Document Sharing Hub

Publication numbering

As from 1 January 1997 all IEC publications are issued with a designation in the 60000 series. For example, IEC 34-1 is now referred to as IEC 60034-1.

Consolidated editions

The IEC is now publishing consolidated versions of its publications. For example, edition numbers 1.0, 1.1 and 1.2 refer, respectively, to the base publication, the base publication incorporating amendment 1 and the base publication incorporating amendments 1 and 2.

Further information on IEC publications

The technical content of IEC publications is kept under constant review by the IEC, thus ensuring that the content reflects current technology. Information relating to this publication, including its validity, is available in the IEC Catalogue of publications (see below) in addition to new editions, amendments and corrigenda. Information on the subjects under consideration and work in progress undertaken by the technical committee which has prepared this publication, as well as the list of publications issued, is also available from the following:

- **IEC Web Site (www.iec.ch)**
- **Catalogue of IEC publications**
The on-line catalogue on the IEC web site (www.iec.ch/searchpub) enables you to search by a variety of criteria including text searches, technical committees and date of publication. On-line information is also available on recently issued publications, withdrawn and replaced publications, as well as corrigenda.
- **IEC Just Published**
This summary of recently issued publications (www.iec.ch/online_news/justpub) is also available by email. Please contact the Customer Service Centre (see below) for further information.
- **Customer Service Centre**
If you have any questions regarding this publication or need further assistance, please contact the Customer Service Centre:

Email: custserv@iec.ch
Tel: +41 22 919 02 11
Fax: +41 22 919 03 00

INTERNATIONAL STANDARD

IEC 61588

First edition
2004-09

IEEE 1588™

Precision clock synchronization protocol for networked measurement and control systems

Copyright © IEEE 2004 — All rights reserved

IEEE is a registered trademark in the U.S. Patent & Trademark Office, owned by the Institute of Electrical and Electronics Engineers, Inc.

No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher.

International Electrotechnical Commission, 3, rue de Varembé, PO Box 131, CH-1211 Geneva 20, Switzerland
Telephone: +41 22 919 02 11 Telefax: +41 22 919 03 00 E-mail: inmail@iec.ch Web: www.iec.ch

The Institute of Electrical and Electronics Engineers, Inc, 3 Park Avenue, New York, NY 10016-5997, USA
Telephone: +1 732 562 3800 Telefax: +1 732 562 1571 E-mail: stds-info@ieee.org Web: www.standards.ieee.org



Commission Electrotechnique Internationale
International Electrotechnical Commission
Международная Электротехническая Комиссия



CONTENTS

Foreword.....	4
IEEE introduction.....	7
1. Overview.....	8
1.1 Scope.....	9
1.2 Purpose.....	9
2. References.....	9
3. Definitions.....	10
4. Conventions	12
4.1 Descriptive syntax.....	12
4.2 Word usage	12
4.3 Behavioral specification notation	13
5. Datatypes in a PTP system.....	15
5.1 Primitive datatypes.....	15
5.2 Derived datatypes.....	16
6. PTP Clock synchronization model.....	17
6.1 Overview	17
6.2 PTP systems	21
7. PTP protocol specification	33
7.1 Protocol model of a clock	33
7.2 Protocol model of a subdomain of PTP clocks	35
7.3 State behavior of clocks	38
7.4 Clock data set	42
7.5 Messaging and internal event behavior of clocks	54
7.6 Best master clock algorithm.....	71
7.7 Clock variance computation	83
7.8 Local clock synchronization	86
7.9 Values for system and clock constants	89
7.10 Physical requirements for PTP implementations	91
7.11 PTP timing and event ordering constraints	91
7.12 Management message semantics	95
8. PTP inter-clock message formats.....	95
8.1 Inter-clock messages	102
8.2 PTP message header specification	102
8.3 Sync and Delay_Req messages.....	105
8.4 Follow_Up messages	109
8.5 Delay_Resp messages.....	110
8.6 Management messages.....	111

9. Conformance 120

9.1 Conformance objective 120

9.2 PTP node conformance requirements 121

9.3 PTP system conformance requirements..... 121

Annex A (informative) Using the PTP protocol 122

Annex B (informative) Time scales and epochs in PTP 128

Annex C (normative) subdomain_name to subdomain_address mapping algorithm 133

Annex D (normative) Ethernet implementation of PTP 135

Annex E (informative) Bibliography 152

Annex F (informative) List of Participants 153

INTERNATIONAL ELECTROTECHNICAL COMMISSION

**PRECISION CLOCK SYNCHRONIZATION PROTOCOL
FOR NETWORKED MEASUREMENT AND CONTROL SYSTEMS**

FOREWORD

- 1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
- 3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
- 5) IEC provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with an IEC Publication.
- 6) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

International Standard IEC/IEEE 61588 has been processed through subcommittee 65C: Digital communications, of IEC technical committee 65: Industrial-process measurement and control.

The text of this standard is based on the following documents:

IEEE Std	FDIS	Report on voting
1588 (2002)	65C/333/FDIS	65C/340/RVD

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

The committee has decided that the contents of this publication will remain unchanged until 2007.

IEC/IEEE Dual Logo International Standards

This Dual Logo International Standard is the result of an agreement between the IEC and the Institute of Electrical and Electronics Engineers, Inc. (IEEE). The original IEEE Standard was submitted to the IEC for consideration under the agreement, and the resulting IEC/IEEE Dual Logo International Standard has been published in accordance with the ISO/IEC Directives.

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. The IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and serve without compensation. While the IEEE administers the process and establishes rules to promote fairness in the consensus development process, the IEEE does not independently evaluate, test, or verify the accuracy of any of the information contained in its standards.

Use of an IEC/IEEE Dual Logo International Standard is wholly voluntary. The IEC and IEEE disclaim liability for any personal injury, property or other damage, of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, or reliance upon this, or any other IEC or IEEE Standard document.

The IEC and IEEE do not warrant or represent the accuracy or content of the material contained herein, and expressly disclaim any express or implied warranty, including any implied warranty of merchantability or fitness for a specific purpose, or that the use of the material contained herein is free from patent infringement. IEC/IEEE Dual Logo International Standards documents are supplied "AS IS".

The existence of an IEC/IEEE Dual Logo International 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 IEC/IEEE Dual Logo International 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.

Every IEEE Standard is subjected to review at least every five years for revision or reaffirmation. When a document is more than five years old and has not been reaffirmed, 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 publishing and making this document available, the IEC and IEEE are not suggesting or rendering professional or other services for, or on behalf of, any person or entity. Neither the IEC nor IEEE is undertaking to perform any duty owed by any other person or entity to another. Any person utilizing this, and any other IEC/IEEE Dual Logo International Standards or IEEE Standards document, should rely upon the advice of a competent professional in determining the exercise of reasonable care in any given circumstances.

Interpretations – Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of concerned interests, it is important to ensure that any interpretation has also received 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 interpretation requests except in those cases where the matter has previously received formal consideration.

Comments for revision of IEC/IEEE Dual Logo International Standards are welcome from any interested party, regardless of membership affiliation with the IEC or IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Comments on standards and requests for interpretations should be addressed to:

Secretary, IEEE-SA Standards Board, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, USA and/or General Secretary, IEC, 3, rue de Varembe, PO Box 131, 1211 Geneva 20, Switzerland.

Authorization to photocopy portions of any individual standard for internal or personal use is granted by the Institute of Electrical and Electronics Engineers, Inc., provided that the appropriate fee is paid to Copyright Clearance Center. To arrange for payment of licensing fee, 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.

NOTE – 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 with respect to the existence or validity of any patent rights in connection therewith. The IEEE shall not be responsible for identifying patents for which a license may be required by an IEEE standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

Precision Clock Synchronization Protocol for Networked Measurement and Control Systems

Sponsor

**TC9—Technical Committee on Sensor Technology
of the
IEEE Instrumentation and Measurement Society**

Approved 12 September 2002

IEEE-SA Standards Board

Abstract: A protocol to synchronize independent clocks running on separate nodes of a distributed measurement and control system to a high degree of accuracy and precision is specified. The protocol is independent of the networking technology, and the system topology is self-configuring.

Keywords: clocks, distributed system, master clock, measurement and control systems, real-time clock, synchronized clocks

IEEE Introduction

The objective of this standard is to specify a protocol to synchronize independent clocks running on separate nodes of a distributed measurement and control system to a high degree of accuracy and precision. The clocks communicate with each other over a communication network. In its basic form, this protocol is intended to be administration free. The protocol generates a master slave relationship among the clocks in the system. Within a given subnet of a network, there will be a single master clock. All clocks ultimately derive their time from a clock known as the grandmaster clock. The communication path between any clock and its grandmaster clock is part of a minimum spanning tree.

Patent notice

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 with respect to the existence or validity of any patent rights in connection therewith. The IEEE shall not be responsible for identifying patents for which a license may be required by an IEEE standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention. A patent holder has filed a statement of assurance that it will grant licenses under these rights without compensation or under reasonable rates and nondiscriminatory, reasonable terms and conditions to all applicants desiring to obtain such licenses. The IEEE makes no representation as to the reasonableness of rates and/or terms and conditions of the license agreements offered by patent holders. Further information may be obtained from the IEEE Standards Department.

Information on the patent declaration can be found on the IEC website under the Technical work heading (see Patent declarations).

PRECISION CLOCK SYNCHRONIZATION PROTOCOL FOR NETWORKED MEASUREMENT AND CONTROL SYSTEMS

1. Overview

This standard is divided into nine clauses. They are as follows:

Clause	Purpose
1	Provides the scope and benefits of this standard
2	Lists references to other standards that are referenced by this standard
3	Provides definitions that are either not found in other standards or have been modified for use with this standard
4	Provides conventions for the notation used in this standard
5	Defines the datatypes used in this standard
6	Provides an overview of the protocol specified by the standard
7	Defines the precision time protocol (PTP)
8	Defines the format of messages passed between participating clocks
9	Defines requirements for conformance

Annexes are provided as follows:

Annex	Purpose
A	Using the PTP
B	Defines time scales and epochs in PTP
C	Defines subdomain_name to address mappings
D	Defines the Ethernet implementation of PTP
E	Bibliography

Annexes defining communication-medium-specific implementation details for additional network implementations are expected to be provided in future revisions of this standard.

1.1 Scope

This standard defines a protocol enabling precise synchronization of clocks in measurement and control systems implemented with technologies such as network communication, local computing, and distributed objects. The protocol will be applicable to systems communicating by local area networks supporting multi-cast messaging including, but not limited to, Ethernet. The protocol will enable heterogeneous systems that include clocks of various inherent precision, resolution, and stability to synchronize. The protocol will support systemwide synchronization accuracy in the submicrosecond range with minimal network and local clock computing resources. The default behavior of the protocol will allow simple systems to be installed and operated without requiring the administrative attention of users.

1.2 Purpose

Measurement and control applications are increasingly using distributed system technologies such as network communication, local computing, and distributed objects. Many of these applications will be enhanced by having an accurate systemwide sense of time achieved by having local clocks in each sensor, actuator, or other system device. Without a standardized protocol for synchronizing these clocks, it is unlikely that the benefits will be realized in the multivendor system component market. Existing protocols for clock synchronization are not optimum for these applications. For example, Network Time Protocol (NTP) targets large distributed computing systems with millisecond synchronization requirements. The protocol proposed in this standard specifically addresses the needs of measurement and control systems:

- Spatially localized
- Microsecond to submicrosecond accuracy
- Administration free
- Accessible for both high-end devices and low-cost, low-end devices

2. References

This standard shall be used in conjunction with the standards listed in this clause. When the following standards are superseded by an approved revision, the revision shall apply.

IEEE Std 802.3™-2002, Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications.¹

ISO 8601:2000, Data elements and interchange formats—Information interchange—Representation of dates and times.²

ISO/IEC 8859-1:1998, Information technology—8-bit single-byte coded graphic character sets—Part 1: Latin alphabet No. 1

¹IEEE Publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, USA and <<http://standards.ieee.org/sds/index.html>>.

²ISO publications are available from the International Organization for Standardization, <<http://www.iso.ch/iso/en/prods-services/ISOstore/store.html>>.

3. Definitions

This clause defines terms that have specific meanings in the context of the PTP.

3.1 boundary clock: A clock with more than a single PTP port, with each PTP port providing access to a separate PTP communication path.

3.2 clock: A node that is capable of providing a measurement of the passage of time since a defined epoch. In this standard, the term *clock* is interpreted to mean either an ordinary clock or a PTP port of a boundary clock unless otherwise stated or obvious from the context.

3.3 clock timestamp point: A point in the network protocol stack of a clock at which timestamps are generated for Sync and Delay_Req messages sent or received by the clock. This point is defined for both the inbound and outbound paths in the protocol stack.

3.4 direct communication: A communication of PTP information between two PTP clocks with no intervening boundary clock.

3.5 epoch: The reference time defining the origin of a time scale.

3.6 event: An abstraction of the mechanism by which signals or conditions are generated and represented.

3.7 external synchronization: The process of bringing two clocks into a synchronized state by means other than the use of the PTP. *See also:* **synchronized clocks.**

3.8 grandmaster clock: Within a PTP subdomain, a PTP clock that is the ultimate source of time for clock synchronization using the PTP protocol.

3.9 interface definition language (IDL): A programming-language-independent method of specifying operation syntax [B1].³

3.10 master clock: In the context of a single PTP communication path, a clock that when viewed from the path appears to be the source of time to which all other clocks synchronize.

3.11 message timestamp point: A distinguished feature of Sync and Delay_Req messages serving as a reference point in these messages. A timestamp is defined by the instant a message timestamp point passes the clock timestamp point in a protocol stack.

3.12 network: A communication mechanism for passing PTP messages among multiple clocks.

3.13 node: A device that can issue or receive PTP communications on a network.

3.14 ordinary clock: A PTP clock with a single PTP port.

3.15 ordinary communication: A communication of non-PTP information between two nodes.

3.16 port number: An index identifying a specific PTP port on a PTP clock.

3.17 preferred master clock set: A set of clocks that will be favored over those not so designated in the selection of the grandmaster clock within a subdomain.

³The numbers in brackets correspond to those of the bibliography in Annex E.

3.18 precision time protocol (PTP): The protocol defined by this standard. As an adjective, it indicates that the modified noun is specified in or interpreted in the context of this standard.

3.19 PTP clock: A clock that participates in the PTP protocol.

3.20 PTP communication: Information used in the operation of the PTP protocol, transmitted in a PTP message over a PTP communication path.

3.21 PTP communication path: A segment of a network enabling direct communication between two PTP clocks.

3.22 PTP domain: A collection of one or more PTP subdomains.

3.23 PTP message: One of the five designated messages types defined in this standard: Sync, Delay_Req, Follow-up, Delay_Resp, and Management.

3.24 PTP multicast communication: A PTP message sent from any PTP port and received and processed by all PTP ports on the same PTP communication path. PTP multicast communications are not automatically forwarded to other PTP communication paths by routers or other similar network components.

3.25 PTP node: A node that issues any message that will be received by a second node resulting in any change in state in the second node that influences any aspect of the PTP protocol.

3.26 PTP port: The logical access point to a PTP clock for PTP communications on a single PTP communication path.

3.27 PTP subdomain: A logical grouping of PTP clocks that synchronize to each other using the PTP protocol but that are not necessarily synchronized to PTP clocks in another PTP subdomain.

NOTE—It should be emphasized that this grouping is logical. Clocks sharing a PTP communication path may or may not be in the same PTP subdomain, and clocks in the same PTP subdomain may or may not share a PTP communication path.

3.28 state transition diagram: A graphical means of expressing the allowed states of an object and the allowed transitions from one state to another (see 4.3).

3.29 subnet: A subset of a network. If a network contains devices whose function is to pass messages and these devices do not pass PTP non-management messages, then the network can be partitioned into regions in such a way that:

- No two nodes in a region communicate via one of these devices, and
- All communication between regions is via one or more of these devices.

Each such region is a subnet. If a network does not contain any such devices, it is a subnet.

3.30 synchronized clocks: Two clocks are synchronized to a specified uncertainty if they have the same epoch, and measurements of any time interval by both clocks differ by no more than the specified uncertainty. The timestamps generated by two synchronized clocks for the same event will differ by no more than the specified uncertainty.

3.31 timeout: A mechanism for terminating requested activity, at least from the requester's perspective, that does not complete within the time specified.

3.32 universally unique identifier (UUID): An identifier that has a unique value within some defined universe. For purposes of this standard, the universe consists of all possible PTP artifacts having a UUID unless otherwise stated.

4. Conventions

The specifications of this clause shall apply to all artifacts defined in this standard.

4.1 Descriptive syntax

The syntax conventions specified in the following subclauses are used in this standard.

4.1.1 Lexical form syntax

A lexical form refers to:

- A name
- A datatype

The conventions illustrated in the following list regarding lexical forms shall be used:

- *Type names:* e.g., TimeRepresentation (no word separation, initial letter of each word capitalized)
- *Enumeration members and global constants:* e.g., PTP_SYNC_MESSAGE (underscore word separation, all letters capitalized)
- *Fields of structures or messages:* e.g., seconds, specialElement (no word separation, initial word not capitalized, initial letter capitalization on subsequent words)
- *Data set members and all other variables:* new_master (underscore word separation, no letters capitalized)
- *<local name for something>:* text enclosed in angle brackets, < >, is used where the standard needs to refer to something whose syntax or lexical form is dependent on the local implementation and language.

When a lexical form appears in text, as opposed to in a signature, 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.

4.2 Word usage

4.2.1 Shall

The word *shall*, equivalent to *is required to*, is used to indicate mandatory requirements, strictly to be followed in order to conform to the standard and from which no deviation is permitted.

4.2.2 Recommended

The word *recommended* is used to indicate flexibility of choice with a strong preference alternative.

4.2.3 Must

The use of the word *must* is deprecated and shall not be used when stating mandatory requirements. The word *must* is only used to describe unavoidable situations.

4.2.4 Should

The word *should*, equivalent to *is recommended that*, is used to indicate:

- Among several possibilities, one is recommended as particularly suitable, without mentioning or excluding others.
- A certain course of action is preferred but not required.
- In the negative form, a certain course of action is deprecated but not prohibited.

4.2.5 May

The word *may*, equivalent to *is permitted*, is used to indicate a course of action permissible within the limits of the standard.

4.2.6 Can

The word *can*, equivalent to *is able to*, is used to indicate possibility and capability, whether material or physical.

4.3 Behavioral specification notation

State transition diagrams are used to specify behavioral characteristics as illustrated in Figure 1. Each state transition diagram is composed of the following components:

- *Named boxes*, representing states
- *Directed arrows*, indicating transitions from one state to the next

Each transition is labeled with:

- The enabling event or predicate label for a transition
- The transition action label for a transition

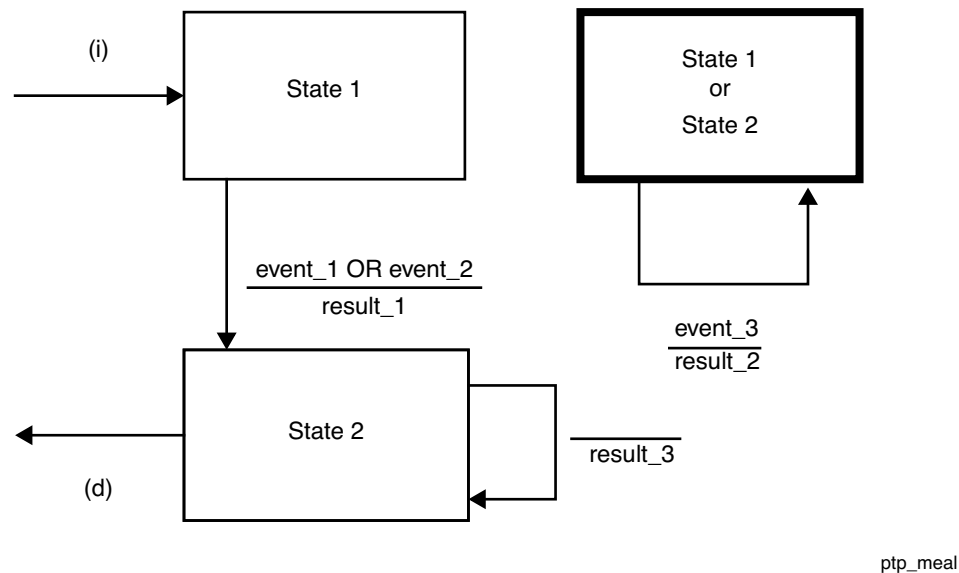


Figure 1 – Mealy state transition diagram

The notation used describes state transition diagrams using the Mealy style, where actions are associated with the transition from one state to another.

Events—for example, event_1, event_2, and event_3—identify the inputs to the state machine. They can be operation requests and responses, or internal occurrences such as timer expirations.

Predicates—for example, event_1 OR event_2—identify enabling conditions for transitions. The first predicate encountered, evaluating from left to right, that is TRUE selects the transition to execute and therefore the next state.

Transition actions—for example, result_1—are the actions that are executed before transitioning to the next state.

The next state identifies the state for the state machine after the selected transition action completes. As the transition to the next state occurs, the value of the current state changes.

A bold line for a state box indicates that the box represents multiple states. Any transition shown that begins and terminates in such a state box indicates that there has been no change in state.

Transitions—for example, the transition resulting in result_3—that have no indicated enabling conditions, occur via unspecified mechanisms. Unless otherwise stated, in PTP these mechanisms are implementation-specific and outside the scope of the standard.

A transition into a state machine—for example, (i)—is indicated by a transition arrow that has no source state. A transition out of a state machine—for example, (d)—is indicated by a transition arrow with no destination state.

Example: As a result of either event_1 or event_2 becoming TRUE, State 1 is replaced with the value of the next state. In this example, the next state is State 2, which is specified as the name of the state box that is the target of the transition arrow. Before the transition, result_1 occurs. event_3 can occur in either State 1 or State 2. The state is unchanged but an action, result_2, occurs as the result of event_3.

5. Datatypes in a PTP system

For every datatype defined in a PTP system there shall be a default value designated. Quantities declared as having a type shall be instantiated with the default value unless otherwise specified.

The datatypes specified for the various PTP variables and message fields define logical properties, such as rollover for integers, necessary for correct operation of the protocol or interpretation of PTP message content.

The definition of data of types defined in this standard shall be correctly represented in the local system after such data has been communicated over the network. In any PTP implementation, these datatypes shall be mapped to implementation language-dependent primitives.

Implementations are free to use any internal representation preserving these characteristics provided that the internal representation shall not change the semantics of any quantity visible via PTP communications.

The mapping of datatypes into their on-the-wire format is network specific. For each network used in a PTP system, an annex to this standard shall be required that includes this mapping. Unless otherwise specified in this standard, the mapping in the network specific annexes shall be used for all PTP communications.

The following normative definitions are used in this clause.

Definition	Meaning
IDL: typedef <datatype> <datatypeArray>[];	An implementation of PTP Array of the datatype: <datatype>
IDL: typedef <datatype1> <datatype2>;	Where <datatype1> is a primitive or previously defined PTP datatype, shall mean that <datatype2> is a synonym for <datatype1>

5.1 Primitive datatypes

The datatypes defined in Table 1 shall be defined in all PTP systems. All other datatypes shall be derived from these primitive types.

Table 1—Primitive PTP datatypes

Datatype	Default value	Definition
Boolean	FALSE	TRUE or FALSE
Integer8	0	8-bit signed integer
UInteger8	0	8-bit unsigned integer
Integer16	0	16-bit signed integer
UInteger16	0	16-bit unsigned integer
Integer32	0	32-bit signed integer
UInteger32	0	32-bit unsigned integer
Integer64	0	64-bit signed integer
UInteger64	0	64-bit unsigned integer
Octet	All bits set to 0	8-bit field not interpreted as a number

5.2 Derived datatypes

The datatypes in this clause are based on the primitive datatypes. Only the primitive or derived datatypes defined in this standard are allowed as part of a PTP message.

5.2.1 Specification of arrays of primitive types

All arrays defined by this standard shall be of fixed length. Any of the following array types may be used provided the length is specified. Unless otherwise specified in this standard or network specific annexes, all arrays shall be marshaled into an on-the-wire format with the member having the lowest numerical index first.

IDL: typedef Boolean	BooleanArray[];
IDL: typedef Integer8	Integer8Array[];
IDL: typedef UInteger8	UInteger8Array[];
IDL: typedef Integer16	Integer16Array[];
IDL: typedef UInteger16	UInteger16Array[];
IDL: typedef Integer32	Integer32Array[];
IDL: typedef UInteger32	UInteger32Array[];
IDL: typedef Integer64	Integer64Array[];
IDL: typedef UInteger64	UInteger64Array[];
IDL: typedef Octet	OctetArray[];

5.2.2 Time type specifications

The TimeRepresentation type shall be used to specify both:

- Timestamps (Time with respect to an epoch)
- Time increments

IDL: struct TimeRepresentation

```
{
    UInteger32 seconds;
    Integer32 nanoseconds;
};
```

The range of the absolute value of the nanoseconds member shall be restricted to:

$$0 \leq |\text{nanoseconds}| < 10^9$$

The sign of the nanoseconds member shall be interpreted as the sign of the entire representation.

A negative timestamp shall indicate time prior to the epoch.

A negative increment shall result for example from subtracting a timestamp A from a timestamp B , where A is an instant later in time than B .

Example: A timestamp whose seconds member value is 10 and whose nanoseconds member value is -5×10^8 represents an instant 10.5 seconds before the epoch.

5.2.3 Enumeration type specifications

An enumeration member of a PTP enumeration type shall be represented as a `UInteger8`. Unless otherwise noted, the assignment of values to the enumeration members shall start at 0 (zero) for the first member listed of an enumeration and shall increment by +1 (one) for each succeeding member. Unless otherwise stated, the default value of an enumeration member shall be 0.

Any enumeration used in a PTP system shall have members of the form `PTP_<suffix>`. All PTP enumerations shall be unique.

5.2.3.1 Enumeration semantics

Enumerations defined in this standard always appear in message formats as `UInteger8` types. For each enumeration, the value of the `UInteger8` has been assigned for each member of the enumeration. Enumerations have no print form specified. Implementations shall not include any operational dependency on the particular textual description used in the definition of the enumeration. Any textual description of the semantics associated with the enumeration value shall not appear in the signature of any operation, or in the format of any PTP message, except for informational purposes.

5.2.3.2 Enumeration summary list

Most enumerations are defined in the clause where they are first used. This clause provides a list of all enumerations defined in this standard along with a reference to the defining clause.

Enumeration name	Reference clause
CommunicationId	6.2.4.1
ControlField	8.2.9
ManagementMessage	7.12
PTPState	7.3.1

6. PTP Clock synchronization model

6.1 Overview

6.1.1 General overview

The PTP standard specifies a clock synchronization protocol. This protocol is applicable to distributed systems consisting of one or more nodes, communicating over some set of communication media. Nodes are modeled as containing a real-time clock that may be used by applications within the node for various purposes, such as generating timestamps for data or ordering events managed by the node. The PTP protocol provides a mechanism for synchronizing the clocks of participating nodes to a high degree of accuracy. Guidelines for the design and use of systems supporting PTP are given in Annex A.

The standard specifies the following:

- The PTP protocol
- The node and communication properties necessary to support PTP

A PTP system is a distributed system consisting of ordinary clocks, possibly boundary clocks, and possibly administrative nodes. A boundary clock is a clock with a clock port for each of two or more distinct PTP communication paths. There are two aspects of the PTP protocol: the synchronization aspect and the administrative aspect.

Externally, all ordinary clocks appear identical in all aspects of the protocol. Furthermore, each PTP port of a boundary clock externally appears to be an ordinary clock in terms of the protocol. An administrative node is not required to implement the synchronization aspect of the protocol unless it is also a clock node.

6.1.2 Operational overview

A PTP communication path supports the direct communication among the PTP ports of a set of ordinary clocks and the PTP ports of any boundary clocks accessing the communication path. The general behavior of the clocks accessing a given communication path follows. In this discussion, the term *ports* will include the ports of any boundary clocks attached to the communication path.

A clock containing one of these ports will be selected as the *master clock* for the path. This selection is made by each port examining information contained in special messages termed Sync messages. A Sync message is sent periodically by any port associated with a clock claiming to be the master clock. All ports use the same algorithm, termed the *best master clock* algorithm. If a port of a master clock receives a Sync message from a *better* clock then that clock will cease to claim to be a master and the receiving port will assume the status of a *slave*. Likewise if a clock with a port acting as a slave determines that it would make a better master than the current master clock, it assumes the status of *master* and begins to send Sync messages.

To provide more orderly behavior when a clock comes on line, a clock will listen for Sync messages from a master clock for a system-specified time. If no Sync message is received within this time, the clock will assume it is the master clock until such time as a better clock appears. An additional mechanism to support more orderly reconfiguration of systems when clocks are added or deleted, clock characteristics change or connection topology changes is embodied in the PTP_PRE_MASTER state. In this state a clock port behaves exactly as it would if it were in the PTP_MASTER state except that it does not place certain classes of messages on the port communication path. A clock port remains in this premaster state long enough to allow changes to occur at points in the system between the local clock and possible master clocks visible from the port.

Sync messages from a port of the master also contain an estimate of the time the Sync message will be placed on the PTP communication path. This estimate is based on the time kept by the local clock of the master.

Master clock ports may also send Follow_Up messages. A Follow_Up message is always associated with a specific Sync message and contains a more precise estimate of the time the Sync message was placed on the PTP communication path.

Slave ports use the information contained in the Sync or Follow_Up messages to correct their local clock in such a way as to synchronize the local clock to the clock of the master.

This correction may be improved by accounting for the propagation delay experienced by messages in traveling from the master to the slave. This additional correction may be estimated by reversing the previous

process. That is, the slave sends a message termed a Delay_Req message to the master. No Follow_Up message is used for this procedure. The master then reports the time of receipt back to the slave in a Delay_Resp message. This process is normally exercised infrequently to reduce network traffic.

In addition to issuing Sync messages, a master clock, and only a master clock, may also provide an external timing signal. These external timing signals shall be issued on the seconds' transition of the local clock. The external timing signal is transmitted only to slaves in the same subdomain, preferably in the same PTP communication path, and is transmitted on a medium other than the medium communicating the Sync messages. The behavior of systems where external timing signals are propagated to clocks on PTP communication paths not associated with a port on the issuing master clock is beyond the scope of this standard.

Boundary clocks serve to implement a time distribution tree (or set of trees). A boundary clock is required wherever there is a change of communication technology or a network element which blocks Sync, Follow_Up, Delay_Req, or Delay_Response messages. It is recommended that a boundary clock be used wherever there is a network element that inserts significant delay fluctuation. The boundary clock is a slave to the path containing the best clock it can see and asserts that it is the master for all other accessed paths. Since boundary clocks like ordinary clocks contain an internal clock, it is possible that this internal clock is the best clock it can see. In this case the boundary clock itself will be the master in all accessed paths. Thus in a system with boundary clocks, there may be several masters, one for each communication path. The best clock of all is designated the grandmaster. Sufficient information is contained in each Sync message to allow the best master clock algorithm to produce the correct overall system configuration and behavior.

6.1.3 Assumptions

The PTP standard makes several assumptions about the environment in which it operates. The following assumptions must be met to ensure correct operation of the protocol:

- The network must support multicast communication (see 6.2.2.2).
- It must be possible to prevent multicast messages from propagating beyond a subnet.
- Each clock implementing the protocol must meet the performance requirements of 7.10.
- A clock's stated properties, including its stratum (see 6.2.4.3) and identifier (see 6.2.4.5), must accurately describe the clock.

The following assumptions must be met to achieve optimal clock synchronization performance:

- Network delay between master and slave on a subnet must be symmetric (see 7.8.1.2).
- A clock may contain asymmetric delays in its timestamping mechanism or protocol path. If these asymmetries are not negligible, they must be correctly accounted for (see 6.2.4.9).
- Network delay between master and slave on a subnet must be constant over a PTP_DELAY_REQ_INTERVAL.
- Boundary clocks must be used to synchronize across subnets (see 6.2.1).
- Delay fluctuation due to network components and due to the protocol stack within clocks must be reduced by two techniques:
 - 1) The timestamps used in PTP are generated as close to the physical layer as practical for a given clock implementation. In cases where the most accurate timestamps can be generated only after a message has actually been transmitted, the actual value is communicated in the Follow_Up message from the master clock (see 8.4.1 and Annex A). See [B4], [B7], [B8] for mechanisms to aid in generating these timestamps.
 - 2) Remaining delay fluctuation introduced by the protocol stack and by network components not isolated by a boundary clock can be reduced by averaging (see A.5.2). The averaging algorithms are outside the scope of this standard.

- The computing power of clocks implementing the protocol must be great enough, and the number of clocks per subnet must be small enough, to meet the constraints of 7.11.
- The inherent stability of a clock's oscillator must be adequate (see A.4 and A.5.4).

6.2 PTP systems

A diagram of a typical distributed system of nodes containing clocks is shown in Figure 2.

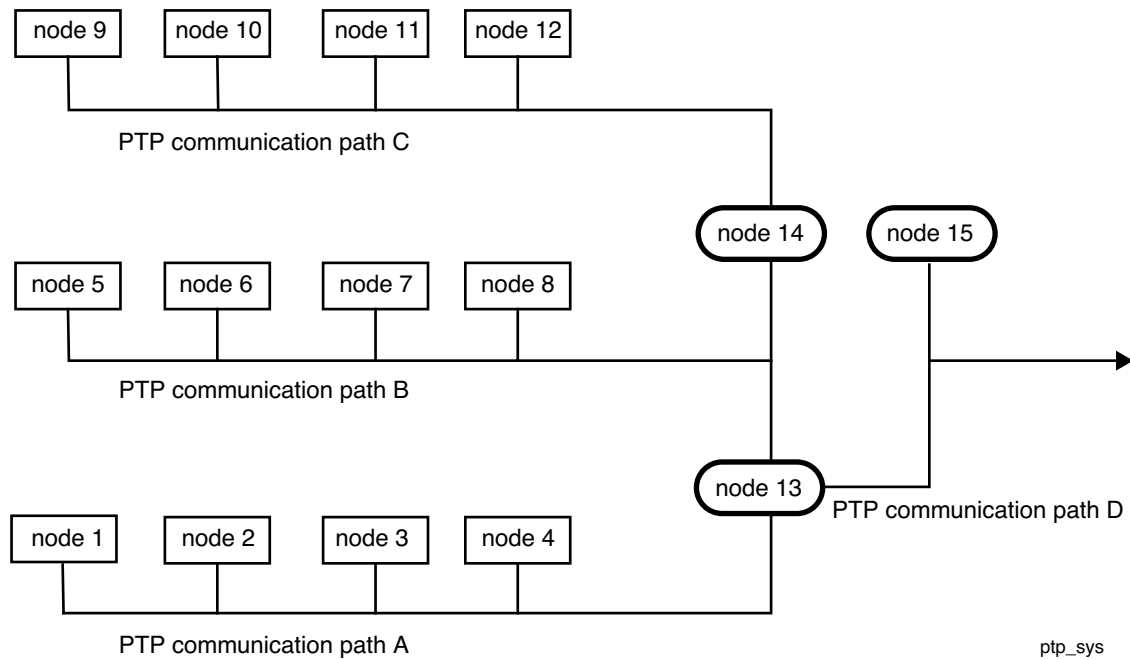


Figure 2—Typical system of synchronized clocks

In Figure 2, the rectangles represent nodes each containing a clock; rounded rectangles represent nodes each containing a boundary clock. These clocks communicate PTP messages with each other via the communication paths illustrated by the solid lines in the figure.

6.2.1 Clock types

There are two types of clocks in a PTP system:

- *Ordinary clocks:* Ordinary clocks shall communicate with other clocks over a single communication path. For example in Figure 2, clocks in nodes 1 through 4 communicate over a single communication path, path A.
- *Boundary clocks:* Boundary clocks, such as node 13 in Figure 2, may communicate with multiple sets of clocks (for example, node set 1, 2, 3, and 4 via A, set 5, 6, 7, 8, and 14 via B, and 15 via D). Each set may contain a mixture of ordinary and boundary clocks. A boundary clock shall communicate with each set using a distinct communication path. Communication between clocks (for example, 9 and 5) in these disjoint sets shall only occur via the boundary clock mechanisms specified in the protocol. Boundary clock to boundary clock communication is illustrated in Figure 2 by path B between boundary clocks 13 and 14 and path D between 13 and 15.

The logical communication access point of a clock to a PTP communication path is termed a PTP port. Ordinary clocks have a single PTP port while boundary clocks have two or more PTP ports.

6.2.2 PTP communications

6.2.2.1 PTP communication topology

The operation of the PTP protocol generates a topology of PTP communication paths forming an acyclic graph structure. That is, there will be no alternate PTP communication paths between any pair of PTP clocks. An example of prohibited path topology is the cyclic path that includes nodes 13, 14, and 15 in Figure 3. Again, rectangles represent nodes each containing a clock; rounded rectangles represent nodes each containing a boundary clock.

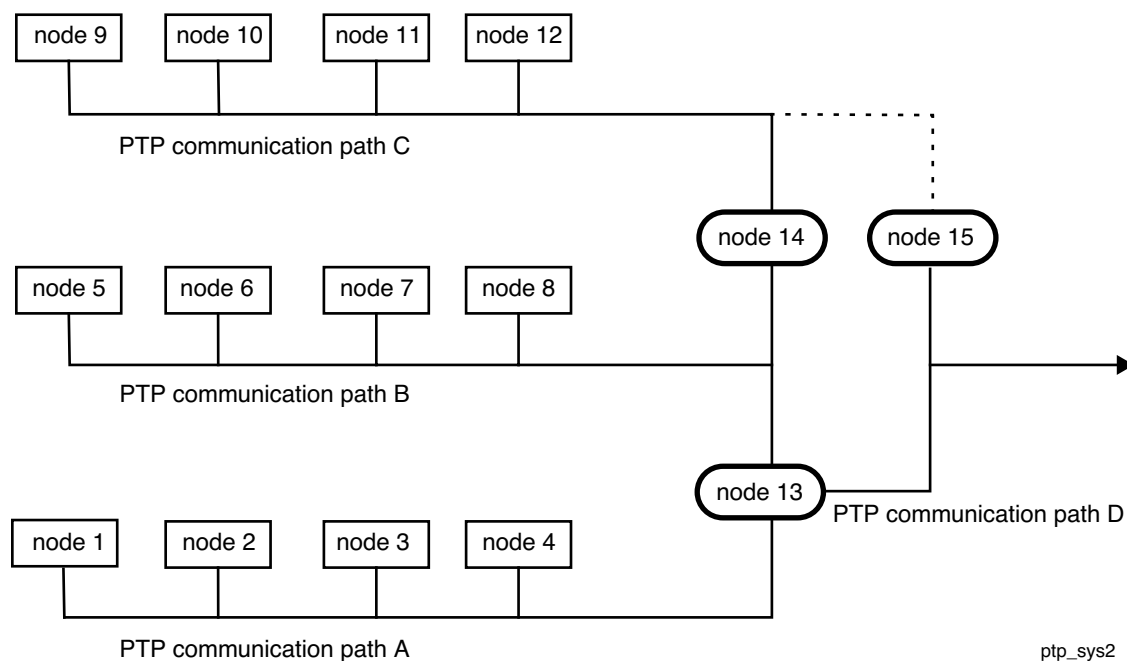


Figure 3—Prohibited topology

The PTP protocol detects such cyclic PTP communication paths. The PTP protocol changes cyclic graphs into acyclic graphs by changing the state of ports on the involved boundary clocks. In this example, the protocol has disabled the communication between nodes 14 and 15.⁴ These state changes result in actual communication being over an acyclic topology even though the physical connection topology is cyclic.

The operation of the PTP protocol to produce this topology follows techniques described by Perlman [B2].

There is no requirement that all of the PTP communication paths use the same underlying communication media or technology. However, if different communication media or technologies are present, a set of clocks communicating with each other using a given communication medium or technology shall be separated from a set using a different communication medium or technology by a boundary clock.

⁴More precisely, it has put the appropriate port on node 14 or 15 into the PTP_PASSIVE state. Which node is put into that state depends on details of the network configuration and properties of the clocks and cannot be determined in this simplified example.

Boundary clocks separate PTP subdomains into distinct PTP communication paths. Communications involving boundary clocks shall be restricted as follows.

- PTP messages of type Sync, Delay_Req, Follow_Up, and Delay_Resp shall not be propagated between PTP communication paths separated by a boundary clock. This restriction applies to boundary clocks and any communication technology specific components enabling ordinary communication between devices on the separate PTP communication paths.
- PTP Management messages shall be communicated between PTP communication paths separated by a boundary clock. These PTP Management messages shall be communicated by the boundary clocks but shall not be communicated by any communication technology specific components enabling ordinary communication between devices on the separate PTP communication paths. The absolute value of the boundaryHops field of the management message shall be decremented by 1 by the boundary clock before retransmitting the message. If the boundaryHops field value is zero, the management message shall not be retransmitted. If the boundaryHops field value is positive the message shall be retransmitted only via boundary clock ports not in the PTP_INITIALIZING, PTP_FAULTY, or PTP_DISABLED states. If the boundaryHops field value is negative the message shall be retransmitted only via boundary clock ports not in the PTP_INITIALIZING state. In no case shall the message be retransmitted on the port on which it was received. Ordinary clocks shall not retransmit management messages.

NOTE—Management messages will be retransmitted by clock ports in the PTP_PASSIVE state and may thus be delivered to some clocks more than once (see Figure 8). Management applications should take this fact into account.

6.2.2.2 PTP communication protocol

All PTP non-management messages shall be communicated as multicast communications. A multicast communication results in a single transmission of a PTP non-management message being received by all clocks and only those clocks sharing a PTP communication path. Management messages may be communicated using either a multicast or a point-to-point communication. All clocks shall be capable of receiving Management messages via multicast communication.

It is recommended that all implementations use a true multicast implementation as the actual communication protocol for PTP multicast communications. The behavior of systems building multicast out of point-to-point communications will:

- Not scale well as the number of clocks increases
- Require more complex logic in clocks to implement the requirements of the protocol

PTP communications take place in PTP domains characterized by a subdomain name and a port category (see 6.2.3, 6.2.5.1, and 6.2.5.3). Each message contains this information and each clock maintains values of these quantities that pertain to it. A clock shall accept and process a message if and only if the subdomain name and port category associated with the message are identical to the corresponding values pertaining to the receiving clock.

6.2.2.3 PTP message model, message timestamp point, and clock timestamp point

PTP communications are implemented with PTP messages. In all communication technologies, PTP communication paths are modeled to contain distinguishing locations known as the *clock timestamp points*, and certain messages are modeled to contain a distinguishing feature known as the *message timestamp point* (see 6.2.4.9). The precise definition of these terms shall be defined in this standard or its revisions for each network protocol supporting PTP. All PTP time stamps shall refer to the time at which the message timestamp point of either a Sync or a Delay_Req message passes a clock timestamp point. This time shall be corrected for latency (see 7.8.1.3).

6.2.3 PTP domains

A PTP domain, hereafter referred to as simply a *domain*, is a set of one or more PTP subdomains. A PTP subdomain, hereafter referred to as simply a *subdomain*, consists of one or more clocks communicating with each other, as defined by the PTP protocol, in order to synchronize these clocks. Except for certain PTP Management messages, the nodes in one subdomain shall not communicate with the nodes in another subdomain for purposes related to PTP. Multiple subdomains may be used to create independent collections of synchronized clocks sharing common PTP communication paths. The clocks within a subdomain will synchronize with each other, but there is no requirement that the clocks in one subdomain be synchronized with the clocks in another subdomain. The purpose of subdomains is to allow users to create localized sets of clocks, typically sharing a single communication path, that maintain a time base independent of the rest of the domain.

There are four subdomains defined by this standard. These subdomains shall be:

- DefaultPTPdomain: This shall be the default subdomain if a domain consists of only a single subdomain. It may be a subdomain in a domain consisting of one or more subdomains. Unless modified by a PTP Management message or clock-specific means outside the scope of this standard (e.g., a switch on the clock), the clock shall use this subdomain.
- AlternatePTPdomain1: This may be a subdomain in a domain consisting of one or more subdomains.
- AlternatePTPdomain2: This may be a subdomain in a domain consisting of one or more subdomains.
- AlternatePTPdomain3: This may be a subdomain in a domain consisting of one or more subdomains.

A domain may contain subdomains, termed *optional subdomains*, other than those given previously. Such optional subdomains shall implement all other requirements of this standard. The administration of domains containing optional subdomains is beyond the scope of this standard.

Examples:

- In Figure 2, if boundary clock 14 and its connections are removed, there are two PTP domains. One consists of the set 9, 10, 11, and 12, while the other consists of the set 1, 2, 3, 4, 5, 6, 7, 8, 13, and 15.
- If nodes 14 and 15 are present as shown, then there is a single PTP domain consisting of all nodes.
- If nodes 14 and 15 are present and nodes 1, 2, and 3 are designated as belonging to AlternatePTPdomain1 while the remaining nodes belong to the DefaultPTPdomain, then the system consists of two subdomains.

There is no requirement that a subdomain be implemented in a single communication medium or technology. However, by 6.2.2, if more than a single communication medium or technology exists within a subdomain, the clocks in the subdomain must consist of two or more disjoint sets, each with its own PTP communication path, communicating via one or more boundary clocks.

Boundary clocks shall be capable of implementing all aspects of the PTP protocol for the DefaultPTPdomain. Boundary clocks may in addition be capable of implementing the PTP protocol for additional PTP subdomains.

Within a subdomain a master clock, in addition to issuing a Sync message, may issue an external timing signal to any or all of clocks within the subdomain having ports in the PTP_SLAVE state, *slave clocks*. This external timing signal shall be transmitted on a medium other than the medium communicating the Sync messages. This external timing signal shall be issued on the seconds' transition of the local clock. Slave clocks receiving the external timing signals may use this information to synchronize to the master. It is recommended that slave clocks be configured to accept external timing signals only from a master clock sharing the PTP communication path with the slave clock. Because the external timing signal uses communication media other than the media used for PTP communications, such configuration is outside of the scope of this standard. The behavior of systems allowing the external timing signal to be delivered to

clocks outside of the subdomain or not on the same PTP communication path of the issuing master clock is outside the scope of this standard.

6.2.4 Clock properties

The clocks in a PTP system are characterized as defined in the following subclauses.

6.2.4.1 UUID

Each clock and each port in a PTP system shall have an identifier, called a UUID. A port's identifier is termed the *Port-UUID*, and a clock's identifier is termed the *Clock-UUID*. UUIDs shall be implemented as three fields, as follows:

- *communication_technology_field*: This field shall specify the communication media and technology used in PTP communications of the port. The values shall be selected from the *CommunicationId* enumeration.
- *uuid_field*: This field shall consist of an *OctetArray* of length *PTP_UUID_LENGTH*. The value of this field for a given clock shall be unique within the communication technology defined by the *communication_technology_field* value. The value consisting of all zeros for this field shall be reserved in all technologies as a special value used by the PTP protocol. Technologies for which the length of the unique identifier is less than *PTP_UUID_LENGTH* shall left-justify the identifier in the field and pad the unused octets with zeroes. That is, the most significant octet of *uuid_field* shall be significant in all technologies. UUIDs longer than *PTP_UUID_LENGTH* shall not be permitted.
- *port_id_field*: The datatype of this field shall be *UInteger16*. The value of this field for a port on an ordinary clock shall be 1. The value of this field for the *N* ports on a boundary clock shall have the values 1, 2, ..., *N*. The value of this field shall be zero when used as part of the *Clock-UUID*. Note that the type of this field limits the number of ports on a boundary clock to 65 535.

There is no requirement on the in-memory layout of these fields. Their on-the-wire layout for Ethernet is given in D.1, and the algorithm for comparing two UUIDs is given later in this clause.

The default value of any UUID shall have the following field values:

- *communication_technology_field*: *PTP_DEFAULT*
- *uuid_field*: Each octet shall have all bits zero.
- *port_id_field*: Zero

The three fields of a Port-UUID shall be selected such that no two ports in a PTP domain have the same Port-UUID.

For an ordinary clock, the Clock-UUID shall be the Port-UUID of its single PTP port, but with a *port_id_field* value of zero.

For a boundary clock *C*, the Clock-UUID shall be constructed as follows:

- Find the smallest Port-UUID of *C*'s ports, using the comparison algorithm of this clause (note that these ports may have different *communication_technology_fields*). Call this smallest Port-UUID *S*.
- *C*'s Clock-UUID shall consist of:
 - 1) A *communication_technology_field* equal to that of *S*
 - 2) A *uuid_field* equal to that of *S*
 - 3) A *port_id_field* of zero

It is recommended that all clocks in a PTP domain using a given technology use the same specification for determining the Clock-UUID. The behavior of systems where this recommendation is not followed is outside the scope of this standard.

It is expected that the organization responsible for standardizing each network technology will provide a method for choosing a uuid_field meeting the requirements of this clause. The method for Ethernet is given in D.2. If a UUID meeting the requirements of this clause is not specified by the organization responsible for standardizing a network technology, then the specification of the uuid_field shall be as specified in D.2, and the communication_technology_field shall be PTP_ETHER.

IDL: enumeration CommunicationId;

Table 2—CommunicationId enumeration

Enumeration	Value	Communication protocol
PTP_CLOSED	0	Closed system outside the scope of this standard
PTP_ETHER	1	IEEE 802.3™ (Ethernet) ^a
reserved	2	Reserved
reserved	3	Reserved
PTP_FFBUS	4	FOUNDATION fieldbus
PTP_PROFIBUS	5	PROFIBUS
PTP_LON	6	LonTalk® protocol ^b
PTP_DNET	7	DeviceNet
PTP_SDS	8	SmartDistributedSystem™ ^c
PTP_CONTROLNET	9	ControlNet™ ^d
PTP_CANOPEN	10	CANopen
reserved	11 – 242	Reserved
PTP_IEEE1394	243	IEEE 1394™
PTP_IEEE802.11A	244	IEEE 802.11a™
PTP_IEEE_WIRELESS	245	IEEE 802.11b™
PTP_INFINIBAND	246	InfiniBand™ ^e
PTP_BLUETOOTH	247	Bluetooth™ wireless ^f
PTP_IEEE802.15.1	248	IEEE 802.15.1™
PTP_IEEE1451.3	249	IEEE 1451.3™
PTP_ IEEE1451.5	250	IEEE 1451.5™
PTP_USB	251	USB bus
PTP_ISA	252	ISA bus
PTP_PCI	253	PCI bus
PTP_VXI	254	VXI bus
PTP_DEFAULT	255	Default value

^aThe IEEE products referred to in Table 2 are trademarks owned by the Institute of Electrical and Electronics Engineers, Incorporated.

^bLonTalk is a registered trademark of Echelon Corporation.

^cSmartDistributedSystem is a trademark owned by Honeywell, Inc.

^dControlNet is a trademark owned by ControlNet International, Ltd.

^eInfiniBand is a trademark owned by the InfiniBand Trade Association.

^fBluetooth is a trademark owned by Bluetooth SIG, Inc.

NOTE 1 The IEEE products referred to in Table 2 are trademarks owned by the IEEE.

NOTE 2 LonTalk, SmartDistributedSystem, ControlNet, InfiniBand and BlueTooth are trade marks and are examples of suitable products available commercially. This information is given for the convenience of users of this International Standard and does not constitute an endorsement by IEC of these products.

The following algorithm shall define the ordering properties of all UUIDs:

- communication_technology_field and port_id_field values shall be treated as unsigned integers.
- uuid_fields shall be compared as follows:
 - 1) Let X and Y be two uuid_fields.
 - 2) If every octet in X is equal to the corresponding octet in Y , $X = Y$.
 - 3) Otherwise, consider the most significant position in which the octets differ, and treat the octets in that position as unsigned integers. If the octet belonging to X is smaller than the octet belonging to Y , $X < Y$; otherwise $X > Y$.

Now, given UUIDs A and B :

- If the communication_technology_field of A is less than the communication_technology_field of B , $A < B$.
- Otherwise, if the communication_technology_field of A is greater than the communication_technology_field of B , $A > B$.
- Otherwise, if the uuid_field of A is greater than the uuid_field of B , $A > B$.
- Otherwise, if the uuid_field of A is less than the uuid_field of B , $A < B$.
- Otherwise, if the port_id_field of A is greater than the port_id_field of B , $A > B$.
- Otherwise, if the port_id_field of A is less than the port_id_field of B , $A < B$.
- Otherwise, $A = B$.

6.2.4.2 Clock type

Each clock shall be typed as either an ordinary clock or a boundary clock (see 7.4.2.11).

6.2.4.3 Clock stratum

The clock stratum, or stratum number, describes one measure of the quality of a clock. Each clock shall be characterized by a stratum number to be used by the best master clock algorithm as one parameter of clock quality (see 7.4.2.4).

The interpretation and allowed values of stratum numbers shall be as defined in Table 3. The clock stratum shall have type UInteger8.

Unless a clock is specifically designed to maintain clock accuracy in the face of power loss, a power cycle on the clock shall preclude assigning stratum numbers less than 3 on powerup.

If the inherent characteristics of a clock degrade such that the clock stratum number and clock identifier designations no longer apply, the clock shall either:

- Degrade its stratum numbers and clock identifiers in such a way as to correctly specify the current clock characteristics, or
- Be placed in the PTP_FAULTY state.

If the inherent characteristics of a clock improve such that the clock stratum and clock identifier designations no longer apply, the clock may upgrade its stratum numbers and clock identifiers. Such upgrades shall remain consistent with all requirements of this standard. Improved characteristics may result from a variety of causes. For example, the clock becomes directly (not via PTP) synchronized to a primary reference clock that is traceable to a recognized standard source of time.

Table 3—Stratum number definitions

Stratum number	Specification
0	May be used temporarily for special purposes by PTP implementations to force a clock to be deemed better than other clocks in the system.
1	Designates the clock as a primary reference standard traceable to a recognized standard source of time. A stratum 1 clock may be either a boundary clock or an ordinary clock. (NOTE—GPS clocks, calibrated atomic clocks, etc. fall into this stratum). A stratum 1 clock shall not be synchronized using the PTP protocol to another clock in a PTP system.
2	Designates the clock as a secondary standard reference clock. The clock shall be: — Directly (not via PTP) synchronized to a stratum 1 clock or another source deemed to be a correct source of time for the PTP subdomain or — Previously directly synchronized to a stratum 1 clock or another source deemed to be a correct source of time for the PTP subdomain and is still providing time information consistent with this clock or source as specified by the clock_identifier associated with the clock (see 6.2.4.5).
3	The lowest possible clock_stratum value if not 1 or 2 for a clock that is capable of issuing external timing signals and possibly setting the PTP_EXT_SYNC flag to TRUE (see 8.2.10).
4	The lowest possible clock_stratum value if not 1 or 2 for a clock that does not have the capability of issuing external timing signals and therefore sets the PTP_EXT_SYNC flag to FALSE (see 8.2.10).
5–254	Reserved.
255	The default value. A clock with this stratum number shall never be the best master clock.

For boundary clocks, the clock stratum number shall be identical for each PTP port.

While not required, it is recommended that in a given PTP subdomain clocks with clock stratum numbers less than 4 be inherently at least as stable (low default variance) as any clock with a greater clock stratum number.

6.2.4.4 Preferred master clock

A clock may be administratively designated as part of a *preferred master clock set*. This creates a set of clocks that will be favored over those not so designated in the selection of master clocks within a subdomain. The purpose of this designation is to allow users to specify a clock that will remain master in the presence of disconnection or connection of other clocks.

6.2.4.5 Clock identifier

The clock_identifier shall indicate the nature and expected absolute accuracy and epoch of a given clock. The datatype of the clock_identifier shall be an OctetArray of length PTP_CODE_STRING_LENGTH. Each octet of the field shall be interpreted as an ASCII character, with the most significant octet the leading character. The interpretation and allowed values of clock identifiers shall be in accordance with Table 4. For clock identifiers of less than PTP_CODE_STRING_LENGTH characters, the clock_identifier is left justified (toward the most significant octet) in the field, with the unused octets having value \0, (hex value 0x0). Clock identifiers shall be used to establish which of several clocks with identical clock stratum numbers is selected as the best master clock.

Clock identifiers indicating that the time scale is universal coordinated time (UTC) (see B.2) may become invalid for clocks with clock stratum values of 2 or greater under the following conditions:

- The possible drift of the clock since last synchronized to a UTC source of time exceeds the given specification, or
- The time interval since last synchronized to a UTC source of time includes times when possible leap second adjustments occur (see Annex B), or
- A fault potentially degrading these specifications occurs.

The ordering of clock identifiers used in the selection of the best master clock (see 7.6) is summarized in Table 5.

Unless specifically designed to maintain clock accuracy, a power cycle on a clock shall preclude clock identifiers other than DFLT on powerup.

For boundary clocks, the clock identifier shall be the same for each PTP port.

In Table 4, the listed accuracy specifications shall be interpreted as the sum of the mean and standard deviation from the applicable time base.

Table 4—Clock identifier definitions

Clock identifier (ASCII)	Applicable to clock_stratum number	Specification
ATOM	1	Time is derived from a calibrated atomic clock maintaining a UTC time base accurate to better than 25 ns
GPS	1	Time is derived from a correctly operating GPS receiver maintaining a UTC time base accurate to better than 100 ns.
ATOM	2	The stability of the clock is such that it is accurate to within 100 ns of the UTC time base established the last time it was synchronized directly to a stratum 1 clock with clock_identifier ATOM. A power cycle may preclude this designation.
GPS	2	The stability of the clock is such that it is accurate to within 100 ns of the UTC time base established the last time it was synchronized directly to a stratum 1 clock with clock_identifier GPS. A power cycle may preclude this designation.
NTP	2	The clock shall meet one of the following specifications: — The clock is correctly and actively participating in a suite of clocks using the NTP or equivalent protocol to maintain a UTC time base accurate to better than 15 ms, or — The stability of the clock is such that it is consistent to within 50ms of the time base established the last time it was correctly and actively participating in a suite of clocks using the NTP or equivalent protocol to maintain time consistent with UTC. A power cycle may preclude this designation. Examples of protocols providing time bases and accuracies equivalent to NTP are SNTP, link to NIST time server, etc.
HAND	2 or greater	The clock has been set to the correct UTC time to accuracy better than 10 seconds by an administrative procedure and is consistent with that time except for normal drift of this clock. A power cycle may preclude this designation.
INIT	2 or greater	The clock has been set with unspecified accuracy to an arbitrary or user-defined time by an administrative procedure and is consistent with that time except for normal drift of this clock. A power cycle may preclude this designation.
DFLT	3 or greater	Applicable if none of the other clock_identifiers apply.

Table 5—Clock identifier precedence

Stratum	Clock identifier ordering
1	ATOM equivalent to GPS
2	ATOM equivalent to GPS takes precedence over NTP takes precedence over HAND takes precedence over INIT.
3 or greater	HAND takes precedence over INIT takes precedence over DFLT.

6.2.4.6 Follow_Up capability

Each clock shall be designated as either supporting or not supporting the capability of providing, for each Sync message issued, a Follow_Up message conveying a more accurate value for the originTimestamp, namely the preciseOriginTimestamp (see 8.3.1.2 and 8.4.1.3).

For boundary clocks, this capability shall be separately specified for each PTP port potentially used by the clock. It is recommended that all PTP ports for a boundary clock carry the same specifications.

This designation shall be encoded into the flags field of all Sync or Delay_Req messages issued by the clock (see 8.2.10) as follows:

- If the PTP port supports the issuing of Follow_Up messages, the PTP_ASSIST flag shall be TRUE.
- Otherwise the PTP_ASSIST flag shall be FALSE.

6.2.4.7 State

For each ordinary clock and for each PTP port of a boundary clock, the state machine of 7.3 shall be supported.

6.2.4.8 Variance

Two variance estimates, as specified in 7.7, characterize clocks in a PTP system:

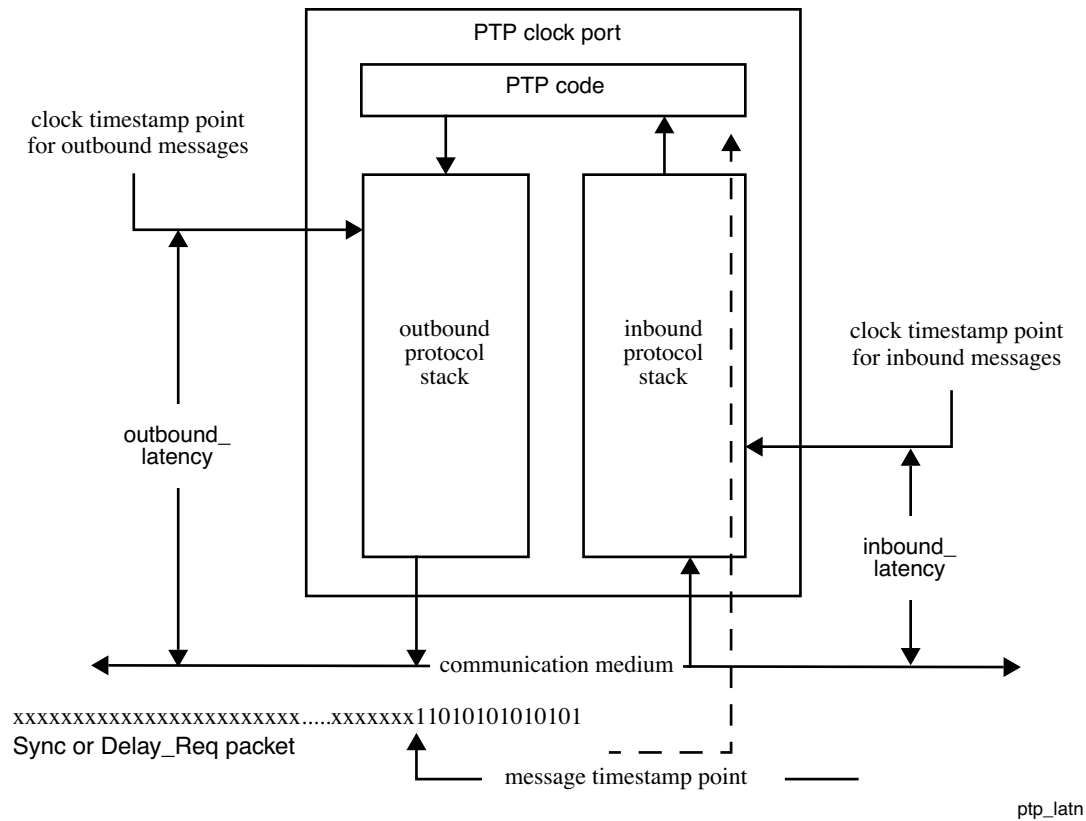
- Each clock shall maintain an estimate, the clock_variance (see 7.4.2.6) of its inherent stability properties (that is, its stability when it is not synchronized to another PTP clock using the PTP protocol).
- If a clock is synchronized to another using the PTP protocol, it may maintain an estimate, the observed_variance (see 7.4.4.9) of the stability properties of the clock to which it is synchronized.

6.2.4.9 Latency

Each PTP port shall be characterized by two constants known as outbound_latency and inbound_latency, used in the delay correction process specified in 7.8.1.3. In a clock, Sync and Delay_Req messages shall be assigned a timestamp on both issuance and receipt. These timestamps are instantiated at a point, the clock timestamp point, in the path between the code executing the PTP protocol and the communication media, as shown in Figure 4.

The outbound_latency constant shall be the propagation time between the clock timestamp point and the communication medium for outbound Sync or Delay_Req messages. The inbound_latency constant shall be the propagation time between the communication medium and the clock timestamp point for inbound Sync or Delay_Req messages.

Note that in general, the values of `outbound_latency` and `inbound_latency` will not be identical. For both inbound and outbound Sync and Delay_Req messages, timestamps are generated at the instant the message timestamp point passes the corresponding clock timestamp point.



Clause 6.2.2.3 requires that all timestamps reflect the time at which a message timestamp point passes a clock timestamp point. If an implementation detects Sync or Delay_Req messages at a point other than the message timestamp point, then the generated timestamps shall be appropriately corrected by the time interval between the time of detection and the time the message timestamp point passed a clock timestamp point.

A PTP subdomain is characterized as defined in the following subclauses.

6.2.5.1 Subdomain name

Each subdomain shall be characterized by a name recognized by all PTP clocks in the subdomain as the basis for sending and receiving PTP communications. The subdomains shall be as defined in 6.2.3. The combination of the subdomain name and the port category shall specify the end points for a PTP communication. All PTP clocks in the subdomain shall use this name as the basis for all PTP communications.

The value of subdomain name member shall be of type OctetArray [PTP_SUBDOMAIN_NAME_LENGTH]. The actual value shall be restricted as follows:

- The default values of the octets shall correspond to the hex values of the five ASCII characters _DFLT followed by (PTP_SUBDOMAIN_NAME_LENGTH – 5) null characters, 0x00. This shall result in the clock operating in the subdomain DefaultPTPdomain (see 6.2.3).
- The values of the octets corresponding to the hex values of the five ASCII characters _ALT1, _ALT2, and _ALT3 followed by (PTP_SUBDOMAIN_NAME_LENGTH – 5) null characters, 0x00, may be used and shall result in the clock operating in the subdomains AlternatePTPdomain1, AlternatePTPdomain2, and AlternatePTPdomain3 respectively (see 6.2.3).
- Other values of the octets may be chosen from the set consisting of the values of the printable ASCII characters starting with hex value 0x21 (!) up to and including hex value 0x7E (~). If the number of desired characters for a subdomain name is less than PTP_SUBDOMAIN_NAME_LENGTH, the desired characters shall be augmented by appending, in the least significant octets, sufficient value 0x00 octets to bring the total to PTP_SUBDOMAIN_NAME_LENGTH. The selection of such a subdomain name shall result in the clock operating in one of the domains AlternatePTPdomain1, AlternatePTPdomain2, or AlternatePTPdomain3 or any optionally named subdomains that exist in a PTP domain (see 6.2.3). The mapping of these octets onto the subdomain addresses shall be as specified in Annex C.

Modification of this member's value shall not take effect until PTP enters the Initialization state. Subclause 7.4, Note 1 applies.

Two subdomain names shall match if and only if an octet-by-octet comparison for all PTP_SUBDOMAIN_NAME_LENGTH octets results in a match for all comparisons.

6.2.5.2 Subdomain address

Each subdomain_name shall have a corresponding subdomain address. The representation of this address and the implementation of PTP communication based on this address are communication technology specific.

6.2.5.3 Port category

PTP communications shall be characterized by two port categories recognized by all PTP clocks in the subdomain as the basis for specifying the contents of PTP communications. The names shall be:

- EventPort: This name shall designate the access point for communication of PTP Sync or Delay_Req messages.
- GeneralPort: This name shall designate the access point for communication of PTP Follow_Up, Delay_Resp and Management messages.

The combination of the subdomain name and the port category shall specify the end points for a PTP communication. All PTP clocks shall use these names as the basis for all PTP communications.

6.2.5.4 Port address

Each communication technology shall have an address corresponding to each port category. The representation of these addresses and the implementation of PTP communication based on these addresses are communication technology specific.

6.2.5.5 Sync interval

The sync interval shall be the interval in seconds between successive Sync messages issued by master clocks. It shall have the same value for all clocks in a subdomain.

The value of sync interval is a compromise between the inherent stability of the clocks, the responsiveness of the clocks in a subdomain to change, and the communication load imposed by PTP.

The values of sync interval shall be taken from the set {1, 2, 8, 16, and 64 seconds}. Specific communication technologies may designate a subset of these values for use in that technology. Within a technology, all clocks shall be capable of correct PTP operation in a system with any of these designated sync intervals.

The behavior of PTP subdomains where clocks do not all have the same value of sync interval is outside the scope of this standard.

6.2.5.6 Epoch

The epoch is the origin of the timescale supported in a subdomain. The timescale of a subdomain shall be measured in cumulative seconds and nanoseconds since the epoch. Since all times within a subdomain are ultimately derived from the grandmaster clock in the subdomain, the subdomain epoch is the epoch of the grandmaster clock.

The clock_identifier of the grandmaster clock shall specify the value of the epoch as follows:

Clock identifier (ASCII)	Epoch specification
ATOM	PTP epoch
GPS	PTP epoch
NTP	PTP epoch
HAND	PTP epoch
INIT	The epoch shall be as defined by the initialization procedure.
DFLT	The epoch is unknown.

Where the epoch is the PTP epoch, the cumulative seconds and nanoseconds supported by the PTP protocol may be converted into UTC current month, day, year, time of day format by utilizing the current_utc_offset field value of the global time properties data set. The PTP epoch began at 0 hours on 1 January 1970. See Annex B for information on how to convert times measured from the PTP epoch to UTC and other commonly used time scales.

6.2.5.7 Epoch number

When the epoch is the PTP epoch, the value of the epoch number shall be the current number of times the 32-bit seconds clock has rolled over since the PTP epoch. For any epoch, the epoch number may be treated as the most significant part of the total number of seconds since the epoch where the least significant part is the 32-bit integer seconds portion of the TimeRepresentation of PTP timestamps.

7. PTP protocol specification

7.1 Protocol model of a clock

The model of an ordinary clock and of a boundary clock used in the definition of the protocol is illustrated in Figure 5 and Figure 6, respectively.

Figure 5 illustrates an ordinary clock. This clock is presumably one of several using a communication path implemented using communication technology q . The clock has two access points to the communication path modeled by the event and general ports with their associated addresses, together termed the *PTP port* or more simply, *port*. This port communicates with the protocol engine of the clock. The protocol engine in turn has access to port-specific information as well as the general information about the clock. The protocol engine can also read the clock's current time as well as adjust the time behavior of the local clock.

For all clocks, both ordinary and boundary:

- Each PTP port of the clock is modeled to have an index represented by the `port_id_field`. For ordinary clocks that have only a single port, the `port_id_field` shall have the value 1. For boundary clocks, each port has a distinct `port_id_field` with the ports numbered sequentially starting with `port_id_field` = 1.
- The *local clock* of Figure 5 and Figure 6 is assigned an *internal* `port_id_field` value of 0.
- All clocks have at most one port in the PTP_SLAVE state during normal operation. The local clock shall be synchronized based on the information received on this port from the clock recognized as the master clock visible from that port. In the absence of a port in the PTP_SLAVE state, the local clock shall not synchronize to any other clock via the PTP protocol. It may synchronize to a local oscillator or to another clock, usually a UTC time source such as GPS via means outside the scope of this standard.
- Each clock maintains the data sets shown.

Figure 6 illustrates a boundary clock having several PTP ports $\{p\}$, each port being one of several using communication paths implemented using communication technologies $\{q\}$. Each PTP port is implemented as in an ordinary clock with the following exceptions:

- A separate set of clock port configuration information and foreign master information is maintained for each PTP port.
- A set of configuration information is maintained for the boundary clock itself. Any PTP port may access this data. If each port maintains a local copy of the clock datasets, implementations shall ensure that all such copies are consistent whenever protocol decisions are made at any port.
- All ports can read a single, common local clock.

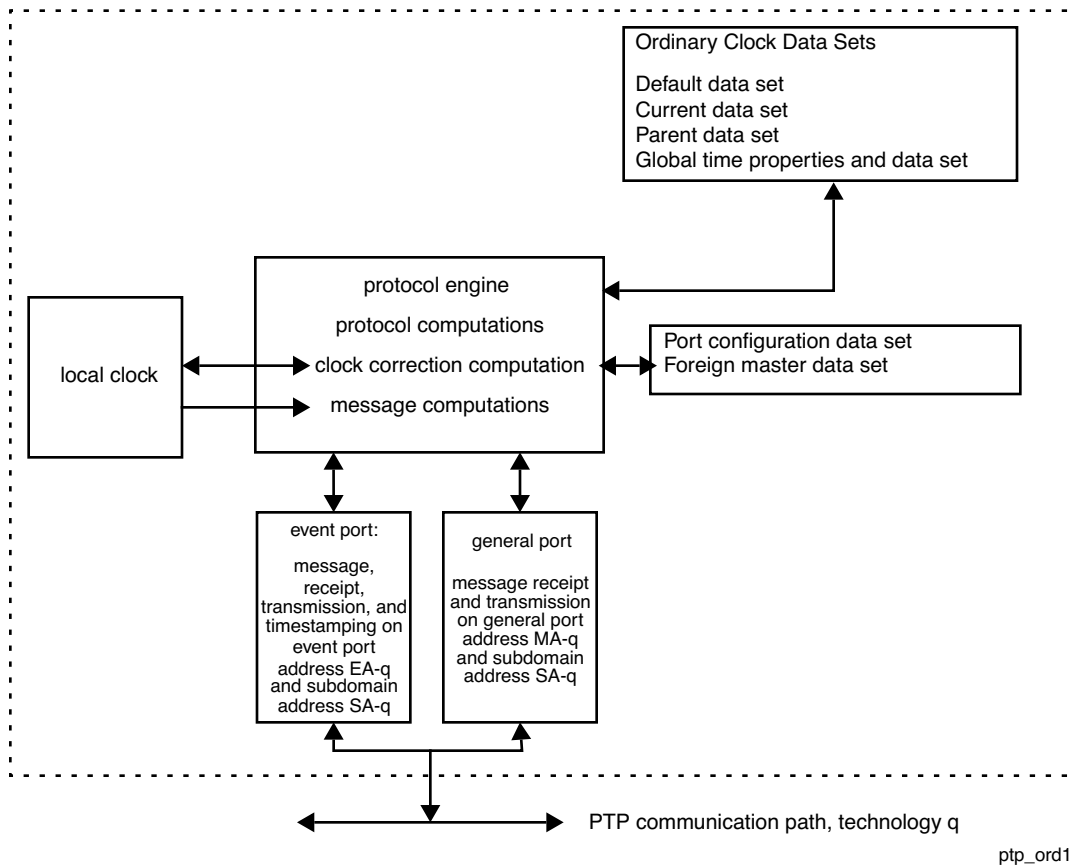


Figure 5—Ordinary clock protocol model

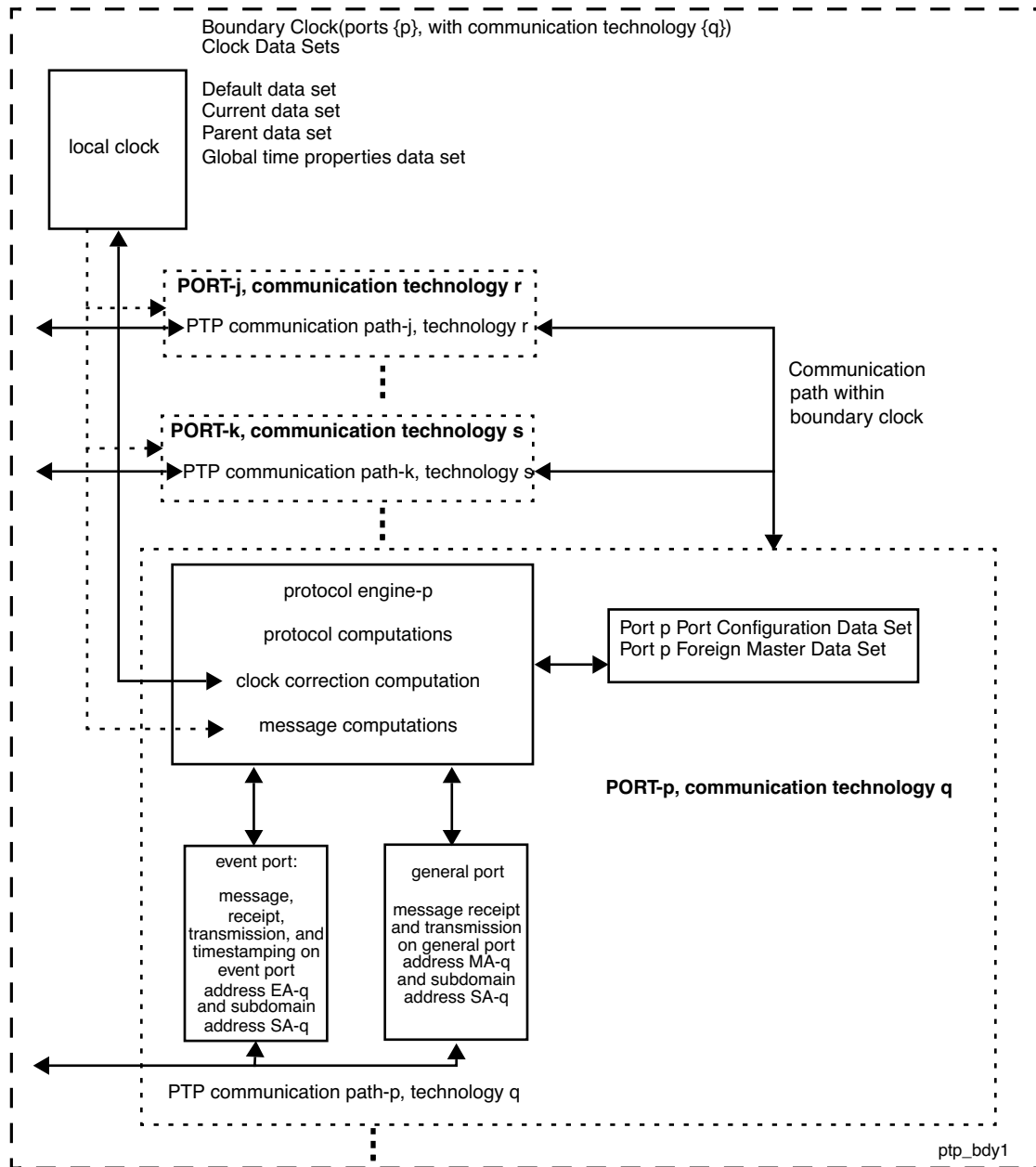


Figure 6—Boundary clock protocol model

7.2 Protocol model of a subdomain of PTP clocks

In a subdomain, the PTP protocol recognizes clocks with five categories of ports. Excluded from this discussion are clock ports in the PTP_INITIALIZING, PTP_DISABLED, or PTP_FAULTY states. These categories are as follows:

- Slave ports: These may be the ports of ordinary clocks or the external view of a single PTP port of a boundary clock that synchronize to a particular master port. A master port to which a slave is synchronized is termed the *slave's parent clock port*. A slave port must be in the PTP_SLAVE state specified in 7.3.

- Master ports: These may be the ports of ordinary clocks or the external view of a single PTP port of a boundary clock that serve as the parent clock port to zero or more slave ports sharing a PTP communication path with the master. A master port must be in the PTP_MASTER or PTP_PRE_MASTER states specified in 7.3.
- Grandmaster ports: These may be the ports of an ordinary clock or the external view of all PTP master ports of a boundary clock. There is a single set of grandmaster ports, all on the same clock in a subdomain. This set serves as the master clock ports to zero or more slave ports sharing the PTP subdomain with the grandmaster. A grandmaster port must be in the PTP_MASTER state specified in 7.3.
- Uncalibrated ports: These may be the ports of an ordinary clock or the external view of a single PTP port of a boundary clock. These are ports for which the protocol has not yet assigned a master or designated the port as a master port. An uncalibrated port must be in the PTP_LISTENING or PTP_UNCALIBRATED state specified in 7.3.
- Passive ports: These may be the ports of an ordinary clock or the external view of a single PTP port of a boundary clock. These ports may be associated with a stratum 1 or 2 clock that has detected the presence of other stratum 1 or 2 clocks in the system. One of the other stratum 1 or 2 clocks has been designated by the protocol as the master clock. Alternatively, a port may be designated as passive by the protocol to avoid cyclic topology. A passive port must be in the PTP_PASSIVE state specified in 7.3.

Every stable subdomain forms a parent-child hierarchy of clock ports. The root of this hierarchy is termed the *grandmaster clock*. At each branch point (necessarily a boundary clock), a clock port must be the parent and master port for all clock ports on the branch further removed from the root and have a slave port synchronizing to the next clock closer to the root. A port, not the root, at the extreme of any branch of the hierarchy must be a slave or passive port. A branch port must be a boundary clock. A stable subdomain is a subdomain in which all ports participating in PTP have been designated by the protocol as either a master, passive, or a slave and a single grandmaster clock has been designated.

For subdomains containing multiple stratum 1 or 2 clocks, the protocol will segment the subdomain into multiple disjoint parent-child hierarchies, each containing a single stratum 1 or 2 clock.

Examples of various subdomain master-slave configurations are illustrated in Figure 7 and Figure 8. In both figures, the state of each clock port is indicated by a M, S, or P for master, slave, and passive, respectively.

Figure 7 illustrates a subdomain in which one of the ordinary clocks, node-5, is designated as the grandmaster clock. In this case, all of the other clock ports will form an acyclic parent-child relationship with node-5 as the root.

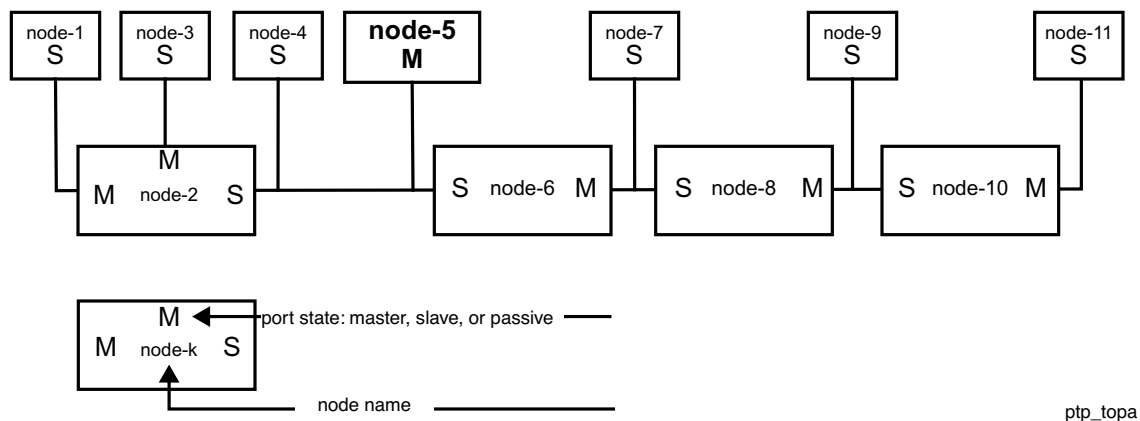


Figure 7—Master-slave configuration, example 1

Figure 8 illustrates a subdomain in which two of the clocks; C-1 and BC-9 are both stratum 1 clocks. In this case the system will segment into two disjoint subdomains, one with C-1 as grandmaster the other with BC-9 as grandmaster, as shown. Within each of the subdomains, the clock ports form an acyclic parent-child hierarchy. Where multiple paths occur, all but one of the paths will be cut by a port in the passive state, for example, between BC-11 and BC-12.

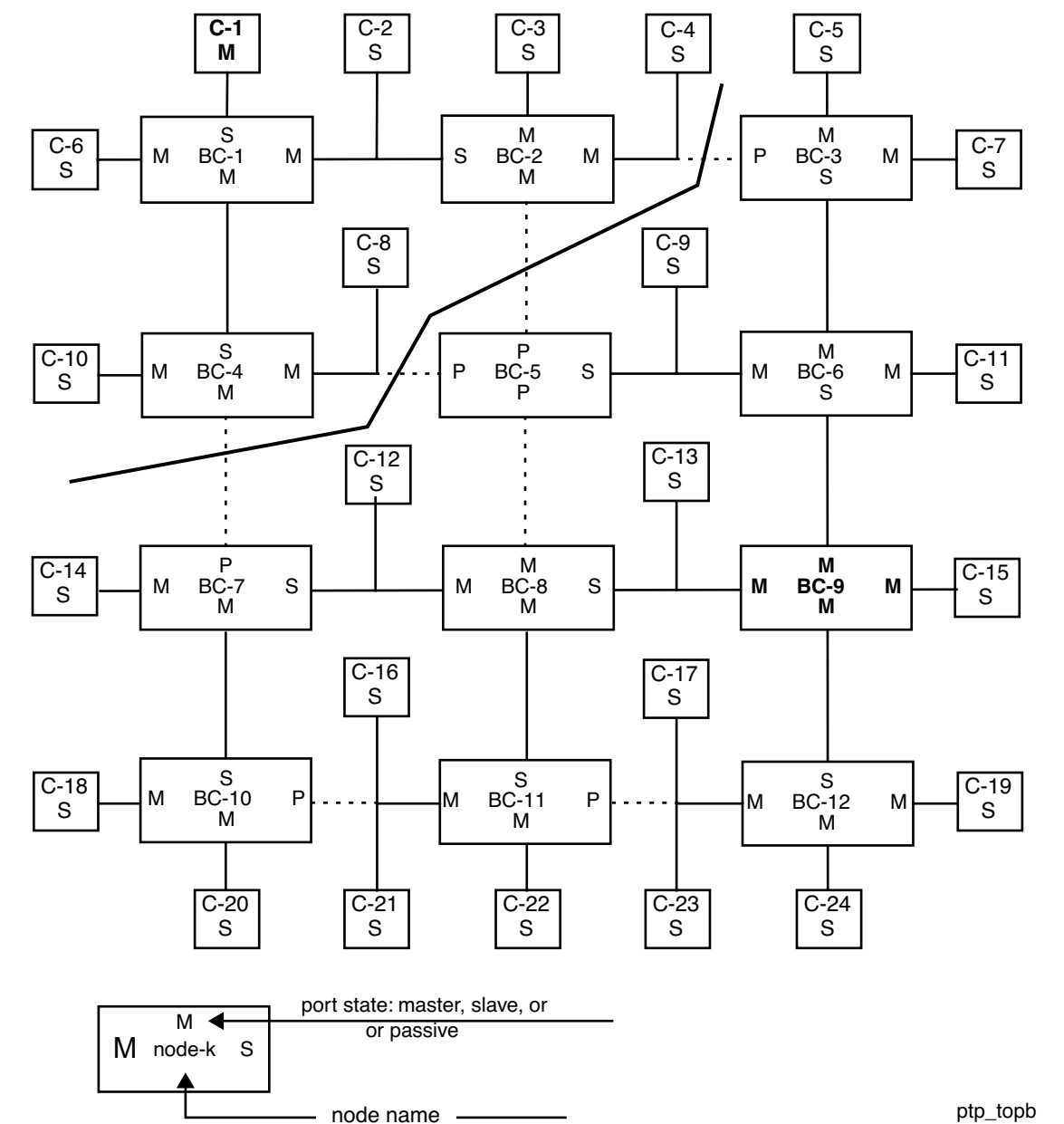


Figure 8—Master-slave configuration, example 2

The allowed possibilities for ordinary clocks and for each PTP port of a boundary clock are summarized in Table 6. The clock numbers refer to clocks in Figure 8 as an example.

Table 6—Allowed combinations of port categories

Port State	Ordinary clock	Boundary clock port
Grandmaster (and therefore a master and potentially a parent)	Possible (C-1)	Possible only if no port is a slave (BC-9)
Master (and therefore potentially a parent)	Possible and if so it is also a grandmaster (C-1)	Possible if one (BC-6) or no (BC-9) port is a slave.
Slave	Possible, (C-10)	Possible only for a single port, (all but BC-9)
Uncalibrated	Possible (not shown)	Possible (not shown)
Passive	Possible (not shown)	Possible (BC-11)

The protocol determines which of the ports in a system are masters, slaves, uncalibrated, passive, and which clock is the grandmaster. The transition from one stable configuration to another is governed by the protocol rules. Such transitions may be caused by the introduction or removal of a clock or the change in the properties, such as the clock identifier of one of the clocks. The addition or removal of a connection to a boundary clock may respectively join two subdomains or segment a single subdomain, thereby causing a transition between two stable configurations.

7.3 State behavior of clocks

This subclause describes the state machine which ordinary clocks and boundary clocks shall implement.

7.3.1 Protocol engine state machine

The protocol engine of each ordinary clock or each PTP port of a boundary clock shall implement the state machine illustrated in Figure 9 (see 4.3 for a key to interpreting the figure).

When a fault is cleared, the engine makes a transition to the PTP_INITIALIZING state. Before making the transition to PTP_LISTENING, an implementation is not required to perform all the steps that it would after an INITIALIZE command or a powerup or reset. It is required only to achieve the *effect* of performing those steps.

The behavior of the seven major states of a port associated with the protocol engine shall be as defined in the PTPState enumeration.

IDL: enumeration PTPState;

Table 7—PTP State enumeration

State enumeration	Description
PTP_INITIALIZING	The associated port may initialize the data sets, hardware, and communication properties of the clock.
PTP_FAULTY	The fault state of the protocol. The associated port shall not participate in the synchronization aspects of the protocol but may take implementation-specific measures to clear the fault.
PTP_DISABLED	The associated port shall not place any messages on its communication path. In a boundary clock, no activity at the associated port shall be allowed to affect the activity at any other port of the boundary clock.
PTP_LISTENING	The associated port is waiting for the Sync message receipt timeout to expire or to receive a Sync message from a master. The purpose of this state is to allow orderly addition of clocks to a subdomain.
PTP_PRE_MASTER	The associated port shall behave in all respects as though it were in the PTP_MASTER state except that it shall not place any non-Management messages on its communication path.
PTP_MASTER	The associated port is behaving as a master port. The local clock is used to timestamp the receipt or departure of messages associated with the port.
PTP_PASSIVE	The associated port shall not place any messages on its communication path unless otherwise specified.
PTP_UNCALIBRATED	One or more master ports have been detected in the subdomain. The appropriate master port is being selected and the local port is preparing to synchronize to the selected master port. This is a transient state to allow initialization of synchronization servos, updating of data sets when a new master port has been selected, and other implementation-specific activity.
PTP_SLAVE	The associated port shall synchronize to the selected master port.



7.3.2 State machine constraints for boundary clocks

A boundary clock consists of a single local clock and N PTP ports.

Each PTP port of a boundary clock shall have its own value for state. This state shall be governed by a copy of the state machine of Figure 9 associated with each port.

Only the combinations of states and behaviors for the PTP ports of a boundary clock shown in Table 8 shall be allowed in a subdomain. For purposes of this clause, an active port is a port not in the PTP_INITIALIZING, PTP_FAULTY, PTP_DISABLED, or PTP_LISTENING states.

Table 8—Boundary clock state constraints

Port state	Constraints on a boundary clock with N ports
PTP_INITIALIZING	If one port is in this state, all ports shall be in this state.
PTP_FAULTY	A port in this state is considered inactive. No activity on this port shall affect the active ports of the boundary clock or of any other port in the subdomain. If fault activity on a port in this state cannot be confined to the port, then all ports shall be in this state.
PTP_DISABLED	A port in this state is considered inactive. No activity on this port shall affect the active ports of the boundary clock or of any other port in the subdomain.
PTP_LISTENING	Any or all ports may be in this state.
PTP_PRE_MASTER	Same as for PTP_MASTER, except that the port shall not place any non-management messages on its associated communication path.
PTP_MASTER	If one port is in this state either: — No active port is in the PTP_SLAVE state and the boundary clock is also the grandmaster clock of the subdomain for all communication paths for which the boundary clock communicates via a port in the PTP_MASTER state, or — One active port is in the PTP_SLAVE state and the grandmaster clock of the subdomain is a clock communicating with the boundary clock directly or indirectly via the slave port. The boundary clock will be the master (but not the grandmaster) for all communication paths for which the boundary clock communicates via a port in the PTP_MASTER state.
PTP_PASSIVE	Any or all ports may be in this state.
PTP_UNCALIBRATED	Any port may be in this state.
PTP_SLAVE	At most, one port shall be in this state. The local clock of the boundary clock shall be synchronized to the master port in the communication path associated with the port in the PTP_SLAVE state.

The events shown Figure 9 shall be applied to the state machine's ports of the boundary clock as specified in Table 9.

Table 9—State machine event applicability

Event name	Applicability
POWERUP	State machines of all ports.
INITIALIZE	State machine of all ports.
FAULT_DETECTED	State machine of all ports affected by the fault.
FAULT_CLEARED	State machine of all ports affected by the fault.
STATE_CHANGE_EVENT	State machine associated with the port signaling the event. The information may affect all state machines due to other constraints of this clause.
BEST_MASTER_CLOCK	The execution of the best master clock algorithm associated with a port directly affects only that port. The execution potentially affects the states of other ports if required by the other constraints of this clause.
SYNC_RECEIPT_TIMEOUT_EXPIRES QUALIFICATION_TIMEOUT_EXPIRES	State machine associated with the expiring timeout mechanism.
DESIGNATED_ENABLED DESIGNATED_DISABLED	State machine specified by the initiating management message.
MASTER_CLOCK_SELECTED	State machine executing the best master clock algorithm making this decision.
SYNCHRONIZATION_FAULT	State machine associated with the port experiencing the fault.

7.4 Clock data set

For each clock, the following clock data sets specified in the subclauses of this clause shall be maintained locally as the basis for protocol decisions and for providing values for message fields:

- Default data set
- Current data set
- Parent data set
- Global time properties data set
- Port configuration data set (one data set for each port of a boundary clock)
- Foreign master data set (one data set for each port of a boundary clock)

Each data set member specification includes the following:

- The formal name for the member
- Semantics associated with the member
- The value assigned at initialization

In these data sets, the following notes apply:

NOTES

1—These values may change infrequently as a result of management messages or degradation of the clock implementation, for example, degradation in the stability of the local clock over time or changes in subdomain topology.

2—These values change frequently during the normal operation of the protocol.

3—This information does not directly affect the PTP protocol but may be used by services built on top of the protocol. This information is typically derived either from management messages or from a stratum 1 clock in the system, for example, a clock synchronized to a GPS receiver or participating in the NTP protocol with other computers with access to standard UTC time sources.

7.4.1 Clock data set initialization properties

There are two kinds of members for the clock data sets: fixed and modifiable.

The values of fixed members are inherent properties of the clock. These members shall have the values as specified in the defining subclause.

The modifiable members may change as a result of management messages, changes in internal properties of the clock resulting, for example, from local synchronization to a standard clock, parametric changes due to aging or temperature, or fault detection and recovery.

Modifiable members shall be initialized to one of the following values:

- The values indicated in the member specifications as initialization values. If no initialization value is specified, the default value for the specified datatype shall be used as the initialization value. This set of values is designated the specification initialization set.
- A set of values established by implementation-specific means such as management messages, non-volatile storage of current operating values, etc. If this option is implemented, the management message PTP_MM_INITIALIZE_CLOCK shall be implemented (see 7.12.4), and the value of the member, initializable, of the default data set shall be TRUE (see 7.4.2.9).

Modifiable members shall be initialized:

- As a result of the initialization event indicated in the state diagram Figure 9.
- As a result of the powerup event indicated in the state diagram Figure 9.
- As a result of the management message PTP_MM_INITIALIZE_CLOCK (see 7.12.4).
- As specified in other clauses of this standard.

7.4.2 Default data set

This data set defines inherent or assumed properties of the local clock. These values are used when the local clock becomes the grandmaster clock in a subdomain. They depend on the specific source of time for the local clock when it is the grandmaster.

The possibilities for the source of time to a grandmaster clock include:

- Directly synchronized (not via PTP) to an atomic clock or a GPS receiver
- Directly synchronized (not via PTP) to a clock participating in NTP
- A local oscillator

The members of this data set are:

- clock_communication_technology
- clock_uuid_field
- clock_port_field
- clock_stratum
- clock_identifier
- clock_variance
- clock_followup_capable
- preferred

- initializable
- external_timing
- is_boundary_clock
- sync_interval
- subdomain_name
- number_ports
- number_foreign_records

7.4.2.1 clock_communication_technology

This member's value shall be the communication_technology_field of the Clock-UUID of the local clock. The datatype shall be as specified in 6.2.4.1. This is a static property of the local clock and shall not be modifiable.

7.4.2.2 clock_uuid_field

This member's value shall be the uuid_field of the Clock-UUID of the local clock. The datatype shall be as specified in 6.2.4.1. This is a static property of the local clock and shall not be modifiable.

7.4.2.3 clock_port_field

This member's value and datatype shall be as specified in 6.2.4.1. This is a static property of the local clock and shall not be modifiable.

7.4.2.4 clock_stratum

This member's value shall characterize those aspects of the accuracy defined in 6.2.4.3. The datatype shall be as specified in 6.2.4.3. Depending on the implementation of the local clock, this value may or may not be modifiable.

7.4.2.5 clock_identifier

This member's value shall be the clock identifier of the local clock when it is the grandmaster clock of the system. The allowed initialization values depend on the value of the member clock_stratum and on the nature of the hardware implementation (see 6.2.4.5). The datatype shall be as specified in 6.2.4.5. Depending on the implementation of the local clock this value may or may not be modifiable. Subclause 7.4, Note 1 applies.

7.4.2.6 clock_variance

This member's value shall be the variance of the local clock when it is the grandmaster clock of the system (see 6.2.4.8). The datatype and computation of variances shall be as specified in 7.7. The initialization value shall reflect the inherent characteristics of the clock. Depending on the implementation of the local clock this value may or may not be modifiable. Subclause 7.4, Note 1 applies.

7.4.2.7 clock_followup_capable

This member's value shall indicate whether the clock is capable of providing Follow_Up messages. The datatype shall be Boolean. The value shall be TRUE if the local clock supports the issuing of Follow_Up messages per 6.2.4.6, and FALSE otherwise. This is a static property of the local clock and shall not be modifiable.

7.4.2.8 preferred

This member's value shall indicate whether the clock is to be preferred in the selection of the grandmaster clock. The datatype shall be Boolean. The value shall be TRUE if the local clock is a member of the preferred master clock set and FALSE if it is not a member of this set (see 6.2.4.4). This value shall be modifiable by the user. The default value shall be TRUE if the clock `clock_stratum` is 1 or 2 and FALSE otherwise. Subclause 7.4, Note 1 applies.

7.4.2.9 initializable

This member's value shall indicate whether the receipt of a management message with a `managementMessageKey` field value `PTP_MM_INITIALIZE_CLOCK` (see 7.12.4) causes the clock to execute the behavior normally executed on a reboot or power cycle of the node containing the clock (see 7.3). The datatype shall be Boolean. The value shall be TRUE if the receipt of a management message with a `managementMessageKey` field value `PTP_MM_INITIALIZE_CLOCK` (see 7.12.4) causes the clock to execute the behavior normally executed on a reboot or power cycle of the node containing the clock (see 7.3). Otherwise, the value shall be FALSE. This is a static property of the local clock and shall not be modifiable.

7.4.2.10 external_timing

This member's value shall indicate whether the clock is capable of sending external timing signals. The datatype shall be Boolean. The value shall be TRUE if the clock is capable of sending external timing signals when it has a port in the master state. Otherwise, the value shall be FALSE. This is a static property of the local clock and shall not be modifiable.

7.4.2.11 is_boundary_clock

This member's value shall indicate whether the clock is a boundary clock. The datatype shall be Boolean. The value shall be TRUE if the clock is a boundary clock. Otherwise, the value shall be FALSE. This is a static property of the local clock and shall not be modifiable.

7.4.2.12 sync_interval

This member's value shall specify the current sync interval (see 6.2.5.5). The datatype shall be Integer8. The value shall be the logarithm base 2 of the current sync interval in seconds. This value may be defined administratively by means of a management message. The default sync interval shall be two seconds, that is, `sync_interval` = 1, irrespective of the communication technology. The `sync_interval` value shall be the default value unless changed by an administrative procedure.

7.4.2.13 subdomain_name

This member's value shall specify the subdomain name. The datatype shall be as specified in 6.2.5.1. This value may be defined administratively by means of a management message. This member's initialization value may be the name of the last PTP subdomain defined administratively. In the absence of prior modification by means of a management message or by implementation-specific reset means, the default shall be the default value defined in 6.2.5.1.

7.4.2.14 number_ports

This member's value shall specify the number of ports on the device. For an ordinary clock this shall be the value 1. The datatype shall be UInteger16. This is a static property of the local clock and shall not be modifiable.

7.4.2.15 number_foreign_records

This member's value shall specify the maximum number of records maintained in the foreign master data set of each port. The datatype shall be UInteger16. This is a static property of the local clock and shall not be modifiable.

7.4.3 Current data set

This data set defines members whose values characterize the current properties of the local clock that describe the source and quality of the local time. These values depend on the specific source of time for the local clock and whether it is the grandmaster. Various possibilities for the source of time are (not an exhaustive list):

- Locally synchronized (not via PTP) to an atomic clock, a GPS receiver
- Locally synchronized (not via PTP) to a clock participating in NTP
- A local oscillator
- Another PTP clock in the subdomain

The members of this data set are:

- steps_removed
- offset_from_master
- one_way_delay

7.4.3.1 steps_removed

This member's value shall be the number of communication paths traversed between the local clock and the grandmaster clock. For example, steps_removed in a slave clock on the same PTP communication path as the grandmaster clock will have a value of 1, indicating that a single path was traversed. The datatype shall be UInteger16.

7.4.3.2 offset_from_master

When the local clock is slaved to another PTP clock using the PTP protocol, this member's value shall be the local clock's estimate of the current time offset between the local clock and the PTP clock to which is directly synchronized. If the local clock is not slaved to another PTP clock using the PTP protocol, the value shall be 0. This value shall be computed as specified in 7.8.1.1. The datatype shall be TimeRepresentation. Subclause 7.4, Note 2 applies.

7.4.3.3 one_way_delay

When the local clock is slaved to another PTP clock using the PTP protocol, this member's value shall be the local clock's estimate of the current one-way propagation delay between the local clock and the PTP clock to which it is synchronized via PTP. Otherwise, the value shall be 0. This value shall be computed as specified in 7.8.1.2. The datatype shall be TimeRepresentation. Subclause 7.4, Note 2 applies.

7.4.4 Parent data set

This data set defines quantities whose values characterize the current properties of the PTP port, the parent or parent port, serving as the source for the time for a PTP slave port via the PTP protocol.