TICTOC Working Group   S. Davari
Internet-Draft   A. Oren
Intended status: Experimental   Broadcom Corp.
Expires: April 18, 2016   M. Bhatia
                                                             P. Roberts
                                                          Alcatel-Lucent
                                                             L. Montini
                                                           Cisco Systems
                                                        October 16, 2015

               Transporting Timing messages over MPLS Networks
                      draft-ietf-tictoc-1588overmpls-07

Abstract

   This document defines a method for transporting timing messages, such
   as Precision Time Protocol (PTP) or Network Time Protocol (NTP), over
   a Multiprotocol Label Switched (MPLS) network. The method
   facilitates efficient recognition of timing packets to enable their
   port level processing in both Label Edge Routers (LERs) and Label
   Switched Routers (LSRs).

   The basic mechanism is to transport timing messages inside "Timing
   LSPs", which are dedicated MPLS Label Switched Paths (LSPs) that
   carry only timing, and possibly related Operations, Administration
   and Maintenance (OAM) or management packets, but do not carry
   customer traffic.

   Two encapsulations methods are defined. The first transports UDP/IP
   encapsulated timing messages directly over the dedicated LSP. The
   second transports Ethernet encapsuled timing messages inside an
   Ethernet pseudowire.

Status of This Memo

This Internet-Draft will expire on April 18, 2016.

Copyright Notice

Table of Contents

1. Introduction

   The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
   "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this
   document are to be interpreted as described in RFC2119 [RFC2119].

   When used in lower case, these words convey their typical use in
   common language, and are not to be interpreted as described in
   RFC2119 [RFC2119].

   The objective of timing distribution protocols, such as Precision
   Time Protocol (PTP) and Network Timing Protocol (NTP), is to
   synchronize clocks running on nodes of a distributed system.

   Timing distribution protocols are presently transported over IP or
   Ethernet. The present document presents a mechanism for transport
   over Multiprotocol Label Switched (MPLS) networks. Our solution
   involves transporting timing messages over dedicated "Timing Label
   Switched Paths (LSPs)". These are ordinary LSPs that carry timing
   messages and MAY carry Operations, Administration and Maintenance
   (OAM) or management messages, but do not carry any other traffic.

   Timing LSPs may be established statically or via signaling. When
   using signaling, extensions to routing protocols (e.g., OSPF, ISIS)
   are required to enable routers to distribute their timing processing
   capabilities, and extensions to path set up protocols (e.g., RSVP-TE)
   are required for establishing the LSPs. All such extensions are
   beyond the scope of this document.

   High accuracy timing distribution requires on-path support, e.g.,
   Transparent Clocks (TCs) or Boundary Clocks (BCs), at intermediate
   nodes. These intermediate nodes need to recognize and appropriately
   process timing distribution packets. To facilitate efficient
   recognition of timing messages transported over MPLS, this document
   restricts the specific encapsulations to be used.

   [IEEE-1588] defines PTP messages for frequency, phase and time
   synchronization. PTP messages may be transported over UDP/IP (Annex
   D and E of [IEEE-1588]) or over Ethernet (Annex F of [IEEE-1588]).
   This document defines two methods to transport PTP messages over MPLS
   networks.

   PTP defines several clock types, including ordinary clocks, boundary
   clocks, end-to-end transparent clocks, and peer-to-peer transparent
   clocks. Transparent clocks are situated at intermediate nodes and

update the Correction Field inside PTP messages in order to reflect
the time required to transit the node.

[RFC5905] defines NTP messages for clock and time synchronization.
NTP messages are transported over UDP/IP. This document defines a
method to transport NTP messages over MPLS networks.

It can be expected that only a subset of LSR ports will be capable of
processing timing messages. Timing LSPs MUST be set up (either by
manual provisioing or via signaling) to traverse these ports. While
Timing LSPs are designed to optimize timing distribution, the
performance of slave clocks is beyond the scope of this document.

Presently on-path support is only defined for PTP, and therefore much
of our discussion will focus on PTP. NTP timing distribution may
benefit from transport in a Timing LSP due to prioritorization or
selection of ports or nodes with minimal delay or delay asymmetry.

2. Terminology

1588: The timing distribution protocol defined in IEEE 1588.

Boundary Clock: A device with one timing port to receive timing
messages and at least one port to re-distribute timing messages.

CF: Correction Field, a field inside certain PTP messages that holds
the accumulated transit time.

Master Clock: The source of 1588 timing messages to a set of slave
clocks.

NTP: The timing distribution protocol defined in RFC 5905.

Ordinary Clock: A master or slave clock. Note that ordinary clocks
have only a single PTP port.

PTP: Precision Time Protocol. See 1588.

Slave Clock: A receiver of 1588 timing messages from a master clock.

Timing LSP: An MPLS LSP dedicated to carry timing messages.

Timing messages: Timing distribution protocol messages that are
exchanged between clocks.

Timing port: A port on a (master, slave, transparent, or boundary)
clock.

Timing PW: A PW within a Timing LSP that is dedicated to carry timing messages.

Transparent Clock: An intermediate node that forwards timing messages while updating their CF.

3. Problem Statement

[IEEE-1588] defines methods for transporting PTP messages over Ethernet and IP networks. [RFC5905] defines a method of transporting NTP messages over IP networks. There is a need to transport timing messages over MPLS networks while supporting the Transparent Clock (TC), Boundary Clock (BC) and Ordinary Clock (OC) functionalities in LER and LSRs of the MPLS network.

There are potentially many ways of transporting timing packets over MPLS. However, it is advisable to limit the number of possible encapsulation options to simplify recognition and processing of timing packets.

The solution herein desscribed transports timing messages over dedicated "Timing Label Switched Paths (LSPs)". Were timing packets to share LSPs with other traffic, intermediate LSRs would be required to perform some deeper inspection to differentiate between timing packets and other packets. The method herein proposed avoids this complexity, and can readily detect all PTP messages (one-step or two-step), and supports ordinary, boundary and transparent clocks.

4. Timing over MPLS Architecture

Timing messages are exchanged between timing ports on ordinary and boundary clocks. Boundary clocks terminate the timing messages and act as master clock for other boundary clocks or slave clocks. End-to-End transparent clocks do not terminate the timing messages but do modify the contents of the timing messages in transit.

OC, BC and TC functionality may be implemented in either LERs or LSRs.

An example is shown in Figure 1, where the LERs act as OCs and are the initiating/terminating points for timing messages. The ingress LER encapsulates timing messages in a Timing LSP and the egress LER terminates this Timing LSP. Intermediate LSRs (only one is shown here) act as TCs, updating the CF of transiting timing messages, as well as performing label switching operations.

```
+--------+     +-------+     +-------+     +-------+     +--------+
|Switch, |     |       |     |       |     |       |     |Switch, |
| Router |-----| LER   |-----| LSR   |-----| LER   |-----| Router |
|        |     | OC    |     | TC    |     | OC    |     |        |
+--------+     +-------+     +-------+     +-------+     +--------+
            /                                    \
+-------+  /                                      \  +-------+
| LER   | /                                        \ | LER   |
| Master|---/                                  \---| Slave |
| Clock |                                          | Clock |
+-------+                                          +-------+
```
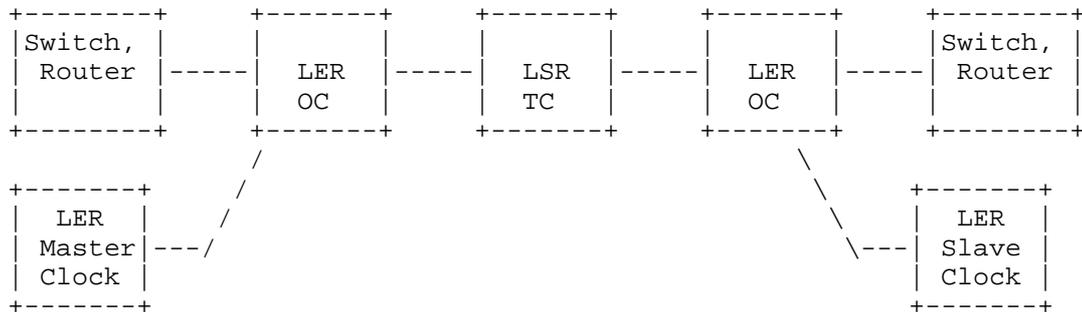
Figure (1) - Deployment example 1 of timing over MPLS network

Another example is shown in Figure 2, where LERs act as BCs, and
switches/routers outside of the MPLS network, act as OCs or BCs. The
ingress LER BC recovers timing and initiates timing messages
encapsulated in the Timing LSP toward the MPLS network, an
intermediate LSR acts as a TC, and the egress LER acts as a BC
sending timing messages to equipment outside the MPLS network.

```
+--------+     +-------+     +-------+     +-------+     +--------+
|Switch, |     |       |     |       |     |       |     |Switch, |
| Router |-----| LER   |-----| LSR   |-----| LER   |-----| Router |
| OC/BC  |     | BC    |     | TC    |     | BC    |     | OC/BC  |
+--------+     +-------+     +-------+     +-------+     +--------+
```
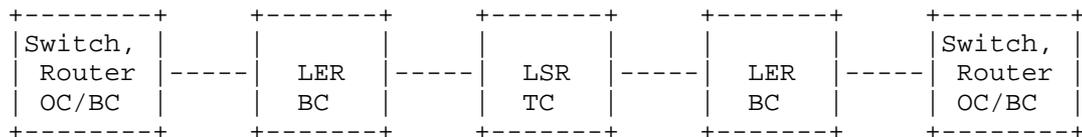
Figure (2) - Deployment example 2 of timing over MPLS network

Yet another example is shown in Figure 3, where both LERs and LSRs
act as TCs. The ingress LER updates the CF and encapsulates the
timing message in an MPLS packet, intermediate LSRs update the CF and
perform label switching, and the egress LER updates the CF and sends
the timing messages to equipment outside the MPLS network.

```
+--------+     +-------+     +-------+     +-------+     +--------+
|Switch, |     |       |     |       |     |       |     |Switch, |
| Router |-----| LER   |-----| LSR   |-----| LER   |-----| Router |
|OC/TC/BC|     | TC    |     | TC    |     | TC    |     |OC/TC/BC|
+--------+     +-------+     +-------+     +-------+     +--------+
```
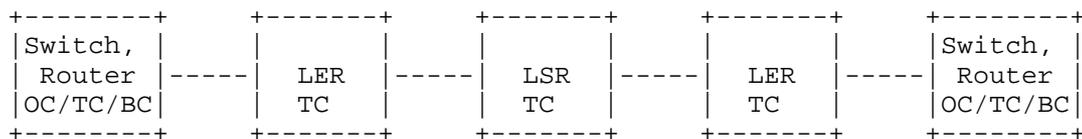
Figure (3) - Deployment example 3 of timing over MPLS network

A final example is shown in Figure 4, where all nodes act as BCs.
Single-hop LSPs are created between every two adjacent LSRs. Of
course, PTP transport over Ethernet MAY be used between two network
elements.

```
+--------+     +-------+     +-------+     +-------+     +--------+
|Switch, |     |       |     |       |     |       |     |Switch, |
| Router |-----|  LER  |-----|  LSR  |-----|  LER  |-----| Router |
| OC/BC  |     |  BC   |     |  BC   |     |  BC   |     | OC/BC  |
+--------+     +-------+     +-------+     +-------+     +--------+
```
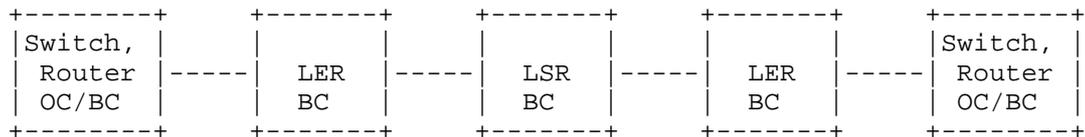
Figure (4) - Deployment example 3 of timing over MPLS network

An MPLS domain MAY serve multiple customers, each having its own
Timing domain. In these cases the MPLS domain (maintained by a
service provider) MUST provide dedicated timing services to each
customer.

The timing over MPLS architecture assumes a full mesh of Timing LSPs
between all LERs supporting this specification. It supports point-
to- point (VPWS) and Multipoint (VPLS) services. This means that a
customer may purchase a point-to-point timing service between two
customer sites or a multipoint timing service between more than two
customer sites.

The Timing over MPLS architecture supports P2P or P2MP Timing LSPs.
This means that the Timing Multicast messages such as PTP Multicast
event messages MAY be transported over P2MP Timing LSPs or MAY be
replicated and transported over multiple P2P Timing LSPs.

Timing LSPs, as defined by this specification, MAY be used for timing
messages that do not require time-stamping or CF updating.

PTP Announce messages that determine the Timing LSP terminating point
behavior such as BC/OC/TC SHOULD be transported over the Timing LSP
to simplify hardware and software.

5. Dedicated LSPs for Timing messages

The method defined in this document is used by LER and LSRs to
identify timing messages by observing the top label of the MPLS label
stack. Compliant implementations MUST use dedicated LSPs to carry
timing messages over MPLS. Such LSPs are herein referred to as
"Timing LSPs" and the labels associated with these LSPs as "Timing
LSP labels".

Timing distribution requires symmetrical bidirectional
communications. Co-routing of the two directions is required to
limit delay asymmetry. Thus timing messages MUST be transported
either over two co-routed unidirectional Timing LSPs, or a single
bidirectional co-routed Timing LSP.

Timing LSPs MAY be configured using RSVP-TE. Extensions to RSVP-TE
are required for this purpose, but are beyond the scope of this
document.

6. Timing over LSP Encapsulation

We define two methods for carrying timing messages over MPLS. The
first method transports UDP/IP-encapsulated timing messages over
Timing LSPs, and the second method transports Ethernet encapsulated
timing messages over Ethernet PWs placed in Timing LSPs.

6.1. Timing over UDP/IP over MPLS Encapsulation

The first method directly encapsulates UDP/IP timing messages in a
Timing LSP. The UDP/IP encapsulation of PTP messages MUST comply to
Annex D and E of [IEEE-1588], and the UDP/IP encapsulation of NTP
messages MUST comply to [RFC5905]. This format is shown in Figure 4.

```
          +---------------------+
          |   Timing LSP Label  |
          +---------------------+
          |      IPv4/6         |
          +---------------------+
          |      UDP            |
          +---------------------+
          |   timing message    |
          +---------------------+
```
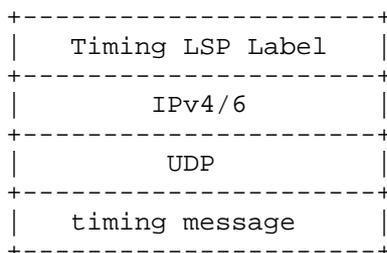
       Figure (4) - Timing over UDP/IP over MPLS Encapsulation

In order for an LER/LSR to process timing messages, the Timing LSP
Label must be the top label of the label stack. The LER/LSR MUST
know that this label is a Timing LSP Label. It can learn this by
static configuration or via RSVP-TE signaling.

6.2. Timing over PW Encapsulation

Another method of transporting timing over MPLS networks is to use
Ethernet encapsulated timing messages, and to transport these in an
Ethernet PW which in turn is transported over a Timing LSP. In the

case of PTP, the Ethernet encapsulation MUST comply to Annex F of
[IEEE-1588] and the Ethernet PW encapsulation to [RFC4448], resulting
in the format shown in Figure 5(A).

Either the Raw mode or Tagged mode defined in [RFC-4448] MAY be used
and the payload MAY have 0, 1, or 2 VLAN tags. The Timing over PW
encapsulation MUST use the Control Word (CW) as specified in
[RFC4448]. The use of Sequence Number in the CW is optional.

NTP MAY be transported using an IP PW (as defined in [RFC4447]) as
shown in Fig 5(B).

```
      +----------------+   +----------------+
      |Timing LSP Label |   |Timing LSP Label|
      +----------------+   +----------------+
      |   PW Label     |   |    PW Label    |
      +----------------+   +----------------+
      |  Control Word  |   |      IP        |
      +----------------+   +----------------+
      |   Ethernet     |   |     UDP        |
      |    Header      |   +----------------+
      +----------------+   | Timing message |
      |S-VLAN (Optional)|  |                |
      +----------------+   +----------------+
      |C-VLAN (Optional)|         (B)
      +----------------+
      |  Timing message |
      |                |
      +----------------+
              (A)
```
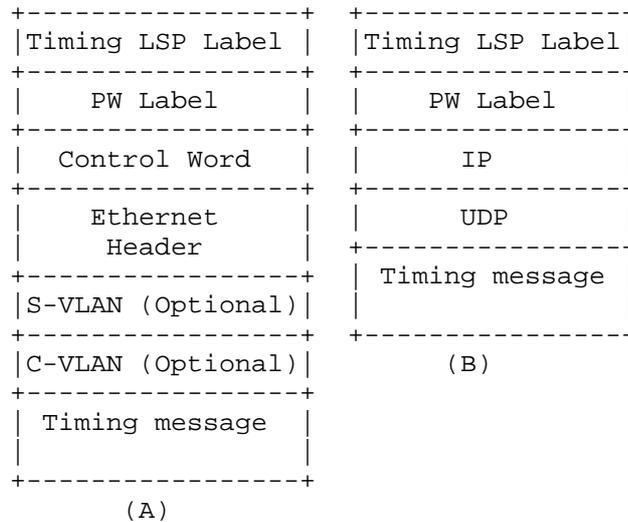
Figure (5) - Timing over PW Encapsulations

7. Timing message Processing

   Each Timing protocol such as PTP and NTP, defines a set of timing
   messages. PTP defines SYNC, DELAY_REQ, DELAY_RESP, FOLLOW_UP, etc.

   Some timing messages require per-packet processing, such as time-
   stamping or CF updating. A compliant LER/LSR parses each timing
   message to determine the required processing.

   For example, the following PTP messages (event messages) require
   time-stamping or CF updating:

   o SYNC

o DELAY_REQ (Delay Request)

o PDELAY_REQ (Peer Delay Request)

o PDELAY_RESP (Peer Delay Response)

SYNC and DELAY_REQ are exchanged between a Master Clock and a Slave
Clock and MUST be transported over Timing LSPs. PDELAY_REQ and
PDELAY_RESP are exchanged between adjacent PTP clocks (master, slave,
boundary, or transparent) and SHOULD be transported over single hop
Timing LSPs. If two-Step PTP clocks are present, then the FOLLOW_UP,
and PDELAY_RESP_FOLLOW_UP messages MUST also be transported over
Timing LSPs.

For a given instance of the 1588 protocol, SYNC and DELAY_REQ MUST be
transported in opposite directions. As aforementioned, two co-routed
unidirectional LSPs or a single bidirectional co-routed LSP MAY be
used.

Except as indicated above for two-step PTP clocks, PTP messages that
are not "event messages" need not be processed by intermediate
routers. These message types MAY be carried in PTP Tunnel LSPs.

8. Protection and Redundancy

In order to ensure continuous uninterrupted operation of timing
distribution, slave clocks often track redundant master clocks.
Prolonged outages of Timing LSPs trigger switching to a redundant
master clock It is the responsibility of the network operator to
ensure that physically disjoint Timing LSPs are established between a
slave clock and redundant master clocks.

LSP or PW layer protection, such as linear protection Switching, ring
protection switching or MPLS Fast Reroute (FRR), will lead to changes
in propagation delay between master and slave clocks. Such a change,
if undetected by the slave clock, would negatively impact timing
performance. While it is expected that slave clocks will often be
able to detect such delay changes, this specification RECOMMENDS that
automatic protection switching NOT be used for Timing LSPs, unless
the operator can ensure that it will not negatively impact timing
performance.

9. ECMP and Entropy

To ensure the correct operation of slave clocks and avoid error
introduced by forward and reverse path delay asymmetry, the physical
path taken by timing messages MUST be the same for all timing

messages. In particular, the PTP event messages listed in section 7
MUST be routed in the same way.

Therefore the Timing LSPs MUST not be subject to ECMP (Equal Cost
Multipath). Entropy labels MUST NOT be used for the Timing LSP
[RFC6790] and MUST NOT be used for PWs inside the Timing LSP
[RFC6391].

10. PHP

To ensure that the label on the top of the label stack is the Timing
LSP Label, PHP MUST not be employed.

11. OAM, Control and Management

In order to monitor Timing LSPs or PWs, it is necessary to enable
them to carry OAM messages. OAM packets MUST be differentiated from
timing messages by already defined IETF methods.

For example BFD [RFC5880], [RFC5884] and LSP-Ping [RFC4389] MAY run
over Timing LSPs via UDP/IP encapsulation or via GAL/G-ACh. These
protocols can easily be identified by the UDP Destination port number
or by GAL/G-ACh respectively.

Also BFD, LSP-Ping and other messages MAY run over Timing PWs via
VCCV [RFC5085]. In this case these messages are recognized according
to the VCCV type.

12. QoS Considerations

There may be deployments where timing messages traverse LSR/LERs that
are not capable of the required processing. In order to minimize the
negative impact on the timing performance of the slave clock timing
messages MUST be treated with the highest priority. This can be
achieved by proper setup of Timing LSPs.

It is recommended that Timing LSPs be configured to indicate EF-PHB
[RFC3246] for the CoS and "green" [RFC2697] for drop eligibility.

13. FCS and Checksum Recalculation

Since Boundary and Transparent Clocks modify packets, when the MPLS
packets are transported over Ethernet the processing MUST include
recalculation of the Ethernet FCS. FCS retention as described in
[RFC4720] MUST NOT be used.

For the UDP/IP encapsulation mode, calculation of the UDP checksum
will generally be required. After updating the CF a Transparent

Clock MUST either incrementally update the UDP checksum or completely
recalculate the checksum before transmission to downstream node.

14. Behavior of LER/LSRs

   Timing-aware LERs or LSRs are MPLS routers that are able to recognize
   timing packets. Timing-capable LERs and LSRs further have one or
   more interfaces that can perform timing processing (OC/BC/TC) on
   timing packets. Timing-capable/aware LERs and LSRs MAY advertise the
   timing capabilities of their interfaces via control plane protocols
   such as OSPF or IS-IS, and timing-aware LERs can then be set up
   Timing LSPs via RSVP-TE signaling. Alternatively the timing
   capabilities of LERs and LSRs may be known by a centralized
   controller or management system, and Timing LSPs may be manually
   configured, or set up by a management platform or a Software Defined
   Networking (SDN) controller.

14.1. Behavior of Timing-capable/aware LERs/LSRs

   When a timing-capable ingress LER acting as a TC receives a timing
   message packet from a timing-capable non-MPLS interface, the LER
   updates the CF, encapsulates and forwards the packet over a
   previously established Timing LSP. When a timing-capable egress LER
   acting as a TC receives a timing message packet on timing-capable
   MPLS interface, the LER updates the CF, decapsulates the MPLS
   encapsulation, and forwards the packet via a non-MPLS interface.
   When a timing-capable LSR acting as a TC receives a timing message
   from a timing-capable MPLS interface, the LSR updates the CF and
   forwards the timing message over another MPLS interface.

   When a timing-capable LER acting as a BC receives a timing message
   packet from a timing-capable interface, the LER time-stamps the
   packet and sends it to the BC processing module.

   When a timing-capable LER acting as an OC receives a timing message
   from a timing-capable MPLS interface, the LER time-stamps the packet
   and sends it to the OC processing module.

14.2. Behavior of non-Timing-capable/aware LSR

   It is most beneficial when all LSRs in the path of a Timing LSP be
   timing-Capable/aware LSRs. This would ensure the highest quality
   time and clock synchronization by slave clocks. However, this
   specification does not mandate that all LSRs in path of a Timing LSP
   be timing-capable/aware.

   Non-timing-capable/aware LSRs just perform label switching on the
   packets encapsulated in Timing LSPs and don't perform any timing

related processing. However, as explained in QoS section, timing
packets MUST be still be treated with the highest priority based on
their Traffic Class marking.

15. Other considerations

   [IEEE-1588] defines an optional peer-to-peer transparent clocking
   (P2P TC) mode that compensates both for residence time in the network
   node and for propagation time on the link between modes. To support
   P2P TC, delay measurement must be performed between two adjacent
   timing-capable/aware LSRs. Thus, in addition to the TC functionality
   detailed above on transit PTP timing messages, adjacent peer to peer
   TCs MUST engage in single-hop peer delay measurement.

   For single hop peer delay measurement a single-hop LSP SHOULD be
   created between the two adjacent LSRs. Other methods MAY be used;
   for example, if the link between the two adjacent routers is
   Ethernet, PTP transport over Ethernet MAY be used.

   To support P2P TC, a timing-capable/ware LSR MUST maintain a list of
   all neighbors to which it needs to send a PDelay_Req, and maintain a
   single-hop timing LSP to each.

   The use of Explicit Null Label (label 0 or 2) is acceptable as long
   as either the Explicit Null label is the bottom of stack label (for
   the UDP/IP encapsulation) or the label below the Explicit Null label
   (for the PW case).

16. Security Considerations

   Security considerations for MPLS and pseudowires are discussed in
   [RFC3985] and [RFC4447]. Security considerations for timing are
   discussed in [RFC7384]. Everything discussed in those documents
   applies to the Timing LSP of this document.

   An experimental security protocol is defined in [IEEE-1588]. The PTP
   security extension and protocol provides group source authentication,
   message integrity, and replay attack protection for PTP messages.

   When the MPLS network (provider network) serves multiple customers,
   it is important to distinguish between timing messages belonging to
   different customers. For example if an LER BC is synchronized to a
   grandmaster belonging to customer A, then the LER MUST only use that
   BC for slaves of customer A, to ensure that customer A cannot
   adversely affect the timing distribution of other customers.

Timing messages MAY be encrypted or authenticated, provided that the timing-capable LERs/LSRs can authenticate/ decrypt the timing messages.

17. Applicability Statement

The Timing over MPLS transport methods described in this document apply to the following network Elements:

o An ingress LER that receives IP or Ethernet encapsulated timing messages from a non-MPLS interface and forwards them as MPLS encapsulated timing messages over Timing LSP, optionally performing TC functionality.

o An egress LER that receives MPLS encapsulated timing messages from a Timing LSP and forwards them to non-MPLS interface as IP or Ethernet encapsulated timing messages, optionally performing TC functionality.

o An ingress LER that receives MPLS encapsulated timing messages from a non-MPLS interface, performs BC functionality, and sends timing messages over a Timing LSP.

o An egress LER that receives MPLS encapsulated timing messages from a Timing LSP, performs BC functionality, and sends timing messages over a non-MPLS interface.

o An LSR on a Timing LSP that receives MPLS encapsulated timing messages from one MPLS interface and forwards them to another MPLS interface, optionally performing TC functionality.

This document also supports the case where not all LSRs are timing-capable/aware, or not all LER/LSR interfaces are timing-capable/ aware.

18. Acknowledgements

The authors would like to thank Yaakov Stein, Luca Martini, Ron Cohen, Tal Mizrahi, Stefano Ruffini, Peter Meyer and other IETF participants for reviewing and providing feedback on this draft.

19. IANA Considerations

There are no IANA requirements in this specification.

20. References

20.1. Normative References

   [IEEE-1588]
              IEEE 1588-2008, "IEEE Standard for a Precision Clock
              Synchronization Protocol for Networked Measurement and
              Control Systems", July 2008.

   [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
              Requirement Levels", BCP 14, RFC 2119,
              DOI 10.17487/RFC2119, March 1997,
              <http://www.rfc-editor.org/info/rfc2119>.

   [RFC3985] Bryant, S., Ed. and P. Pate, Ed., "Pseudo Wire Emulation
              Edge-to-Edge (PWE3) Architecture", RFC 3985,
              DOI 10.17487/RFC3985, March 2005,
              <http://www.rfc-editor.org/info/rfc3985>.

   [RFC4389] Thaler, D., Talwar, M., and C. Patel, "Neighbor Discovery
              Proxies (ND Proxy)", RFC 4389, DOI 10.17487/RFC4389, April
              2006, <http://www.rfc-editor.org/info/rfc4389>.

   [RFC4447] Martini, L., Ed., Rosen, E., El-Aawar, N., Smith, T., and
              G. Heron, "Pseudowire Setup and Maintenance Using the
              Label Distribution Protocol (LDP)", RFC 4447,
              DOI 10.17487/RFC4447, April 2006,
              <http://www.rfc-editor.org/info/rfc4447>.

   [RFC4448] Martini, L., Ed., Rosen, E., El-Aawar, N., and G. Heron,
              "Encapsulation Methods for Transport of Ethernet over MPLS
              Networks", RFC 4448, DOI 10.17487/RFC4448, April 2006,
              <http://www.rfc-editor.org/info/rfc4448>.

   [RFC4720] Malis, A., Allan, D., and N. Del Regno, "Pseudowire
              Emulation Edge-to-Edge (PWE3) Frame Check Sequence
              Retention", RFC 4720, DOI 10.17487/RFC4720, November 2006,
              <http://www.rfc-editor.org/info/rfc4720>.

   [RFC5085] Nadeau, T., Ed. and C. Pignataro, Ed., "Pseudowire Virtual
              Circuit Connectivity Verification (VCCV): A Control
              Channel for Pseudowires", RFC 5085, DOI 10.17487/RFC5085,
              December 2007, <http://www.rfc-editor.org/info/rfc5085>.

   [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection
              (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010,
              <http://www.rfc-editor.org/info/rfc5880>.

   [RFC5884] Aggarwal, R., Kompella, K., Nadeau, T., and G. Swallow,
              "Bidirectional Forwarding Detection (BFD) for MPLS Label
              Switched Paths (LSPs)", RFC 5884, DOI 10.17487/RFC5884,
              June 2010, <http://www.rfc-editor.org/info/rfc5884>.

20.2. Informative References

   [ISO]   ISO/IEC 10589:1992, "Intermediate system to Intermediate
              system routing information exchange protocol for use in
              conjunction with the Protocol for providing the
              Connectionless-mode Network Service (ISO 8473)", April
              1992.

   [RFC1195] Callon, R., "Use of OSI IS-IS for routing in TCP/IP and
              dual environments", RFC 1195, DOI 10.17487/RFC1195,
              December 1990, <http://www.rfc-editor.org/info/rfc1195>.

   [RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328,
              DOI 10.17487/RFC2328, April 1998,
              <http://www.rfc-editor.org/info/rfc2328>.

   [RFC2697] Heinanen, J. and R. Guerin, "A Single Rate Three Color
              Marker", RFC 2697, DOI 10.17487/RFC2697, September 1999,
              <http://www.rfc-editor.org/info/rfc2697>.

   [RFC3246] Davie, B., Charny, A., Bennet, J., Benson, K., Le Boudec,
              J., Courtney, W., Davari, S., Firoiu, V., and D.
              Stiliadis, "An Expedited Forwarding PHB (Per-Hop
              Behavior)", RFC 3246, DOI 10.17487/RFC3246, March 2002,
              <http://www.rfc-editor.org/info/rfc3246>.

   [RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF
              for IPv6", RFC 5340, DOI 10.17487/RFC5340, July 2008,
              <http://www.rfc-editor.org/info/rfc5340>.

   [RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch,
              "Network Time Protocol Version 4: Protocol and Algorithms
              Specification", RFC 5905, DOI 10.17487/RFC5905, June 2010,
              <http://www.rfc-editor.org/info/rfc5905>.

   [RFC6391] Bryant, S., Ed., Filsfils, C., Drafz, U., Kompella, V.,
              Regan, J., and S. Amante, "Flow-Aware Transport of
              Pseudowires over an MPLS Packet Switched Network",
              RFC 6391, DOI 10.17487/RFC6391, November 2011,
              <http://www.rfc-editor.org/info/rfc6391>.

   [RFC6790] Kompella, K., Drake, J., Amante, S., Henderickx, W., and
              L. Yong, "The Use of Entropy Labels in MPLS Forwarding",
              RFC 6790, DOI 10.17487/RFC6790, November 2012,
              <http://www.rfc-editor.org/info/rfc6790>.

   [RFC7384] Mizrahi, T., "Security Requirements of Time Protocols in
              Packet Switched Networks", RFC 7384, DOI 10.17487/RFC7384,
              October 2014, <http://www.rfc-editor.org/info/rfc7384>.

Appendix A. Appendix

A.1. Routing extensions for Timing-aware Routers

   MPLS-TE routing relies on extensions to OSPF [RFC2328] [RFC5340] and
   IS-IS [ISO] [RFC1195] in order to advertise Traffic Engineering (TE)
   link information used for constraint-based routing.

   Timing related capabilities, such as the capability for a router to
   perform time-stamping, and OC, TC or BC processing, need to be
   advertised in order for them to be taken into account during path
   computation. A management system or SDN controller cognizant of
   timing related capabilities, can prefer or even require a Timing LSP
   to traverse links or nodes or intefaces with the required
   capabilities. The optimal path will optimize the performance of the
   slave clock.

   Extensions are required to OSPF and IS-IS in order to advertise
   timing related capabilities of a link. Such extensions are outside
   the scope of this document; however such extensions SHOULD be able to
   signal the following information per Router Link:

   o Capable of processing PTP, NTP or other timing flows

   o Capable of performing TC operation

   o Capable of performing BC operation

A.2. Signaling Extensions for Creating Timing LSPs

   RSVP-TE signaling MAY be used to set up Timing LSPs. Extensions are
   required to RSVP-TE for this purpose. Such extensions are outside
   the scope of this document; however, the following information MAY be
   included in such extensions:

   o Offset from Bottom of Stack (BoS) to the start of the Time-stamp
      field

   o Number of VLANs in case of PW encapsulation

o Time-stamp field Type

   * Correction Field, time-stamp

o Time-stamp Field format

   * 64-bit PTPv1, 80-bit PTPv2, 32-bit NTP, 64-bit NTP, 128-bit
      NTP, etc.

Note that when the above optional information is signaled with RSVP-
TE for a Timing LSP, all the timing packets carried in that LSP must
have the same signaled characteristics. For example if time-stamp
format is signaled as 64-bit PTPv1, then all timing packets must use
64-bit PTPv1 time-stamp.

Authors' Addresses

Shahram Davari
Broadcom Corp.
San Jose, CA 95134
USA

Email: davari@broadcom.com

Amit Oren
Broadcom Corp.
San Jose, CA 95134
USA

Email: amito@broadcom.com

Manav Bhatia
Alcatel-Lucent
Bangalore
India

Email: manav.bhatia@alcatel-lucent.com

Peter Roberts
Alcatel-Lucent
Kanata
Canada

Email: peter.roberts@alcatel-lucent.com

Laurent Montini
Cisco Systems
San Jose CA
USA

Email: lmontini@cisco.com

Network Working Group   A. Shpiner
Internet Draft   Technion - Israel Institute of Technology
Intended status: Experimental   R. Tse
Expires: April 2016   C. Schelp
                                                        PMC-Sierra
                                                        T. Mizrahi
                                                           Marvell
                                                  October 18, 2015

                   Multi-Path Time Synchronization
            draft-ietf-tictoc-multi-path-synchronization-03.txt

Abstract

   Clock synchronization protocols are very widely used in IP-based
   networks. The Network Time Protocol (NTP) has been commonly deployed
   for many years, and the last few years have seen an increasingly
   rapid deployment of the Precision Time Protocol (PTP). As time-
   sensitive applications evolve, clock accuracy requirements are
   becoming increasingly stringent, requiring the time synchronization
   protocols to provide high accuracy. Slave Diversity is a recently
   introduced approach, where the master and slave clocks (also known as
   server and client) are connected through multiple network paths, and
   the slave combines the information received through all paths to
   obtain a higher clock accuracy compared to the conventional one-path
   approach. This document describes a multi-path approach to PTP and
   NTP over IP networks, allowing the protocols to run concurrently over
   multiple communication paths between the master and slave clocks. The
   multi-path approach can significantly contribute to clock accuracy,
   security and fault tolerance. The Multi-Path Precision Time Protocol
   (MPPTP) and Multi-Path Network Time Protocol (MPNTP) define an
   additional layer that extends the existing PTP and NTP without the
   need to modify these protocols. MPPTP and MPNTP also allow backward
   compatibility with nodes that do not support the multi-path
   extension.

Status of this Memo

   This Internet-Draft is submitted to IETF in full conformance with the
   provisions of BCP 78 and BCP 79.

   Internet-Drafts are working documents of the Internet Engineering
   Task Force (IETF), its areas, and its working groups. Note that
   other groups may also distribute working documents as Internet-
   Drafts.

Copyright Notice

Table of Contents

1. Introduction

   The two most common time synchronization protocols in IP networks are
   the Network Time Protocol [NTP], and the Precision Time Protocol
   (PTP), defined in the IEEE 1588 standard [IEEE1588].
   The accuracy of the time synchronization protocols directly depends
   on the stability and the symmetry of propagation delays on both
   directions between the master and slave clocks. Depending on the
   nature of the underlying network, time synchronization protocol
   packets can be subject to variable network latency or path asymmetry
   (e.g. [ASSYMETRY], [ASSYMETRY2]). As time sensitive applications
   evolve, accuracy requirements are becoming increasingly stringent.

   Using a single network path in a clock synchronization protocol
   closely ties the slave clock accuracy to the behavior of the specific
   path, which may suffer from temporal congestion, faults or malicious
   attacks. Relying on multiple clock servers as in NTP solves these
   problems, but requires active maintenance of multiple accurate
   sources in the network, which is not always possible. The usage of
   Transparent Clocks (TC) in PTP solves the congestion problem by
   eliminating the queueing time from the delay calculations, but does
   not address security or fault-tolerance aspects.

```
                                  ____
                        _____/      \_____
                   ___/                     \____
                 ____/                            \
           ____                  /                    /        ___
         /      \       /   _____    \      /    \
        /Master_____/                           _____/Slave\
        \Clock /    /     _____  _____/      \        \Clock/
         \____/        \            path 1               /         \__ /
                        \_____          path 2      __/
                             _____        _____/
                                     _____/
```
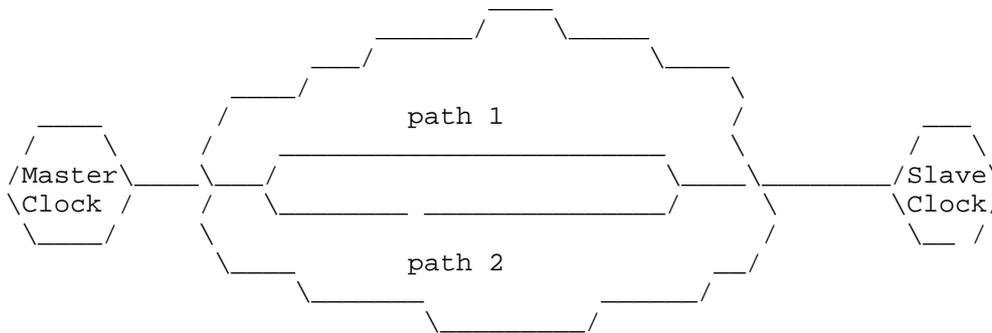
                    Figure 1 Multi-Path Connection

   Since master and slave clocks are often connected through more than
   one path in the network, as shown in Figure 1, [SLAVEDIV] suggested
   that a time synchronization protocol can be run over multiple paths,
   providing several advantages. First, it can significantly increase
   the clock accuracy as shown in [SLAVEDIV]. Second, this approach
   provides additional security, allowing to mitigate man-in-the-middle
   attacks against the time synchronization protocol [DELAY-ATT]. Third,
   using multiple paths concurrently provides an inherent failure
   protection mechanism.

   This document introduces Multi-Path PTP (MPPTP) and Multi-Path NTP
   (MPNTP), respectively. These extensions are defined at the network
   layer and do not require any changes in the PTP or in the NTP
   protocols.

   MPPTP and MPNTP are defined over IP networks. As IP networks
   typically combine ECMP routing, this property is leveraged for the
   multiple paths used in MPPTP and MPNTP. The key property of the
   multi-path extension is that clocks in the network can use more than
   one IP address. Each {master IP, slave IP} address pair defines a
   path. Depending on the network topology and configuration, the IP
   combination pairs can form multiple diverse paths used by the multi-
   path synchronization protocols. It has been shown [MULTI] that using
   multiple IP addresses over the wide Internet indeed allows two end-
   points to attain multiple diverse paths.

   This document introduces two variants for each of the two multi-path
   protocols; a variant that requires both master and slave nodes to
   support the multi-path protocol, referred to as the dual-ended

variant, and a backward compatible variant that allows a multi-path
clock to connect to a conventional single-path clock, referred to as
the single-ended variant.

2. Conventions Used in this Document

2.1. Abbreviations

   BMC   Best Master Clock [IEEE1588]

   ECMP   Equal Cost Multiple Path

   LAN   Local Area Network

   MPNTP   Multi-Path Network Time Protocol

   MPPTP   Multi-Path Precision Time Protocol

   NTP   Network Time Protocol [NTP]

   PTP   Precision Time Protocol [IEEE1588]

2.2. Terminology

   In the NTP terminology, a time synchronization protocol is run
   between a client and a server, while PTP uses the terms master and
   slave. Throughout this document, the sections that refer to both PTP
   and NTP generically use the terms master and slave.

3. Multiple Paths in IP Networks

3.1. Load Balancing

   Traffic sent across IP networks is often load balanced across
   multiple paths. The load balancing decisions are typically based on
   packet header fields: source and destination addresses, Layer 4
   ports, the Flow Label field in IPv6, etc.
   Three common load balancing criteria are per-destination, per-flow
   and per-packet. The per-destination load balancers take a load
   balancing decision based on the destination IP address. Per-flow load
   balancers use various fields in the packet header, e.g., IP addresses
   and Layer 4 ports, for the load balancing decision. Per-packet load
   balancers use flow-blind techniques such as round-robin without
   basing the choice on the packet content.

3.2. Using Multiple Paths Concurrently

   To utilize the diverse paths that traverse per-destination load-
   balancers or per-flow load-balancers, the packet transmitter can vary
   the IP addresses in the packet header. The analysis in [PARIS2] shows
   that a significant majority of the flows on the internet traverse
   per-destination or per-flow load-balancing. It presents statistics
   that 72% of the flows traverse per-destination load balancing and 39%
   of the flows traverse per-flow load-balancing, while only a
   negligible part of the flows traverse per-packet load balancing.
   These statistics show that the vast majority of the traffic on the
   internet is load balanced based on packet header fields.

   The approaches in this draft are based on varying the source and
   destination IP addresses in the packet header. Possible extensions
   have been considered that also vary the UDP ports. However some of
   the existing implementations of PTP and NTP use fixed UDP port values
   in both the source and destination UDP port fields, and thus do not
   allow this approach.

3.3. Two-Way Paths

   A key property of IP networks is that packets forwarded from A to B
   do not necessarily traverse the same path as packets from B to A.
   Thus, we define a two-way path for a master-slave connection as a
   pair of one-way paths: the first from master to slave and the second
   from slave to master.

   If possible, a traffic engineering approach can be used to verify
   that time synchronization traffic is always forwarded through
   bidirectional two-way paths, i.e., that each two-way path uses the
   same route on the forward and reverse directions, thus allowing
   propagation time symmetry. However, in the general case two-way paths
   do not necessarily use the same path for the forward and reverse
   directions.

4. Solution Overview

   The multi-path time synchronization protocols we present are
   comprised of two building blocks; one is the path configuration and
   identification, and the other is the algorithm used by the slave to
   combine the information received from the various paths.

4.1. Path Configuration and Identification

   The master and slave clocks must be able to determine the path of
   transmitted protocol packets, and to identify the path of incoming

protocol packets. A path is determined by a {master IP, slave IP} address pair. The synchronization protocol message exchange is run independently through each path.

Each IP address pair defines a two-way path, and thus allows the clocks to bind a transmitted packet to a specific path, or to identify the path of an incoming packet.

If possible, the routing tables across the network should be configured with multiple traffic engineered paths between the pair of clocks. By carefully configuring the routers in such networks it is possible to create diverse paths for each of the IP address pairs between two clocks in the network. However, in public and provider networks the load balancing behavior is hidden from the end users. In this case the actual number of paths may be less than the number of IP address pairs, since some of the address pairs may share common paths.

4.2. Combining

Various methods can be used for combining the time information received from the different paths. The output of the combining algorithm is the accurate time offset. Combining methods are further discussed in Section 6.

5. Multi-Path Time Synchronization Protocols over IP Networks

This section presents two variants of MPPTP and MPNTP; single-ended multi-path time synchronization and dual-ended multi-path time synchronization. In the first variant, the multi-path protocol is run only by the slave and the master is not aware of its usage. In the second variant, all clocks must support the multi-path protocol.

The dual-ended protocol provides higher path diversity by using multiple IP addresses at both ends, the master and slave, while the single-ended protocol only uses multiple addresses at the slave. On the other hand, the dual-ended protocol can only be deployed when both the master and the slave support this protocol. Dual-ended and single-ended protocols can co-exist in the same network. Each slave selects the connection(s) it wants to make with the available masters. A dual-ended slave could switch to single-ended mode if it does not see any dual-ended masters available. A single-ended slave could connect to a single IP address of a dual-ended master.

Multi-path time synchronization, in both variants, requires clocks to use multiple IP addresses. If possible, the set of IP addresses for each clock should be chosen in a way that enables the establishment

of paths that are the most different. It is applicable if the load
balancing rules in the network are known. Using multiple IP addresses
introduces a tradeoff. A large number of IP addresses allows a large
number of diverse paths, providing the advantages of slave diversity
discussed in Section 1. On the other hand, a large number of IP
addresses is more costly, requires the network topology to be more
redundant, and exacts extra management overhead.

The descriptions in this section refer to the end-to-end scheme of
PTP, but are similarly applicable to the peer-to-peer scheme. The
MPNTP protocol described in this document refers to the NTP client-
server mode, although the concepts described here can be extended to
include the symmetric variant as well.

Multi-path synchronization protocols by nature require protocol
messages to be sent as unicast. Specifically in PTP, the following
messages must be sent as unicast in MPPTP: Sync, Delay_Req,
Delay_Resp, PDelay_Req, PDelay_Resp, Follow_Up, and
PDelay_Resp_Follow_Up. Note that [IEEE1588] allows these messages to
be sent either as multicast or as unicast.

5.1. Single-Ended Multi-Path Synchronization

In the single-ended approach, only the slave is aware of the fact
that multiple paths are used, while the master is agnostic to the
usage of multiple paths. This approach allows a hybrid network, where
some of the clocks are multi-path clocks, and others are conventional
one-path clocks. A single-ended multi-path clock presents itself to
the network as N independent clocks, using N IP addresses, as well as
N clock identity values (in PTP). Thus, the usage of multiple slave
identities by a slave clock is transparent from the master's point of
view, such that it treats each of the identities as a separate slave
clock.

5.1.1. Single-Ended MPPTP Synchronization Message Exchange

The single-ended MPPTP message exchange procedure is as follows.

o Each single-ended MPPTP clock has a fixed set of N IP addresses
   and N corresponding clockIdentities. Each clock arbitrarily
   defines one of its IP addresses and clockIdentity values as the
   clock primary identity.

o A single-ended MPPTP port sends Announce messages only from its
   primary identity, according to the BMC algorithm.

o The BMC algorithm at each clock determines the master, based on
   the received Announce messages.

o A single-ended MPPTP port that is in the 'slave' state uses
   unicast negotiation to request the master to transmit unicast
   messages to each of the N slave clock identities. The slave port
   periodically sends N Signaling messages to the master, using each
   of its N identities. The Signaling message includes the
   REQUEST_UNICAST_TRANSMISSION_TLV.

o The master periodically sends unicast Sync messages from its
   primary identity, identified by the sourcePortIdentity and IP
   address, to each of the slave identities.

o The slave, upon receiving a Sync message, identifies its path
   according to the destination IP address. The slave sends a
   Delay_Req unicast message to the primary identity of the master.
   The Delay_Req is sent using the slave identity corresponding to
   the path the Sync was received through. Note that the rate of
   Delay_Req messages may be lower than the Sync message rate, and
   thus a Sync message is not necessarily followed by a Delay_Req.

o The master, in response to a Delay_Req message from the slave,
   responds with a Delay_Resp message using the IP address and
   sourcePortIdentity from the Delay_Req message.

o Upon receiving the Delay_Resp message, the slave identifies the
   path using the destination IP address and the
   requestingPortIdentity. The slave can then compute the
   corresponding path delay and the offset from the master.

o The slave combines the information from all negotiated paths.

5.1.2. Single-Ended MPNTP Synchronization Message Exchange

   The single-ended MPNTP message exchange procedure is as follows.

o A single-ended MPNTP client has N separate identities, i.e., N IP
   addresses. The assumption is that the server information,
   including its IP address is known to the NTP clients.

o A single-ended MPNTP client initiates the NTP protocol with an NTP
   server N times, using each of its N identities.

o The NTP protocol is maintained between the server and each of the
   N client identities.

o The client sends NTP messages to the master using each of its N
   identities.

o The server responds to the client's NTP messages using the IP
   address from the received NTP packet.

o The client, upon receiving an NTP packet, uses the IP destination
   address to identify the path it came through, and uses the time
   information accordingly.

o The client combines the information from all paths.

5.2. Dual-Ended Multi-Path Synchronization

   In dual-ended multi-path synchronization each clock has N IP
   addresses. Time synchronization messages are exchanged between some
   of the combinations of {master IP, slave IP} addresses, allowing
   multiple paths between the master and slave. Note that the actual
   number of paths between the master and slave may be less than the
   number of chosen {master, slave} IP address pairs.

   Once the multiple two-way connections are established, a separate
   synchronization protocol exchange instance is run through each of
   them.

5.2.1. Dual-Ended MPPTP Synchronization Message Exchange

   The dual-ended MPPTP message exchange procedure is as follows.

o Every clock has N IP addresses, but uses a single clockIdentity.

o The BMC algorithm at each clock determines the master. The master
   is identified by its clockIdentity, allowing other clocks to know
   the multiple IP addresses it uses.

o When a clock sends an Announce message, it sends it from each of
   its IP addresses with its clockIdentity.

o A dual-ended MPPTP port that is in the 'slave' state uses unicast
   negotiation to request the master to transmit unicast messages to
   some or all of its N_s IP addresses. This negotiation is done
   individually between a slave IP address and the corresponding
   master IP address that the slave desires a connection with. The
   slave port periodically sends Signaling messages to the master,
   using some or all of its N_s IP addresses as source, to the
   corresponding master's N_m IP addresses. The Signaling message
   includes the REQUEST_UNICAST_TRANSMISSION_TLV.

o The master periodically sends unicast Sync messages from each of
   its IP addresses to the corresponding slave IP addresses for which
   a unicast connection was negotiated.

o The slave, upon receiving a Sync message, identifies its path
   according to the {source, destination} IP addresses. The slave
   sends a Delay_Req unicast message, swapping the source and
   destination IP addresses from the Sync message. Note that the rate
   of Delay_Req messages may be lower than the Sync message rate, and
   thus a Sync message is not necessarily followed by a Delay_Req.

o The master, in response to a Delay_Req message from the slave,
   responds with a Delay_Resp message using the sourcePortIdentity
   from the Delay_Req message, and swapping the IP addresses from the
   Delay_Req.

o Upon receiving the Delay_Resp message, the slave identifies the
   path using the {source, destination} IP address pair. The slave
   can then compute the corresponding path delay and the offset from
   the master.

o The slave combines the information from all negotiated paths.

5.2.2. Dual-Ended MPNTP Synchronization Message Exchange

The MPNTP message exchange procedure is as follows.

o Each NTP clock has a set of N IP addresses. The assumption is that
   the server information, including its multiple IP addresses is
   known to the NTP clients.

o The MPNTP client chooses N_svr of the N server IP addresses and
   N_c of the N client IP addresses and initiates the N_svr*N_c
   instances of the protocol, one for each {server IP, client IP}
   pair, allowing the client to combine the information from the
   N_s*N_c paths.
   (N_svr and N_c indicate the number of IP addresses of the server
   and client, respectively, which a client chooses to connect with)

o The client sends NTP messages to the master using each of the
   source-destination address combinations.

o The server responds to the client's NTP messages using the IP
   address combination from the received NTP packet.

o Using the {source, destination} IP address pair in the received
   packets, the client identifies the path, and performs its
   computations for each of the paths accordingly.

o The client combines the information from all paths.

5.3. Using Traceroute for Path Discovery

The protocols presented above use multiple IP addresses in a single
clock to create multiple paths. However, although each two-way path
is defined by a different {master, slave} address pair, some of the
IP address pairs may traverse exactly the same network path, making
them redundant. Traceroute-based path discovery can be used for
filtering only the IP addresses that obtain diverse paths. 'Paris
Traceroute' [PARIS] and 'TraceFlow' [TRACEFLOW] are examples of tools
that discover the paths between two points in the network.

The Traceroute-based filtering can be implemented by both master and
slave nodes, or it can be restricted to run only on slave nodes to
reduce the overhead on the master. For networks that guarantee the
path of the timing packets in the forward and reverse direction are
the same, path discovery should only be performed at the slave.

5.4. Using Unicast Discovery for MPPTP

As presented above, MPPTP uses Announce messages and the BMC
algorithm to discover the master. The unicast discovery option of PTP
can be used as an alternative.

When using unicast discovery the MPPTP slave ports maintain a list of
the IP addresses of the master. The slave port uses unicast
negotiation to request unicast service from the master, as follows:

o In single-ended MPPTP, the slave uses unicast negotiation from
   each of its identities to the master's (only) identity.

o In dual-ended MPPTP, the slave uses unicast negotiation from its
   IP addresses, each to a corresponding master IP address to request
   unicast synchronization messages.

Afterwards, the message exchange continues as described in sections
5.1.1. and 5.2.1.

The unicast discovery option can be used in networks that do not
support multicast or in networks in which the master clocks are known
in advance. In particular, unicast discovery avoids multicasting
Announce messages.

6. Combining Algorithm

   Previous sections discussed the methods of creating the multiple
   paths and obtaining the time information required by the slave
   algorithm. Once the time information is received through each of the
   paths, the slave should use a combining algorithm, which consolidates
   the information from the different paths into a single clock.
   Various methods have been suggested for combining information from
   different paths or from different clocks, e.g., [NTP], [SLAVEDIV],
   [HIGH-AVAI], [KALMAN]. The choice of the combining algorithm is local
   to the slave, and does not affect the interoperability of the
   protocol. Hence, this document does not define a specific method to
   be used by the slave.

7. Security Considerations

   The security aspects of time synchronization protocols are discussed
   in detail in [TICTOCSEC]. The methods describe in this document
   propose to run a time synchronization protocol through redundant
   paths, and thus allow to detect and mitigate man-in-the-middle
   attacks, as described in [DELAY-ATT].

8. IANA Considerations

   There are no IANA actions required by this document.

   RFC Editor: please delete this section before publication.

9. Acknowledgments

   The authors gratefully acknowledge the useful comments provided by
   Peter Meyer and Doug Arnold, as well as other comments received from
   the TICTOC working group participants.

   This document was prepared using 2-Word-v2.0.template.dot.

10. References

10.1. Normative References

   [IEEE1588]   IEEE Instrumentation and Measurement Society, "IEEE
                 Standard for a Precision Clock Synchronization
                 Protocol for Networked Measurement and Control
                 Systems", IEEE Std 1588, 2008.

   [NTP]   D. Mills, J. Martin, J. Burbank, W. Kasch, "Network
                      Time Protocol Version 4: Protocol and Algorithms
                      Specification", IETF, RFC 5905, 2010.

10.2. Informative References

   [ASSYMETRY]   Yihua He and Michalis Faloutsos and Srikanth
                      Krishnamurthy and Bradley Huffaker, "On routing
                      asymmetry in the internet", IEEE Globecom, 2005.

   [ASSYMETRY2]   Abhinav Pathak, Himabindu Pucha, Ying Zhang, Y.
                      Charlie Hu, and Z. Morley Mao, "A measurement study of
                      internet delay asymmetry", PAM'08, 2008.

   [DELAY-ATT]   T. Mizrahi, "A Game Theoretic Analysis of Delay
                      Attacks against Time Synchronization Protocols",
                      ISPCS, 2012.

   [HIGH-AVAI]   P. Ferrari, A. Flammini, S. Rinaldi, G. Prytz, "High
                      availability IEEE 1588 nodes over IEEE 802.1 aq
                      Shortest Path Bridging networks" ISPCS, 2013.

   [KALMAN]   G. Giorgi, C. Narduzzi, "Kalman filtering for multi-
                      path network synchronization" ISPCS, 2014.

   [MULTI]   A. Shpiner, Y. Revah, T. Mizrahi, "Multi-path Time
                      Protocols" ISPCS, 2013.

   [PARIS]   Brice Augustin, Timur Friedman and Renata Teixeira,
                      "Measuring Load-balanced Paths in the Internet", IMC,
                      2007.

   [PARIS2]   B. Augustin, T. Friedman, and R. Teixeira, "Measuring
                      Multipath Routing in the Internet", IEEE/ACM
                      Transactions on Networking, 19(3), p. 830 - 840, June
                      2011.

   [SLAVEDIV]   T. Mizrahi, "Slave Diversity: Using Multiple Paths to
                      Improve the Accuracy of Clock Synchronization
                      Protocols", ISPCS, 2012.

   [TICTOCSEC]   T. Mizrahi, "Security Requirements of Time Protocols
                      in Packet Switched Networks", IETF, RFC 7384, 2014.

   [TRACEFLOW]   J. Narasimhan, B. V. Venkataswami, R. Groves and P.
                      Hoose, "Traceflow", IETF, draft-janapath-intarea-
                      traceflow, work in progress, 2012.

Authors' Addresses

   Alex Shpiner
   Department of Electrical Engineering
   Technion - Israel Institute of Technology
   Haifa, 32000 Israel

   Email: shalex@tx.technion.ac.il

   Richard Tse
   PMC-Sierra
   8555 Baxter Place
   Burnaby, BC
   Canada
   V5A 4V7

   Email: Richard.Tse@pmcs.com

   Craig Schelp
   PMC-Sierra
   8555 Baxter Place
   Burnaby, BC
   Canada
   V5A 4V7

   Email: craig.schelp@pmcs.com

   Tal Mizrahi
   Marvell
   6 Hamada St.
   Yokneam, 20692 Israel

   Email: talmi@marvell.com

TICTOC Working Group   Doug Arnold
Internet Draft   Meinberg-USA
Intended status: Standards Track   Heiko Gerstung
                                                        Meinberg
Expires: August 4, 2015   Feb 4, 2015

             Enterprise Profile for the Precision Time Protocol
                 With Mixed Multicast and Unicast Messages

              draft-ietf-tictoc-ptp-enterprise-profile-05.txt

Status of this Memo
     This Internet-Draft is submitted in full conformance with the
     provisions of BCP 78 and BCP 79. This document may not be
     modified, and derivative works of it may not be created, except to
     publish it as an RFC and to translate it into languages other than
     English.

     Internet-Drafts are working documents of the Internet Engineering
     Task Force (IETF), its areas, and its working groups. Note that
     other groups may also distribute working documents as Internet-
     Drafts.

     Internet-Drafts are draft documents valid for a maximum of six
     months and may be updated, replaced, or obsoleted by other
     documents at any time. It is inappropriate to use Internet-Drafts
     as reference material or to cite them other than as "work in
     progress."

     The list of current Internet-Drafts can be accessed at
     http://www.ietf.org/ietf/1id-abstracts.txt

     The list of Internet-Draft Shadow Directories can be accessed at
     http://www.ietf.org/shadow.html

     This Internet-Draft will expire on Agust 4, 2015.

Abstract

   This document describes a profile for the use of the Precision
   Time Protocol in an IPV4 or IPv6 Enterprise information system
   environment. The profile uses the End to End Delay Measurement
   Mechanism, allows both multicast and unicast Delay Request and Delay
   Response Messages.

Table of Contents

1.   Introduction

   The Precision Time Protocol ("PTP"), standardized in IEEE 1588,
   has been designed in its first version (IEEE 1588-2002) with the
   goal to minimize configuration on the participating nodes. Network
   communication was based solely on multicast messages, which unlike
   NTP did not require that a receiving node ("slave clock") in
   [IEEE1588] needs to know the identity of the time sources in the
   network (the Master Clocks).

   The so-called "Best Master Clock Algorithm" ([IEEE1588] Clause
   9.3), a mechanism that all participating PTP nodes must follow,
   set up strict rules for all members of a PTP domain to determine
   which node shall be the active sending time source (Master Clock).
   Although the multicast communication model has advantages in
   smaller networks, it complicated the application of PTP in larger
   networks, for example in environments like IP based
   telecommunication networks or financial data centers. It is
   considered inefficient that, even if the content of a message
   applies only to one receiver, it is forwarded by the underlying

network (IP) to all nodes, requiring them to spend network
bandwidth and other resources like CPU cycles to drop the message.

The second revision of the standard (IEEE 1588-2008) is the
current version (also known as PTPv2) and introduced the
possibility to use unicast communication between the PTP nodes in
order to overcome the limitation of using multicast messages for
the bi-directional information exchange between PTP nodes. The
unicast approach avoided that, in PTP domains with a lot of nodes,
devices had to throw away up to 99% of the received multicast
messages because they carried information for some other node.
PTPv2 also introduced so-called "PTP profiles" ([IEEE1588] Clause
19.3). This construct allows organizations to specify selections
of attribute values and optional features, simplifying the
configuration of PTP nodes for a specific application. Instead of
having to go through all possible parameters and configuration
options and individually set them up, selecting a profile on a PTP
node will set all the parameters that are specified in the profile
to a defined value. If a PTP profile definition allows multiple
values for a parameter, selection of the profile will set the
profile-specific default value for this parameter. Parameters not
allowing multiple values are set to the value defined in the PTP
profile. A number of PTP features and functions are optional and a
profile should also define which optional features of PTP are
required, permitted or prohibited. It is possible to extend the
PTP standard with a PTP profile by using the TLV mechanism of PTP
(see [IEEE1588] Clause 13.4), defining an optional Best Master
Clock Algorithm and a few other ways. PTP has its own management
protocol (defined in [IEEE1588] Clause 15.2) but allows a PTP
profile specify an alternative management mechanism, for example
SNMP.

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL
NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL"
in this document are to be interpreted as described in RFC-2119
[RFC2119].

In this document, these words will appear with that interpretation
only when in ALL CAPS. Lower case uses of these words are not to
be interpreted as carrying RFC-2119 significance.

3. Technical Terms

Acceptable Master Table: A PTP Slave Clock may maintain a list of
masters which it is willing to synchronize to.

Alternate Master: A PTP Master Clock, which is not the Best
Master, may act as a master with the Alternate Master flag set on
the messages it sends.

Announce message: Contains the master clock properties of a Master clock. Used to determine the Best Master.

Best Master: A clock with a port in the master state, operating consistently with the Best Master Clock Algorithm.

Best Master Clock Algorithm: A method for determining which state a port of a PTP clock should be in. The algorithm works by identifying which of several PTP Master capable clocks is the best master. Clocks have priority to become the acting Grandmaster, based on the properties each Master Clock sends in its Announce Message.

Boundary Clock: A device with more than one PTP port. Generally boundary clocks will have one port in slave state to receive timing and then other ports in master state to re-distribute the timing.

Clock Identity: In IEEE 1588-2008 this is a 64-bit number assigned to each PTP clock which must be unique. Often the Ethernet MAC address is used since there is already an international infrastructure for assigning unique numbers to each device manufactured.

Domain: Every PTP message contains a domain number. Domains are treated as separate PTP systems in the network. Slaves, however, can combine the timing information derived from multiple domains.

End to End Delay Measurement Mechanism: A network delay measurement mechanism in PTP facilitated by an exchange of messages between a Master Clock and Slave Clock.

Grandmaster: the primary master clock within a domain of a PTP system

IEEE 1588: The timing and synchronization standard which defines PTP, and describes The node, system, and communication properties necessary to support PTP.

Master clock: a clock with at least one port in the master state.

NTP: Network Time Protocol, defined by RFC 5905, see [NTP].

Ordinary Clock: A clock that has a single Precision Time Protocol (PTP) port in a domain and maintains the timescale used in the domain. It may serve as a master clock, or be a slave clock.

Peer to Peer Delay Measurement Mechanism: A network delay measurement mechanism in PTP facilitated by an exchange of messages between adjacent devices in a network.

Preferred Master: A device intended to act primarily as the Grandmaster of a PTP system, or as a back up to a Grandmaster.

PTP: The Precision Time Protocol, the timing and synchronization
protocol define by IEEE 1588.

PTP port: An interface of a PTP clock with the network. Note that
there may be multiple PTP ports running on one physical interface,
for example a unicast slave which talks to several Grandmaster
clocks in parallel.

PTPv2: Refers specifically to the second version of PTP defined by
IEEE 1588-2008.

Rogue Master: A clock with a port in the master state, even though
it should not be in the master state according to the Best Master
Clock Algorithm, and does not set the alternate master flag.

Slave clock: a clock with at least one port in the slave state,
and no ports in the master state.

Slave Only Clock: An Ordinary clock which cannot become a Master
clock.

TLV: Type Length Value, a mechanism for extending messages in
networked communications.

Transparent Clock. A device that measures the time taken for a
PTP event message to transit the device and then updates the
message with a correction for this transit time.

Unicast Discovery: A mechanism for PTP slaves to establish a
unicast communication with PTP masters using a configures table of
master IP addresses and Unicast Message Negotiation.

Unicast Negotiation: A mechanism in PTP for Slave Clocks to
negotiate unicast Sync, announce and Delay Request Message Rates
from a Master Clock.

4. Problem Statement

This document describes a version of PTP intended to work in large
enterprise networks. Such networks are deployed, for example, in
financial corporations. It is becoming increasingly common in such
networks to perform distributed time tagged measurements, such as
one-way packet latencies and cumulative delays on software
systems spread across multiple computers. Furthermore there is
often a desire to check the age of information time tagged by a
different machine. To perform these measurements it is necessary
to deliver a common precise time to multiple devices on a network.
Accuracy currently required in the Financial Industry range from
100 microseconds to 500 nanoseconds to the Grandmaster. This
profile does not specify timing performance requirements, but such
requirements explain why the needs cannot always be met by NTP, as
commonly implemented. Such accuracy cannot usually be achieved with
a traditional time transfer such as NTP, without adding

non-standard customizations such as hardware time stamping, and on
path support. These features are currently part of PTP, or are
allowed by it. Because PTP has a complex range of features and
options it is necessary to create a profile for enterprise
networks to achieve interoperability between equipment
manufactured by different vendors.

Although enterprise networks can be large, it is becoming
increasingly common to deploy multicast protocols, even across
multiple subnets. For this reason it is desired to make use of
multicast whenever the information going to many destinations is
the same. It is also advantageous to send information which is
unique to one device as a unicast message. The latter can be
essential as the number of PTP slaves becomes hundreds or
thousands.

PTP devices operating in these networks need to be robust. This
includes the ability to ignore PTP messages which can be
identified as improper, and to have redundant sources of time.

5. Network Technology

This PTP profile SHALL operate only in networks characterized by
UDP [RFC768] over either IPv4 [RFC791] or IPv6 [RFC2460], as
described by Annexes D and E in [IEEE1588] respectively. If a
network contains both IPv4 and IPv6, then they SHALL be treated as
separate communication paths. Clocks which communicate using IPv4
can interact with clocks using IPv6 if there is an intermediary
device which simultaneously communicates with both IP versions. A
boundary clock might perform this function, for example. A PTP
domain SHALL use either IPv4 or IPv6 over a communication path,
but not both. The PTP system MAY include switches and routers.
These devices MAY be transparent clocks, boundary clocks, or
neither, in any combination. PTP Clocks MAY be Preferred Masters,
Ordinary Clocks, or Boundary Clocks. The ordinary clocks may be
Slave Only Clocks, or be master capable.

Note that clocks SHOULD always be identified by their clock ID and
not the IP or Layer 2 address. This is important in IPv6 networks
since Transparent clocks are required to change the source address
of any packet which they alter. In IPv4 networks some clocks
might be hidden behind a NAT, which hides their IP addresses from
the rest of the network. Note also that the use of NATs may place
limitations on the topology of PTP networks, depending on the port
forwarding scheme employed. Details of implementing PTP with NATs
are out of scope of this document.

Similar to NTP, PTP makes the assumption that the one way network
delay for Sync Messages and Delay Response Messages are the same.
When this is not true it can cause errors in the transfer of time
from the Master to the Slave. It is up to the system integrator to
design the network so that such effects do not prevent the PTP
system from meeting the timing requirements. The details of

network asymmetry are outside the scope of this document. See for
example, [G8271].

6. Time Transfer and Delay Measurement

   Master clocks, Transparent clocks and Boundary clocks MAY be
   either one-step clocks or two-step clocks. Slave clocks MUST
   support both behaviors. The End to End Delay Measurement Method
   MUST be used.

   Note that, in IP networks, Sync messages and Delay Request
   messages exchanged between a master and slave do not necessarily
   traverse the same physical path. Thus, wherever possible, the
   network SHOULD be traffic engineered so that the forward and
   reverse routes traverse the same physical path. Traffic
   engineering techniques for path consistency are out of scope of
   this document.

   Sync messages MUST be sent as PTP event multicast messages (UDP
   port 319) to the PTP primary IP address.   Two step clocks SHALL
   send Follow-up messages as PTP general messages (UDP port 320).
   Announce messages MUST be sent as multicast messages (UDP port 320)
   to the PTP primary address. The PTP primary IP address is
   224.0.1.129 for IPv4 and FF0X:0:0:0:0:0:0:181 for Ipv6, where X can
   be a value between 0x0 and 0xF, see [IEEE1588] Annex E, Section
   E.3.

   Delay Request Messages MAY be sent as either multicast or unicast
   PTP event messages. Master clocks SHALL respond to multicast Delay
   Request messages with multicast Delay Response PTP general
   messages. Master clocks SHALL respond to unicast Delay Request PTP
   event messages with unicast Delay Response PTP general messages.
   This allow for the use of Ordinary clocks which do not support the
   Enterprise Profile, as long as they are slave Only Clocks.

   Clocks SHOULD include support for multiple domains. The purpose is
   to support multiple simultaneous masters for redundancy. Leaf
   devices (non-forwarding devices) can use timing information from
   multiple masters by combining information from multiple
   instantiations of a PTP stack, each operating in a different
   domain. Redundant sources of timing can be ensembled, and/or
   compared to check for faulty master clocks. The use of multiple
   simultaneous masters will help mitigate faulty masters reporting as
   healthy, network delay asymmetry, and security problems. Security
   problems include man-in-the-middle attacks such as delay attacks,
   packet interception / manipulation attacks. Assuming the path to
   each master is different, failures malicious or otherwise would
   have to happen at more than one path simultaneously. Whenever
   feasible, the underlying network transport technology SHOULD be
   configured so that timing messages in different domains traverse
   different network paths.

7. Default Message Rates

   The Sync, Announce and Delay Request default message rates SHALL
   each be once per second. The Sync and Delay Request message rates
   MAY be set to other values, but not less than once every 128
   seconds, and not more than 128 messages per second. The Announce
   message rate SHALL NOT be changed from the default value. The
   Announce Receipt Timeout Interval SHALL be three Announce
   Intervals for Preferred Masters, and four Announce Intervals for
   all other masters. Unicast Discovery and Unicast Message
   Negotiation options MUST NOT be utilized.

8. Requirements for Master Clocks

   Master clocks SHALL obey the standard Best Master Clock Algorithm
   from [IEEE1588]. PTP systems using this profile MAY support
   multiple simultaneous Grandmasters as long as each active
   Grandmaster is operating in a different PTP domain.

   A port of a clock SHALL NOT be in the master state unless the
   clock has a current value for the number of UTC leap
   seconds. A clock with a port in the master state SHOULD indicate
   the maximum adjustment to its internal clock within one sync
   interval. The maximum phase adjustment is indicated in the
   Enterprise Profile announce TLV field for Maximum Phase Adjustment.

   The Announce Messages SHALL include a TLV which indicates that the
   clock is operating in the Enterprise Profile. The TLV shall have
   the following structure:

   TLV Type (Enumeration16): ORGANIZATION_EXTENSION value = 0003 hex

   Length Field (UInteger16): value = 10. The number of TLV octets

   Organization Unique Identifier (3 Octets): The Organization ID
   value for IETF assigned by IEEE = 00005Ehex

   IETF Profile number (UInteger8): value = 1

   Revision number (UInteger8): value = 1

   Port Number (UInteger16): The Port Number of the port transmitting
   the TLV. The all-ones Port Number, with value FFFFhex, is used to
   indicate that the identified profile is applicable to all ports on
   the clock.

   Maximum Absolute Phase Adjustment Value within one sync interval
   (UInteger16): value

Maximum Phase Adjustment Units (Enumeration8):
            Value 0 = unknown
            Value 1 = seconds
            Value 3 = milliseconds
            Value 6 = microseconds
            Value 9 = nanoseconds
            Value 12 = picoseconds
            Value 15 = femtoseconds
            All other values reserved for future use

Slaves can use the Maximum Phase Adjustment to determine if the
clock is slewing to rapidly for the applications which are of
interest. For example if the clock set by slave is used to
measure time intervals then it might be desired that that the
amount which the time changes during the intervals is limited.

9.   Requirements for Slave Clocks

Slave clocks MUST be able to operate properly in a network which
contains multiple Masters in multiple domains. Slaves SHOULD make
use of information from the all Masters in their clock control
subsystems. Slave Clocks MUST be able to operate properly in the
presence of a Rogue Master. Slaves SHOULD NOT Synchronize to a
Master which is not the Best Master in its domain. Slaves will
continue to recognize a Best Master for the duration of the
Announce Time Out Interval. Slaves MAY use an Acceptable Master
Table. If a Master is not an Acceptable Master, then the Slave
MUST NOT synchronize to it. Note that IEEE 1588-2008 requires
slave clocks to support both two-step or one-step Master clocks.
See [IEEE1588], section 11.2.

Since Announce messages are sent as multicast messages slaves can
obtain the IP addresses of master from the Announce messages. Note
that the IP source addresses of Sync and Follow-up messages may
have been replaced by the source addresses of a transparent clock,
so slaves MUST send Delay Request messages to the IP address in the
Announce message. Sync and Follow-up messages can be correlated
with the Announce message using the clock ID, which is never
altered by Transparent clocks in this profile.

10.   Requirements for Transparent Clocks

Transparent clocks SHALL NOT change the transmission mode of an
Enterprise Profile PTP message. For example a Transparent clock
SHALL NOT change a unicast message to a multicast message.
Transparent clocks SHALL NOT alter the Enterprise Profile TLV of
an Announce message, or any other part of an Announce message.
Transparent Clocks SHOULD support multiple domains. Transparent
Clocks which syntonize to the master clock will need to maintain
separate clock rate offsets for each of the supported domains.

11.   Requirements for Boundary Clocks

    Boundary Clocks SHOULD support multiple simultaneous PTP domains.
    This will require them to maintain servo loops for each of the
    domains supported, at least in software. Boundary clocks MUST NOT
    combine timing information from different domains.

12.   Management and Signaling Messages

    PTP Management messages MAY be used. Any PTP management message
    which is sent with the targetPortIdentity.clockIdentity set to all
    1s (all clocks) MUST be sent as a multicast message. Management
    messages with any other value of for the Clock Identity is
    intended for a specific clock and MUST be sent as a unicast
    message. Similarly, if any signaling messages are used they
    MUST also be sent as unicast messages whenever the message is
    intended for a specific clock.

13.   Forbidden PTP Options

    Clocks operating in the Enterprise Profile SHALL NOT use peer to
    peer timing for delay measurement. Clocks operating in the
    Enterprise Profile SHALL NOT use Unicast Message Negotiation to
    determine message rates. Slave clocks operating in the Enterprise
    Profile SHALL NOT use Unicast Discovery to establish connection to
    Master clocks. Grandmaster Clusters are NOT ALLOWED. The Alternate
    Master option is also forbidden. Clocks operating in the Enterprise
    Profile SHALL NOT use Alternate Timescales.

14.   Interoperation with Other PTP Profiles

    Clocks operating in the Enterprise Profile will not interoperate
    with clocks operating in the Power Profile [C37.238], because the
    Enterprise Profile requires the End to End Delay Measurement
    Mechanism and the Power Profile requires the Peer to Peer Delay
    Measurement Mechanism.

    Clocks operating in the Enterprise Profile will not interoperate
    with clocks operating in the Telecom Profile for Frequency
    Synchronization[G8265.1], because the Enterprise Profile forbids
    Unicast Message Negotiation. This feature is part of the default
    manner of operation for the Telecom Profile for Frequency
    Synchronization.

    Clocks operating in the Enterprise Profile will interoperate with
    clocks operating in the Default Profile described in [IEEE1588]
    Annex J.3. This variant of the Default Profile uses the End to End
    Delay Measurement Mechanism. In addition the Default Profile would
    have to operates over IPv4 or IPv6 networks, and use management
    messages in unicast when those messages are directed at a specific
    clock. If either of these requirements are not met than Enterprise

Profile clocks will not interoperate with Annex J.3 Default Profile
Clocks. The Enterprise Profile Profile will will not interoperate
with the Annex J.4 variant of the Default Profile which requires
use of the Peer to Peer Delay Measurement Mechanism.

Enterprise Profile Clocks will interoperate with clocks operating
in other profiles if the clocks in the other profiles obey the
rules of the Enterprise Profile. These rules MUST NOT be changed
to achieve interoperability with other profiles.

15.   Security Considerations

Protocols used to transfer time, such as PTP and NTP can be
important to security mechanisms which use time windows for keys
and authorization. Passing time through the networks poses a
security risk since time can potentially be manipulated.
The use of multiple simultaneous masters, using multiple PTP
domains can mitigate problems from rogue masters and
man-in-the-middle attacks. See sections 9 and 10. Additional
security mechanisms are outside the scope of this document.

16.   IANA Considerations

There are no IANA requirements in this specification.

17.   References

17.1.   Normative References

[IEEE1588] IEEE std. 1588-2008, "IEEE Standard for a
           Precision Clock Synchronization for Networked
           Measurement and Control Systems." July, 2008.
[RFC768]   Postel, J., "User Datagram Protocol," RFC 768,
           August, 980.

[RFC791]   "Internet Protocol DARPA Internet Program Protocol
           Specification," RFC 791, September, 1981.

[RFC2119] Bradner, S., "Key words for use in RFCs to
           Indicate Requirement Levels", BCP 14, RFC 2119,
           March 1997.

[RFC2460] Deering, S., Hinden, R., "Internet Protocol,
           Version 6 (IPv6) Specification," RFC 2460,
           December, 1998.

17.2.   Informative References

      [C37.238]   IEEE std. C37.238-2011, "IEEE Standard Profile for
                  Use of IEEE 1588 Precision Time Protocol in Power
                  System Applications," July 2011.

      [G8265.1]   ITU-T G.8265.1/Y.1365.1, "Precision time protocol
                  telecom profile for frequency synchronization,"
                  October 2011.

      [G8271]   ITU-T G.8271/Y.1366, "Time and Phase
                  Synchronization Aspects of Packet Networks"
                  February, 2012.

      [NTP]   Mills, D., Martin, J., Burbank, J., Kasch, W.,
                  "Network Time Protocol Version 4: Protocol and
                  Algorithms Specification," RFC 5905, June 2010.

18.   Acknowledgments

   The authors would like to thank members of IETF for reviewing and
   providing feedback on this draft.

   This document was initially prepared using
   2-Word-v2.0.template.dot.

19.   Authors' Addresses

   Doug Arnold
   Meinberg USA
   228 Windsor River Rd
   Windsor, CA 95492
   USA

   Email: doug.arnold@meinberg-usa.com

   Heiko Gerstung
   Meinberg Funkuhren GmbH & Co. KG
   Lange Wand 9
   D-31812 Bad Pyrmont
   Germany

   Email: Heiko.gerstung@meinberg.de

TICTOC Working Group   Vinay Shankarkumar
INTERNET DRAFT   Laurent Montini
Intended status: Standards Track   Cisco Systems

                                                       Tim Frost
                                            Calnex Solutions Ltd.

                                                       Greg Dowd
                                                       Microsemi

                Precision Time Protocol Version 2 (PTPv2)
                       Management Information Base
                     draft-ietf-tictoc-ptp-mib-07.txt

Status of this Memo

   This Internet-Draft is submitted to IETF in full conformance with the
   provisions of BCP 78 and BCP 79.

   Internet-Drafts are working documents of the Internet Engineering
   Task Force (IETF), its areas, and its working groups. Note that
   other groups may also distribute working documents as Internet-
   Drafts.

   Internet-Drafts are draft documents valid for a maximum of six months
   and may be updated, replaced, or obsoleted by other documents at any
   time. It is inappropriate to use Internet-Drafts as reference
   material or to cite them other than as "work in progress."

   The list of current Internet-Drafts can be accessed at
   http://www.ietf.org/ietf/1id-abstracts.txt

   The list of Internet-Draft Shadow Directories can be accessed at
   http://www.ietf.org/shadow.html

   This Internet-Draft will expire on September 24, 2015.

(http://trustee.ietf.org/license-info) in effect on the date of
publication of this document. Please review these documents
carefully, as they describe your rights and restrictions with respect
to this document. Code Components extracted from this document must
include Simplified BSD License text as described in Section 4.e of
the Trust Legal Provisions and are provided without warranty as
described in the Simplified BSD License.

Abstract

   This memo defines a portion of the Management Information Base (MIB)
   for use with network management protocols in TCP/IP-based internets.
   In particular, it defines objects for managing networks using
   Precision Time Protocol, specified in IEEE Std. 1588(TM)-2008.

   This memo specifies a MIB module in a manner that is both compliant
   to the SNMPv2 SMI, and semantically identical to the peer SNMPv1
   definitions.

Table of Contents

1. Introduction

   This memo defines a portion of the Management Information Base (MIB)
   for use with network management protocols in the Internet Community.
   In particular, it describes managed objects used for managing PTP
   devices including the ordinary clock, transparent clock, boundary
   clocks.

   This MIB is restricted to reading standard PTP data elements, as
   described in [IEEE 1588-2008]. This enables it to monitor the
   operation of PTP clocks within the network. It is envisioned this
   MIB will complement other managed objects to be defined that will
   provide more detailed information on the performance of PTP clocks

supporting the Telecom Profile defined in [G.8265.1], and any future
profiles that may be defined. Those objects are considered out of
scope for the current draft.

Similarly, this MIB is read-only and not intended to provide the
ability to configure PTP clocks. Since PTP clocks are often embedded
in other network elements such as routers, switches and gateways,
this ability is generally provided via the configuration interface
for the network element.

1.1. Relationship to other Profiles and MIBs

This MIB is intended to be used with the default PTP profile
described in [IEEE 1588-2008] when running over the IP network layer.
As stated above, it is envisioned this MIB will complement other
managed objects to be defined to monitor and measure the performance
of PTP clocks supporting specific PTP profiles, e.g. the Telecom
Profile defined in [G.8265.1].

Some other PTP profiles have their own MIBs defined as part of the
profile, and this MIB is not intended to replace those MIBs.

1.2. Change Log

This section tracks changes made to the revisions of the Internet
Drafts of this document. It will be *deleted* when the document is
published as an RFC.

draft-vinay-tictoc-ptp-mib

-00 Mar 11 Initial version; showed structure of MIB

draft-ietf-tictoc-ptp-mib

-00 Jul 11 First full, syntactically correct and compileable MIB

-01 Jan 12 Revised following comments from Bert Wijnen:
             - revised introduction to clarify the scope, and the
               relationship to other MIBs and profiles
             - changed name to "ptpbase"
             - corrected some data types
             - corrected references and typos

-02 Jul 12 Revised following comment at IETF83:
             - changed "ptpbaseClockPortRunningIPversion" to the more
               generic "ptpbaseClockPortRunningTransport", covering
               all transport types defined in [IEEE 1588-2008]
               (i.e. IPv4, IPv6, Ethernet, DeviceNet and ControlNet).
             - changed addresses associated with transports from

                    "InetAddress" (for the IP transport) to a string,
                    to allow for the different transport types.

      -03   Jul 12 Minor changes following comments from Andy Bierman:
                    - corrected some compilation errors
                    - moved OBJECT-GROUP and MODULE-COMPLIANCE macros to the
                    end

      -04   Jan 13 Changes:
                    - Use of 'AutonomousType' import
                    - Display hint being specified for ClockIdentity,
                    ClockInterval, ClockPortTransportTypeAddress
                    Textual Conventions
                    - Removal of the Textual convention
                    ClockPortTransportType, replaced with the
                    wellKnownTransportTypes
                    - Modified ptpbaseClockPortCurrentPeerAddressType,
                    ptpbaseClockPortRunningTransport,
                    ptpbaseClockPortAssociateAddressType, to use
                    AutonomousType.
                    - various textual changes to descriptive text in
                    response to comments

      -05   Feb 13 Several changes in response to comments from Alun Luchuk
                    and Kevin Gross:
                    - Modified the use of wellKnownTransportTypes and
                    wellKnownEncapsulationTypes
                    - changed ptpbaseClockPortSyncOneStep to
                    ptpbaseClockPortSyncTwoStep to match IEEE1588
                    semantics
                    - Re-ordered textual conventions to be alphabetic
                    - Changed some types from Integer32 to use defined
                    textual conventions
                    - various minor descriptive text changes

      -06   Mar 14 Updated author information, and fixed typos

      -07   Mar 15 Updated author information, and fixed typo/enum

2. The SNMP Management Framework

   The SNMP Management Framework presently consists of five major
   components:

   o An overall architecture, described in STD62, [RFC 3411].

   o Mechanisms for describing and naming objects and events for the
      purpose of management. The first version of this Structure of

   Management Information (SMI) is called SMIv1 and described in
   STD 16: [RFC 1155], [RFC 1212] and [RFC 1215].
   The second version, called SMIv2, is described in STD 58:
   [RFC 2578], [RFC 2579] and [RFC 2580].

   o Message protocols for transferring management information. The
      first version of the SNMP message protocol is called SNMPv1 and
      described in STD 15 [RFC 1157]. A second version of the SNMP
      message protocol, which is not an Internet standards track
      protocol, is called SNMPv2c and described in [RFC 1901] and
      [RFC 1906]. The third version of the message protocol is called
      SNMPv3 and described in STD62: [RFC 3417], [RFC 3412] and [RFC
      3414].

   o Protocol operations for accessing management information. The
      first set of protocol operations and associated PDU formats is
      described in STD 15 [RFC 1157]. A second set of protocol
      operations and associated PDU formats is described in STD 62
      [RFC 3416].

   o A set of fundamental applications described in STD 62 [RFC 3413]
      and the view-based access control mechanism described in STD 62
      [RFC 3415].

   Managed objects are accessed via a virtual information store, termed
   the Management Information Base or MIB. Objects in the MIB are
   defined using the mechanisms defined in the SMI.

   This memo specifies a MIB module that is compliant to the SMIv2. A
   MIB conforming to the SMIv1 can be produced through the appropriate
   translations. The resulting translated MIB must be semantically
   equivalent, except where objects or events are omitted because no
   translation is possible (e.g., use of Counter64). Some machine
   readable information in SMIv2 will be converted into textual
   descriptions in SMIv1 during the translation process. However, this
   loss of machine readable information is not considered to change the
   semantics of the MIB.

3. Overview

   The objects defined in this MIB are to be used when describing the
   Precision Time Protocol (PTPv2).

4. IETF PTP MIB Definition

PTPBASE-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY,
    OBJECT-TYPE,
    OBJECT-IDENTITY,
    Gauge32,
    Unsigned32,
    Counter32,
    Counter64,
    mib-2,
    Integer32
        FROM SNMPv2-SMI
    OBJECT-GROUP,
    MODULE-COMPLIANCE
        FROM SNMPv2-CONF
    TEXTUAL-CONVENTION,
    TruthValue,
    DisplayString,
    AutonomousType
        FROM SNMPv2-TC
    InterfaceIndexOrZero
        FROM IF-MIB;

ptpbaseMIB MODULE-IDENTITY
    LAST-UPDATED   "201207230000Z"
    ORGANIZATION   "TICTOC Working Group"
    CONTACT-INFO
          "WG Email: tictoc@ietf.org

           Vinay Shankarkumar
           Cisco Systems,
           Email: vinays@cisco.com

           Laurent Montini,
           Cisco Systems,
           Email: lmontini@cisco.com

           Tim Frost,
           Calnex Solutions Ltd.,

          Email: tim.frost@calnexsol.com

          Greg Dowd,
          Microsemi Inc.,
          Email: greg.dowd@microsemi.com"

     DESCRIPTION
          "The MIB module for PTP version 2 (IEEE Std. 1588(TM)-2008)

          Overview of PTP version 2 (IEEE Std. 1588(TM)-2008)

          [IEEE 1588-2008] defines a protocol enabling precise
          synchronization of clocks in measurement and control systems
          implemented with packet-based networks, the Precision Time
          Protocol Version 2 (PTPv2). This MIB does not address the
          earlier version IEEE Std. 1588(TM)-2002 (PTPv1). The protocol
          is applicable to network elements communicating using IP. The
          protocol enables heterogeneous systems that include clocks of
          various inherent precision, resolution, and stability to
          synchronize to a grandmaster clock.

          The protocol supports system-wide synchronization accuracy in
          the sub-microsecond range with minimal network and local clock
          computing resources. [IEEE 1588-2008] uses UDP/IP or
          Ethernet and can be adapted to other mappings. It includes
          formal mechanisms for message extensions, higher sampling rates,
          correction for asymmetry, a clock type to reduce error
          accumulation in large topologies, and specifications on how to
          incorporate the resulting additional data into the
          synchronization protocol. The [IEEE 1588-2008] defines
          conformance and management capability also.

          MIB description

          This MIB is to support the Precision Time Protocol version 2
          (PTPv2, hereafter designated as PTP) features of network element
          system devices, when using the default PTP profile described in
          [IEEE 1588-2008] when running over the IP network layer.

          It is envisioned this MIB will complement other managed objects
          to be defined to monitor and measure the performance of the PTP
          devices and telecom clocks supporting specific PTP profiles.

Some other PTP profiles have their own MIBs defined as part of
the profile, and this MIB is not intended to replace those MIBs.

Acronyms:
          ARB   Arbitrary Timescale
          E2E   End-to-End
          EUI   Extended Unique Identifier.
          GPS   Global Positioning System
          IANA   Internet Assigned Numbers Authority
          IP   Internet Protocol
          MAC   Media Access Control
                    according to [IEEE 802.3-2008]
          NIST   National Institute of Standards and Technology
          NTP   Network Time Protocol (see IETF [RFC 5905])
          OUI   Organizational Unique Identifier
                    (allocated by the IEEE)
          P2P   Peer-to-Peer
          PTP   Precision Time Protocol
          TAI   International Atomic Time
          TC   Transparent Clock
          UDP   User Datagram Protocol
          UTC   Coordinated Universal Time

References:
[IEEE 1588-2008] IEEE Standard for A Precision Clock
     Synchronization Protocol for Networked Measurement and
     Control Systems, IEEE Std. 1588(TM)-2008, 24 July 2008.

As defined in [IEEE 1588-2008]:

Accuracy:
The mean of the time or frequency error between the clock under
test and a perfect reference clock, over an ensemble of
measurements. Stability is a measure of how the mean varies
with respect to variables such as time, temperature, and so on,
while the precision is a measure of the deviation of the error
from the mean.

Atomic process:
A process is atomic if the values of all inputs to the process
are not permitted to change until all of the results of the

process are instantiated, and the outputs of the process are
not visible to other processes until the processing of each
output is complete.

Boundary clock:
A clock that has multiple Precision Time Protocol (PTP) ports in
a domain and maintains the timescale used in the domain. It
may serve as the source of time, i.e., be a master clock, and
may synchronize to another clock, i.e., be a slave clock.

Boundary node clock:
A clock that has multiple Precision Time Protocol(PTP) ports in
a domain and maintains the timescale used in the domain. It
differs from a boundary clock in that the clock roles can
change.

Clock:
A node participating in the Precision Time Protocol (PTP) that
is capable of providing a measurement of the passage of time
since a defined epoch.

Domain:
A logical grouping of clocks that synchronize to each other
using the protocol, but that are not necessarily synchronized
to clocks in another domain.

End-to-end transparent clock:
A transparent clock that supports the use of the end-to-end
delay measurement mechanism between slave clocks and the master
clock. Each node must measure the residence time of PTP event
messages and accumulate it in Correction Field.

Epoch:
The origin of a timescale.

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

Foreign master:
An ordinary or boundary clock sending Announce messages to
another clock that is not the current master recognized by the
other clock.

Grandmaster clock:
Within a domain, a clock that is the ultimate source of time
for clock synchronization using the protocol.

Holdover:
A clock previously synchronized/syntonized to another clock
(normally a primary reference or a master clock) but now
free-running based on its own internal oscillator, whose
frequency is being adjusted using data acquired while it had
been synchronized/syntonized to the other clock. It is said to
be in holdover or in the holdover mode, as long as it is within
its accuracy requirements.

Link:
A network segment between two Precision Time Protocol ports
supporting the peer delay mechanism of this standard. The peer
delay mechanism is designed to measure the propagation time
over such a link.

Management node:
A device that configures and monitors clocks.

Master clock:
In the context of a single Precision Time Protocol
communication path, a clock that is the source of time to which
all other clocks on that path synchronize.

Message timestamp point:
A point within a Precision Time Protocol event message serving
as a reference point in the message. A timestamp is defined by
the instant a message timestamp point passes the reference
plane of a clock.

Multicast communication:
A communication model in which each Precision Time Protocol
message sent from any PTP port is capable of being received and
processed by all PTP ports on the same PTP communication path.

Node:
A device that can issue or receive Precision Time Protocol
communications on a network.

One-step clock:
A clock that provides time information using a single event
message.

On-pass support:
Indicates that each node in the synchronization chain from
master to slave can support IEEE-1588.

Ordinary clock:
A clock that has a single Precision Time Protocol port in a
domain and maintains the timescale used in the domain. It may
serve as a source of time, i.e., be a master clock, or may
synchronize to another clock, i.e., be a slave clock.

Parent clock:
The master clock to which a clock is synchronized.

Peer-to-peer transparent clock:
A transparent clock that, in addition to providing Precision
Time Protocol event transit time information, also provides
corrections for the propagation delay of the link connected to
the port receiving the PTP event message. In the presence of
peer-to-peer transparent clocks, delay measurements between
slave clocks and the master clock are performed using the
peer-to-peer delay measurement mechanism.

Phase change rate:
The observed rate of change in the measured time with respect
to the reference time. The phase change rate is equal to the
fractional frequency offset between the measured frequency and
the reference frequency.

PortNumber:
An index identifying a specific Precision Time Protocol port on
a PTP node.

Primary reference:
A source of time and or frequency that is traceable to
international standards.

Profile:
The set of allowed Precision Time Protocol features applicable
to a device.

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

Precision Time Protocol communication path:
The signaling path portion of a particular network enabling
direct communication among ordinary and boundary clocks.

Precision Time Protocol node:
PTP ordinary, boundary, or transparent clock or a device that
generates or parses PTP messages.

Precision Time Protocol port:
A logical access point of a clock for PTP communications to the
communications network.

Recognized standard time source:
A recognized standard time source is a source external to
Precision Time Protocol that provides time and/or frequency as
appropriate that is traceable to the international standards
laboratories maintaining clocks that form the basis for the
International Atomic Time and Universal Coordinated Time
timescales. Examples of these are GPS, NTP, and NIST
timeservers.

Requestor:
The port implementing the peer-to-peer delay mechanism that
initiates the mechanism by sending a Pdelay_Req message.

Responder:
The port responding to the receipt of a Pdelay_Req message as
part of the operation of the peer-to-peer delay mechanism.

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

Syntonized clocks:
Two clocks are syntonized if the duration of the second is the
same on both, which means the time as measured by each advances

at the same rate. They may or may not share the same epoch.

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

Timescale:
A linear measure of time from an epoch.

Traceability:
A property of the result of a measurement or the value of a
standard whereby it can be related to stated references,
usually national or international standards, through an
unbroken chain of comparisons all having stated uncertainties.

Translation device:
A boundary clock or, in some cases, a transparent clock that
translates the protocol messages between regions implementing
different transport and messaging protocols, between different
versions of [IEEE 1588-2008], or different PTP profiles.

Transparent clock:
A device that measures the time taken for a Precision Time
Protocol event message to transit the device and provides this
information to clocks receiving this PTP event message.

Two-step clock:
A clock that provides time information using the combination of
an event message and a subsequent general message.

The below table specifies the object formats of the various
textual conventions used.

Data type mapping	Textual Convention	SYNTAX
5.3.2 TimeInterval	ClockTimeInterval	OCTET STRING(SIZE(1..255))
5.3.3 Timestamp	ClockTimestamp	OCTET STRING(SIZE(6))
5.3.4 ClockIdentity	ClockIdentity	OCTET STRING(SIZE(1..255))
5.3.5 PortIdentity	ClockPortNumber	INTEGER(1..65535)
5.3.7 ClockQuality	ClockQualityClassType	

Simple master-slave hierarchy, section 6.6.2.4 [IEEE 1588-2008]:

```
                    +---------------+
                    |   Ordinary    |
                    |   Clock -1    |
                    | (GrandMaster) |
                    +-------M-------+
                            |
                            1
                            |
          +---------------S-----------------+
          |              Boundary           |
          |              Clock -1           |
          +-----M--------------------M-----+
                |                     |
                2                     3
                |                     |
          +-----S-----+  +-----------S------------+
          | Ordinary  |  |         Boundary       |
          | Clock -2  |  |         Clock -2       |
          +----------+  +-----M-------------M-----+
                              |             |
                              4             5
                              |             |
                        +-----S-----+  +-----S-----+
                        | Ordinary  |  | Ordinary  |
                        | Clock -3  |  | Clock -4  |
                        +----------+  +----------+
```
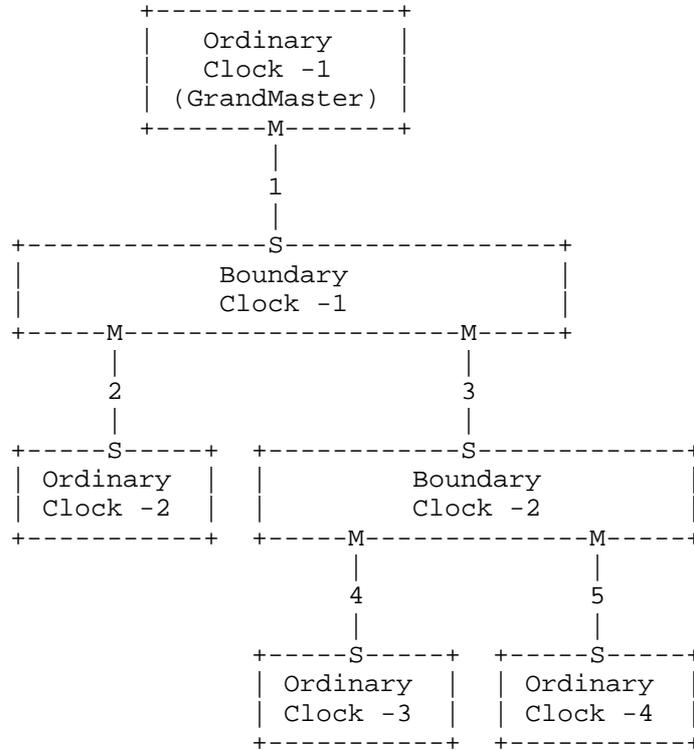
Grandmaster

Boundary Clock(0-N)   Ordinary Clocks(0-N)
Ordinary Clocks(0-N)

Relationship cardinality:
   PTP system 1 : N PTP Clocks
   PTP Clock 1 : 1 Domain
   PTP Clock 1 : N PTP Ports
   PTP Ports N : M Physical Ports (interface in IF-MIB)

Transparent clock diagram, section 6.7.1.3 of [IEEE 1588-2008]:

```
                      +----------------------------+
                      |     Boundary clock - 1     |
                      +----------------------------+
                         |                    |
                         |                    |
              +-- A --+                       B
              |                               |
  +---------------------+                     |
  |   Ordinary clock    |                     |
  +---------------------+                     |
                                              |
                                    +---------------------+
  +---------------------+           |     End-to-end      |
  |  Ordinary clock 1-1 |-----------|  transparent clock- |
  +---------------------+           |        1 - 1        |
                                    +---------------------+
                                              |
                                              C
                                              |
                                    +---------------------+
  +---------------------+           |     End-to-end      |
  |  Ordinary clock 1-2 |-----------|  transparent clock- |
  +---------------------+           |        1 - 2        |
                                    +---------------------+
```
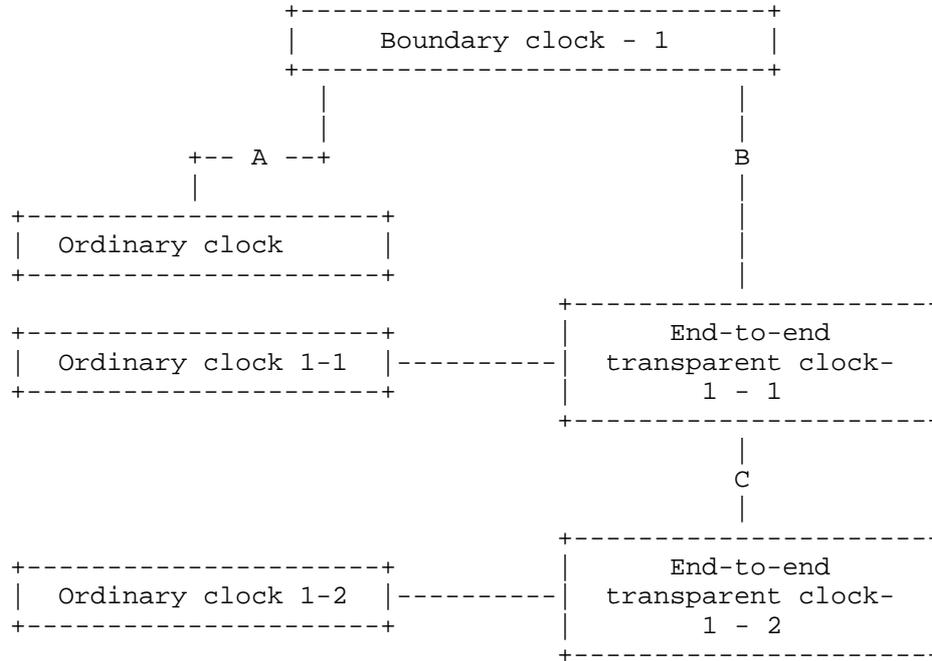
The MIB refers to the sections of [IEEE 1588-2008]."

     -- revision log

     ::= { mib-2 XXX } -- XXX to be assigned by IANA

-- Textual Conventions

ClockDomainType ::= TEXTUAL-CONVENTION
    DISPLAY-HINT   "d"
    STATUS   current
    DESCRIPTION
        "The Domain is identified by an integer, the domainNumber, in
        the range of 0 to 255. An integer value that is used to assign
        each PTP device to a particular domain. The following values

define the valid domains.

```
          Value       Definition
          ---------   -------------------
          0           Default domain
          1           Alternate domain 1
          2           Alternate domain 2
          3           Alternate domain 3
          4 - 127     User-defined domains
          128 - 255   Reserved"
```

    REFERENCE   "Section 7.1 Domains, Table 2 of [IEEE 1588-2008]"
    SYNTAX   Unsigned32 (0..255)

ClockIdentity ::= TEXTUAL-CONVENTION
    DISPLAY-HINT   "255a"
    STATUS   current
    DESCRIPTION
        "The clock Identity is an 8-octet array and will be presented in
        the form of a character array. Network byte order is assumed.

        The value of the ClockIdentity should be taken from the
        IEEE EUI-64 individual assigned numbers as indicated in
        Section 7.5.2.2.2 of [IEEE 1588-2008].

        The EUI-64 address is divided into the following fields:

            OUI bytes (0-2)
            Extension identifier bytes (3-7)

        The clock identifier can be constructed from existing EUI-48
        assignments and here is an abbreviated example extracted from
        section 7.5.2.2.2 [IEEE 1588-2008].

            Company EUI-48 = 0xACDE4823456716
            EUI-64 = ACDE48FFFE23456716

        It is important to note the IEEE Registration Authority has
        deprecated the use of MAC-48 in any new design."

    REFERENCE   "Section 7.5.2.2.1 of [IEEE 1588-2008]"
    SYNTAX   OCTET STRING (SIZE (1..255))

ClockInstanceType ::= TEXTUAL-CONVENTION
    DISPLAY-HINT   "d"
    STATUS   current
    DESCRIPTION
        "The instance of the Clock of a given clock type in a given
        domain."
    SYNTAX   Unsigned32 (0..255)

ClockIntervalBase2 ::= TEXTUAL-CONVENTION
    DISPLAY-HINT   "d"
    STATUS   current
    DESCRIPTION
        "The interval included in message types Announce, Sync,
        Delay_Req, and Pdelay_Req as indicated in section 7.7.2.1 of
        [IEEE 1588-2008].

        The mean time interval between successive messages shall be
        represented as the logarithm to the base 2 of this time
        interval measured in seconds on the local clock of the device
        sending the message. The values of these logarithmic attributes
        shall be selected from integers in the range -128 to 127 subject
        to further limits established in an applicable PTP profile."

    REFERENCE   "Section 7.7.2.1 General interval specification of
                  [IEEE 1588-2008]"
    SYNTAX   Integer32 (-128..127)

ClockMechanismType ::= TEXTUAL-CONVENTION
    STATUS   current
    DESCRIPTION
        "The clock type based on whether End to End or peer to peer
        mechanisms are used. The mechanism used to calculate the Mean
        Path Delay as indicated in Table 9 of [IEEE 1588-2008].

        Delay mechanism   Value(hex) Specification
        ---------------   ---------- -------------
        E2E   01   The port is configured to use the
                                       delay request-response mechanism.
        P2P   02   The port is configured to use the
                                       peer delay mechanism.
        DISABLED   FE   The port does not implement the
                                       delay mechanism."

```
    REFERENCE     "Sections 8.2.5.4.4, 6.6.4, 7.4.2 of [IEEE 1588-2008]."
    SYNTAX        INTEGER {
                  e2e(1),
                  p2p(2),
                  disabled(254)
                  }

ClockPortNumber ::= TEXTUAL-CONVENTION
    DISPLAY-HINT   "d"
    STATUS         current
    DESCRIPTION
        "An index identifying a specific Precision Time Protocol (PTP)
        port on a PTP node."

    REFERENCE     "Sections 7.5.2.3 and 5.3.5 of [IEEE 1588-2008]"
    SYNTAX        Unsigned32 (0..65535)

ClockPortState ::= TEXTUAL-CONVENTION
    STATUS         current
    DESCRIPTION
        "This is the value of the current state of the protocol engine
        associated with this port.
        Port state      Value    Description
        -----------------------------------------------------------
        initializing     1       In this state a port initializes
                                 its data sets, hardware, and
                                 communication facilities.
        faulty           2       The fault state of the protocol.
        disabled         3       The port shall not place any
                                 messages on its communication path.
        listening        4       The port is waiting for the
                                 announceReceiptTimeout to expire or
                                 to receive an Announce message from
                                 a master.
        preMaster        5       The port shall behave in all respects
                                 as though it were in the MASTER state
                                 except that it shall not place any
                                 messages on its communication path
                                 except for Pdelay_Req, Pdelay_Resp,
                                 Pdelay_Resp_Follow_Up, signaling, or
                                 management messages.
        master           6       The port is behaving as a master port.
        passive          7       The port shall not place any messages
```

```
                                on its communication path except for
                                Pdelay_Req, Pdelay_Resp,
                                Pdelay_Resp_Follow_Up, or signaling
                                messages, or management messages that
                                are a required response to another
                                management message
          uncalibrated     8      The local port is preparing to
                                synchronize to the master port.
          slave            9      The port is synchronizing to the
                                selected master port."

      REFERENCE          "Section 8.2.5.3.1 portState and 9.2.5 of
                          [IEEE 1588-2008]"
      SYNTAX             INTEGER {
                            initializing(1),
                            faulty(2),
                            disabled(3),
                            listening(4),
                            preMaster(5),
                            master(6),
                            passive(7),
                            uncalibrated(8),
                            slave(9)
                          }

ClockPortTransportTypeAddress ::= TEXTUAL-CONVENTION
      DISPLAY-HINT    "255a"
      STATUS          current
      DESCRIPTION
          "The Clock port transport protocol address used for this
           communication between the clock nodes. This is a string
           corresponding to the address type as specified by the
           Transport type used. The transport types can be defined
           elsewhere, in addition to the ones defined in this document.
           This can be address of type IP version 4, IP version 6,
           Ethernet, DeviceNET, ControlNET and IEC61158."

      REFERENCE    "Annex D (IPv4), Annex E (IPv6), Annex F (Ethernet),
                    Annex G (DeviceNET), Annex H (ControlNET) and
                    Annex I (IEC61158) of [IEEE 1588-2008]"
      SYNTAX          OCTET STRING (SIZE (1..255))

ClockProfileType ::= TEXTUAL-CONVENTION
```

```
    STATUS          current
    DESCRIPTION
        "Clock Profile used. A profile is the set of allowed Precision
        Time Protocol (PTP) features applicable to a device."

    REFERENCE       "Section 3.1.30 and 19.3 PTP profiles of
                     [IEEE 1588-2008]"
    SYNTAX          INTEGER  {
                        default(1),
                        telecom(2),
                        vendorspecific(3)
                    }

ClockQualityAccuracyType ::= TEXTUAL-CONVENTION
    STATUS          current
    DESCRIPTION
        "The ClockQuality as specified in section 5.3.7, 7.6.2.5 and
        Table 6 of [IEEE 1588-2008].

        The following values are not represented in the enumerated
        values.

                0x01-0x1F Reserved
                0x32-0x7F Reserved

        It is important to note that section 7.1.1 RFC2578 allows for
        gaps and enumerate values to start with zero when indicated by
        the protocol."

    REFERENCE       "Section 5.3.7, 7.6.2.5 and Table 6 of
                     [IEEE 1588-2008]"
    SYNTAX          INTEGER  {
                        -- reserved00(0:31), -- 0x00 to 0x1F
                        nanoSecond25(32),    -- 0x20
                        nanoSecond100(33),   -- 0x21
                        nanoSecond250(34),   -- 0x22
                        microSec1(35),       -- 0x23
                        microSec2dot5(36),   -- 0x24
                        microSec10(37),      -- 0x25
                        microSec25(38),      -- 0x26
                        microSec100(39),     -- 0x27
                        microSec250(40),     -- 0x28
                        milliSec1(41),       -- 0x29
```

```
                        milliSec2dot5(42),   -- 0x2A
                        milliSec10(43),      -- 0x2B
                        milliSec25(44),      -- 0x2C
                        milliSec100(45),     -- 0x2D
                        milliSec250(46),     -- 0x2E
                        second1(47),         -- 0x2F
                        second10(48),        -- 0x30
                        secondGreater10(49), -- 0x31
                        unknown(254),        -- 0xFE
                        -- reserved255(255)  -- 0xFF
                    }

ClockQualityClassType ::= TEXTUAL-CONVENTION
    DISPLAY-HINT      "d"
    STATUS            current
    DESCRIPTION
        "The ClockQuality as specified in section 5.3.7, 7.6.2.4 and
        Table 5 of [IEEE 1588-2008].

        Value     Description
        --------  -------------------------------------------------
              0   Reserved to enable compatibility with future
                  versions.
            1-5   Reserved
              6   Shall designate a clock that is synchronized
                  to a primary reference time source.  The
                  timescale distributed shall be PTP.  A
                  clockClass 6 clock shall not be a slave to
                  another clock in the domain.
              7   Shall designate a clock that has previously
                  been designated as clockClass 6 but that has
                  lost the ability to synchronize to a primary
                  reference time source and is in holdover mode
                  and within holdover specifications. The
                  timescale distributed shall be PTP.  A
                  clockClass 7 clock shall not be a slave to
                  another clock in the domain.
              8   Reserved.
           9-10   Reserved to enable compatibility with future
                  versions.
          11-12   Reserved.
             13   Shall designate a clock that is synchronized
                  to an application-specific source of time.
```

The timescale distributed shall be ARB. A
clockClass 13 clock shall not be a slave to
another clock in the domain.

14   Shall designate a clock that has previously
       been designated as clockClass 13 but that
       has lost the ability to synchronize to an
       application-specific source of time and is
       in holdover mode and within holdover
       specifications. The timescale distributed
       shall be ARB. A clockClass 14 clock shall
       not be a slave to another clock in the domain.

15-51   Reserved.

52   Degradation alternative A for a clock of
       clockClass 7 that is not within holdover
       specification. A clock of clockClass 52
       shall not be a slave to another clock in
       the domain.

53-57   Reserved.

58   Degradation alternative A for a clock of
       clockClass 14 that is not within holdover
       specification. A clock of clockClass 58 shall
       not be a slave to another clock in the domain.

59-67   Reserved.

68-122   For use by alternate PTP profiles.

123-127 Reserved.

128-132 Reserved.

133-170 For use by alternate PTP profiles.

171-186 Reserved.

187   Degradation alternative B for a clock of
       clockClass 7 that is not within holdover
       specification. A clock of clockClass 187 may
       be a slave to another clock in the domain.

188-192 Reserved.

193   Degradation alternative B for a clock of
       clockClass 14 that is not within holdover
       specification. A clock of clockClass 193 may
       be a slave to another clock in the domain.

194-215 Reserved.

216-232 For use by alternate PTP profiles.

233-247 Reserved.

248   Default. This clockClass shall be used if
       none of the other clockClass definitions apply.

```
          249-250  Reserved.
              251  Reserved for version 1 compatibility; see Clause 18.
          252-254  Reserved.
              255  Shall be the clockClass of a slave-only clock; see
                   9.2.2."

    REFERENCE         "Section 5.3.7, 7.6.2.4 and Table 5 of
                       [IEEE 1588-2008]."
    SYNTAX            Unsigned32 (0..255)

ClockRoleType ::= TEXTUAL-CONVENTION
    STATUS            current
    DESCRIPTION
        "The Clock Role. The protocol generates a Master Slave
        relationship among the clocks in the system.

        Clock Role        Value      Description
        ----------------------------------------------------------------
        Master clock      1          A clock that is the source of
                                     time to which all other clocks on
                                     that path synchronize.

        Slave clock       2          A clock which synchronizes to
                                     another clock (master)."
    SYNTAX            INTEGER {
                         master(1),
                         slave(2)
                      }

ClockStateType ::= TEXTUAL-CONVENTION
    STATUS            current
    DESCRIPTION
        "The clock state returned by PTP engine.

        Clock State       Value    Description
        ----------------------------------------------------------------
        Freerun state        1     Applies to a slave device that is not
                                   locked to a master. This is the initial
                                   state a slave starts out with when it
                                   is not getting any PTP packets from the
                                   master or because of some other input
                                   error (erroneous packets, etc).
```

```
            Holdover state      2   In this state the slave device is
                                    locked to a master but communication
                                    with the master has been lost or the
                                    timestamps in the ptp packets are
                                    incorrect.  Since the slave was
                                    locked to the master, it can run in this
                                    state, with similar accuracy for some
                                    time.  If communication with the master
                                    is not restored for an extended period
                                    (dependent on the clock implementation),
                                    the device should move to the FREERUN
                                    state.

            Acquiring state     3   The slave device is receiving packets
                                    from a master and is trying to acquire
                                    a lock.

            Freq_locked state   4   Slave device is locked to the Master
                                    with respect to frequency, but not phase
                                    aligned

            Phase_aligned state 5   Locked to the master with respect to
                                    frequency and phase."
    SYNTAX              INTEGER  {
                           freerun(1),
                           holdover(2),
                           acquiring(3),
                           frequencyLocked(4),
                           phaseAligned(5)
                       }

ClockTimeInterval ::= TEXTUAL-CONVENTION
    DISPLAY-HINT      "255a"
    STATUS            current
    DESCRIPTION
        "This textual convention corresponds to the TimeInterval
        structure indicated in section 5.3.2 of [IEEE 1588-2008].
        It will be presented in the form of a character array.
        Network byte order is assumed.

        The TimeInterval type represents time intervals.

            struct TimeInterval
```

```
            {
                  Integer64 scaledNanoseconds;
            };
```

        The scaledNanoseconds member is the time interval expressed in
        units of nanoseconds and multiplied by 2**16.

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

        For example, 2.5 ns is expressed as 0000 0000 0002 8000 in
        Base16."

    REFERENCE
        "Section 5.3.2 and setion 7.7.2.1 Timer interval
         specification of [IEEE 1588-2008]"
    SYNTAX   OCTET STRING (SIZE (1..255))

ClockTimeSourceType ::= TEXTUAL-CONVENTION
    STATUS   current
    DESCRIPTION
        "The ClockQuality as specified in section 5.3.7, 7.6.2.6 and
        Table 7 of [IEEE 1588-2008].

        The following values are not represented in the enumerated
        values.

        0xF0-0xFE For use by alternate PTP profiles
        0xFF   Reserved

        It is important to note that section 7.1.1 RFC2578 allows for
        gaps and enumerate values to start with zero when indicated by
        the protocol."

    REFERENCE   "Section 5.3.7, 7.6.2.6 and Table 7 of
                        [IEEE 1588-2008]."
    SYNTAX   INTEGER {
                          atomicClock(16), -- 0x10
                          gps(32), -- 0x20
                          terrestrialRadio(48), -- 0x22
                          ptp(64), -- 0x40
                          ntp(80), -- 0x50
```

```
                          handSet(96), -- 0x60
                          other(144), -- 0x90
                          internalOscillator(160) -- 0xA0
                      }

ClockTxModeType ::= TEXTUAL-CONVENTION
    STATUS          current
    DESCRIPTION
        "Transmission mode.

        Unicast:       Using unicast communication channel.
        Multicast:     Using Multicast communication channel.
        multicast-mix: Using multicast-unicast communication channel"
    SYNTAX          INTEGER  {
                        unicast(1),
                        multicast(2),
                        multicastmix(3)
                    }

ClockType ::= TEXTUAL-CONVENTION
    STATUS          current
    DESCRIPTION
        "The clock types as defined in the MIB module description."

    REFERENCE       "Section 6.5.1 of [IEEE 1588-2008]."
    SYNTAX          INTEGER  {
                        ordinaryClock(1),
                        boundaryClock(2),
                        transparentClock(3),
                        boundaryNode(4)
                    }

ptpbaseMIBNotifs  OBJECT IDENTIFIER
    ::= { ptpbaseMIB 0 }

ptpbaseMIBObjects  OBJECT IDENTIFIER
    ::= { ptpbaseMIB 1 }

ptpbaseMIBConformance  OBJECT IDENTIFIER
    ::= { ptpbaseMIB 2 }

ptpbaseMIBSystemInfo  OBJECT IDENTIFIER
    ::= { ptpbaseMIBObjects 1 }
```

ptpbaseMIBClockInfo OBJECT IDENTIFIER
    ::= { ptpbaseMIBObjects 2 }

ptpbaseSystemTable OBJECT-TYPE
    SYNTAX   SEQUENCE OF PtpbaseSystemEntry
    MAX-ACCESS   not-accessible
    STATUS   current
    DESCRIPTION
        "Table of count information about the PTP system for all
        domains."
    ::= { ptpbaseMIBSystemInfo 1 }

ptpbaseSystemEntry OBJECT-TYPE
    SYNTAX   PtpbaseSystemEntry
    MAX-ACCESS   not-accessible
    STATUS   current
    DESCRIPTION
        "An entry in the table, containing count information about a
        single domain. New row entries are added when the PTP clock for
        this domain is configured, while the unconfiguration of the PTP
        clock removes it."
    INDEX   {
                        ptpDomainIndex,
                        ptpInstanceIndex
                    }
    ::= { ptpbaseSystemTable 1 }

PtpbaseSystemEntry ::= SEQUENCE {
        ptpDomainIndex   ClockDomainType,
        ptpInstanceIndex   ClockInstanceType,
        ptpDomainClockPortsTotal Gauge32
}

ptpDomainIndex OBJECT-TYPE
    SYNTAX   ClockDomainType
    MAX-ACCESS   not-accessible
    STATUS   current
    DESCRIPTION
        "This object specifies the domain number used to create logical
        group of PTP devices. The Clock Domain is a logical group of
        clocks and devices that synchronize with each other using the

        PTP protocol.

        0   Default domain
        1   Alternate domain 1
        2   Alternate domain 2
        3   Alternate domain 3
        4 - 127   User-defined domains
        128 - 255 Reserved"
    ::= { ptpbaseSystemEntry 1 }

ptpInstanceIndex OBJECT-TYPE
    SYNTAX   ClockInstanceType
    MAX-ACCESS   not-accessible
    STATUS   current
    DESCRIPTION
        "This object specifies the instance of the Clock for this
        domain."
    ::= { ptpbaseSystemEntry 2 }

ptpDomainClockPortsTotal OBJECT-TYPE
    SYNTAX   Gauge32
    UNITS   "ptp ports"
    MAX-ACCESS   read-only
    STATUS   current
    DESCRIPTION
        "This object specifies the total number of clock ports
        configured within a domain in the system."
    ::= { ptpbaseSystemEntry 3 }

ptpbaseSystemDomainTable OBJECT-TYPE
    SYNTAX   SEQUENCE OF PtpbaseSystemDomainEntry
    MAX-ACCESS   not-accessible
    STATUS   current
    DESCRIPTION
        "Table of information about the PTP system for all clock modes
        -- ordinary, boundary or transparent."
    ::= { ptpbaseMIBSystemInfo 2 }

ptpbaseSystemDomainEntry OBJECT-TYPE
    SYNTAX   PtpbaseSystemDomainEntry
    MAX-ACCESS   not-accessible

```
     STATUS          current
     DESCRIPTION
         "An entry in the table, containing information about a single
         clock mode for the PTP system. A row entry gets added when PTP
         clocks are configured on the router."
     INDEX          { ptpbaseSystemDomainClockTypeIndex }
     ::= { ptpbaseSystemDomainTable 1 }

PtpbaseSystemDomainEntry ::= SEQUENCE {
        ptpbaseSystemDomainClockTypeIndex ClockType,
        ptpbaseSystemDomainTotals        Unsigned32
}

ptpbaseSystemDomainClockTypeIndex OBJECT-TYPE
     SYNTAX          ClockType
     MAX-ACCESS      not-accessible
     STATUS          current
     DESCRIPTION
         "This object specifies the clock type as defined in the
         Textual convention description."
     ::= { ptpbaseSystemDomainEntry 1 }

ptpbaseSystemDomainTotals OBJECT-TYPE
     SYNTAX          Unsigned32
     UNITS           "domains"
     MAX-ACCESS      read-only
     STATUS          current
     DESCRIPTION
         "This object specifies the total number of PTP domains for this
         particular clock type configured in this node."
     ::= { ptpbaseSystemDomainEntry 2 }



ptpbaseSystemProfile OBJECT-TYPE
     SYNTAX          ClockProfileType
     MAX-ACCESS      read-only
     STATUS          current
     DESCRIPTION
         "This object specifies the PTP Profile implemented on the
         system."
     REFERENCE       "Section 19.3 PTP profiles of [IEEE 1588-2008]"
     ::= { ptpbaseMIBSystemInfo 3 }
```

ptpbaseClockCurrentDSTable OBJECT-TYPE
    SYNTAX   SEQUENCE OF PtpbaseClockCurrentDSEntry
    MAX-ACCESS   not-accessible
    STATUS   current
    DESCRIPTION
        "Table of information about the PTP clock Current Datasets for
        all domains."
    ::= { ptpbaseMIBClockInfo 1 }

ptpbaseClockCurrentDSEntry OBJECT-TYPE
    SYNTAX   PtpbaseClockCurrentDSEntry
    MAX-ACCESS   not-accessible
    STATUS   current
    DESCRIPTION
        "An entry in the table, containing information about a single
        PTP clock Current Datasets for a domain."
    REFERENCE
        "1588 Version 2.0 Section 8.2.2 currentDS data set member
        specifications of [IEEE 1588-2008]"
    INDEX   {
                        ptpbaseClockCurrentDSDomainIndex,
                        ptpbaseClockCurrentDSClockTypeIndex,
                        ptpbaseClockCurrentDSInstanceIndex
                      }
    ::= { ptpbaseClockCurrentDSTable 1 }

PtpbaseClockCurrentDSEntry ::= SEQUENCE {
        ptpbaseClockCurrentDSDomainIndex   ClockDomainType,
        ptpbaseClockCurrentDSClockTypeIndex   ClockType,
        ptpbaseClockCurrentDSInstanceIndex   ClockInstanceType,
        ptpbaseClockCurrentDSStepsRemoved   Unsigned32,
        ptpbaseClockCurrentDSOffsetFromMaster ClockTimeInterval,
        ptpbaseClockCurrentDSMeanPathDelay   ClockTimeInterval
}

ptpbaseClockCurrentDSDomainIndex OBJECT-TYPE
    SYNTAX   ClockDomainType
    MAX-ACCESS   not-accessible
    STATUS   current
    DESCRIPTION
        "This object specifies the domain number used to create logical
        group of PTP devices."

```
    ::= { ptpbaseClockCurrentDSEntry 1 }

ptpbaseClockCurrentDSClockTypeIndex OBJECT-TYPE
    SYNTAX          ClockType
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "This object specifies the clock type as defined in the
        Textual convention description."
    ::= { ptpbaseClockCurrentDSEntry 2 }

ptpbaseClockCurrentDSInstanceIndex OBJECT-TYPE
    SYNTAX          ClockInstanceType
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "This object specifies the instance of the clock for this clock
        type in the given domain."
    ::= { ptpbaseClockCurrentDSEntry 3 }

ptpbaseClockCurrentDSStepsRemoved OBJECT-TYPE
    SYNTAX          Unsigned32
    UNITS           "Steps"
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "The current clock dataset StepsRemoved value.

        This object specifies the distance measured by the number of
        Boundary clocks between the local clock and the Foreign master
        as indicated in the stepsRemoved field of Announce messages."
    REFERENCE       "1588 Version 2.0 Section 8.2.2.2 stepsRemoved"
    ::= { ptpbaseClockCurrentDSEntry 4 }

ptpbaseClockCurrentDSOffsetFromMaster OBJECT-TYPE
    SYNTAX          ClockTimeInterval
    UNITS           "Time Interval"
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "This object specifies the current clock dataset ClockOffset
        value. The value of the computation of the offset in time
        between a slave and a master clock."
```

     REFERENCE   "1588 Version 2.0 Section 8.2.2.3 of
                         [IEEE 1588-2008]"
     ::= { ptpbaseClockCurrentDSEntry 5 }

ptpbaseClockCurrentDSMeanPathDelay OBJECT-TYPE
     SYNTAX   ClockTimeInterval
     UNITS   "Time Interval"
     MAX-ACCESS   read-only
     STATUS   current
     DESCRIPTION
         "This object specifies the current clock dataset
         MeanPathDelay value.

         The mean path delay between a pair of ports as measure by the
         delay request-response mechanism."
     REFERENCE   "1588 Version 2.0 Section 8.2.2.4 mean path delay"
     ::= { ptpbaseClockCurrentDSEntry 6 }

ptpbaseClockParentDSTable OBJECT-TYPE
     SYNTAX   SEQUENCE OF PtpbaseClockParentDSEntry
     MAX-ACCESS   not-accessible
     STATUS   current
     DESCRIPTION
         "Table of information about the PTP clock Parent Datasets for
         all domains."
     ::= { ptpbaseMIBClockInfo 2 }

ptpbaseClockParentDSEntry OBJECT-TYPE
     SYNTAX   PtpbaseClockParentDSEntry
     MAX-ACCESS   not-accessible
     STATUS   current
     DESCRIPTION
         "An entry in the table, containing information about a single
         PTP clock Parent Datasets for a domain."
     REFERENCE
         "Section 8.2.3 parentDS data set member specifications of
         [IEEE 1588-2008]"
     INDEX   {
                         ptpbaseClockParentDSDomainIndex,
                         ptpbaseClockParentDSClockTypeIndex,
                         ptpbaseClockParentDSInstanceIndex

```
                   }
      ::= { ptpbaseClockParentDSTable 1 }

PtpbaseClockParentDSEntry ::= SEQUENCE {
    ptpbaseClockParentDSDomainIndex         ClockDomainType,
    ptpbaseClockParentDSClockTypeIndex      ClockType,
    ptpbaseClockParentDSInstanceIndex       ClockInstanceType,
    ptpbaseClockParentDSParentPortIdentity  OCTET STRING,
    ptpbaseClockParentDSParentStats         TruthValue,
    ptpbaseClockParentDSOffset              ClockIntervalBase2,
    ptpbaseClockParentDSClockPhChRate       Integer32,
    ptpbaseClockParentDSGMClockIdentity     ClockIdentity,
    ptpbaseClockParentDSGMClockPriority1    Unsigned32,
    ptpbaseClockParentDSGMClockPriority2    Unsigned32,
    ptpbaseClockParentDSGMClockQualityClass ClockQualityClassType,
    ptpbaseClockParentDSGMClockQualityAccuracy ClockQualityAccuracyType,
    ptpbaseClockParentDSGMClockQualityOffset Unsigned32
}

ptpbaseClockParentDSDomainIndex OBJECT-TYPE
    SYNTAX          ClockDomainType
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "This object specifies the domain number used to create logical
        group of PTP devices."
    ::= { ptpbaseClockParentDSEntry 1 }

ptpbaseClockParentDSClockTypeIndex OBJECT-TYPE
    SYNTAX          ClockType
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "This object specifies the clock type as defined in the
        Textual convention description."
    ::= { ptpbaseClockParentDSEntry 2 }

ptpbaseClockParentDSInstanceIndex OBJECT-TYPE
    SYNTAX          ClockInstanceType
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "This object specifies the instance of the clock for this clock
```

          type in the given domain."
    ::= { ptpbaseClockParentDSEntry 3 }

ptpbaseClockParentDSParentPortIdentity OBJECT-TYPE
    SYNTAX   OCTET STRING(SIZE(1..256))
    MAX-ACCESS   read-only
    STATUS   current
    DESCRIPTION
        "This object specifies the value of portIdentity of the port on
        the master that issues the Sync messages used in synchronizing
        this clock."
    REFERENCE
        "Section 8.2.3.2 parentDS.parentPortIdentity of
         [IEEE 1588-2008]"
    ::= { ptpbaseClockParentDSEntry 4 }

ptpbaseClockParentDSParentStats OBJECT-TYPE
    SYNTAX   TruthValue
    MAX-ACCESS   read-only
    STATUS   current
    DESCRIPTION
        "This object specifies the Parent Dataset ParentStats value.

        This value indicates whether the values of ParentDSOffset
        and ParentDSClockPhChRate have been measured and are valid.
        A TRUE value shall indicate valid data."
    REFERENCE   "Section 8.2.3.3 parentDS.parentStats of
                     [IEEE 1588-2008]"
    ::= { ptpbaseClockParentDSEntry 5 }

ptpbaseClockParentDSOffset OBJECT-TYPE
    SYNTAX   ClockIntervalBase2 (-128..127)
    MAX-ACCESS   read-only
    STATUS   current
    DESCRIPTION
        "This object specifies the Parent Dataset
        ParentOffsetScaledLogVariance value.

        This value is the variance of the parent clocks phase as
        measured by the local clock."
    REFERENCE
        "Section 8.2.3.4
        parentDS.observedParentOffsetScaledLogVariance

```
        [IEEE 1588-2008]"
    ::= { ptpbaseClockParentDSEntry 6 }

ptpbaseClockParentDSClockPhChRate OBJECT-TYPE
    SYNTAX          Integer32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "This object specifies the clock's parent dataset
        ParentClockPhaseChangeRate value.

        This value is an estimate of the parent clocks phase change
        rate as measured by the slave clock."
    REFERENCE
        "Section 8.2.3.5
        parentDS.observedParentClockPhaseChangeRate of
        [IEEE 1588-2008]"
    ::= { ptpbaseClockParentDSEntry 7 }

ptpbaseClockParentDSGMClockIdentity OBJECT-TYPE
    SYNTAX          ClockIdentity
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "This object specifies the parent dataset Grandmaster clock
        identity."
    REFERENCE
        "Section 8.2.3.6 parentDS.grandmasterIdentity of
         [IEEE 1588-2008]"
    ::= { ptpbaseClockParentDSEntry 8 }

ptpbaseClockParentDSGMClockPriority1 OBJECT-TYPE
    SYNTAX          Unsigned32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "This object specifies the parent dataset Grandmaster clock
        priority1."
    REFERENCE
        "Section 8.2.3.8 parentDS.grandmasterPriority1 of
         [IEEE 1588-2008]"
    ::= { ptpbaseClockParentDSEntry 9 }
```

ptpbaseClockParentDSGMClockPriority2 OBJECT-TYPE
    SYNTAX   Unsigned32
    MAX-ACCESS   read-only
    STATUS   current
    DESCRIPTION
        "This object specifies the parent dataset grandmaster clock
        priority2."
    REFERENCE
        "Section 8.2.3.9 parentDS.grandmasterPriority2 of
         [IEEE 1588-2008]"
    ::= { ptpbaseClockParentDSEntry 10 }

ptpbaseClockParentDSGMClockQualityClass OBJECT-TYPE
    SYNTAX   ClockQualityClassType (0..255)
    MAX-ACCESS   read-only
    STATUS   current
    DESCRIPTION
        "This object specifies the parent dataset grandmaster clock
        quality class."
    REFERENCE
        "Section 8.2.3.7 parentDS.grandmasterClockQuality of
         [IEEE 1588-2008]"
    ::= { ptpbaseClockParentDSEntry 11 }

ptpbaseClockParentDSGMClockQualityAccuracy OBJECT-TYPE
    SYNTAX   ClockQualityAccuracyType
    MAX-ACCESS   read-only
    STATUS   current
    DESCRIPTION
        "This object specifies the parent dataset grandmaster clock
        quality accuracy."
    REFERENCE
        "Section 8.2.3.7 parentDS.grandmasterClockQuality of
         [IEEE 1588-2008]"
    ::= { ptpbaseClockParentDSEntry 12 }

ptpbaseClockParentDSGMClockQualityOffset OBJECT-TYPE
    SYNTAX   Unsigned32
    MAX-ACCESS   read-only
    STATUS   current
    DESCRIPTION
        "This object specifies the parent dataset grandmaster clock
        quality offset."

```
     REFERENCE
         "Section 8.2.3.7 parentDS.grandmasterClockQuality of
          [IEEE 1588-2008]"
     ::= { ptpbaseClockParentDSEntry 13 }



ptpbaseClockDefaultDSTable OBJECT-TYPE
     SYNTAX          SEQUENCE OF PtpbaseClockDefaultDSEntry
     MAX-ACCESS      not-accessible
     STATUS          current
     DESCRIPTION
         "Table of information about the PTP clock Default Datasets for
         all domains."
     ::= { ptpbaseMIBClockInfo 3 }

ptpbaseClockDefaultDSEntry OBJECT-TYPE
     SYNTAX          PtpbaseClockDefaultDSEntry
     MAX-ACCESS      not-accessible
     STATUS          current
     DESCRIPTION
         "An entry in the table, containing information about a single
         PTP clock Default Datasets for a domain."
     INDEX           {
                         ptpbaseClockDefaultDSDomainIndex,
                         ptpbaseClockDefaultDSClockTypeIndex,
                         ptpbaseClockDefaultDSInstanceIndex
                     }
     ::= { ptpbaseClockDefaultDSTable 1 }

PtpbaseClockDefaultDSEntry ::= SEQUENCE {
         ptpbaseClockDefaultDSDomainIndex      ClockDomainType,
         ptpbaseClockDefaultDSClockTypeIndex   ClockType,
         ptpbaseClockDefaultDSInstanceIndex    ClockInstanceType,
         ptpbaseClockDefaultDSTwoStepFlag      TruthValue,
         ptpbaseClockDefaultDSClockIdentity    ClockIdentity,
         ptpbaseClockDefaultDSPriority1        Unsigned32,
         ptpbaseClockDefaultDSPriority2        Unsigned32,
         ptpbaseClockDefaultDSSlaveOnly        TruthValue,
         ptpbaseClockDefaultDSQualityClass     ClockQualityClassType,
         ptpbaseClockDefaultDSQualityAccuracy ClockQualityAccuracyType,
         ptpbaseClockDefaultDSQualityOffset    Integer32
}
```

```
ptpbaseClockDefaultDSDomainIndex OBJECT-TYPE
    SYNTAX          ClockDomainType
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "This object specifies the domain number used to create logical
        group of PTP devices."
    ::= { ptpbaseClockDefaultDSEntry 1 }

ptpbaseClockDefaultDSClockTypeIndex OBJECT-TYPE
    SYNTAX          ClockType
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "This object specifies the clock type as defined in the
        Textual convention description."
    ::= { ptpbaseClockDefaultDSEntry 2 }

ptpbaseClockDefaultDSInstanceIndex OBJECT-TYPE
    SYNTAX          ClockInstanceType
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "This object specifies the instance of the clock for this clock
        type in the given domain."
    ::= { ptpbaseClockDefaultDSEntry 3 }

ptpbaseClockDefaultDSTwoStepFlag OBJECT-TYPE
    SYNTAX          TruthValue
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "This object specifies whether the Two Step process is used."
    ::= { ptpbaseClockDefaultDSEntry 4 }

ptpbaseClockDefaultDSClockIdentity OBJECT-TYPE
    SYNTAX          ClockIdentity
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "This object specifies the default Datasets clock identity."
    ::= { ptpbaseClockDefaultDSEntry 5 }
```

```
ptpbaseClockDefaultDSPriority1 OBJECT-TYPE
    SYNTAX          Unsigned32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "This object specifies the default Datasets clock Priority1."
    ::= { ptpbaseClockDefaultDSEntry 6 }

ptpbaseClockDefaultDSPriority2 OBJECT-TYPE
    SYNTAX          Unsigned32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "This object specifies the default Datasets clock Priority2."
    ::= { ptpbaseClockDefaultDSEntry 7 }

ptpbaseClockDefaultDSSlaveOnly OBJECT-TYPE
    SYNTAX          TruthValue
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "Whether the SlaveOnly flag is set."
    ::= { ptpbaseClockDefaultDSEntry 8 }

ptpbaseClockDefaultDSQualityClass OBJECT-TYPE
    SYNTAX          ClockQualityClassType (0..255)
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "This object specifies the default dataset Quality Class."
    ::= { ptpbaseClockDefaultDSEntry 9 }

ptpbaseClockDefaultDSQualityAccuracy OBJECT-TYPE
    SYNTAX          ClockQualityAccuracyType
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "This object specifies the default dataset Quality Accuracy."
    ::= { ptpbaseClockDefaultDSEntry 10 }

ptpbaseClockDefaultDSQualityOffset OBJECT-TYPE
    SYNTAX          Integer32
```

```
    MAX-ACCESS        read-only
    STATUS            current
    DESCRIPTION
        "This object specifies the default dataset Quality offset."
    ::= { ptpbaseClockDefaultDSEntry 11 }



ptpbaseClockRunningTable OBJECT-TYPE
    SYNTAX            SEQUENCE OF PtpbaseClockRunningEntry
    MAX-ACCESS        not-accessible
    STATUS            current
    DESCRIPTION
        "Table of information about the PTP clock Running Datasets for
        all domains."
    ::= { ptpbaseMIBClockInfo 4 }

ptpbaseClockRunningEntry OBJECT-TYPE
    SYNTAX            PtpbaseClockRunningEntry
    MAX-ACCESS        not-accessible
    STATUS            current
    DESCRIPTION
        "An entry in the table, containing information about a single
        PTP clock running Datasets for a domain."
    INDEX            {
                        ptpbaseClockRunningDomainIndex,
                        ptpbaseClockRunningClockTypeIndex,
                        ptpbaseClockRunningInstanceIndex
                     }
    ::= { ptpbaseClockRunningTable 1 }

PtpbaseClockRunningEntry ::= SEQUENCE {
        ptpbaseClockRunningDomainIndex      ClockDomainType,
        ptpbaseClockRunningClockTypeIndex   ClockType,
        ptpbaseClockRunningInstanceIndex    ClockInstanceType,
        ptpbaseClockRunningState            ClockStateType,
        ptpbaseClockRunningPacketsSent       Counter64,
        ptpbaseClockRunningPacketsReceived Counter64
}

ptpbaseClockRunningDomainIndex OBJECT-TYPE
    SYNTAX            ClockDomainType
    MAX-ACCESS        not-accessible
```

     STATUS   current
     DESCRIPTION
         "This object specifies the domain number used to create logical
         group of PTP devices."
     ::= { ptpbaseClockRunningEntry 1 }

ptpbaseClockRunningClockTypeIndex OBJECT-TYPE
     SYNTAX   ClockType
     MAX-ACCESS   not-accessible
     STATUS   current
     DESCRIPTION
         "This object specifies the clock type as defined in the
         Textual convention description."
     ::= { ptpbaseClockRunningEntry 2 }

ptpbaseClockRunningInstanceIndex OBJECT-TYPE
     SYNTAX   ClockInstanceType
     MAX-ACCESS   not-accessible
     STATUS   current
     DESCRIPTION
         "This object specifies the instance of the clock for this clock
         type in the given domain."
     ::= { ptpbaseClockRunningEntry 3 }

ptpbaseClockRunningState OBJECT-TYPE
     SYNTAX   ClockStateType
     MAX-ACCESS   read-only
     STATUS   current
     DESCRIPTION
         "This object specifies the Clock state returned by PTP engine
         which was described earlier.

         Freerun state. Applies to a slave device that is not locked to
         a master. This is the initial state a slave starts out with
         when it is not getting any PTP packets from the master, or
         because of some other input error (erroneous packets, etc).

         Holdover state. In this state the slave device is locked to a
         master but communication with the master has been lost or the
         timestamps in the ptp packets are incorrect. Since the
         slave was previously locked to the master, it can run in this
         state, with similar accuracy for some time. If communication
         with the master is not restored for an extended period

        (dependent on the clock implementation), the device should move
        to the FREERUN state.

        Acquiring state. The slave device is receiving packets from a
        master and is trying to acquire a lock.

        Freq_locked state. Slave device is locked to the Master with
        respect to frequency, but not phase aligned

        Phase_aligned state. Locked to the master with respect to
        frequency and phase."
     ::= { ptpbaseClockRunningEntry 4 }

ptpbaseClockRunningPacketsSent OBJECT-TYPE
     SYNTAX   Counter64
     MAX-ACCESS   read-only
     STATUS   current
     DESCRIPTION
        "This object specifies the total number of all packet Unicast
        and multicast that have been sent out for this clock in this
        domain for this type."
     ::= { ptpbaseClockRunningEntry 5 }

ptpbaseClockRunningPacketsReceived OBJECT-TYPE
     SYNTAX   Counter64
     MAX-ACCESS   read-only
     STATUS   current
     DESCRIPTION
        "This object specifies the total number of all packet Unicast
        and multicast that have been received for this clock in this
        domain for this type."
     ::= { ptpbaseClockRunningEntry 6 }

ptpbaseClockTimePropertiesDSTable OBJECT-TYPE
     SYNTAX   SEQUENCE OF PtpbaseClockTimePropertiesDSEntry
     MAX-ACCESS   not-accessible
     STATUS   current
     DESCRIPTION
        "Table of information about the PTP clock time properties
        datasets for all domains."
     ::= { ptpbaseMIBClockInfo 5 }

ptpbaseClockTimePropertiesDSEntry OBJECT-TYPE
     SYNTAX   PtpbaseClockTimePropertiesDSEntry
     MAX-ACCESS   not-accessible
     STATUS   current
     DESCRIPTION
         "An entry in the table, containing information about a single
         PTP clock timeproperties Datasets for a domain."
     REFERENCE   "Section 8.2.4 of [IEEE 1588-2008]"
     INDEX   {
                         ptpbaseClockTimePropertiesDSDomainIndex,
                         ptpbaseClockTimePropertiesDSClockTypeIndex,
                         ptpbaseClockTimePropertiesDSInstanceIndex
                      }
     ::= { ptpbaseClockTimePropertiesDSTable 1 }

PtpbaseClockTimePropertiesDSEntry ::= SEQUENCE {
  ptpbaseClockTimePropertiesDSDomainIndex   ClockDomainType,
  ptpbaseClockTimePropertiesDSClockTypeIndex   ClockType,
  ptpbaseClockTimePropertiesDSInstanceIndex   ClockInstanceType,
  ptpbaseClockTimePropertiesDSCurrentUTCOffsetValid TruthValue,
  ptpbaseClockTimePropertiesDSCurrentUTCOffset   Integer32,
  ptpbaseClockTimePropertiesDSLeap59   TruthValue,
  ptpbaseClockTimePropertiesDSLeap61   TruthValue,
  ptpbaseClockTimePropertiesDSTimeTraceable   TruthValue,
  ptpbaseClockTimePropertiesDSFreqTraceable   TruthValue,
  ptpbaseClockTimePropertiesDSPTPTimescale   TruthValue,
  ptpbaseClockTimePropertiesDSSource   ClockTimeSourceType
}

ptpbaseClockTimePropertiesDSDomainIndex OBJECT-TYPE
     SYNTAX   ClockDomainType
     MAX-ACCESS   not-accessible
     STATUS   current
     DESCRIPTION
         "This object specifies the domain number used to create logical
         group of PTP devices."
     ::= { ptpbaseClockTimePropertiesDSEntry 1 }

ptpbaseClockTimePropertiesDSClockTypeIndex OBJECT-TYPE
     SYNTAX   ClockType
     MAX-ACCESS   not-accessible
     STATUS   current

```
     DESCRIPTION
         "This object specifies the clock type as defined in the
         Textual convention description."
     ::= { ptpbaseClockTimePropertiesDSEntry 2 }


ptpbaseClockTimePropertiesDSInstanceIndex OBJECT-TYPE
     SYNTAX          ClockInstanceType
     MAX-ACCESS      not-accessible
     STATUS          current
     DESCRIPTION
         "This object specifies the instance of the clock for this clock
         type in the given domain."
     ::= { ptpbaseClockTimePropertiesDSEntry 3 }


ptpbaseClockTimePropertiesDSCurrentUTCOffsetValid OBJECT-TYPE
     SYNTAX          TruthValue
     MAX-ACCESS      read-only
     STATUS          current
     DESCRIPTION
         "This object specifies the timeproperties dataset value of
         whether current UTC offset is valid."
     REFERENCE       "Section 8.2.4.2 of [IEEE 1588-2008]"
     ::= { ptpbaseClockTimePropertiesDSEntry 4 }


ptpbaseClockTimePropertiesDSCurrentUTCOffset OBJECT-TYPE
     SYNTAX          Integer32
     MAX-ACCESS      read-only
     STATUS          current
     DESCRIPTION
         "This object specifies the timeproperties dataset value of
         current UTC offset.

         In PTP systems whose epoch is the PTP epoch, the value of
         timePropertiesDS.currentUtcOffset is the offset
         between TAI and UTC; otherwise the value has no meaning. The
         value shall be in units of seconds.
         The initialization value shall be selected as follows:
         a) If the timePropertiesDS.ptpTimescale (see 8.2.4.8) is TRUE,
         the value is the value obtained from a
         primary reference if the value is known at the time of
         initialization, else,
         b) The value shall be the current number of leap seconds (7.2.3)
         when the node is designed."
```

```
    REFERENCE        "Section 8.2.4.3 of [IEEE 1588-2008]"
    ::= { ptpbaseClockTimePropertiesDSEntry 5 }

ptpbaseClockTimePropertiesDSLeap59 OBJECT-TYPE
    SYNTAX          TruthValue
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "This object specifies the Leap59 value in the clock Current
        Dataset."
    REFERENCE        "Section 8.2.4.4 of [IEEE 1588-2008]"
    ::= { ptpbaseClockTimePropertiesDSEntry 6 }

ptpbaseClockTimePropertiesDSLeap61 OBJECT-TYPE
    SYNTAX          TruthValue
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "This object specifies the Leap61 value in the clock Current
        Dataset."
    REFERENCE        "Section 8.2.4.5 of [IEEE 1588-2008]"
    ::= { ptpbaseClockTimePropertiesDSEntry 7 }

ptpbaseClockTimePropertiesDSTimeTraceable OBJECT-TYPE
    SYNTAX          TruthValue
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "This object specifies the Timetraceable value in the clock
        Current Dataset."
    REFERENCE        "Section 8.2.4.6 of [IEEE 1588-2008]"
    ::= { ptpbaseClockTimePropertiesDSEntry 8 }

ptpbaseClockTimePropertiesDSFreqTraceable OBJECT-TYPE
    SYNTAX          TruthValue
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "This object specifies the Frequency Traceable value in the
        clock Current Dataset."
    REFERENCE        "Section 8.2.4.7 of [IEEE 1588-2008]"
    ::= { ptpbaseClockTimePropertiesDSEntry 9 }
```

ptpbaseClockTimePropertiesDSPTPTimescale OBJECT-TYPE
    SYNTAX   TruthValue
    MAX-ACCESS   read-only
    STATUS   current
    DESCRIPTION
        "This object specifies the PTP Timescale value in the clock
        Current Dataset."
    REFERENCE   "Section 8.2.4.8 of [IEEE 1588-2008]"
    ::= { ptpbaseClockTimePropertiesDSEntry 10 }

ptpbaseClockTimePropertiesDSSource OBJECT-TYPE
    SYNTAX   ClockTimeSourceType
    MAX-ACCESS   read-only
    STATUS   current
    DESCRIPTION
        "This object specifies the Timesource value in the clock Current
        Dataset."
    REFERENCE   "Section 8.2.4.9 of [IEEE 1588-2008]"
    ::= { ptpbaseClockTimePropertiesDSEntry 11 }

ptpbaseClockTransDefaultDSTable OBJECT-TYPE
    SYNTAX   SEQUENCE OF PtpbaseClockTransDefaultDSEntry
    MAX-ACCESS   not-accessible
    STATUS   current
    DESCRIPTION
        "Table of information about the PTP Transparent clock Default
        Datasets for all domains."
    ::= { ptpbaseMIBClockInfo 6 }

ptpbaseClockTransDefaultDSEntry OBJECT-TYPE
    SYNTAX   PtpbaseClockTransDefaultDSEntry
    MAX-ACCESS   not-accessible
    STATUS   current
    DESCRIPTION
        "An entry in the table, containing information about a single
        PTP Transparent clock Default Datasets for a domain."
    REFERENCE   "Section 8.3.2 of [IEEE 1588-2008]"
    INDEX   {
                        ptpbaseClockTransDefaultDSDomainIndex,
                        ptpbaseClockTransDefaultDSInstanceIndex
                    }

    ::= { ptpbaseClockTransDefaultDSTable 1 }

PtpbaseClockTransDefaultDSEntry ::= SEQUENCE {
        ptpbaseClockTransDefaultDSDomainIndex   ClockDomainType,
        ptpbaseClockTransDefaultDSInstanceIndex ClockInstanceType,
        ptpbaseClockTransDefaultDSClockIdentity ClockIdentity,
        ptpbaseClockTransDefaultDSNumOfPorts   Counter32,
        ptpbaseClockTransDefaultDSDelay   ClockMechanismType,
        ptpbaseClockTransDefaultDSPrimaryDomain ClockDomainType
}

ptpbaseClockTransDefaultDSDomainIndex OBJECT-TYPE
    SYNTAX   ClockDomainType
    MAX-ACCESS   not-accessible
    STATUS   current
    DESCRIPTION
        "This object specifies the domain number used to create logical
        group of PTP devices."
    ::= { ptpbaseClockTransDefaultDSEntry 1 }

ptpbaseClockTransDefaultDSInstanceIndex OBJECT-TYPE
    SYNTAX   ClockInstanceType
    MAX-ACCESS   not-accessible
    STATUS   current
    DESCRIPTION
        "This object specifies the instance of the clock for this clock
        type in the given domain."
    ::= { ptpbaseClockTransDefaultDSEntry 2 }

ptpbaseClockTransDefaultDSClockIdentity OBJECT-TYPE
    SYNTAX   ClockIdentity
    MAX-ACCESS   read-only
    STATUS   current
    DESCRIPTION
        "This object specifies the value of the clockIdentity attribute
        of the local clock."
    REFERENCE   "Section 8.3.2.2.1 of [IEEE 1588-2008]"
    ::= { ptpbaseClockTransDefaultDSEntry 3 }

ptpbaseClockTransDefaultDSNumOfPorts OBJECT-TYPE
    SYNTAX   Counter32
    MAX-ACCESS   read-only
    STATUS   current

      DESCRIPTION
          "This object specifies the number of PTP ports of the device."
      REFERENCE   "Section 8.3.2.2.2 of [IEEE 1588-2008]"
      ::= { ptpbaseClockTransDefaultDSEntry 4 }

  ptpbaseClockTransDefaultDSDelay OBJECT-TYPE
      SYNTAX   ClockMechanismType
      MAX-ACCESS   read-only
      STATUS   current
      DESCRIPTION
          "This object, if the transparent clock is an end-to-end
          transparent clock, has the value shall be E2E; If the
          transparent clock is a peer-to-peer transparent clock, the
          value
          shall be P2P."
      REFERENCE   "Section 8.3.2.3.1 of [IEEE 1588-2008]"
      ::= { ptpbaseClockTransDefaultDSEntry 5 }

  ptpbaseClockTransDefaultDSPrimaryDomain OBJECT-TYPE
      SYNTAX   ClockDomainType
      MAX-ACCESS   read-only
      STATUS   current
      DESCRIPTION
          "This object specifies the value of the primary syntonization
          domain. The initialization value shall be 0."
      REFERENCE   "Section 8.3.2.3.2 of [IEEE 1588-2008]"
      ::= { ptpbaseClockTransDefaultDSEntry 6 }

  ptpbaseClockPortTable OBJECT-TYPE
      SYNTAX   SEQUENCE OF PtpbaseClockPortEntry
      MAX-ACCESS   not-accessible
      STATUS   current
      DESCRIPTION
          "Table of information about the clock ports for a particular
          domain."
      ::= { ptpbaseMIBClockInfo 7 }

  ptpbaseClockPortEntry OBJECT-TYPE
      SYNTAX   PtpbaseClockPortEntry
      MAX-ACCESS   not-accessible
      STATUS   current

```
    DESCRIPTION
        "An entry in the table, containing information about a single
        clock port."
    INDEX           {
                        ptpbaseClockPortDomainIndex,
                        ptpbaseClockPortClockTypeIndex,
                        ptpbaseClockPortClockInstanceIndex,
                        ptpbaseClockPortTablePortNumberIndex
                    }
    ::= { ptpbaseClockPortTable 1 }

PtpbaseClockPortEntry ::= SEQUENCE {
        ptpbaseClockPortDomainIndex         ClockDomainType,
        ptpbaseClockPortClockTypeIndex      ClockType,
        ptpbaseClockPortClockInstanceIndex  ClockInstanceType,
        ptpbaseClockPortTablePortNumberIndex ClockPortNumber,
        ptpbaseClockPortName                DisplayString,
        ptpbaseClockPortRole                ClockRoleType,
        ptpbaseClockPortSyncTwoStep         TruthValue,
        ptpbaseClockPortCurrentPeerAddressType AutonomousType,
        ptpbaseClockPortCurrentPeerAddress
ClockPortTransportTypeAddress,
        ptpbaseClockPortNumOfAssociatedPorts   Gauge32
}

ptpbaseClockPortDomainIndex OBJECT-TYPE
    SYNTAX          ClockDomainType
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "This object specifies the domain number used to create logical
        group of PTP devices."
    ::= { ptpbaseClockPortEntry 1 }

ptpbaseClockPortClockTypeIndex OBJECT-TYPE
    SYNTAX          ClockType
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "This object specifies the clock type as defined in the
        Textual convention description."
    ::= { ptpbaseClockPortEntry 2 }
```

ptpbaseClockPortClockInstanceIndex OBJECT-TYPE
    SYNTAX   ClockInstanceType
    MAX-ACCESS   not-accessible
    STATUS   current
    DESCRIPTION
        "This object specifies the instance of the clock for this clock
        type in the given domain."
    ::= { ptpbaseClockPortEntry 3 }

ptpbaseClockPortTablePortNumberIndex OBJECT-TYPE
    SYNTAX   ClockPortNumber
    MAX-ACCESS   not-accessible
    STATUS   current
    DESCRIPTION
        "This object specifies the PTP Portnumber for this port."
    ::= { ptpbaseClockPortEntry 4 }

ptpbaseClockPortName OBJECT-TYPE
    SYNTAX   DisplayString (SIZE (1..64))
    MAX-ACCESS   read-only
    STATUS   current
    DESCRIPTION
        "This object specifies the PTP clock port name configured on the
        router."
    ::= { ptpbaseClockPortEntry 5 }

ptpbaseClockPortRole OBJECT-TYPE
    SYNTAX   ClockRoleType
    MAX-ACCESS   read-only
    STATUS   current
    DESCRIPTION
        "This object describes the current role (slave/master) of the
        port."
    ::= { ptpbaseClockPortEntry 6 }

ptpbaseClockPortSyncTwoStep OBJECT-TYPE
    SYNTAX   TruthValue
    MAX-ACCESS   read-only
    STATUS   current
    DESCRIPTION
        "This object specifies that two-step clock operation between
        the PTP master and slave device is enabled."
    ::= { ptpbaseClockPortEntry 7 }

ptpbaseClockPortCurrentPeerAddressType OBJECT-TYPE
    SYNTAX   AutonomousType
    MAX-ACCESS   read-only
    STATUS   current
    DESCRIPTION
        "This object specifies the current peer's network address type
         used for PTP communication."
    ::= { ptpbaseClockPortEntry 8 }

ptpbaseClockPortCurrentPeerAddress OBJECT-TYPE
    SYNTAX   ClockPortTransportTypeAddress
    MAX-ACCESS   read-only
    STATUS   current
    DESCRIPTION
        "This object specifies the current peer's network address used
        for PTP communication."
    ::= { ptpbaseClockPortEntry 9 }

ptpbaseClockPortNumOfAssociatedPorts OBJECT-TYPE
    SYNTAX   Gauge32
    MAX-ACCESS   read-only
    STATUS   current
    DESCRIPTION
        "This object specifies -
        For a master port - the number of PTP slave sessions (peers)
        associated with this PTP port.
        For a slave port - the number of masters available to this slave
        port (might or might not be peered)."
    ::= { ptpbaseClockPortEntry 10 }

ptpbaseClockPortDSTable OBJECT-TYPE
    SYNTAX   SEQUENCE OF PtpbaseClockPortDSEntry
    MAX-ACCESS   not-accessible
    STATUS   current
    DESCRIPTION
        "Table of information about the clock ports dataset for a
        particular domain."
    ::= { ptpbaseMIBClockInfo 8 }

ptpbaseClockPortDSEntry OBJECT-TYPE

```
    SYNTAX          PtpbaseClockPortDSEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "An entry in the table, containing port dataset information for
        a single clock port."
    INDEX           {
                        ptpbaseClockPortDSDomainIndex,
                        ptpbaseClockPortDSClockTypeIndex,
                        ptpbaseClockPortDSClockInstanceIndex,
                        ptpbaseClockPortDSPortNumberIndex
                    }
    ::= { ptpbaseClockPortDSTable 1 }

PtpbaseClockPortDSEntry ::= SEQUENCE {
        ptpbaseClockPortDSDomainIndex         ClockDomainType,
        ptpbaseClockPortDSClockTypeIndex      ClockType,
        ptpbaseClockPortDSClockInstanceIndex  ClockInstanceType,
        ptpbaseClockPortDSPortNumberIndex     ClockPortNumber,
        ptpbaseClockPortDSName                DisplayString,
        ptpbaseClockPortDSPortIdentity        OCTET STRING,
        ptpbaseClockPortDSlogAnnouncementInterval ClockIntervalBase2,
        ptpbaseClockPortDSAnnounceRctTimeout  Integer32,
        ptpbaseClockPortDSlogSyncInterval     ClockIntervalBase2,
        ptpbaseClockPortDSMinDelayReqInterval Integer32,
        ptpbaseClockPortDSPeerDelayReqInterval Integer32,
        ptpbaseClockPortDSDelayMech           ClockMechanismType,
        ptpbaseClockPortDSPeerMeanPathDelay   ClockTimeInterval,
        ptpbaseClockPortDSGrantDuration       Unsigned32,
        ptpbaseClockPortDSPTPVersion          Unsigned32
}

ptpbaseClockPortDSDomainIndex OBJECT-TYPE
    SYNTAX          ClockDomainType
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "This object specifies the domain number used to create logical
        group of PTP devices."
    ::= { ptpbaseClockPortDSEntry 1 }

ptpbaseClockPortDSClockTypeIndex OBJECT-TYPE
    SYNTAX          ClockType
```

```
    MAX-ACCESS        not-accessible
    STATUS            current
    DESCRIPTION
        "This object specifies the clock type as defined in the
        Textual convention description."
    ::= { ptpbaseClockPortDSEntry 2 }

ptpbaseClockPortDSClockInstanceIndex OBJECT-TYPE
    SYNTAX            ClockInstanceType
    MAX-ACCESS        not-accessible
    STATUS            current
    DESCRIPTION
        "This object specifies the instance of the clock for this clock
        type in the given domain."
    ::= { ptpbaseClockPortDSEntry 3 }

ptpbaseClockPortDSPortNumberIndex OBJECT-TYPE
    SYNTAX            ClockPortNumber
    MAX-ACCESS        not-accessible
    STATUS            current
    DESCRIPTION
        "This object specifies the PTP portnumber associated with this
        PTP port."
    ::= { ptpbaseClockPortDSEntry 4 }

ptpbaseClockPortDSName OBJECT-TYPE
    SYNTAX            DisplayString (SIZE  (1..64))
    MAX-ACCESS        read-only
    STATUS            current
    DESCRIPTION
        "This object specifies the PTP clock port dataset name."
    ::= { ptpbaseClockPortDSEntry 5 }

ptpbaseClockPortDSPortIdentity OBJECT-TYPE
    SYNTAX            OCTET STRING(SIZE(1..256))
    MAX-ACCESS        read-only
    STATUS            current
    DESCRIPTION
        "This object specifies the PTP clock port Identity."
    ::= { ptpbaseClockPortDSEntry 6 }

ptpbaseClockPortDSlogAnnouncementInterval OBJECT-TYPE
    SYNTAX            ClockIntervalBase2
```

```
    UNITS           "Time Interval"
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "This object specifies the Announce message transmission
        interval associated with this clock port."
    ::= { ptpbaseClockPortDSEntry 7 }

ptpbaseClockPortDSAnnounceRctTimeout OBJECT-TYPE
    SYNTAX          Integer32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "This object specifies the Announce receipt timeout associated
        with this clock port."
    ::= { ptpbaseClockPortDSEntry 8 }

ptpbaseClockPortDSlogSyncInterval OBJECT-TYPE
    SYNTAX          ClockIntervalBase2
    UNITS           "Time Interval"
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "This object specifies the Sync message transmission interval."
    ::= { ptpbaseClockPortDSEntry 9 }

ptpbaseClockPortDSMinDelayReqInterval OBJECT-TYPE
    SYNTAX          Integer32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "This object specifies the Delay_Req message transmission
        interval."
    ::= { ptpbaseClockPortDSEntry 10 }

ptpbaseClockPortDSPeerDelayReqInterval OBJECT-TYPE
    SYNTAX          Integer32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "This object specifies the Pdelay_Req message transmission
        interval."
    ::= { ptpbaseClockPortDSEntry 11 }
```

ptpbaseClockPortDSDelayMech OBJECT-TYPE
    SYNTAX   ClockMechanismType
    MAX-ACCESS   read-only
    STATUS   current
    DESCRIPTION
        "This object specifies the delay mechanism used. If the clock
        is an end-to-end clock, the value of the is e2e, else if the
        clock is a peer to-peer clock, the value shall be p2p."
    ::= { ptpbaseClockPortDSEntry 12 }

ptpbaseClockPortDSPeerMeanPathDelay OBJECT-TYPE
    SYNTAX   ClockTimeInterval
    UNITS   "Time Interval"
    MAX-ACCESS   read-only
    STATUS   current
    DESCRIPTION
        "This object specifies the peer meanPathDelay."
    ::= { ptpbaseClockPortDSEntry 13 }

ptpbaseClockPortDSGrantDuration OBJECT-TYPE
    SYNTAX   Unsigned32
    UNITS   "seconds"
    MAX-ACCESS   read-only
    STATUS   current
    DESCRIPTION
        "This object specifies the grant duration allocated by the
        master."
    ::= { ptpbaseClockPortDSEntry 14 }

ptpbaseClockPortDSPTPVersion OBJECT-TYPE
    SYNTAX   Unsigned32
    MAX-ACCESS   read-only
    STATUS   current
    DESCRIPTION
        "This object specifies the PTP version being used."
    ::= { ptpbaseClockPortDSEntry 15 }

ptpbaseClockPortRunningTable OBJECT-TYPE
    SYNTAX   SEQUENCE OF PtpbaseClockPortRunningEntry
    MAX-ACCESS   not-accessible

```
     STATUS           current
     DESCRIPTION
         "Table of information about the clock ports running dataset for
         a particular domain."
     ::= { ptpbaseMIBClockInfo 9 }

ptpbaseClockPortRunningEntry OBJECT-TYPE
     SYNTAX           PtpbaseClockPortRunningEntry
     MAX-ACCESS       not-accessible
     STATUS           current
     DESCRIPTION
         "An entry in the table, containing runing dataset information
         about a single clock port."
     INDEX            {
                         ptpbaseClockPortRunningDomainIndex,
                         ptpbaseClockPortRunningClockTypeIndex,
                         ptpbaseClockPortRunningClockInstanceIndex,
                         ptpbaseClockPortRunningPortNumberIndex
                     }
     ::= { ptpbaseClockPortRunningTable 1 }

PtpbaseClockPortRunningEntry ::= SEQUENCE {
         ptpbaseClockPortRunningDomainIndex        ClockDomainType,
         ptpbaseClockPortRunningClockTypeIndex     ClockType,
         ptpbaseClockPortRunningClockInstanceIndex ClockInstanceType,
         ptpbaseClockPortRunningPortNumberIndex    ClockPortNumber,
         ptpbaseClockPortRunningName               DisplayString,
         ptpbaseClockPortRunningState              ClockPortState,
         ptpbaseClockPortRunningRole               ClockRoleType,
         ptpbaseClockPortRunningInterfaceIndex     InterfaceIndexOrZero,
         ptpbaseClockPortRunningTransport          AutonomousType,
         ptpbaseClockPortRunningEncapsulationType  AutonomousType,
         ptpbaseClockPortRunningTxMode             ClockTxModeType,
         ptpbaseClockPortRunningRxMode             ClockTxModeType,
         ptpbaseClockPortRunningPacketsReceived    Counter64,
         ptpbaseClockPortRunningPacketsSent        Counter64
}

ptpbaseClockPortRunningDomainIndex OBJECT-TYPE
     SYNTAX           ClockDomainType
     MAX-ACCESS       not-accessible
     STATUS           current
     DESCRIPTION
```

```
    "This object specifies the domain number used to create logical
    group of PTP devices."
::= { ptpbaseClockPortRunningEntry 1 }
```

ptpbaseClockPortRunningClockTypeIndex OBJECT-TYPE
```
    SYNTAX          ClockType
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "This object specifies the clock type as defined in the
        Textual convention description."
::= { ptpbaseClockPortRunningEntry 2 }
```

ptpbaseClockPortRunningClockInstanceIndex OBJECT-TYPE
```
    SYNTAX          ClockInstanceType
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "This object specifies the instance of the clock for this clock
        type in the given domain."
::= { ptpbaseClockPortRunningEntry 3 }
```

ptpbaseClockPortRunningPortNumberIndex OBJECT-TYPE
```
    SYNTAX          ClockPortNumber
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "This object specifies the PTP portnumber associated with this
        clock port."
::= { ptpbaseClockPortRunningEntry 4 }
```

ptpbaseClockPortRunningName OBJECT-TYPE
```
    SYNTAX          DisplayString (SIZE  (1..64))
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "This object specifies the PTP clock port name."
::= { ptpbaseClockPortRunningEntry 5 }
```

ptpbaseClockPortRunningState OBJECT-TYPE
```
    SYNTAX          ClockPortState
    MAX-ACCESS      read-only
    STATUS          current
```

        DESCRIPTION
            "This object specifies the port state returned by PTP engine.

            initializing - In this state a port initializes
                           its data sets, hardware, and
                           communication facilities.
            faulty   - The fault state of the protocol.
            disabled   - The port shall not place any
                           messages on its communication path.
            listening   - The port is waiting for the
                           announceReceiptTimeout to expire or
                           to receive an Announce message from
                           a master.
            preMaster   - The port shall behave in all respects
                           as though it were in the MASTER state
                           except that it shall not place any
                           messages on its communication path
                           except for Pdelay_Req, Pdelay_Resp,
                           Pdelay_Resp_Follow_Up, signaling, or
                           management messages.
            master   - The port is behaving as a master port.
            passive   - The port shall not place any
                           messages on its communication path
                           except for Pdelay_Req, Pdelay_Resp,
                           Pdelay_Resp_Follow_Up, or signaling
                           messages, or management messages
                           that are a required response to
                           another management message
            uncalibrated - The local port is preparing to
                           synchronize to the master port.
            slave   - The port is synchronizing to the
                           selected master port."
        ::= { ptpbaseClockPortRunningEntry 6 }

    ptpbaseClockPortRunningRole OBJECT-TYPE
        SYNTAX   ClockRoleType
        MAX-ACCESS   read-only
        STATUS   current
        DESCRIPTION
            "This object specifies the Clock Role."
        ::= { ptpbaseClockPortRunningEntry 7 }

    ptpbaseClockPortRunningInterfaceIndex OBJECT-TYPE

```
    SYNTAX          InterfaceIndexOrZero
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "This object specifies the interface on the router being used by
        the PTP Clock for PTP communication."
    ::= { ptpbaseClockPortRunningEntry 8 }

ptpbaseClockPortRunningTransport OBJECT-TYPE
    SYNTAX          AutonomousType
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "This object specifies the transport protocol being used for PTP
        communication (the mapping used)."
    ::= { ptpbaseClockPortRunningEntry 9 }

ptpbaseClockPortRunningEncapsulationType OBJECT-TYPE
    SYNTAX          AutonomousType
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "This object specifies the type of encapsulation if the
        interface is adding extra layers (eg. VLAN, Pseudowire
        encapsulation...) for the PTP messages."
    ::= { ptpbaseClockPortRunningEntry 10 }

ptpbaseClockPortRunningTxMode OBJECT-TYPE
    SYNTAX          ClockTxModeType
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "This object specifies the clock transmission mode as

        unicast:       Using unicast communication channel.
        multicast:     Using Multicast communication channel.
        multicast-mix: Using multicast-unicast communication channel"
    ::= { ptpbaseClockPortRunningEntry 11 }

ptpbaseClockPortRunningRxMode OBJECT-TYPE
    SYNTAX          ClockTxModeType
    MAX-ACCESS      read-only
    STATUS          current
```

```
    DESCRIPTION
        "This object specifie the clock receive mode as

        unicast:       Using unicast communication channel.
        multicast:     Using Multicast communication channel.
        multicast-mix: Using multicast-unicast communication channel"
    ::= { ptpbaseClockPortRunningEntry 12 }

ptpbaseClockPortRunningPacketsReceived OBJECT-TYPE
    SYNTAX          Counter64
    UNITS           "packets"
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "This object specifies the packets received on the clock port
        (cummulative)."
    ::= { ptpbaseClockPortRunningEntry 13 }

ptpbaseClockPortRunningPacketsSent OBJECT-TYPE
    SYNTAX          Counter64
    UNITS           "packets"
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "This object specifies the packets sent on the clock port
        (cummulative)."
    ::= { ptpbaseClockPortRunningEntry 14 }



ptpbaseClockPortTransDSTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF PtpbaseClockPortTransDSEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "Table of information about the Transparent clock ports running
        dataset for a particular domain."
    ::= { ptpbaseMIBClockInfo 10 }

ptpbaseClockPortTransDSEntry OBJECT-TYPE
    SYNTAX          PtpbaseClockPortTransDSEntry
    MAX-ACCESS      not-accessible
    STATUS          current
```

```
     DESCRIPTION
         "An entry in the table, containing clock port Transparent
         dataset information about a single clock port"
     INDEX          {
                        ptpbaseClockPortTransDSDomainIndex,
                        ptpbaseClockPortTransDSInstanceIndex,
                        ptpbaseClockPortTransDSPortNumberIndex
                    }
     ::= { ptpbaseClockPortTransDSTable 1 }

PtpbaseClockPortTransDSEntry ::= SEQUENCE {
        ptpbaseClockPortTransDSDomainIndex      ClockDomainType,
        ptpbaseClockPortTransDSInstanceIndex    ClockInstanceType,
        ptpbaseClockPortTransDSPortNumberIndex  ClockPortNumber,
        ptpbaseClockPortTransDSPortIdentity     ClockIdentity,
        ptpbaseClockPortTransDSlogMinPdelayReqInt ClockIntervalBase2,
        ptpbaseClockPortTransDSFaultyFlag       TruthValue,
        ptpbaseClockPortTransDSPeerMeanPathDelay  ClockTimeInterval
}

ptpbaseClockPortTransDSDomainIndex OBJECT-TYPE
     SYNTAX          ClockDomainType
     MAX-ACCESS      not-accessible
     STATUS          current
     DESCRIPTION
         "This object specifies the domain number used to create logical
         group of PTP devices."
     ::= { ptpbaseClockPortTransDSEntry 1 }

ptpbaseClockPortTransDSInstanceIndex OBJECT-TYPE
     SYNTAX          ClockInstanceType
     MAX-ACCESS      not-accessible
     STATUS          current
     DESCRIPTION
         "This object specifies the instance of the clock for this clock
         type in the given domain."
     ::= { ptpbaseClockPortTransDSEntry 2 }

ptpbaseClockPortTransDSPortNumberIndex OBJECT-TYPE
     SYNTAX          ClockPortNumber
     MAX-ACCESS      not-accessible
     STATUS          current
     DESCRIPTION
```

          "This object specifies the PTP port number associated with this
          port."
       REFERENCE   "Section 7.5.2 Port Identity [IEEE 1588-2008]"
       ::= { ptpbaseClockPortTransDSEntry 3 }

   ptpbaseClockPortTransDSPortIdentity OBJECT-TYPE
       SYNTAX   ClockIdentity
       MAX-ACCESS   read-only
       STATUS   current
       DESCRIPTION
          "This object specifies the value of the PortIdentity
          attribute of the local port."
       REFERENCE   "Section 8.3.3.2.1 of [IEEE 1588-2008]"
       ::= { ptpbaseClockPortTransDSEntry 4 }

   ptpbaseClockPortTransDSlogMinPdelayReqInt OBJECT-TYPE
       SYNTAX   ClockIntervalBase2
       MAX-ACCESS   read-only
       STATUS   current
       DESCRIPTION
          "This object specifies the value of the logarithm to the
          base 2 of the minPdelayReqInterval."
       REFERENCE   "Section 8.3.3.3.1 of [IEEE 1588-2008]"
       ::= { ptpbaseClockPortTransDSEntry 5 }

   ptpbaseClockPortTransDSFaultyFlag OBJECT-TYPE
       SYNTAX   TruthValue
       MAX-ACCESS   read-only
       STATUS   current
       DESCRIPTION
          "This object specifies the value TRUE if the port is faulty
          and FALSE if the port is operating normally."
       REFERENCE   "Section 8.3.3.3.2 of [IEEE 1588-2008]"
       ::= { ptpbaseClockPortTransDSEntry 6 }

   ptpbaseClockPortTransDSPeerMeanPathDelay OBJECT-TYPE
       SYNTAX   ClockTimeInterval
       UNITS   "Time Interval"
       MAX-ACCESS   read-only
       STATUS   current
       DESCRIPTION
          "This object specifies, (if the delayMechanism used is P2P) the
          value is the estimate of the current one-way propagation delay,

        i.e., <meanPathDelay> on the link attached to this port,
        computed using the peer delay mechanism. If the value of the
        delayMechanism used is E2E, then the value will be zero."
    REFERENCE   "Section 8.3.3.3.3 of [IEEE 1588-2008]"
    ::= { ptpbaseClockPortTransDSEntry 7 }

ptpbaseClockPortAssociateTable OBJECT-TYPE
    SYNTAX   SEQUENCE OF PtpbaseClockPortAssociateEntry
    MAX-ACCESS   not-accessible
    STATUS   current
    DESCRIPTION
        "Table of information about a given port's associated ports.

        For a master port: multiple slave ports which have established
                           sessions with the current master port.
        For a slave port: the list of masters available for a given
                           slave port.

        Session information (packets, errors) to be displayed based on
        availability and scenario."
    ::= { ptpbaseMIBClockInfo 11 }

--
-- Well Known transport types for PTP communication.
--
ptpbaseWellKnownTransportTypes OBJECT IDENTIFIER ::= {
ptpbaseMIBClockInfo 12 }

ptpbaseTransportTypeIPversion4 OBJECT-IDENTITY
    STATUS current
    DESCRIPTION
        "IP version 4"
    ::= { ptpbaseWellKnownTransportTypes 1 }

ptpbaseTransportTypeIPversion6 OBJECT-IDENTITY
   STATUS current
     DESCRIPTION
        "IP version 6"
     ::= { ptpbaseWellKnownTransportTypes 2 }

```
ptpbaseTransportTypeEthernet OBJECT-IDENTITY
    STATUS current
     DESCRIPTION
        "Ethernet"
     ::= { ptpbaseWellKnownTransportTypes 3 }

ptpbaseTransportTypeDeviceNET OBJECT-IDENTITY
    STATUS current
     DESCRIPTION
        "Device NET"
     ::= { ptpbaseWellKnownTransportTypes 4 }

ptpbaseTransportTypeControlNET OBJECT-IDENTITY
    STATUS current
     DESCRIPTION
        "Control NET"
     ::= { ptpbaseWellKnownTransportTypes 5 }


ptpbaseTransportTypeIEC61158 OBJECT-IDENTITY
    STATUS current
     DESCRIPTION
        "IEC61158"
     ::= { ptpbaseWellKnownTransportTypes 6 }


--
-- Well Known encapsulation types for PTP communication.
--
ptpbaseWellKnownEncapsulationTypes OBJECT IDENTIFIER ::= {
ptpbaseMIBClockInfo 13 }

ptpbaseEncapsulationTypeEthernet OBJECT-IDENTITY
    STATUS current
    DESCRIPTION
        "Ethernet Encapsulation type."
    ::= { ptpbaseWellKnownEncapsulationTypes 1 }


ptpbaseEncapsulationTypeVLAN OBJECT-IDENTITY
    STATUS current
    DESCRIPTION
        "VLAN Encapsulation type."
```

```
    ::= { ptpbaseWellKnownEncapsulationTypes 2 }


ptpbaseEncapsulationTypeUDPIPLSP OBJECT-IDENTITY
   STATUS current
     DESCRIPTION
       "UDP/IP over MPLS Encapsulation type."
     ::= { ptpbaseWellKnownEncapsulationTypes 3 }


ptpbaseEncapsulationTypePWUDPIPLSP OBJECT-IDENTITY
   STATUS current
     DESCRIPTION
       "UDP/IP Pseudowire over MPLS Encapsulation type."
     ::= { ptpbaseWellKnownEncapsulationTypes 4 }


ptpbaseEncapsulationTypePWEthernetLSP OBJECT-IDENTITY
   STATUS current
     DESCRIPTION
       "Ethernet Pseudowire over MPLS Encapsulation type."
     ::= { ptpbaseWellKnownEncapsulationTypes 5 }




ptpbaseClockPortAssociateEntry OBJECT-TYPE
    SYNTAX          PtpbaseClockPortAssociateEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "An entry in the table, containing information about a single
        associated port for the given clockport."
    INDEX           {
                        ptpClockPortCurrentDomainIndex,
                        ptpClockPortCurrentClockTypeIndex,
                        ptpClockPortCurrentClockInstanceIndex,
                        ptpClockPortCurrentPortNumberIndex,
                        ptpbaseClockPortAssociatePortIndex
                    }
    ::= { ptpbaseClockPortAssociateTable 1 }

PtpbaseClockPortAssociateEntry ::= SEQUENCE {
        ptpClockPortCurrentDomainIndex          ClockDomainType,
        ptpClockPortCurrentClockTypeIndex       ClockType,
        ptpClockPortCurrentClockInstanceIndex   ClockInstanceType,
        ptpClockPortCurrentPortNumberIndex      ClockPortNumber,
```

```
        ptpbaseClockPortAssociatePortIndex      Unsigned32,
        ptpbaseClockPortAssociateAddressType    AutonomousType,
        ptpbaseClockPortAssociateAddress
ClockPortTransportTypeAddress,
        ptpbaseClockPortAssociatePacketsSent    Counter64,
        ptpbaseClockPortAssociatePacketsReceived Counter64,
        ptpbaseClockPortAssociateInErrors       Counter64,
        ptpbaseClockPortAssociateOutErrors      Counter64
}

ptpClockPortCurrentDomainIndex OBJECT-TYPE
    SYNTAX          ClockDomainType
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "This object specifies the given port's domain number."
    ::= { ptpbaseClockPortAssociateEntry 1 }

ptpClockPortCurrentClockTypeIndex OBJECT-TYPE
    SYNTAX          ClockType
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "This object specifies the given port's clock type."
    ::= { ptpbaseClockPortAssociateEntry 2 }

ptpClockPortCurrentClockInstanceIndex OBJECT-TYPE
    SYNTAX          ClockInstanceType
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "This object specifies the instance of the clock for this clock
        type in the given domain."
    ::= { ptpbaseClockPortAssociateEntry 3 }

ptpClockPortCurrentPortNumberIndex OBJECT-TYPE
    SYNTAX          ClockPortNumber
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "This object specifies the PTP Port Number for the given port."
    ::= { ptpbaseClockPortAssociateEntry 4 }
```

```
ptpbaseClockPortAssociatePortIndex OBJECT-TYPE
    SYNTAX          Unsigned32 (1..65535)
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "This object specifies the associated port's serial number in
        the current port's context."
    ::= { ptpbaseClockPortAssociateEntry 5 }

ptpbaseClockPortAssociateAddressType OBJECT-TYPE
    SYNTAX          AutonomousType
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "This object specifies the peer port's network address type used
        for PTP communication."
    ::= { ptpbaseClockPortAssociateEntry 6 }

ptpbaseClockPortAssociateAddress OBJECT-TYPE
    SYNTAX          ClockPortTransportTypeAddress
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "This object specifies the peer port's network address used for
        PTP communication."
    ::= { ptpbaseClockPortAssociateEntry 7 }

ptpbaseClockPortAssociatePacketsSent OBJECT-TYPE
    SYNTAX          Counter64
    UNITS           "packets"
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "The number of packets sent to this peer port from the current
        port."
    ::= { ptpbaseClockPortAssociateEntry 8 }

ptpbaseClockPortAssociatePacketsReceived OBJECT-TYPE
    SYNTAX          Counter64
    UNITS           "packets"
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
```

        "The number of packets received from this peer port by the
        current port."
    ::= { ptpbaseClockPortAssociateEntry 9 }

ptpbaseClockPortAssociateInErrors OBJECT-TYPE
    SYNTAX   Counter64
    UNITS   "packets"
    MAX-ACCESS   read-only
    STATUS   current
    DESCRIPTION
        "This object specifies the input errors associated with the
        peer port."
    ::= { ptpbaseClockPortAssociateEntry 10 }

ptpbaseClockPortAssociateOutErrors OBJECT-TYPE
    SYNTAX   Counter64
    UNITS   "packets"
    MAX-ACCESS   read-only
    STATUS   current
    DESCRIPTION
        "This object specifies the output errors associated with the
        peer port."
    ::= { ptpbaseClockPortAssociateEntry 11 }

-- Conformance Information Definition

ptpbaseMIBCompliances OBJECT IDENTIFIER
    ::= { ptpbaseMIBConformance 1 }

ptpbaseMIBGroups OBJECT IDENTIFIER
    ::= { ptpbaseMIBConformance 2 }

ptpbaseMIBCompliancesSystemInfo MODULE-COMPLIANCE
    STATUS   current
    DESCRIPTION
        "Compliance statement for agents that provide read-only support
        for PTPBASE-MIB to provide system level information of clock
        devices.
        Such devices can only be monitored using this MIB module.

        The Module is implemented with support for read-only. In other

```
        words, only monitoring is available by implementing this
        MODULE-COMPLIANCE."
    MODULE          -- this module
    MANDATORY-GROUPS { ptpbaseMIBSystemInfoGroup }
    ::= { ptpbaseMIBCompliances 1 }

ptpbaseMIBCompliancesClockInfo MODULE-COMPLIANCE
    STATUS          current
    DESCRIPTION
        "Compliance statement for agents that provide read-only support
        for PTPBASE-MIB to provide clock related information.
        Such devices can only be monitored using this MIB module.

        The Module is implemented with support for read-only. In other
        words, only monitoring is available by implementing this
        MODULE-COMPLIANCE."
    MODULE          -- this module
    MANDATORY-GROUPS {
                    ptpbaseMIBClockCurrentDSGroup,
                    ptpbaseMIBClockParentDSGroup,
                    ptpbaseMIBClockDefaultDSGroup,
                    ptpbaseMIBClockRunningGroup,
                    ptpbaseMIBClockTimepropertiesGroup
                }
    ::= { ptpbaseMIBCompliances 2 }

ptpbaseMIBCompliancesClockPortInfo MODULE-COMPLIANCE
    STATUS          current
    DESCRIPTION
        "Compliance statement for agents that provide read-only support
        for PTPBASE-MIB to provide clock port related information.
        Such devices can only be monitored using this MIB module.

        The Module is implemented with support for read-only. In other
        words, only monitoring is available by implementing this
        MODULE-COMPLIANCE."
    MODULE          -- this module
    MANDATORY-GROUPS {
                    ptpbaseMIBClockPortGroup,
                    ptpbaseMIBClockPortDSGroup,
                    ptpbaseMIBClockPortRunningGroup,
                    ptpbaseMIBClockPortAssociateGroup
                }
```

```
    ::= { ptpbaseMIBCompliances 3 }

ptpbaseMIBCompliancesTransparentClockInfo MODULE-COMPLIANCE
    STATUS          current
    DESCRIPTION
        "Compliance statement for agents that provide read-only support
        for PTPBASE-MIB to provide Transparent clock related
        information.
        Such devices can only be monitored using this MIB module.

        The Module is implemented with support for read-only. In other
        words, only monitoring is available by implementing this
        MODULE-COMPLIANCE."
    MODULE          -- this module
    MANDATORY-GROUPS {
                    ptpbaseMIBClockTranparentDSGroup,
                    ptpbaseMIBClockPortTransDSGroup
                }
    ::= { ptpbaseMIBCompliances 4 }

ptpbaseMIBSystemInfoGroup OBJECT-GROUP
    OBJECTS         {
                    ptpbaseSystemDomainTotals,
                    ptpDomainClockPortsTotal,
                    ptpbaseSystemProfile
                }
    STATUS          current
    DESCRIPTION
        "Group which aggregates objects describing system-wide
        information"
    ::= { ptpbaseMIBGroups 1 }

ptpbaseMIBClockCurrentDSGroup OBJECT-GROUP
    OBJECTS         {
                    ptpbaseClockCurrentDSStepsRemoved,
                    ptpbaseClockCurrentDSOffsetFromMaster,
                    ptpbaseClockCurrentDSMeanPathDelay
                }
    STATUS          current
    DESCRIPTION
        "Group which aggregates objects describing PTP Current Dataset
        information"
    ::= { ptpbaseMIBGroups 2 }
```

```
ptpbaseMIBClockParentDSGroup OBJECT-GROUP
    OBJECTS          {
                        ptpbaseClockParentDSParentPortIdentity,
                        ptpbaseClockParentDSParentStats,
                        ptpbaseClockParentDSOffset,
                        ptpbaseClockParentDSClockPhChRate,
                        ptpbaseClockParentDSGMClockIdentity,
                        ptpbaseClockParentDSGMClockPriority1,
                        ptpbaseClockParentDSGMClockPriority2,
                        ptpbaseClockParentDSGMClockQualityClass,
                        ptpbaseClockParentDSGMClockQualityAccuracy,
                        ptpbaseClockParentDSGMClockQualityOffset
                     }
    STATUS           current
    DESCRIPTION
        "Group which aggregates objects describing PTP Parent Dataset
        information"
    ::= { ptpbaseMIBGroups 3 }

ptpbaseMIBClockDefaultDSGroup OBJECT-GROUP
    OBJECTS          {
                        ptpbaseClockDefaultDSTwoStepFlag,
                        ptpbaseClockDefaultDSClockIdentity,
                        ptpbaseClockDefaultDSPriority1,
                        ptpbaseClockDefaultDSPriority2,
                        ptpbaseClockDefaultDSSlaveOnly,
                        ptpbaseClockDefaultDSQualityClass,
                        ptpbaseClockDefaultDSQualityAccuracy,
                        ptpbaseClockDefaultDSQualityOffset
                     }
    STATUS           current
    DESCRIPTION
        "Group which aggregates objects describing PTP Default Dataset
        information"
    ::= { ptpbaseMIBGroups 4 }

ptpbaseMIBClockRunningGroup OBJECT-GROUP
    OBJECTS          {
                        ptpbaseClockRunningState,
                        ptpbaseClockRunningPacketsSent,
                        ptpbaseClockRunningPacketsReceived
                     }
```

     STATUS   current
     DESCRIPTION
         "Group which aggregates objects describing PTP running state
         information"
     ::= { ptpbaseMIBGroups 5 }

ptpbaseMIBClockTimepropertiesGroup OBJECT-GROUP
     OBJECTS   {
                 ptpbaseClockTimePropertiesDSCurrentUTCOffsetValid,
                 ptpbaseClockTimePropertiesDSCurrentUTCOffset,
                 ptpbaseClockTimePropertiesDSLeap59,
                 ptpbaseClockTimePropertiesDSLeap61,
                 ptpbaseClockTimePropertiesDSTimeTraceable,
                 ptpbaseClockTimePropertiesDSFreqTraceable,
                 ptpbaseClockTimePropertiesDSPTPTimescale,
                 ptpbaseClockTimePropertiesDSSource
                }
     STATUS   current
     DESCRIPTION
         "Group which aggregates objects describing PTP Time Properties
         information"
     ::= { ptpbaseMIBGroups 6 }

ptpbaseMIBClockTranparentDSGroup OBJECT-GROUP
     OBJECTS   {
                       ptpbaseClockTransDefaultDSClockIdentity,
                       ptpbaseClockTransDefaultDSNumOfPorts,
                       ptpbaseClockTransDefaultDSDelay,
                       ptpbaseClockTransDefaultDSPrimaryDomain
                     }
     STATUS   current
     DESCRIPTION
         "Group which aggregates objects describing PTP Transparent
         Dataset
         information"
     ::= { ptpbaseMIBGroups 7 }

ptpbaseMIBClockPortGroup OBJECT-GROUP
     OBJECTS   {
                       ptpbaseClockPortName,
                       ptpbaseClockPortSyncTwoStep,
                       ptpbaseClockPortCurrentPeerAddress,
                       ptpbaseClockPortNumOfAssociatedPorts,

```
                        ptpbaseClockPortCurrentPeerAddressType,
                        ptpbaseClockPortRole
                    }
    STATUS          current
    DESCRIPTION
        "Group which aggregates objects describing information for a
        given PTP Port."
    ::= { ptpbaseMIBGroups 8 }

ptpbaseMIBClockPortDSGroup OBJECT-GROUP
    OBJECTS         {
                        ptpbaseClockPortDSName,
                        ptpbaseClockPortDSPortIdentity,
                        ptpbaseClockPortDSlogAnnouncementInterval,
                        ptpbaseClockPortDSAnnounceRctTimeout,
                        ptpbaseClockPortDSlogSyncInterval,
                        ptpbaseClockPortDSMinDelayReqInterval,
                        ptpbaseClockPortDSPeerDelayReqInterval,
                        ptpbaseClockPortDSDelayMech,
                        ptpbaseClockPortDSPeerMeanPathDelay,
                        ptpbaseClockPortDSGrantDuration,
                        ptpbaseClockPortDSPTPVersion
                    }
    STATUS          current
    DESCRIPTION
        "Group which aggregates objects describing PTP Port Dataset
        information"
    ::= { ptpbaseMIBGroups 9 }

ptpbaseMIBClockPortRunningGroup OBJECT-GROUP
    OBJECTS         {
                        ptpbaseClockPortRunningName,
                        ptpbaseClockPortRunningState,
                        ptpbaseClockPortRunningRole,
                        ptpbaseClockPortRunningInterfaceIndex,
                        ptpbaseClockPortRunningTransport,
                        ptpbaseClockPortRunningEncapsulationType,
                        ptpbaseClockPortRunningTxMode,
                        ptpbaseClockPortRunningRxMode,
                        ptpbaseClockPortRunningPacketsReceived,
                        ptpbaseClockPortRunningPacketsSent
                    }
    STATUS          current
```

```
     DESCRIPTION
         "Group which aggregates objects describing PTP running interface
         information"
     ::= { ptpbaseMIBGroups 10 }

ptpbaseMIBClockPortTransDSGroup OBJECT-GROUP
     OBJECTS          {
                         ptpbaseClockPortTransDSPortIdentity,
                         ptpbaseClockPortTransDSlogMinPdelayReqInt,
                         ptpbaseClockPortTransDSFaultyFlag,
                         ptpbaseClockPortTransDSPeerMeanPathDelay
                      }
     STATUS          current
     DESCRIPTION
         "Group which aggregates objects describing PTP TransparentDS
         Dataset
         information"
     ::= { ptpbaseMIBGroups 11 }

ptpbaseMIBClockPortAssociateGroup OBJECT-GROUP
     OBJECTS          {
                         ptpbaseClockPortAssociatePacketsSent,
                         ptpbaseClockPortAssociatePacketsReceived,
                         ptpbaseClockPortAssociateAddress,
                         ptpbaseClockPortAssociateAddressType,
                         ptpbaseClockPortAssociateInErrors,
                         ptpbaseClockPortAssociateOutErrors
                      }
     STATUS          current
     DESCRIPTION
         "Group which aggregates objects describing information on peer
         PTP ports for a given PTP clock-port."
     ::= { ptpbaseMIBGroups 12 }


END
```

5. Security Considerations

   This MIB contains readable objects whose values provide information
   related to PTP objects. It does not contain writable objects.

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

   SNMP versions prior to SNMPv3 did not include adequate security.
   Even if the network itself is secure (for example by using IPSec),
   even then, there is no control as to who on the secure network is
   allowed to access and GET (read) the objects in this MIB module.

   It is recommended that the implementers consider the security
   features as provided by the SNMPv3 framework (see [RFC 3410], section
   8). Specifically, the use of the User-based Security Model
   [RFC 3414] and the View-based Access Control Model [RFC 3415] is
   recommended.

   Further, deployment of SNMP versions prior to SNMPv3 is NOT
   recommended. Instead, it is recommended to deploy SNMPv3 and to
   enable cryptographic security. It is then a customer/operator
   responsibility to ensure that the SNMP entity giving access to an
   instance of this MIB is properly configured to give access to those
   objects only to those principals (users) that have legitimate rights
   to access them.

6. IANA Considerations

   The MIB module defined in this document uses the following IANA-
   assigned OBJECT IDENTIFIER value recorded in the SMI Numbers
   registry:

           Descriptor   OBJECT IDENTIFIER value
           ----------   -----------------------
           ptpbaseMIB   { mib-2 xxx }

   [NOTE for IANA: Please allocate an object identifier at
   http://www.iana.org/assignments/smi-numbers for object ptpbaseMIB.]

7. References

7.1. Normative References

   [IEEE 1588-2008] "IEEE Standard for A Precision Clock Synchronization
   Protocol for Networked Measurement and Control Systems", IEEE Std.
   1588(TM)-2008, 24 July 2008

7.2. Informative References

   [RFC 1155] Rose, M., and K. McCloghrie, "Structure and Identification
   of Management Information for TCP/IP-based Internets", STD 16, RFC
   1155, Performance Systems International, Hughes LAN Systems, May 1990

   [RFC 1157] Case, J., Fedor, M., Schoffstall, M., and J. Davin,
   "Simple Network Management Protocol", STD 15, RFC 1157, SNMP
   Research, Performance Systems International, Performance Systems
   International, MIT Laboratory for Computer Science, May 1990.

   [RFC 1212] Rose, M., and K. McCloghrie, "Concise MIB Definitions",
   STD 16, RFC 1212, Performance Systems International, Hughes LAN
   Systems, March 1991

   [RFC 1215] M. Rose, "A Convention for Defining Traps for use with the
   SNMP", RFC 1215, Performance Systems International, March 1991

   [RFC 1901] SNMPv2 Working Group, Case, J., McCloghrie, K., Rose, M.,
   and S. Waldbusser, "Introduction to Community-based SNMPv2", RFC
   1901, SNMP Research, Inc., Cisco Systems, Inc., Dover Beach
   Consulting, Inc., International Network Services, January 1996.

   [RFC 1906] SNMPv2 Working Group, Case, J., McCloghrie, K., Rose, M.,
   and S. Waldbusser, "Transport Mappings for Version 2 of the Simple
   Network Management Protocol (SNMPv2)", RFC 1906, SNMP Research, Inc.,
   Cisco Systems, Inc., Dover Beach Consulting, Inc., International
   Network Services, January 1996.

   [RFC 2578] McCloghrie, K., Perkins, D., and J. Schoenwaelder,
   "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC
   2578, April 1999.

   [RFC 2579] McCloghrie, K., Perkins, D., and J. Schoenwaelder,
   "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.

   [RFC 2580] McCloghrie, K., Perkins, D., and J. Schoenwaelder,
   "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.

   [RFC 3410] Case, J., Mundy, R., Partain, D., and B. Stewart,
   "Introduction and Applicability Statements for Internet Standard

Management Framework", RFC 3410 SNMP Research, Inc., Network
Associates Laboratories, Ericsson, December 2002.

[RFC 3411] Harrington, D., Presuhn, R., and B. Wijnen, "An
Architecture for Describing Simple Network Management Protocol (SNMP)
Management Frameworks", STD 62, RFC 3411, Enterasys Networks, BMC
Software, Inc., Lucent Technologies, December 2002

[RFC 3412] Case, J., Harrington D., Presuhn R., and B. Wijnen,
"Message Processing and Dispatching for the Simple Network Management
Protocol (SNMP)", STD 62, RFC 3412, SNMP Research, Inc., Enterasys
Networks, BMC Software, Inc., Lucent Technologies, December 2002.

[RFC 3413] Levi, D., Meyer, P., and B. Stewart, "Simple Network
Management Protocol (SNMP) Applications", STD 62, RFC 3413, Nortel
Networks, Secure Computing Corporation, December 2002.

[RFC 3414] Blumenthal, U., and B. Wijnen, "User-based Security Model
(USM) for version 3 of the Simple Network Management Protocol
(SNMPv3)", STD 62, RFC 3414, Lucent Technologies, December 2002.

[RFC 3415] Wijnen, B., Presuhn, R., and K. McCloghrie, "View-based
Access Control Model (VACM) for the Simple Network Management
Protocol (SNMP)", STD 62, RFC 3415, Lucent Technologies, BMC
Software, Inc., Cisco Systems, Inc., December 2002.

[RFC 3416] Presuhn, R. (Ed.), "Version 2 of the Protocol Operations
for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3416,
BMC Software, Inc., December 2002.

[RFC 3417] Presuhn, R. (Ed.), "Transport Mappings for the Simple
Network Management Protocol (SNMP)", STD 62, RFC 3417, BMC Software,
Inc., December 2002.

[RFC 5905] David L. Mills, " Network Time Protocol Version 4:
Protocol and Algorithms Specification", RFC 5905, University of
Delaware, June 2010.

[IEEE 802.3-2008] "IEEE Standard for 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", IEEE Std. 802.3 -
2008, 26 December 2008

[G.8265.1] "Precision time protocol telecom profile for frequency
synchronization", ITU-T Recommendation G.8265.1, October 2010.

8. Acknowledgements

   Thanks to John Linton and Danny Lee for valuable comments, and to
   Bert Wijnen, Kevin Gross, Alan Luchuk and Chris Elliot for their
   reviews of this MIB.

9. Author's Addresses

   Vinay Shankarkumar
   Cisco Systems,
   7025-4 Kit Creek Road,
   Research Triangle Park,
   NC 27560,
   USA.

   Email: vinays@cisco.com

   Laurent Montini,
   Cisco Systems,
   11, rue Camille Desmoulins,
   92782 Issy-les-Moulineaux,
   France.

   Email: lmontini@cisco.com

   Tim Frost,
   Calnex Solutions Ltd.,
   Oracle Campus,
   Linlithgow,
   EH49 7LR,
   UK.

   Email: tim.frost@calnexsol.com

   Greg Dowd,
   Microsemi Inc.,
   3870 N. First Str.,
   San Jose,
   CA 95134,
   USA.

   Email: greg.dowd@microsemi.com