Abstract

This informational document analyzes the accuracy issues with time synchronization protocols when time synchronization packets are encrypted during transmission. In addition, several candidate solutions on such issues are introduced.

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/1id-abstracts.html

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

Copyright and License Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.
1 Introduction

Time synchronization protocols (e.g., PTP [IEEE 1588] and NTP [RFC5905]) have been widely adopted in LAN or Internet to synchronize the clocks of geographically distributed network devices. Depending on different application requirements, the accuracy of the synchronization services can be various from sub-microseconds to milliseconds.

In most conditions, timing information is not confidential and transported over networks in plaintext. However, there are also many cases where the time synchronization packets need to be encrypted during transmission for better security (e.g., to prevent MITM attacks [I-D.ietf-tictoc-security-requirements]) or higher efficiency (e.g., to take advantage of existing IPsec ESP channels). For instance, the [3GPP.33.320] specification mandates that all the packets exchanged between a femtocell (a home access cellular base station) and the security gateway must be encrypted and transported through an IPsec tunnel as the femtocell and the security gateway are connected with an un-secure backhaul network. In a typical deployment, a security gateway may need to process the packets sent from several hundred thousand femtocells. In this case, it is reasonable to for a security gateway to exchange time synchronization packets with its femtocells using already established IPsec tunnels instead of generating additional security tunnels only providing integrity protection although the timing information itself is not confidential.

The encryption and decryption operations on time information will introduce additional errors and negatively influence the accuracy of the time synchronization mechanisms. This issue is discussed in Section 2 with a simple example scenario. Then in Section 3, three candidate solutions are introduced and compared.

2 Motivation Scenario

This section makes use of a typical PTP (IEEE 1588) implementation as an example to explain how a time synchronization mechanism works and why encryption and decryption operations may introduce errors on clock synchronization.

To perform a high accurate synchronization service, the PTP system implements the time-critical operations in the hardware component and relatively slow operations in the software component. The hardware component consists of a high-precision real-time clock and a timestamp unit (TSU) for generating timestamps. The timestamps are struck on the received timing packets, in the middle of the MAC layer and PHY layer, such as, at the Media Independent Interface.
(MII). The software component implements the actual PTP packets processing functions which makes use of the real-time clock and the TSU.

Messages in the PTP protocol include Master sync/follow up message, Slave clock delay request messages, and Master delay response message.

2.1 Time Synchronization Operations of a Two-Step Protocol

Clock synchronization normally requires one Master, and one or multiple Slaves. The Master clock provides synchronization messages to correct slaves’ clocks. Both the master and the slaves generate precise timestamps using their clocks and exchange them to calculate network latency. Typically, the Master sends a sync message to the slaves every two seconds, and a Slave sends a delay request message every minute.

Figure 1 illustrate the initializing process of the protocol, Master first sends a sync message to Slave. A timestamp T1 is struck as the precise time when sync message is transmitted from Master. T1 is sent to Slave in the follow up sync message. When Slave receives the first sync message, the precise time is recorded in a timestamp T2. Then, Slave replies with a delay request message. A timestamp T3 is struck when the message is transmitted from Slave. Finally, a timestamp T4 is struck when the delay request message arrives at Master.

```
Master | Slave
---|---
T1 | T2
   |------------------\ sync message
   |------------------>
   |------------------\ sync follow up message
   |------------------>
   |------------------\ delay request message
   |------------------>
T4 | T3 | time
   |------------------\ delay response message
   |------------------>
```

Figure 1. Timestamping Procedure of PTP Protocol

During the above process, the outbound and inbound transmission delay is T2-T1, and T4-T3 respectively. The one-way delay could be calculated as \([(T2-T1)+(T4-T3)])/2, i.e.
one-way delay = \((T2 - T1) + (T4 - T3)/2\) \hspace{1cm} (1)

Thus, the offset to correct Slave clock could be "outbound delay" - "one-way delay", i.e.

\[ \text{offset} = ((T2 - T1) - (T4 - T3))/2 \] \hspace{1cm} (2)

By notifying Slave that four timestamps T1-T4, it is able to calculate the offset in order to adjust the clock or frequency with Master clock.

2.2 Errors Imposed by Encryption and Decryption in Two-Step Protocols

From the above introduction, it is obvious to see that the execution speed is significantly crucial to the accuracy of synchronization. The time-critical part of protocol, such as timestamping, is typically required to be implemented by hardware.

A direct impact of confidentiality protection of synchronization lies in the fact that the encryption and decryption operations introduce error during Slave time correction, and thus influences the precision of the results. Moreover, it is difficult for TSU to strike the timestamp on the packets including synchronization information in this case, because the encrypted packet is not explicitly distinguishable.

\[
\begin{array}{c|c|c}
\text{Master} & \text{Slave} \\
T1 & \text{---------} & \text{(sync message)} & \text{---------} \\
& \text{------} & \text{----------------} & \text{------} \\
+e1 & \text{-----------} & \text{(sync follow up message)} & \text{-----------} \\
& \text{------} & \text{----------------} & \text{------} \\
& \text{-------------} & \text{----------------} & \text{-------------} \\
+e4 & \text{---------------} & \text{(delay request message)} & \text{---------------} \\
T4 & \text{---------------} & \text{(delay response message)} & \text{---------------} \\
\end{array}
\]

\(\text{Figure 2. Timestamping with Encrypted Packets}\)

Compared with the normal timestamping process of PTP, confidentiality protection in Figure 2 increases (denoted by "+")) processing time from point T1 to T4, including: e1, d2, d’, e3, d4, in which e1 and e3 is the time needed to encrypt follow up message and delay request message.
message, respectively; d2, d’ and d4 is the time needed to decrypt ciphertext of sync message, follow up message and delay request message, respectively.

From the equation (1), (2), it is easy to see that local time correction is subject to the value of time difference (T2−T1) and (T4−T3), but not those absolute value. Denote T1’, T2’, T3’ and T4’ as the new timestamps captured for encrypted packets. Since T1’ is recorded when sync message is sent out, and transmitted by sync follow up message instead of sync message itself, encryption time e1 does not affect the accuracy of T1’. Also, the sync follow up message is only used for transmission of T1’, decryption time d’ does not affect the calculation of time difference (T2’−T1’) nor (T4’−T3’). Besides, encryption corresponding to e3 is done before the T3’ is struck and T3’ is not delivered to Master, thus it does not change the time difference of (T4’−T3’), as well.

Thus, encryption time e1, e3 can be excluded where new time differences to calculate the offset are as follows:

\[ T2’−T1’=(T2+d2)−T1, \]
\[ T4’−T3’=(T4+d4)−T3 \]

Both d2 and d4 are typically in an order of milliseconds and may be varied from one time to another. Therefore, it may seriously influence the performance of a time synchronization mechanism with accuracy in sub-milliseconds.

2.3 Errors Imposed by Encryption in One-Step Protocols

According to the above analysis, the two-step method that utilizes sync message and sync follow up message in sequence is not subject to the delay of encryption. However, it is doubtful whether one-step protocol (i.e. timestamp T1’ is included and delivered in one sync message) is immune from the additional delay by encryption.

Both PTP and NTP synchronization protocol permit one-step process for sync request delivery. In this case the encryption delay occurs since that encryption must be run after timestamp is struck.

A straightforward solution to mitigate this problem is to exclude an estimated time of e1 in the final calculation. However, this solution would be infeasible in the scenarios where high precision is required. Another simple solution is to use the one-step protocol to simulate a two-step one. In this solution, the sync message in the one-step protocol is used to perform the function of sync follow up message in two-step protocols. That is, the time contained in a sync
message is the time when the previous sync packet is generated.

3 Candidate Solutions

This section discusses three candidate methods that could be used to eliminate the errors brought by encryption/decryption time for two-step protocols, which could efficiently synchronize the Slave clock for encrypted synchronization protocol.

3.1 Strike Timestamp on Each Encrypted Packet (STEP)

The most intuitive solution is to strike a timestamp on each encrypted packet (STEP) when it is being received, and timing message will be decoded later for local time correction. The STEP method is able to avoid the delay produced by decryption (such as, d2 and d4 in Sec. 2.1), because it captured timestamps first and then decoded the encrypted messages, not further introducing delay of decrypting operation. In other words, for PTP, there is T2′=T2, T4′=T4, and (T2′−T1′)=(T2−T1) and (T4′−T3′)=(T4−T3), Slave time precision avoids the delay by decryption.

Another advantage is that needs not select the synchronization packets before decrypting them, which is beyond the capability of current synchronization protocol.

In this exhaustive way of timestamp capture, the STEP method is able to synchronize with encrypted protocols. On the other hand, however, it raises the cost of timestamping, from a level of one time per minute (typical frequency of PTP Slave) to a number of mega or kilo times per second, which increases along with the actual network throughput. It may be practical for hardware timestamp, but not appropriate for software timestamp. Even for hardware implementation, it is low efficient and degrades performance.

3.2 Strike Timestamp on Identified Encrypted Packets (STIP)

If timestamp striker is able to know which packet includes time message for synchronization, then the protocol can work correctly as before. To avoid the decryption time delay, such a solution could be achieved by setting an explicit identifier on encrypted packets which contain synchronization message.

From a timestamping point of view, it is the most efficient way to capture the stamp only for those including time message. However, it additionally requires the extension to the current ESP protocol. For example, it is possible to include such an identifier in ESP header or extended ESP (such as WESP, RFC5840) header, or IPv6 header, as
those parts are not encrypted and can carry additional information. A specific solution by using STIP method has been proposed [I-D.xu-tictoc-ipsec-security-for-synchronization].

One concern may be raised that synchronization packets with identifier could help distinguish crucial packets for both valid and malicious users. Malicious users may make use of identifiers to delay or block the synchronization protocol. However, it should be clarified that the STIP method does not breach the security against the underlying delay or block attacks, because such attacks exists no matter whether STIP is employed or not. Attackers can delay or block synchronization packets can, in principle, delay or block all the traffic between Master and Slave. Countermeasures should be considered further but is out of the scope of this draft.

Moreover, in the scenarios where timing information is not confidential but still needs to be transported in an encrypted way to reuse existing security channels, STIP shows its advantage. Compared with the solution of generating new IPsec AH channels to transport timing information, STIP effectively reduce the number of IPsec channels but do not introduce any new security drawbacks since the timing information transported through IPsec AH channels can be easily detected by attackers as well.

3.3 Strike Timestamp on Selective Encrypted Packets (STSP)

As mentioned above, the STEP method is costly and low efficient. And STIP method may give malicious users additional information in encryption transfer mode, even though this does not breach the security as we have analyzed.

One more solution is like a hybrid of STEP and STIP, which strike on some of packets without using identifiers. More precisely, the protocol predicts a time window that the next one will reach after receiving a time synchronization packet and then strike the timestamps on those packets received during that notified period. This solution does not need to capture all the traffic packets for timestamp, thus is more efficient than the STEP method. However, it is only feasible in the conditions where the frequency of transiting time synchronization messages is steady and known by both Master and Slave, thus is limited.

3.4 Comparison

Synchronization methods on encrypted timing packets have been introduced. STEP is straightforward and practical when the performance is acceptable. STIP impacts the precision of synchronization the least, and should be recommended when delay
attack is not a concern. STSP is a trade-off of STEP and STIP, but useful only in limited cases.

4 IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

5 Security Considerations

In encrypted transfer mode (such as an IPsec ESP tunnel), both confidentiality and integrity of the synchronization packets are protected.

Note that in both unencrypted mode and encrypted mode, it is still hard to prevent attackers who intend to block or delay the synchronization protocol. Besides, if relay nodes in routing are corrupted, then MITM (Man-in-the-Middle) attack to synchronization also needs to be dealt with. Thus, a suite of security countermeasures should be worked for preventing the underlying attacks in the future.

6 Acknowledgements

7 References

7.1 Normative References


7.2 Informative References

[I-D.ietf-tictoc-security-requirements]

[I-D.xu-tictoc-ipsec-security-for-synchronization]
Xu, Y., "IPsec security for packet based synchronization", draft-xu-tictoc-ipsec-security-for-synchronization-02 (work in progress), September 2011.

[3GPP.33.320]
3GPP, "Security of Home Node B (HNB) / Home evolved Node B (HeNB)", 3GPP TS 33.320 10.3.0, June 2011.


Authors’ Addresses

Yang Cui
Huawei
Beijing,
China
Email: cuiyang@huawei.com

Manav Bhatia
Alcatel-Lucent
Bangalore
India
Email: manav.bhatia@alcatel-lucent.com

Dacheng Zhang
Huawei
Beijing,
China
Email: zhangdacheng@huawei.com
Transporting PTP messages (1588) over MPLS Networks
draft-ietf-tictoc-1588overmpls-02

Abstract

This document defines the method for transporting PTP messages (PDUs) over an MPLS network. The method allows for the easy identification of these PDUs at the port level to allow for port level processing of these PDUs in both LERs and LSRs.

The basic idea is to transport PTP messages inside dedicated MPLS LSPs. These LSPs only carry PTP messages and possibly Control and Management packets, but they do not carry customer traffic.

Two methods for transporting 1588 over MPLS are defined. The first method is to transport PTP messages directly over the dedicated MPLS LSP via UDP/IP encapsulation, which is suitable for IP/MPLS networks. The second method is to transport PTP messages inside a PW via Ethernet encapsulation, which is more suitable for MPLS-TP networks.

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on April 9, 2012.
Table of Contents

1. Introduction .................................................. 6
2. Terminology ..................................................... 7
3. Problem Statement ................................................. 8
4. 1588 over MPLS Architecture ..................................... 9
5. Dedicated LSPs for PTP messages ................................. 10
6. 1588 over MPLS Encapsulation ................................... 11
   6.1. 1588 over LSP Encapsulation ................................. 11
   6.2. 1588 over PW Encapsulation ................................. 11
7. 1588 Message Transport .......................................... 14
8. Protection and Redundancy ........................................ 16
9. ECMP .............................................................. 17
10. OAM, Control and Management ..................................... 18
11. QoS Considerations ............................................... 19
12. FCS Recalculation ............................................... 20
13. UDP Checksum Correction ........................................ 21
14. Routing extensions for 1588aware LSRs .......................... 22
   14.1. 1588aware Link Capability for OSPF ....................... 22
   14.2. 1588aware Link Capability for IS-IS ....................... 23
15. RSVP-TE Extensions for support of 1588 ......................... 25
16. Behavior of LER/LSR ............................................. 26
   16.1. Behavior of 1588-aware LER ............................... 26
   16.2. Behavior of 1588-aware LSR ............................... 26
   16.3. Behavior of non-1588-aware LSR ......................... 26
17. Other considerations ............................................ 28
18. Security Considerations ......................................... 29
19. Acknowledgements ............................................... 30
20. IANA Considerations ............................................. 31
20.1. IANA Considerations for OSPF ......................... 31
20.2. IANA Considerations for IS-IS .......................... 31
20.3. IANA Considerations for RSVP .......................... 31

21. References .................................................. 32
   21.1. Normative References ................................. 32
   21.2. Informative References ............................... 32

Authors’ Addresses .............................................. 34
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].
1. Introduction

The objective of Precision Time Protocol (PTP) is to synchronize independent clocks running on separate nodes of a distributed system. [IEEE] defines PTP messages for clock and time synchronization. The PTP messages include PTP PDUs over UDP/IP (Annex D and E of [IEEE]) and PTP PDUs over Ethernet (Annex F of [IEEE]). This document defines mapping and transport of the PTP messages defined in [IEEE] over MPLS networks.

PTP defines several clock types: ordinary clocks, boundary clocks, end-to-end transparent clocks, and peer-to-peer transparent clocks. One key attribute of all of these clocks is the recommendation for PTP messages processing to occur as close as possible to the actual transmission and reception at the physical port interface. This targets optimal time and/or frequency recovery by avoiding variable delay introduced by queues internal to the clocks. To facilitate the fast and efficient recognition of PTP messages at the port level when the PTP messages are carried over MPLS LSPs, this document defines the specific encapsulations that should be used. In addition, it can be expected that there will exist LSR/LERs where only a subset of the physical ports will have the port based PTP message processing capabilities. In order to ensure that the PTP carrying LSPs always enter and exit ports with this capability, routing extensions are defined to advertise this capability on a port basis and to allow for the establishment of LSPs that only transit such ports. While this path establishment restriction may be applied only at the LER ingress/egress ports, it becomes more important when using Transparent Clock capable LSRs in the path.

The port based PTP message processing involves PTP event message recognition. Once the PTP event messages are recognized they can be modified based on the reception or transmission timestamp. An alternative technique to actual packet modification could include the enforcement of a fixed delay time across the LSR to remove variability in the transit delay. This latter would be applicable in a LSR which does not contain a PTP transparent Clock function.

This document provides two methods for transporting PTP messages over MPLS. One is principally focused on an IP/MPLS environment and the second is focused on the MPLS-TP environment.

While the techniques included herein allow for the establishment of paths optimized to include PTP Timestamping capable links, the performance of the Slave clocks is outside the scope of this document.
2. Terminology

1588: The timing and synchronization as defined by IEEE 1588

PTP: The timing and synchronization protocol used by 1588

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

Master Port: A port on an ordinary or boundary clock that is in Master state. This is the source of timing toward slave ports.

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

Slave Port: A port on a boundary clock or ordinary clock that is receiving timing from a master clock.

Ordinary Clock: A device with a single PTP port.

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

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.

PTP LSP: An LSP dedicated to carry PTP messages

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

CW: Pseudowire Control Word

LAG: Link Aggregation

ECMP: Equal Cost Multipath

CF: Correction Field, a field inside certain PTP messages (message type 0-3) that holds the accumulative transit time inside intermediate switches
3. Problem Statement

When PTP messages are transported over MPLS networks, there is a need for PTP message processing at the physical port level. This requirement exists to minimize uncertainty in the transit delays. If PTP message processing occurs interior to the MPLS routers, then the variable delay introduced by queuing between the physical port and the PTP processing will add noise to the timing distribution. Port based processing applies at both the originating and terminating LERs and also at the intermediate LSRs if they support transparent clock functionality.

PTP messages over Ethernet or IP can always be tunneled over MPLS. However there is a requirement to limit the possible encapsulation options to simplify the PTP message processing required at the port level. This applies to all 1588 clock types implemented in MPLS routers. But this is particularly important in LSRs that provide transparent clock functionality.

When 1588-awareness is needed, PTP messages should not be transported over LSPs or PWs that are carrying customer traffic because LSRs perform Label switching based on the top label in the stack. To detect PTP messages inside such LSPs requires special hardware to do deep packet inspection at line rate. Even if such hardware exists, the payload can’t be deterministically identified by LSRs because the payload type is a context of the PW label and the PW label and its context are only known to the Edge routers (PEs); LSRs don’t know what is a PW’s payload (Ethernet, ATM, FR, CES, etc). Even if one restricts an LSP to only carry Ethernet PWs, the LSRs don’t have the knowledge of whether PW Control Word (CW) is present or not and therefore can’t deterministically identify the payload.

Therefore a generic method is defined in this document that does not require deep packet inspection at line rate, and can deterministically identify PTP messages. The defined method is applicable to both MPLS and MPLS-TP networks.
4. 1588 over MPLS Architecture

1588 communication flows map onto MPLS nodes as follows: 1588 messages are exchange between PTP ports on Ordinary and boundary clocks. Transparent clocks do not terminate the PTP messages but they do modify the contents of the PTP messages as they transit across the Transparent clock. SO Ordinary and boundary clocks would exist within LERs as they are the termination points for the PTP messages carried in MPLS. Transparent clocks would exist within LSRs as they do not terminate the PTP message exchange.

Perhaps a picture would be good here.
5. Dedicated LSPs for PTP messages

Many methods were considered for identifying the 1588 messages when they are encapsulated in MPLS such as by using GAL/ACH or a new reserved label. These methods were not attractive since they either required deep packet inspection and snooping at line rate or they required use of a scarce new reserved label. Also one of the goals was to reuse existing OAM and protection mechanisms.

The method defined in this document can be used by LER/LSRs to identify PTP messages in MPLS tunnels by using dedicated LSPs to carry PTP messages.

Compliant implementations MUST use dedicated LSPs to carry PTP messages over MPLS. These LSPs are herein referred to as "PTP LSPs" and the labels associated with these LSPs as "PTP labels". These LSPs could be P2P or P2MP LSPs. The PTP LSP between Master Clocks and Slave Clocks MAY be P2MP or P2P LSP while the PTP LSP between each Slave Clock and Master Clock SHOULD be P2P LSP. The PTP LSP between a Master Clock and a Slave Clock and the PTP LSP between the same Slave Clock and Master Clock MUST be co-routed. Alternatively, a single bidirectional co-routed LSP can be used. The PTP LSP MAY be MPLS LSP or MPLS-TP LSP. This co-routing is required to limit differences in the delays in the Master clock to Slave clock direction compared to the Slave clock to Master clock direction.

The PTP LSPs could be configured or signaled via RSVP-TE/GMPLS. New RSVP-TE/GMPLS TLVs and objects are defined in this document to indicate that these LSPs are PTP LSPs.

The PTP LSPs MAY carry essential MPLS/MPLS-TP control plane traffic such as BFD and LSP Ping but the LSP user plane traffic MUST be PTP only.
6. 1588 over MPLS Encapsulation

This document defines two methods for carrying PTP messages over MPLS. The first method is carrying IP encapsulated PTP messages over PTP LSPs and the second method is to carry PTP messages over dedicated Ethernet PWs (called PTP PWs) inside PTP LSPs.

6.1. 1588 over LSP Encapsulation

The simplest method of transporting PTP messages over MPLS is to encapsulate PTP PDUs in UDP/IP and then encapsulate them in PTP LSP. The 1588 over LSP format is shown in Figure 1.

```
+----------------------+
|   PTP Tunnel Label   |
+----------------------+
|        IPv4/6        |
+----------------------+
|         UDP          |
+----------------------+
|        PTP PDU       |
+----------------------+
```

Figure 1 - 1588 over LSP Encapsulation

This encapsulation is very simple and is useful when the networks between 1588 Master Clock and Slave Clock are IP/MPLS networks.

In order for an LSR to process PTP messages, the PTP Label must be the top label of the label stack.

The UDP/IP encapsulation of PTP MUST follow Annex D and E of [IEEE].

6.2. 1588 over PW Encapsulation

Another method of transporting 1588 over MPLS networks is by encapsulating PTP PDUs in Ethernet and then transporting them over Ethernet PW (PTP PW) as defined in [RFC4448], which in turn is transported over PTP LSPs. Alternatively PTP PDUs MAY be encapsulated in UDP/IP/Ethernet and then transported over Ethernet PW.

Both Raw and Tagged modes for Ethernet PW are permitted. The 1588 over PW format is shown in Figure 2.
The Control Word (CW) as specified in [RFC4448] SHOULD be used to ensure a more robust detection of PTP messages inside the MPLS packet. If CW is used, the use of Sequence number is optional.

The use of VLAN and UDP/IP are optional. Note that 1 or 2 VLANs MAY exist in the PW payload.

In order for an LSR to process PTP messages, the top label of the label stack (the Tunnel Label) MUST be from PTP label range. However in some applications the PW label may be the top label in the stack, such as cases where there is only one-hop between PEs or in case of PHP. In such cases, the PW label SHOULD be chosen from the PTP Label range.

In order to ensure congruency between the two directions of PTP message flow, ECMP should not be used for the PTP LSPs. Therefore, no Entropy label [I-D.ietf-pwe3-fat-pw] is necessary and it SHOULD NOT be present in the stack.

The Ethernet encapsulation of PTP MUST follow Annex F of [IEEE] and the UDP/IP encapsulation of PTP MUST follow Annex D and E of [IEEE].

For 1588 over MPLS encapsulations that are PW based, there are some cases which the PTP LSP label may not be present:

Figure 2 - 1588 over PW Encapsulation
When PHP is applied to the PTP LSP, and the packet is received without PTP LSP label at PW termination point.

When the PW is established between two routers directly connected to each other and no PTP LSP is needed.

In such cases it is required for a router to identify these packets as PTP packets. This would require the PW label to also be a label that is distributed specifically for carrying PTP traffic (aka PTP PW label). Therefore there is a need to add extension to LDP/BGP PW label distribution protocol to indicate that a PW label is a PTP PW labels.
7. 1588 Message Transport

1588 protocol comprises of the following message types:

- Announce
- SYNC
- FOLLOW UP
- DELAY_REQ (Delay Request)
- DELAY_RESP (Delay Response)
- PDELAY_REQ (Peer Delay Request)
- PDELAY_RESP (Peer Delay Response)
- PDELAY_RESP_FOLLOW_UP (Peer Delay Response Follow up)
- Management
- Signaling

A subset of PTP message types that require timestamp processing are
called Event messages:

- SYNC
- DELAY_REQ (Delay Request)
- PDELAY_REQ (Peer Delay Request)
- PDELAY_RESP (Peer Delay Response)

SYNC and DELAY_REQ are exchanged between Master Clock and Slave Clock
and MUST be transported over PTP LSPs. PDELAY_REQ and PDELAY_RESP
are exchanged between adjacent PTP clocks (i.e. Master, Slave,
Boundary, or Transparent) and MAY be transported over single hop PTP
LSPs. If Two Step PTP clocks are present, then the FOLLOW_UP,
DELAY_RESP, and PDELAY_RESP_FOLLOW_UP messages must also be
transported over the PTP LSPs.

For a given instance of 1588 protocol, SYNC and DELAY_REQ MUST be
transported over two PTP LSPs that are in opposite directions. These
PTP LSPs, which are in opposite directions MUST be congruent and co-
routed. Alternatively, a single bidirectional co-routed LSP can be
used.
Except as indicated above for the two-step PTP clocks, Non-Event PTP message types don’t need to 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 1588 Slaves, usually as a general practice, Redundant Masters are tracked by each Slave. It is the responsibility of the network operator to ensure that physically disjoint PTP tunnels that don’t share any link are used between the redundant Masters and a Slave.

When redundant Masters are tracked by a Slave, any prolonged PTP LSP or PTP PW outage will trigger the Slave Clock to switch to the Redundant Master Clock. However LSP/PW protection such as Linear Protection Switching (1:1,1+1), Ring protection switching or MPLS Fast Reroute (FRR) SHOULD still be used to provide resiliency to individual network segment failures.

Note that any protection or reroute mechanism that adds additional label to the label stack, such as Facility Backup Fast Reroute, MUST ensure that the pushed label is a PTP Label to ensure recognition of the MPLS frame as containing PTP messages as it transits the backup path.
9. ECMP

To ensure the optimal operation of 1588 Slave clocks and avoid errors introduced by forward and reverse path delay asymmetry, the physical path for PTP messages from Master Clock to Slave Clock and vice versa must be the same for all PTP messages listed in section 7 and must not change even in the presence of ECMP in the MPLS network.

To ensure the forward and reverse paths are the same PTP LSPs and PWs MUST NOT be subject to ECMP.
10. OAM, Control and Management

In order to manage PTP LSPs and PTP PWs, they MAY carry OAM, Control and Management messages. These control and management messages can be differentiated from PTP messages via already defined IETF methods.

In particular BFD [RFC5880], [RFC5884] and LSP-Ping [RFC4389] MAY run over PTP LSPs via UDP/IP encapsulation or via GAL/G-ACH. These Management protocols are easily identified by the UDP Destination Port number or by GAL/ACH respectively.

Also BFD, LSP-Ping and other Management messages MAY run over PTP PW via one of the defined VCCVs (Type 1, 2 or 3) [RFC5085]. In this case G-ACH, Router Alert Label (RAL), or PW label (TTL=1) are used to identify such management messages.
11. QoS Considerations

In network deployments where not every LSR/LER is PTP-aware, then it is important to reduce the impact of the non-PTP-aware LSR/LERs on the timing recovery in the slave clock. The PTP messages are time critical and must be treated with the highest priority. Therefore, 1588 over MPLS messages must be treated with the highest priority in the routers. This can be achieved by proper setup of PTP tunnels. It is recommended that the PTP LSPs are setup and marked properly to indicate EF-PHB for the CoS and Green for drop eligibility.

In network deployments where every LSR/LER supports PTP LSPs, then it MAY NOT be required to apply the same level of prioritization as specified above.
12. FCS Recalculation

Ethernet FCS of the outer encapsulation MUST be recalculated at every LSR that performs the Transparent Clock processing and FCS retention for the payload Ethernet described in [RFC4720] MUST NOT be used.
13. UDP Checksum Correction

For UDP/IP encapsulation mode of 1588 over MPLS, the UDP checksum is optional when used for IPv4 encapsulation and mandatory in case of IPv6. When IPv4/v6 UDP checksum is used each 1588-aware LSR must either incrementally update the UDP checksum after the CF update or should verify the UDP checksum on reception from upstream and recalculate the checksum completely on transmission after CF update to downstream node.
14. Routing extensions for 1588-aware LSRs

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.

Indeed, it is useful to advertise data plane TE router link capabilities, such as the capability for a router to be 1588-aware. This capability MUST then be taken into account during path computation to prefer or even require links that advertise themselves as 1588-aware. In this way the path can ensure the entry and exit points into the LERs and, if desired, the links into the LSRs are able to perform port based timestamping thus minimizing their impact on the performance of the slave clock.

For this purpose, the following sections specify extensions to OSPF and IS-IS in order to advertise 1588 aware capabilities of a link.

14.1. 1588-aware Link Capability for OSPF

OSPF uses the Link TLV (Type 2) that is itself carried within either the Traffic Engineering LSA specified in [RFC3630] or the OSPFv3 Intra-Area-TE LSA (function code 10) defined in [RFC5329] to advertise the TE related information for the locally attached router links. For an LSA Type 10, one LSA can contain one Link TLV information for a single link. This extension defines a new 1588-aware capability sub-TLV that can be carried as part of the Link TLV.

The 1588-aware capability sub-TLV is OPTIONAL and MUST NOT appear more than once within the Link TLV. If a second instance of the 1588-aware capability sub-TLV is present, the receiving system MUST only process the first instance of the sub-TLV. It is defined as follows:

```
0                   1                   2                   3
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Type              |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Flags     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: 1588-aware Capability TLV

Where:

Type, 16 bits: 1588-aware Capability TLV where the value is TBD
Length, 16 bits: Gives the length of the flags field in octets, and is currently set to 1.

Flags, 8 bits: The bits are defined least-significant-bit (LSB) first, so bit 7 is the least significant bit of the flags octet.

```
0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|   Reserved   |C|
+-+-+-+-+-+-+-+-+
```

Figure 4: Flags Format

Correction (C) field Update field, 1 bit: Setting the C bit to 1 indicates that the link is capable of recognizing the PTP event packets and can compensate for residence time by updating the PTP packet Correction Field. When this is set to 0, it means that this link cannot perform the residence time correction but is capable of performing MPLS frame forwarding of the frames with PTP labels using a method that support the end to end delivery of accurate timing. The exact method is not defined herein.

Reserved, 7 bits: Reserved for future use. The reserved bits must be ignored by the receiver.

The 1588-aware Capability sub-TLV is applicable to both OSPFv2 and OSPFv3.

14.2. 1588aware Link Capability for IS-IS

The IS-IS Traffic Engineering [RFC3784] defines the intra-area traffic engineering enhancements and uses the Extended IS Reachability TLV (Type 22) [RFC5305] to carry the per link TE-related information. This extension defines a new 1588-aware capability sub-TLV that can be carried as part of the Extended IS Reachability TLV.

The 1588-aware capability sub-TLV is OPTIONAL and MUST NOT appear more than once within the Extended IS Reachability TLV or the Multi-Topology (MT) Intermediate Systems TLV (type 222) specified in [RFC5120]. If a second instance of the 1588-aware capability sub-TLV is present, the receiving system MUST only process the first instance of the sub-TLV.

The format of the IS-IS 1588-aware sub-TLV is identical to the TLV format used by the Traffic Engineering Extensions to IS-IS [RFC3784]. That is, the TLV is comprised of 1 octet for the type, 1 octet
specifying the TLV length, and a value field. The Length field defines the length of the value portion in octets.

<table>
<thead>
<tr>
<th>Type</th>
<th>Length</th>
<th>Flags</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

Figure 5: 1588-aware Capability sub-TLV

Where:

Type, 8 bits: 1588-aware Capability sub-TLV where the value is TBD

Length, 8 bits: Gives the length of the flags field in octets, and is currently set to 1

Flags, 8 bits: The bits are defined least-significant-bit (LSB) first, so bit 7 is the least significant bit of the flags octet.

<table>
<thead>
<tr>
<th>Reserved</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 2 3 4 5 6 7</td>
<td></td>
</tr>
</tbody>
</table>

Figure 6: Flags Format

Correction (C) field Update field, 1 bit: Setting the C bit to 1 indicates that the link is capable of recognizing the PTP event packets and can compensate for residence time by updating the PTP packet Correction Field. When this is set to 0, it means that this link cannot perform the residence time correction but is capable of performing MPLS frame forwarding of the frames with PTP labels using a method that support the end to end delivery of accurate timing. The exact method is not defined herein.

Reserved, 7 bits: Reserved for future use. The reserved bits must be ignored by the receiver.
15. RSVP-TE Extensions for support of 1588

RSVP-TE signaling MAY be used to setup the PTP LSPs. A new RSVP object is defined to signal that this is a PTP LSP. The OFFSET to the start of the PTP message header MAY also be signaled. Implementations can trivially locate the correctionField (CF) location given this information. The OFFSET points to the start of the PTP header as a node may want to check the PTP messageType before it touches the correctionField (CF). The OFFSET is counted from TBD.

The LSRs that receive and process the RSVP-TE/GMPLS messages MAY use the OFFSET to locate the start of the PTP message header.

Note that the new object/TLV Must be ignored by LSRs that are not compliant to this specification.

The new RSVP 1588_PTP_LSP object should be included in signaling PTP LSPs and is defined as follows:

<table>
<thead>
<tr>
<th>Length (bytes)</th>
<th>Class-Num</th>
<th>C-Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Offset to locate the start of the PTP message header</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 7: RSVP 1588_PTP_LSP object

The ingress LSR MUST include this object in the RSVP PATH Message. It is just a normal RSVP path that is exclusively set up for PTP messages.
16. Behavior of LER/LSR

16.1. Behavior of 1588-aware LER

A 1588-aware LER advertises its 1588-awareness via the OSPF procedure explained in earlier section of this specification. The 1588-aware LER then signals PTP LSPs by including the 1588_PTP_LSP object in the RSVP-TE signaling.

When a 1588 message is received from a non-MPLS interface, the LER MUST redirect them to a previously established PTP LSP. When a 1588 over MPLS message is received from an MPLS interface, the processing is similar to 1588-aware LSR processing.

16.2. Behavior of 1588-aware LSR

1588-aware LSRs are LSRs that understand the 1588_PTP_LSP RSVP object and can perform 1588 processing (e.g. Transparent Clock processing).

A 1588-aware LSR advertises its 1588-awareness via the OSPF procedure explained in earlier section of this specification.

When a 1588-aware LSR distributes a label for PTP LSP, it maintains this information. When the 1588-aware LSR receives an MPLS packet, it performs a label lookup and if the label lookup indicates it is a PTP label then further parsing must be done to positively identify that the payload is 1588 and not OAM, BFD or control and management. Ruling out non-1588 messages can easily be done when parsing indicates the presence of GAL, ACH or VCCV (Type 1, 2, 3) or when the UDP port number does not match one of the 1588 UDP port numbers.

After a 1588 message is positively identified in a PTP LSP, the PTP message type indicates whether any timestamp processing is required. After 1588 processing the packet is forwarded as a normal MPLS packet to downstream node.

16.3. Behavior of non-1588-aware LSR

It is most beneficial that all LSRs in the path of a PTP LSP be 1588-aware LSRs. This would ensure the highest quality time and clock synchronization by 1588 Slave Clocks. However, this specification does not mandate that all LSRs in path of a PTP LSP be 1588-aware.

Non-1588-aware LSRs are LSRs that either don’t have the capability to process 1588 packets (e.g. perform Transparent Clock processing) or don’t understand the 1588_PTP_LSP RSVP object.

Non-1588-aware LSRs ignore the RSVP 1588_PTP_LSP object and just
switch the MPLS packets carrying 1588 messages as data packets and don't perform any timestamp related processing. However as explained in QoS section the 1588 over MPLS packets MUST be still be treated with the highest priority.
17. Other considerations

The use of Explicit Null (Label= 0 or 2) is acceptable as long as either the Explicit Null label is the bottom of stack label (applicable only to UDP/IP encapsulation) or the label below the Explicit Null label is a PTP label.

The use of Penultimate Hop Pop (PHP) is acceptable as long as either the PHP label is the bottom of stack label (applicable only to UDP/IP encapsulation) or the label below the PHP label is a PTP label.
18. Security Considerations

MPLS PW security considerations in general are discussed in [RFC3985] and [RFC4447], and those considerations also apply to this document.

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

The authors would like to thank Luca Martini, Ron Cohen, Yaakov Stein, Tal Mizrahi and other members of the TICTOC WG for reviewing and providing feedback on this draft.
20. IANA Considerations

20.1. IANA Considerations for OSPF

IANA has defined a sub-registry for the sub-TLVs carried in an OSPF TE Link TLV (type 2). IANA is requested to assign a new sub-TLV codepoint for the 1588aware capability sub-TLV carried within the Router Link TLV.

<table>
<thead>
<tr>
<th>Value</th>
<th>Sub-TLV</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBD</td>
<td>1588aware node sub-TLV</td>
<td>(this document)</td>
</tr>
</tbody>
</table>

20.2. IANA Considerations for IS-IS

IANA has defined a sub-registry for the sub-TLVs carried in the IS-IS Extended IS Reacability TLV. IANA is requested to assign a new sub-TLV code-point for the 1588aware capability sub-TLV carried within the Extended IS Reacability TLV.

<table>
<thead>
<tr>
<th>Value</th>
<th>Sub-TLV</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBD</td>
<td>1588aware node sub-TLV</td>
<td>(this document)</td>
</tr>
</tbody>
</table>

20.3. IANA Considerations for RSVP

IANA is requested to assign a new Class Number for 1588 PTP LSP object that is used to signal PTP LSPs.

1588 PTP LSP Object

Class-Num of type 1b1bbbbbb

Suggested value TBD

Defined CType: 1 (1588 PTP LSP)
21. References

21.1. Normative References


21.2. Informative References


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
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 August 6, 2012.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents.
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]. It is envisioned this MIB will complement other managed objects to be defined to monitor, measure the performance of the PTP devices and telecom clocks. 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], and the Telecom Profile described in [G.8265.1], 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, measure the performance of the PTP devices and telecom clocks.

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. This section tracks changes made to the visions 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

2. The SNMP Management Framework

The SNMP Management Framework presently consists of five major components:

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

- 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].

- 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].

- 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,
  Integer32,
  Gauge32,
  Unsigned32,
  Counter32,
  Counter64
  FROM SNMPv2-SMI
  OBJECT-GROUP,
  MODULE-COMPLIANCE
  FROM SNMPv2-CONF
  TEXTUAL-CONVENTION,
  TruthValue,
  DisplayString
  FROM SNMPv2-TC
  InterfaceIndexOrZero
  FROM IF-MIB
  InetAddressType,
  InetAddress
  FROM INET-ADDRESS-MIB;

ptpbaseMIB MODULE-IDENTITY
  LAST-UPDATED    "201201230000Z"
  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,
     Symmetricom Inc.,
     Email: tfrost@symmetricom.com

     Greg Dowd,
     Symmetricom Inc.,
     Email: gdowd@symmetricom.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], or the Telecom Profile described in [G.8265.1], when running over the IP network layer.

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

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:

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARB</td>
<td>Arbitrary Timescale</td>
</tr>
<tr>
<td>E2E</td>
<td>End-to-End</td>
</tr>
<tr>
<td>EUI</td>
<td>Extended Unique Identifier.</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>IANA</td>
<td>Internet Assigned Numbers Authority</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
</tbody>
</table>

Shankarkumar et al. Expires August 6, 2012
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

[G.8265.1] Precision Time Protocol Telecom Profile for
    Frequency Synchronization, ITU-T Recommendation G.8265.1,
    October 2010.

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.

<table>
<thead>
<tr>
<th>Data type mapping</th>
<th>Textual Convention</th>
<th>SYNTAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.3.2 TimeInterval</td>
<td>ClockTimeInterval</td>
<td>OCTET STRING(SIZE(1..255))</td>
</tr>
</tbody>
</table>
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  |       Boundary
         /       |
Ordinary  |       Boundary
         /       |
Ordinary  |       Ordinary
         /       |
Ordinary  |       Ordinary

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]:
The MIB refers to the sections of [IEEE 1588-2008].

-- revision log

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

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.

<table>
<thead>
<tr>
<th>Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Default domain</td>
</tr>
<tr>
<td>1</td>
<td>Alternate domain 1</td>
</tr>
<tr>
<td>2</td>
<td>Alternate domain 2</td>
</tr>
<tr>
<td>3</td>
<td>Alternate domain 3</td>
</tr>
<tr>
<td>4 - 127</td>
<td>User-defined domains</td>
</tr>
</tbody>
</table>

ClockIdentity ::= TEXTUAL-CONVENTION

STATUS          current
DESCRIPTION
"The clock Identity is an 8-octet array and will be presented in
the form of a character array. 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))

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]."

<table>
<thead>
<tr>
<th>Delay mechanism</th>
<th>Value(hex)</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>E2E</td>
<td>01</td>
<td>The port is configured to use the delay request-response mechanism.</td>
</tr>
<tr>
<td>P2P</td>
<td>02</td>
<td>The port is configured to use the peer delay mechanism.</td>
</tr>
<tr>
<td>DISABLED</td>
<td>FE</td>
<td>The port does not implement the delay mechanism.</td>
</tr>
</tbody>
</table>

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)
}

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)

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)
}

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(1),       -- 0
    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]."

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Reserved to enable compatibility with future versions.</td>
</tr>
<tr>
<td>1-5</td>
<td>Reserved</td>
</tr>
<tr>
<td>6</td>
<td>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.</td>
</tr>
<tr>
<td>7</td>
<td>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.</td>
</tr>
<tr>
<td>8</td>
<td>Reserved</td>
</tr>
<tr>
<td>9-10</td>
<td>Reserved to enable compatibility with future versions.</td>
</tr>
<tr>
<td>11-12</td>
<td>Reserved</td>
</tr>
<tr>
<td>13</td>
<td>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.</td>
</tr>
<tr>
<td>14</td>
<td>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.</td>
</tr>
<tr>
<td>15-51</td>
<td>Reserved</td>
</tr>
</tbody>
</table>
| 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.

<table>
<thead>
<tr>
<th>Clock Role</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Master clock</td>
<td>1</td>
<td>A clock that is the source of time to which all other clocks on that path synchronize.</td>
</tr>
</tbody>
</table>
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 is lost or the timestamps in the ptp packets are incorrect. But since the slave was locked to the master, it can run with the same accuracy for sometime. The slave can continue to operate in this state for some time. If communication with the master is not restored for a while, the device is moved 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       Lock to the master with respect to frequency and phase.

SYNTAX       INTEGER  { 
  freerun(1),
  holdover(2),
  acquiring(3),
  frequencyLocked(4),
  phaseAligned(5)
}
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
    internalOsillator(160) -- 0xA0
}

ClockTimeInterval ::= TEXTUAL-CONVENTION
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.

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 section 7.7.2.1 Timer interval specification of [IEEE 1588-2008]"

SYNTAX OCTET STRING (SIZE (1..255))

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 }

-- Conformance Information Definition

ptpbaseMIBCompliances  OBJECT IDENTIFIER
  ::= { ptpbaseMIBConformance 1 }

ptpbaseMIBGroups  OBJECT IDENTIFIER
  ::= { ptpbaseMIBConformance 2 }

ptpbaseMIBCompliances1 MODULE-COMPLIANCE
  STATUS current
  DESCRIPTION
  "Compliance statement for agents that provide read-only support for PTPBASE-MIB. 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 }

ptpbaseMIBCompliances2 MODULE-COMPLIANCE
  STATUS current
  DESCRIPTION
  "Compliance statement for agents that provide read-only support for PTPBASE-MIB. 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 }

ptpbaseMIBCompliances3 MODULE-COMPLIANCE
  STATUS current
  DESCRIPTION
"Compliance statement for agents that provide read-only support for PTPBASE-MIB. 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 }

ptpbaseMIBCompliances4 MODULE-COMPLIANCE
STATUS current
DESCRIPTION "Compliance statement for agents that provide read-only support for PTPBASE-MIB. 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

}{ ptpbaseMIBGroups 2 }

ptpbaseMIBClockParentDSGroup OBJECT-GROUP
OBJECTS
{ ptpbaseClockParentDSParentPortIdentity,
  ptpbaseClockParentDSParentStats,
  ptpbaseClockParentDSOffset,
  ptpbaseClockParentDSClockPhChRate,
  ptpbaseClockParentDSGMClockIdentity,
  ptpbaseClockParentDSGMClockPriority1,
  ptpbaseClockParentDSGMClockPriority2,
  ptpbaseClockParentDSGMClockQualityClass,
  ptpbaseClockParentDSGMClockQualityAccuracy,
  ptpbaseClockParentDSGMClockQualityOffset

}{ ptpbaseMIBGroups 3 }

ptpbaseMIBClockDefaultDSGroup OBJECT-GROUP
OBJECTS
{ ptpbaseClockDefaultDSTwoStepFlag,
  ptpbaseClockDefaultDSClockIdentity,
  ptpbaseClockDefaultDSPriority1,
  ptpbaseClockDefaultDSPriority2,
  ptpbaseClockDefaultDSSlaveOnly,
  ptpbaseClockDefaultDSQualityClass,
  ptpbaseClockDefaultDSQualityAccuracy,
  ptpbaseClockDefaultDSQualityOffset

}{ ptpbaseMIBGroups 4 }

ptpbaseMIBClockRunningGroup OBJECT-GROUP
OBJECTS
{ ptpbaseClockRunningState,
  ptpbaseClockRunningPacketsSent,
  ptpbaseClockRunningPacketsReceived

"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 }

ptpbaseMIBClockTransparentDSGroup 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,
         ptpbaseClockPortSyncOneStep,
         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,
    ptpbaseClockPortDSSPortIdentity,
    ptpbaseClockPortDSAnnouncementInterval,
    ptpbaseClockPortDSAnnounceRctTimeout,
    ptpbaseClockPortDSSyncInterval,
    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,
    ptpbaseClockPortRunningIPversion,
    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
Internet-Draft  draft-ietf-tictoc-ptp-mib-01.txt  February 6, 2012

}  
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 } 

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 } 

Shankarkumar et al. Expires August 6, 2012
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."
::= { ptpbaseSystemEntry 3 }

ptpbaseSystemDomainTable OBJECT-TYPE
SYNTAX        SEQUENCE OF PtpbaseSystemDomainEntry
MAX-ACCESS    not-accessible
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,
    ptpbaseClockCurrentDSSStepsRemoved    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

Shankarkumar et al. Expires August 6, 2012 [Page 31]
SYNTAX        ClockType
MAX-ACCESS    not-accessible
STATUS        current
DESCRIPTION   "This object specifies the clock type as defined in the
                Textual convention description."
::= { ptphbaseClockCurrentDSEntry 2 }

ptphbaseClockCurrentDSInstanceIndex 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."
::= { ptphbaseClockCurrentDSEntry 3 }

ptphbaseClockCurrentDSStepsRemoved 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"
::= { ptphbaseClockCurrentDSEntry 4 }

ptphbaseClockCurrentDSOffsetFromMaster 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]"
::= { ptphbaseClockCurrentDSEntry 5 }

ptphbaseClockCurrentDSMeanPathDelay OBJECT-TYPE
SYNTAX        ClockTimeInterval
MAX-ACCESS    read-only
STATUS        current
DESCRIPTION   Shankarkumar et al. Expires August 6, 2012 [Page 32]
"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 Shankarkumar et al. Expires August 6, 2012 [Page 35]"
::= { 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]"
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 12 }

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

Shankarkumar et al. Expires August 6, 2012 [Page 38]
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

Shankarkumar et al.   Expires August 6, 2012
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 }

Shankarkumar et al. Expires August 6, 2012 [Page 40]
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 is lost or the
  timestamps in the ptp packets are incorrect. But since the
  slave was locked to the master, it can run with the same
  accuracy for sometime. The slave can continue to operate in this state for
  some time. If communication with the master is not restored for
  a while, the device is moved 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
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 Timeproperties 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

Shankarkumar et al. Expires August 6, 2012 [Page 44]
"This object specifies the Frequency Traceable value in the clock Current Dataset."
REFERENCE "Section 8.2.4.7 of [IEEE 1588-2008]"
::= { ptpbaseClockTimePropertiesDSEntry 9 }

ptpbaseClockTimePropertiesDSPTimescale 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 Integer32
}

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
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              Integer32
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 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 1 }

ptpbaseClockPortClockTypeIndex OBJECT-TYPE
SYNTAX ClockType
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "This object specifies the domain number used to create logical group of PTP devices."
::= { ptpbaseClockPortEntry 2 }

ptpbaseClockPortClockInstancenIndex 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 (1..65535)
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
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 }

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

ptpbaseClockPortCurrentPeerAddressType OBJECT-TYPE
SYNTAX InetAddressType
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"This object specifies the current peer’s network address used for PTP communication. Based on the scenario and the setup involved, the values might look like these -
Scenario Value
-------------- ------------
Single Master master port
Multiple Masters selected master port
Single Slave slave port
Multiple Slaves <empty>

(In relevant setups, information on available slaves and available masters will be available through ptpClockPortAssociateTable)"
::= { ptpbaseClockPortEntry 8 }

ptpbaseClockPortCurrentPeerAddress OBJECT-TYPE
SYNTAX InetAddress
MAX-ACCESS read-only
STATUS current
DESCRIPTION

Shankarkumar et al. Expires August 6, 2012 [Page 49]
This object specifies the current peer’s network address used for PTP communication. Based on the scenario and the setup involved, the values might look like these:

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Master</td>
<td>master port</td>
</tr>
<tr>
<td>Multiple Masters</td>
<td>selected master port</td>
</tr>
<tr>
<td>Single Slave</td>
<td>slave port</td>
</tr>
<tr>
<td>Multiple Slaves</td>
<td>&lt;empty&gt;</td>
</tr>
</tbody>
</table>

(In relevant setups, information on available slaves and available masters will be available through ptpClockPortAssociateTable)

::= { 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,
  ptpbaseClockPortDSAnnouncementInterval Integer32,
  ptpbaseClockPortDSAnnounceRctTimeout   Integer32,
  ptpbaseClockPortDSSyncInterval         Integer32,
  ptpbaseClockPortDSMinDelayReqInterval  Integer32,
  ptpbaseClockPortDSPeerDelayReqInterval Integer32,
  ptpbaseClockPortDSDelayMech            ClockMechanismType,
  ptpbaseClockPortDSPeerMeanPathDelay    ClockTimeInterval,
  ptpbaseClockPortDSGrantDuration        Unsigned32,
  ptpbaseClockPortDSPTPVersion           Integer32
}

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 }

Shankarkumar et al.     Expires August 6, 2012                [Page 51]
ptpbaseClockPortDSPortNumberIndex OBJECT-TYPE
SYNTAX ClockPortNumber (1..65535)
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 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 }

ptpbaseClockPortDSAnnouncementInterval OBJECT-TYPE
SYNTAX Integer32
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 }

ptpbaseClockPortDSSyncInterval OBJECT-TYPE
SYNTAX Integer32
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 }

ptpbaseClockPortDelayMech 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 }

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

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

ptpbaseClockPortSPTPVersion OBJECT-TYPE
SYNTAX          Integer32
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 running 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,
    ptpbaseClockPortRunningIPversion          Integer32,
    ptpbaseClockPortRunningEncapsulationType  Integer32,
    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 (1..65535)
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

Shankarkumar et al. Expires August 6, 2012 [Page 55]
"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 }

ptpbaseClockPortRunningIPversion OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "This object specifies the IP version being used for PTP communication (the mapping used)."
 ::= { ptpbaseClockPortRunningEntry 9 }

ptpbaseClockPortRunningEncapsulationType OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "This object specifies the type of encapsulation if the interface is adding extra layers (e.g. 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 specifies 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 Integer32,
   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 (1..65535)
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 Integer32
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 }

Shankarkumar et al. Expires August 6, 2012 [Page 59]
Internet-Draft      draft-ietf-tictoc-ptp-mib-01.txt   February 6, 2012

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 8.3.3.2 of [IEEE 1588-2008]"
 ::= { ptpbaseClockPortTransDSEntry 6 }

ptpbaseClockPortTransDSPeerMeanPathDelay OBJECT-TYPE
SYNTAX          ClockTimeInterval
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 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 (pkts, errors) to be displayed based on
    availability and scenario."
 ::= { ptpbaseMIBClockInfo 11 }

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     InetAddressType,  
  ptpbaseClockPortAssociateAddress         InetAddress,  
  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 }

Shankarkumar et al.     Expires August 6, 2012                [Page 61]
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 InetAddressType
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 InetAddress
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

5. Security Considerations

This MIB contains readable objects whose values provide information related to PTP objects. While unauthorized access to the readable objects is relatively innocuous, unauthorized access to the writeable objects could cause a denial of service, or could cause unauthorized creation and/or manipulation of tunnels. Hence, the support for SET operations in a non-secure environment without proper protection can have a negative effect on network operations.

SNMPv1 by itself is such an insecure environment. 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 SET (change/create/delete) the objects in this MIB.

It is recommended that the implementers consider the security features as provided by the SNMPv3 framework. Specifically, the use of the User-based Security Model [RFC 3414] and the View-based Access Control Model [RFC 3415] is recommended.
It is then a customer/user responsibility to ensure that the SNMP entity giving access to 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

To be added.

7. References

7.1. Normative References


7.2. Informative References


[G.8265.1] "Precision time protocol telecom profile for frequency
8. Acknowledgements

Thanks to John Linton and Danny Lee for valuable comments.

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,  
Symmetricom Inc.,  
2300 Orchard Parkway,  
San Jose,  
CA 95131,  
USA.  
Email: tfrost@symmetricom.com

Greg Dowd,  
Symmetricom Inc.,  
2300 Orchard Parkway,  
San Jose,  
CA 95131,  
USA.  
Email: gdowd@symmetricom.com
Transporting PTP messages over MPLS networks using a link local addressing
draft-jobert-tictoc-ptp-link-local-00.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 contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or 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 September 5, 2012.
Abstract

This document introduces a method for transporting PTP messages over an MPLS network supported by an Ethernet physical layer. The MPLS layer itself is not used to carry the PTP messages with this method; instead, a link local Ethernet channel is used. Several advantages related to this method are highlighted in this document. The method targets in particular telecom applications requiring accurate phase/time synchronization, with "link-by-link" PTP architectures, where all the network nodes support a PTP function, such as Boundary Clock or Transparent Clock.

Table of Contents

1. Introduction ........................................ 3
2. Conventions used in this document................ 4
3. Analysis of the PTP frequency telecom profile with MPLS networks ........................................ 4
4. Transporting PTP messages over MPLS networks with a "link-by-link" PTP architecture ......................... 5
   4.1. Need for identifying the PTP messages in MPLS networks... 6
   4.2. Use of a link local addressing over MPLS networks supported by an Ethernet physical layer.................. 7
   4.3. Use of link local addressing with Transparent Clocks..... 8
5. Security Considerations................................. 12
6. IANA Considerations ................................... 12
7. References .............................................. 13
   7.1. Normative References.............................. 13
   7.2. Informative References............................ 13
8. Acknowledgments ....................................... 13
1. Introduction

The Precision Time Protocol version 2 (PTPv2), defined by the [IEEE1588-2008] standard, is used to support telecom applications that may include MPLS networks. Telecoms applications may require frequency synchronization only or accurate phase/time synchronization.

This has led to the definition of two PTP telecom profiles at the ITU-T: the Recommendation [G.8265.1] (finalized) defines a PTP telecom profile for frequency synchronization in an "end-to-end" mode (the intermediate network nodes do not support PTP functions) and the future Recommendation G.8275.1 (under development) will define a PTP telecom profile for phase/time synchronization in a "link-by-link" mode (all the intermediate network nodes support PTP functions).

For frequency applications using the ITU-T frequency profile, there is no particular need to identify the PTP messages in case they are carried in an MPLS layer. The use of a high priority class of service is in general sufficient to minimize the Packet Delay Variation (PDV) introduced by the network nodes. The identification of the PTP messages in a network node which does not support PTP functions is not expected in general to provide a better performance than the positioning of the PTP messages in a dedicated high priority queue.

For phase/time applications with stringent requirements (e.g. sub-micro-second accuracy), it is in general recognized that PTP support from the network nodes is required to avoid the generation of Packet Delay Variation. Therefore, being able to identify the PTP messages is considered important. This is the one of the objectives of the definition of a PTP mapping. Some mappings are already defined in the [IEEE1588-2008] standard, and may be applicable to an MPLS network.

This document introduces a method for transporting PTP messages over an MPLS network supported by an Ethernet physical layer. The MPLS layer itself is not used to carry the PTP messages with this method; instead, a link local Ethernet channel is used.

Several advantages related to this method are highlighted in this document. The method targets in particular telecom applications requiring accurate phase/time synchronization, with "link-by-link" PTP architectures, where all the network nodes support a PTP function, such as Boundary Clock (BC) or Transparent Clock (TC).
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.

PTP: Precision Time Protocol

PDV: Packet Delay Variation

BC: Boundary Clock

TC: Transparent Clock

3. Analysis of the PTP frequency telecom profile with MPLS networks

For applications requiring frequency synchronization only, when the use of physical layer synchronization methods such as Synchronous Ethernet is not possible, the ITU-T PTP frequency telecom profile defined in the Recommendation G.8265.1 is in general relevant, especially in order to address mobile networks needs.

This PTP telecom profile is based on an "end-to-end" PTP architecture: the intermediate network nodes do not support PTP functions such as Boundary Clock (BC) or Transparent Clock (TC). As such, they generate Packet Delay Variation (PDV). The PTP communication is only performed between a PTP master function and a PTP slave function.

This PTP dialog may involve different layers, due to different encodings. In particular, it is common that PTP messages are carried within an MPLS layer when using this PTP profile.

In order to minimize the PDV generated by the intermediate network nodes, PTP messages MUST be marked as high priority traffic, and MUST be positioned in high priority queues. This marking does not involve new PTP functions in the network nodes; it corresponds simply to the usual DiffServ functions supported in these devices.
In particular, the intermediate network nodes do not identify the PTP messages among the rest of the traffic; only the marking of the packets is considered to position them in the relevant queues.

The identification of the PTP messages by an intermediate network node which does not support PTP functions with this PTP frequency telecom profile is not expected in general to provide real performance improvements compared to the prioritization of the PTP traffic and the positioning of the PTP messages in a dedicated high priority queue.

Indeed, more specialized treatment of the PTP messages would make the network node very close to a node supporting PTP functions such as Boundary Clocks or Transparent Clocks. This would be quite contradictory to the architecture assumptions of this PTP frequency telecom profile.

In conclusion, when the ITU-T PTP frequency telecom profile defined in the Recommendation G.8265.1 is used, the identification of the PTP messages among the rest of the MPLS traffic does neither appear necessary, nor providing real performance benefits.

4. Transporting PTP messages over MPLS networks with a "link-by-link" PTP architecture

For applications requiring accurate phase/time synchronization, the use of the future ITU-T PTP phase/time telecom profile under definition in the Recommendation G.8275.1 is foreseen to be relevant to address the needs of mobile networks.

This PTP telecom profile is based on a "link-by-link" PTP architecture: the intermediate network nodes MUST support PTP functions such as Boundary Clock or Transparent Clock. This architecture is considered as necessary to avoid the generation of Packet Delay Variation, due to the stringent accuracy requirements that are targeted. The PTP communication is therefore performed between different PTP entities: PTP master function, PTP slave function, PTP Boundary Clocks, PTP Transparent Clocks.

Hence, being able to identify the PTP messages is considered important, in order to allow the intermediate network nodes to apply the special treatment on the PTP packets corresponding to the PTP function that they implement (BC or TC).
This is one of the objectives of the definition of a PTP mapping. Some mappings are already defined in the [IEEE1588-2008] standard, and may be applicable to an MPLS network. The transport of PTP messages over MPLS networks SHOULD NOT involve the MPLS layer itself in this type of "link-by-link" PTP architecture.

4.1. Need for identifying the PTP messages in MPLS networks

The "link-by-link" PTP architecture described above may be applicable over MPLS networks. As such, it is relevant to discuss the mapping options for transporting the PTP messages over MPLS networks when considering this type of PTP architecture.

Two PTP operations may be necessary in the MPLS nodes in order to handle the PTP packets in the general case:

- **PTP packets detection**: how to detect that a packet contains PTP payload? (this question is applicable to both Boundary Clock or Transparent Clock types of PTP support)

- **PTP payload position in the packet**: how to determine where the PTP payload is in the message once the relevant packets have been detected? (this question is applicable only to Transparent Clock PTP support, because Boundary Clocks terminate and process the PTP payload)

Regarding the first point listed above (PTP packets detection), the three following mappings could be considered in the general case:

- **in case of an Ethernet mapping**, the PTP packets can be detected thanks to a specific Ethertype. Some PTP mappings already defined in [IEEE1588-2008] already cover this point (see Annex F).

- **in case of an IP/UDP mapping**, the PTP packets can be detected thanks to specific UDP port numbers. Some PTP mapping already defined in [IEEE1588-2008] already cover this point (see Annexes D and E). This mapping corresponds to the mapping specified for the PTP frequency telecom profile defined in [G.8265.1].
In case of MPLS mapping, if relevant, the draft [4] ("Transporting PTP messages (1588) over MPLS Networks") currently discussed in the IETF TICTOC Working Group aims at specifying new MPLS mappings enabling to detect the PTP packets among the traffic. Note that these new PTP mappings are not defined in [IEEE1588-2008).

This document advocates that the third type of mapping (MPLS mappings) is not necessary for carrying PTP messages over MPLS networks supported by an Ethernet physical layer when using a "link-by-link" PTP architecture as depicted above in this document. Instead, it is considered that the use of a link local addressing is more relevant when the MPLS network is supported by an Ethernet physical layer. This point will be discussed further in the next sections of this document.

Regarding the second point (PTP payload position in the packet), it should be stressed the network nodes may not know exactly where the PTP payload is in the packet in some cases (e.g. when tunnels are used), because of other potential encapsulations beyond the layer handled by the node. This situation may happen in the case of MPLS network nodes. In particular, as mentioned above, it raises problems for modifying the PTP payload in case of a Transparent Clock PTP support.

This document explains that the use of a link local addressing simplifies this point, since the PTP payload is in this case at a fixed location in the message. It is moreover in line with the principles of a "link-by-link" PTP architecture, where the PTP messages are sent to the next network node, and are not assumed to be forwarded through a tunnel. This point will be discussed further in the next sections of this document.

4.2. Use of a link local addressing over MPLS networks supported by an Ethernet physical layer

This section introduces a solution to carry PTP messages over an MPLS network supported by an Ethernet physical layer, using a link local Ethernet addressing. This solution fits very well with the "link-by-link" PTP architecture depicted before.

With this solution, Ethernet interfaces supporting MPLS traffic MUST use the Ethernet multicast address: ’01-80-C2-00-00-0E’ based on the Annex F of IEEE1588-2008 for all the PTP messages that are sent.
This type of addressing aims at making sure that the PTP messages will be sent to the next network node in the chain (which may be or not an MPLS node).

This solution has several advantages:

- It prevents unwanted forwarding of PTP messages over network nodes which do not provide PTP support: indeed, such a network node is assumed in general to drop the PTP messages, and not to forward them. It is useful in order to avoid the generation of PDV. This property is considered in line with the “link-by-link” PTP architecture principles depicted earlier.

- It facilitates the configuration for the operator, since no particular addressing needs to be configured in the network nodes.

- It allows having a consistent PTP mapping all along the chain: all the PTP messages are transported the same way, using the same mapping, whatever the actual layers used to transport the user plane. In particular, an MPLS node may establish a PTP dialog with an IP node or a node working at the layer 2 with this type of solution.

- It facilitates the PTP payload identification, since the PTP payload is necessarily at a fixed location.

Note: in case of MPLS nodes connected together via a different physical layer than Ethernet, another link local channel linked to the physical layer might be used. This is beyond the scope of this document.

4.3. Use of link local addressing with Transparent Clocks

The case of Transparent Clock type of PTP support deserves a specific analysis when considering the use of a link local addressing. Indeed, some designs of Transparent Clock may not terminate the PTP messages; it creates issues in order to forward the PTP messages when link local addressing is used.

This section highlights however that some simple mechanisms might be implemented in Transparent Clocks to ensure their compatibility with the use of a link local addressing as proposed in the previous
section. It also shows that a link local addressing may avoid the layer violation issues with TCs.

Three main steps are observed in a standard Transparent Clock which does not terminate the PTP messages in order to treat and forward them:

1- Detection of the PTP packet among the rest of the traffic on an active PTP port, and precise timestamping of the arrival instant of the packet in the network node.

2- The PTP packet is treated/forwarded in the network node as a standard packet, e.g. analysis of the network header of the packet corresponding to the layer treated by the network node, in order to determine using the forwarding engine towards which output port the packet must be forwarded (for instance: IP lookup operation in a routing table). In summary: the output port is determined based on information contained in the PTP packet itself, using standard forwarding functions in the network node.

3- Transmission of the PTP packet at the output of the network node on the port determined before, and precise timestamping of the emission instant of the packet in the network. Modification of the "correction field" of the packet to include the residence time calculation.

The layer violation is due here to the fact that the PTP packet has been modified (correction field update) by an intermediate node which was assumed only to forward it. Moreover, there might be some difficulties to determine where the PTP payload is located, as mentioned earlier.

The use of a link local addressing might not be suitable with this model of TC. Indeed, it can be observed that the step 2 requires in the general case that the necessary information (e.g. final destination address) would be contained in the network header of the PTP messages to determine the output port where each PTP message must be forwarded. This is not the case with link local addressing, because each message is sent to the next node over a single link.

However, there are easy ways to overcome this issue. One possible straightforward solution could be to include locally in the network node the necessary information for the forwarding of the PTP messages. This might correspond to a "PTP local forwarding"
function", which could be part of the network node configuration (manual configuration would be possible, but automatic procedures would also work).

As for the case of a standard TC, three main steps are observed in order to treat and forward a PTP message in a Transparent Clock implementing a PTP local forwarding function:

- The step 1 is similar in both cases (standard TC and TC with PTP local forwarding function).

- The step 2 would differ in this example (TC with PTP local forwarding function): the standard forwarding function of the network node (forwarding engine) MUST NOT be used in this case to forward the PTP packets; instead, the PTP local forwarding function MUST be used. This allows handling PTP packets without forwarding information in the network header of the packet.

- The step 3 is quite similar in both cases (standard TC and TC with PTP local forwarding function).

It must be stressed that the use of link local addressing leads to terminate the PTP packets that are received by the network node, since the recipient of the PTP messages is the network node itself. The PTP packets sent at the output of the TC with PTP local forwarding function are therefore new PTP packets, similarly to a BC. This is the reason why it can be considered as a way to avoid the layer violation issue.

In practice, the operations are similar between standard TC and TC with PTP local forwarding function for generating a new PTP packet based on the PTP packet received (e.g. update of the correction field, etc...).

Moreover, it must also be stressed that the use of link local addressing leads to a fixed location of the PTP payload in the packet. This is expected to greatly simplify the operations.

The PTP local forwarding function includes locally in the network node all the necessary information for forwarding the PTP packets. For instance, it may associate one or several output ports to an input port. An example of what could be a PTP local forwarding function is provided in the figure 1 below.
In the figure 1 above, three configurations are possible for a PTP port in a TC with PTP local forwarding function:
o Disabled PTP port: any potential PTP packet received on this port MUST be discarded.

o Enabled PTP upstream port: corresponds to a port where upstream PTP packets are received (e.g. the PTP packets generated by a PTP master port). When a PTP packet is received on an enabled PTP upstream port, a new PTP packet MUST be transmitted by one or several enabled PTP downstream ports of the network node associated to the enabled PTP upstream port. This/these new PTP packet(s) is/are formed using the information of the original PTP packet that was received, and by modifying the fields normally modified by a TC (the correction field in particular).

o Enabled PTP downstream port: corresponds to a port where downstream PTP packets are received (e.g. the PTP packets generated by a PTP slave port). When a PTP packet is received on an enabled PTP downstream port, a new PTP packet MUST be transmitted by the enabled PTP upstream port of the network node associated to the enabled PTP downstream port. This new PTP packet is formed using the information of the original PTP packet that was received, and by modifying the fields normally modified by a TC (the correction field in particular).

Note that the case of a two-port device is an example where implicit PTP local forwarding function exists: every port PTP packet received on one port must be forwarded by the other port.

The advantages of this type of mechanism are that it allows mixing BCs and TCs in a chain in a consistent way, using link local addressing. It also allows avoiding layer violation issues, since the PTP messages are terminated and processed by each network node, including the TC with PTP local forwarding function.

5. Security Considerations

<Add any security considerations>

6. IANA Considerations

<Add any IANA considerations>
7. References

7.1. Normative References


7.2. Informative References


8. Acknowledgments

This document was prepared using 2−Word−v2.0.template.dot.
Authors’ Addresses

Sebastien Jobert
France Telecom Orange
2 avenue Pierre Marzin
22300 LANNION, FRANCE

Email: sebastien.jobert@orange.com
Abstract

As time synchronization protocols are becoming increasingly common and widely deployed, concern about their exposure to various security threats is increasing. This document defines a set of requirements for security solutions for time synchronization protocols, focusing on the IEEE 1588 and NTP. This document also discusses the security impacts of time synchronization protocol practices, the time synchronization performance implications of external security practices, the dependencies between other security services and time synchronization.

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 April 24, 2012.
Copyright Notice

Copyright (c) 2011 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (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.

Table of Contents

1. Introduction ................................................ 3
2. Conventions Used in this Document ........................... 4
   2.1. Terminology ............................................ 4
   2.2. Abbreviations .......................................... 4
3. Security Threats ............................................ 4
   3.1. Packet interception and manipulation ................... 5
   3.2. Spoofing ............................................... 5
   3.3. Replay attack .......................................... 5
   3.4. Rogue master attack .................................... 5
   3.5. Packet Interception and Removal ....................... 5
   3.6. Packet delay manipulation .............................. 5
   3.7. Cryptographic performance attacks ...................... 6
   3.8. DoS attacks ............................................ 6
   3.9. Time source spoofing (e.g. GPS fraud) .................. 6
4. Security Requirements ....................................... 6
   4.1. Clock Identity Authentication .......................... 6
   4.1.1. Authentication and Proventication of Masters ...... 6
   4.1.2. Authentication of Slaves ........................... 7
   4.1.3. PTP: Authentication of Transparent Clocks.......... 7
   4.1.4. PTP: Authentication of Announce Messages .......... 8
   4.2. Data integrity ......................................... 8
   4.2.1. PTP: Hop-by-hop vs. End-to-end Integrity Protection 8
          4.2.1.1. Hop by Hop Integrity Protection .............. 9
          4.2.1.2. End to End Integrity Protection ............. 9
   4.3. Availability .......................................... 10
   4.4. Replay Protection ..................................... 10
4.5. Cryptographic Keys & Security Associations ..........10
   4.5.1. Security Association ............................. 10
   4.5.2. Unicast and Multicast ............................ 10
   4.5.3. Key Freshness .................................... 11

Mizrahi, O'Donoghue    Expires April 24, 2012                 [Page 2]
1. Introduction

As time synchronization protocols are becoming increasingly common and widely deployed, concern about the resulting exposure to various security threats is increasing. If a time synchronization protocol is compromised, the applications it serves are prone to a range of possible attacks including Denial-of-Service or incorrect behavior.

This document focuses on the security aspects of the Precision Time Protocol ([IEEE 1588]) and the Network Time Protocol ([NTPv4]). The Network Time Protocol was defined with an inherent security protocol, defined in [NTPv4] and in [AutoKey]. The IEEE 1588 includes an experimental security protocol, defined in Annex K of the standard, but this Annex was never formalized into a fully defined security protocol.

This document attempts to add clarity to the time synchronization protocol security requirements discussion by addressing a series of questions. It is expected that this document will evolve into possibly two documents including one on requirements and one providing clarity around the additional questions raised below. Until the discussion has matured sufficiently, it will be captured in this document. The four primary questions addressed by this draft include:

(1) What are the threats that need to be addressed for the time synchronization protocol, and thus what security services need to be provided? (e.g. a malicious NTP server or PTP master)

(2) What external security practices impact the security and performance of time keeping, and what can be done to mitigate these impacts? (e.g. an IPSec tunnel in the synchronization traffic path)

(3) What are the security impacts of time synchronization protocol practices? (e.g. on-the-fly modification of timestamps)
What are the dependencies between other security services and time synchronization? (e.g. which comes first - the certificate or the timestamp?)

It is expected that the final version of this document will define a set of requirements for security solutions for time synchronization protocols, focusing on the IEEE 1588 and NTP.

2. Conventions Used in this Document

2.1. Terminology

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 [KEYWORDS].

This document describes security requirements, and thus requirements are phrased in the document in the form "the security mechanism MUST/SHOULD/...". Note, that the phrasing does not imply that this document defines a specific security mechanism, but defines the requirements that every security mechanism should comply to.

This document refers to both PTP and NTP. For the sake of consistency, throughout the document the term "master" applies to both a PTP master and an NTP server. Similarly, the term "slave" applies to both PTP slaves and NTP clients. The general term "clock" refers to masters, slaves and PTP Transparent Clocks (TC). The term "protocol packets" is refers generically to PTP and NTP messages.

2.2. Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BC</td>
<td>Boundary Clock</td>
</tr>
<tr>
<td>MITM</td>
<td>Man In The Middle</td>
</tr>
<tr>
<td>NTP</td>
<td>Network Time Protocol</td>
</tr>
<tr>
<td>OC</td>
<td>Ordinary Clock</td>
</tr>
<tr>
<td>PTP</td>
<td>Precision Time Protocol</td>
</tr>
<tr>
<td>TC</td>
<td>Transparent Clock</td>
</tr>
</tbody>
</table>

3. Security Threats

The following section defines the security threats that are discussed in subsequent sections.
3.1. Packet interception and manipulation

A packet interception and manipulation attack results when a Man-In-The-Middle (MITM) attacker intercepts timing protocol packets, alters them and relays them to their destination, allowing the attacker to maliciously tamper with the protocol. This can result in a situation where the time protocol is apparently operational but providing intentionally inaccurate information.

3.2. Spoofing

In spoofing, an attacker masquerades as a legitimate node in the network. For example, an attacker can impersonate the master, allowing malicious distribution of false timing information. As with packet interception and manipulation, this can result in a situation where the time protocol is apparently operational but providing intentionally inaccurate information.

3.3. Replay attack

In a replay attack, an attacker records protocol packets and replays them at a later time. This can also result in a situation where the time protocol is apparently operational but providing intentionally inaccurate information.

3.4. Rogue master attack

In a rogue master attack, an attacker causes other nodes in the network to believe it is a legitimate master. As opposed to the spoofing attack, in the Rogue Master attack the attacker does not fake its identity, but rather manipulates the master election process. For example, in PTP, an attacker can manipulate the Best Master Clock Algorithm (BMCA), and cause other nodes in the network to believe it is the most eligible candidate to be a grandmaster.

3.5. Packet Interception and Removal

A packet interception and removal attack results when a Man-In-The-Middle attacker intercepts and drops protocol packets, preventing the destination node from receiving the timing information.

3.6. Packet delay manipulation

In a packet delay manipulation scenario, a Man-In-The-Middle attacker intercepts protocol packets, and relays them to their destination after adding a maliciously computed delay.
3.7. Cryptographic performance attacks

In cryptographic performance attacks, an attacker transmits fake protocol packet, causing high utilization of the cryptographic engine at the receiver, which attempts to verify the integrity of these fake packets.

3.8. DoS attacks

There are many possible Layer 2 and Layer 3 Denial of Service attacks. As the target’s availability is compromised, the timing protocol is affected accordingly.

3.9. Time source spoofing (e.g. GPS fraud)

In time source spoofing, an attacker spoofs the accurate time source of the master. For example, if the master uses a GPS based clock as its reference source, an attacker can spoof the GPS satellites, causing the master to use a false reference time.

4. Security Requirements

4.1. Clock Identity Authentication

Requirement

The security mechanism MUST provide a means for each clock to authenticate the sender of a protocol packet.

Discussion

In the context of this document, authentication refers to:

- Identification: verifying the identity of the peer clock.
- Authorization: verifying that the peer clock is permitted to play the role that it plays in the protocol. For example, some nodes may be permitted to be masters, while other nodes are only permitted to be slaves or TCs.

The following subsections describe 4 distinct cases of clock authentication.

4.1.1. Authentication and Provention of Masters

Requirement
The security mechanism MUST support a proventication mechanism, to be used in cases where end-to-end authentication is not possible.

Discussion

Slaves and transparent clocks authenticate masters in order to ensure the authenticity of the time source.

In some cases a slave is connected to an intermediate master, that is not the primary time source. For example, in PTP a slave can be connected to a Boundary Clock (BC), which in turn is connected to a grandmaster. A similar example in NTP is when a client is connected to a stratum 2 server, which is connected to a stratum 1 server. In both the PTP and the NTP cases, the slave authenticates the intermediate master, and the intermediate master authenticates the primary master. This inductive authentication process is referred to in [AutoKey] as proventication.

4.1.2. Authentication of Slaves

Requirement

The security mechanism SHOULD provide a means for a master to authenticate its slaves.

Discussion

Slaves are authenticated by masters in order to verify that the slave is authorized to receive timing services from the master.

Authentication of slaves prevents unauthorized clocks from receiving time services, and also reduces unnecessary load on the master clock, by preventing the master from serving unauthorized clocks. It could be argued that the authentication of slaves could put a higher load on the master then serving the unauthorized clock. This tradeoff will need to be discussed further.

4.1.3. PTP: Authentication of Transparent Clocks

Requirement

The security mechanism for PTP SHOULD provide a means for a master to authenticate the TCs.

Discussion

Transparent clocks are authenticated by peer masters, slaves and TCs.
Authentication of TCs, much like authentication of slaves, reduces unnecessary load on the master clock and peer TCs, by preventing the master from serving unauthorized clocks. It also prevents malicious TCs from attacking the protocol by manipulating the correctionField. It could also be argued that the authentication could result in a higher load than merely serving the unauthorized devices. This tradeoff will need to be discussed further.

4.1.4. PTP: Authentication of Announce Messages

Requirement

The security mechanism for PTP MUST support authentication of Announce messages.

Discussion

Master election is performed in PTP using the Best Master Clock Algorithm (BMCA). Each Ordinary Clock (OC) announces its clock attributes using Announce messages, and the best master is elected based on the information gathered from all the candidates. Announce messages must be authenticated in order to prevent malicious master attacks.

Note, that this subsection specifies a requirement that is not necessarily included in 4.1.1. or in 4.1.2., since the BMCA is initiated before clocks have been defined as masters or slaves.

4.2. Data integrity

Requirement

The security mechanism MUST protect the integrity of protocol packets.

Discussion

While subsection 4.1. refers to ensuring WHO sent the protocol packet, this subsection refers to ensuring that the packet arrived intact. The integrity protection mechanism ensures the authenticity and completeness of data from the data originator.

4.2.1. PTP: Hop-by-hop vs. End-to-end Integrity Protection

Requirement
A security mechanism for PTP MUST support hop-by-hop integrity protection.

Requirement

A security mechanism for PTP SHOULD support end-to-end integrity protection.

Discussion

Specifically in PTP, when protocol packets are subjected to modification by TCs, the integrity protection can be enforced in one of two approaches, end-to-end or hop-by-hop.

4.2.1.1. Hop by Hop Integrity Protection

Each hop that needs to modify a protocol packet:

- Verifies its integrity.
- Modifies the packet, i.e., modifies the correctionField.
- Re-generates the integrity protection, e.g., re-computes a Message Authentication Code.

In the hop-by-hop approach, the integrity of protocol packets is protected by induction on the path from the originator to the receiver.

This approach is simple, but allows malicious TCs to modify protocol packets.

4.2.1.2. End to End Integrity Protection

In this approach, the integrity protection is maintained on the path from the originator of a protocol packet to the receiver. This allows the receiver to validate the protocol packet without the ability of intermediate TCs to manipulate the packet.

Since TCs need to modify the correctionField, a separate integrity protection mechanism is used specifically for the correctionField.

The end-to-end approach limits the TC’s impact to the correctionField alone, while the rest of the protocol packet is protected on an end-to-end basis.
4.3. Availability

Requirement
The security mechanism MUST be resistant to DoS attacks from an external attacker.

Discussion
This requirement is attained by clock authentication, as described in 4.1.

4.4. Replay Protection

Requirement
Protocol messages MUST be resistant to replay attacks.

4.5. Cryptographic Keys & Security Associations

4.5.1. Security Association

Requirement
The security protocol MUST support an association protocol where:

o Two or more clocks authenticate each other.

o The clocks generate and agree on a cryptographic session key.

Discussion
The security requirements in 4.1. and 4.2. require usage of cryptographic mechanisms, deploying cryptographic keys. A security association is an essential building block in these mechanisms.

4.5.2. Unicast and Multicast

Requirement
The security mechanism MUST support security association protocols for unicast and for multicast associations.

Discussion
A unicast protocol requires an association protocol between two clocks, whereas a multicast protocol requires an association protocol among two or more clocks, where one of the clocks is a master.

### 4.5.3. Key Freshness

**Requirement**

The cryptographic keys MUST be refreshed periodically.

**Requirement**

The association protocol MUST be invoked periodically, where each instance of the association protocol MUST produce a different session key.

### 4.6. Performance

**Requirement**

The security mechanism MUST be designed in such a way that it does not degrade the quality of the time transfer.

**Requirement**

The mechanism SHOULD be relatively lightweight, as client restrictions often dictate a low processing and memory footprint, and because the server may have extensive fan-out.

**Requirement**

The mechanism also SHOULD not require excessive storage of client state in the master, nor significantly increase bandwidth consumption.

### 4.7. Confidentiality

**Requirement**

The security mechanism MAY provide confidentiality protection of the protocol packets.

**Discussion**

In the context of time synchronization, confidentiality is typically of low importance, since timing information is typically not considered secret information.
Confidentiality can play an important role when service providers charge payment for time synchronization services, but these cases are rather esoteric.

Confidentiality can also prevent an MITM attacker from identifying protocol packets. Thus, confidentiality can assist in protecting the timing protocol against packet delay attacks, where the attacker selectively adds delay to time protocol packets.

4.8. Protection against packet delay attacks

Requirement

The security mechanism MAY include a means to detect packet delay attacks.

Requirement

The security mechanism MAY include a protection switching mechanism that allows a node that detects a delay attack to switch over to a secondary master.

5. Summary of Requirements

<table>
<thead>
<tr>
<th>Section</th>
<th>Requirement</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1.</td>
<td>Authentication of sender.</td>
<td>MUST</td>
</tr>
<tr>
<td></td>
<td>Proventication.</td>
<td>MUST</td>
</tr>
<tr>
<td></td>
<td>Authentication of slaves.</td>
<td>SHOULD</td>
</tr>
<tr>
<td></td>
<td>PTP: Authentication of TCs.</td>
<td>SHOULD</td>
</tr>
<tr>
<td></td>
<td>PTP: Authentication of Announce messages.</td>
<td>SHOULD</td>
</tr>
<tr>
<td>4.2.</td>
<td>Integrity protection.</td>
<td>MUST</td>
</tr>
<tr>
<td></td>
<td>PTP: hop-by-hop integrity protection.</td>
<td>MUST</td>
</tr>
<tr>
<td></td>
<td>PTP: end-to-end integrity protection.</td>
<td>SHOULD</td>
</tr>
</tbody>
</table>
4.3. Protection against DoS attacks. MUST

4.4. Replay protection. MUST

4.5. Security association. MUST

| Unicast and multicast associations. MUST |
| Key freshness. MUST |

4.6. Performance: no degradation in quality of time transfer. MUST

| Performance: lightweight. SHOULD |
| Performance: storage, bandwidth. MUST |

4.7. Confidentiality protection. MAY

4.8. Protection against delay attacks. MAY

Table 1 Summary of Security Requirements

6. Additional security implications

This section will discuss additional security implications as outlined in the questions below. Contributions are welcome and encouraged.

- What external security practices impact the security and performance of time keeping? (and what can be done to mitigate these impacts?)

- What are the security impacts of time synchronization protocol practices? (e.g. on-the-fly modification of timestamps)

- What are the dependencies between other security services and time synchronization?

7. Issues for Further Discussion

This section will discuss additional issues as identified below. Again, contributions are welcome and encouraged.
o Integrity - end-to-end vs. hop-by-hop.

o Supporting a hybrid network, where some nodes are security enabled and others are not.

o The key distribution is outside the scope of this document. Although this is a cardinal element in any security system, it is not a security requirement, and is thus not described here.

8. Security Considerations

The security considerations of network timing protocols are presented throughout this document.

9. IANA Considerations

There are no new IANA considerations implied by this document.

10. Acknowledgments

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

11. References

11.1. Normative References


11.2. Informative References


Authors’ Addresses

Tal Mizrahi
Marvell
6 Hamada St.
Yokneam, 20692 Israel

Email: talmi@marvell.com

Karen O’Donoghue
7167 Goby Lane
King George, VA 22485

Email:odonoghue@isoc.org
Control Messages Protocol for Use with Network Time Protocol Version 4
draft-odonoghue-ntpv4-control-01

Abstract

This document describes the structure of the control messages used with the Network Time Protocol. These control messages can be used to monitor and control the Network Time Protocol application running on any IP network attached computer. The information in this informational RFC was originally described in Appendix B of RFC 1305.

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on May 3, 2012.

Copyright Notice

Copyright (c) 2011 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (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.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.
1. Introduction

Editor’s Note (to be removed prior to publication): The text below is taken directly from RFC 1305. Input is requested to update the text to reflect current practice. This is required to fully obsolete RFC 1305.

In a comprehensive network-management environment, facilities are presumed available to perform routine NTP control and monitoring functions, such as setting the leap-indicator bits at the primary servers, adjusting the various system parameters and monitoring regular operations. Ordinarily, these functions can be implemented using a network-management protocol such as SNMP and suitable extensions to the MIB database. However, in those cases where such facilities are not available, these functions can be implemented using special NTP control messages described herein. These messages are intended for use only in systems where no other management facilities are available or appropriate, such as in dedicated-function bus peripherals. Support for these messages is not required in order to conform to this specification.

The NTP Control Message has the value 6 specified in the mode field of the first octet of the NTP header and is formatted as shown below. The format of the data field is specific to each command or response; however, in most cases the format is designed to be constructed and viewed by humans and so is coded in free-form ASCII. This facilitates the specification and implementation of simple management tools in the absence of fully evolved network-management facilities. As in ordinary NTP messages, the authenticator field follows the data field. If the authenticator is used the data field is zero-padded to a 32-bit boundary, but the padding bits are not considered part of the data field and are not included in the field count.

IP hosts are not required to reassemble datagrams larger than 576 octets; however, some commands or responses may involve more data than will fit into a single datagram. Accordingly, a simple reassembly feature is included in which each octet of the message data is numbered starting with zero. As each fragment is transmitted the number of its first octet is inserted in the offset field and the number of octets is inserted in the count field. The more-data (M) bit is set in all fragments except the last.

Most control functions involve sending a command and receiving a response, perhaps involving several fragments. The sender chooses a distinct, nonzero sequence number and sets the status field and R and E bits to zero. The responder interprets the opcode and additional information in the data field, updates the status field, sets the R bit to one and returns the three 32-bit words of the header along
with additional information in the data field. In case of invalid message format or contents the responder inserts a code in the status field, sets the R and E bits to one and, optionally, inserts a diagnostic message in the data field.

Some commands read or write system variables and peer variables for an association identified in the command. Others read or write variables associated with a radio clock or other device directly connected to a source of primary synchronization information. To identify which type of variable and association a 16-bit association identifier is used. System variables are indicated by the identifier zero. As each association is mobilized a unique, nonzero identifier is created for it. These identifiers are used in a cyclic fashion, so that the chance of using an old identifier which matches a newly created association is remote. A management entity can request a list of current identifiers and subsequently use them to read and write variables for each association. An attempt to use an expired identifier results in an exception response, following which the list can be requested again.

Some exception events, such as when a peer becomes reachable or unreachable, occur spontaneously and are not necessarily associated with a command. An implementation may elect to save the event information for later retrieval or to send an asynchronous response (called a trap) or both. In case of a trap the IP address and port number is determined by a previous command and the sequence field is set as described below. Current status and summary information for the latest exception event is returned in all normal responses. Bits in the status field indicate whether an exception has occurred since the last response and whether more than one exception has occurred.

Commands need not necessarily be sent by an NTP peer, so ordinary access-control procedures may not apply; however, the optional mask/match mechanism suggested in [RFC5905] provides the capability to limit mode 6 processing to selected address ranges.

The Network Time Protocol reference implementation maintained by the University of Delaware and ntp.org provides a utility program, ntpq which enables management and configuration of the ntpd daemon using NTP Control Messages (mode 6). A related utility program, ntpdc, uses an earlier, deprecated implementation-specific binary management protocol using NTP mode 7 datagrams. Due to the implementation complexity of the earlier protocol, the reference implementation has added support for all operations that previously were exposed only via mode 7 to the preferred mode 6 interface. Support for mode 7 requests is likely to be disabled by default in the reference implementation’s daemon.
2. NTP Control Message Format

The format of the NTP Control Message header, which immediately follows the UDP header, is shown below. Following is a description of its fields. Bit positions marked as zero are reserved and should always be transmitted as zero.

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+-----+-----+-----+---------+-------------------------------+
|LI | VN  |Mode |R E M|    Op   |           Sequence            |
+---+-----+-----+-----+---------+-------------------------------+
        Status             |        Association ID         |
+-------------------------------+-------------------------------+
        Offset             |             Count             |
+-------------------------------+-------------------------------+
                                                                 |
                                                                 Data                              |
                                                                 (468 octets or less)                  |
                                                                 +-----------------------------------------------+
                                                                 | Padding as needed to next multiple of 32 bits |
                                                                 +---------------------------------------------------------------+
                                                                 | Authenticator (optional, 96 octets or less)                    |
                                                                 +---------------------------------------------------------------+
```

LI: This is a two-bit integer that must be zero for control message requests and responses. The Leap Indicator value used at this position in most NTP modes is in the System Status Word provided in some control message responses.

Version Number (VN): This is a three-bit integer indicating a minimum NTP version number. NTP servers should not respond to control messages with an unrecognized version number. Requests may intentionally use a lower version number to enable interoperability with earlier versions. The reference implementation utility ntpq uses version 2 by default. Responses must carry the same version as the corresponding request.

Mode: This is a three-bit integer indicating the mode. It must have the value 6, indicating an NTP control message.

Response Bit (R): Set to zero for commands, one for responses.

Error Bit (E): Set to zero for normal response, one for error response.

More Bit (M): Set to zero for last fragment, one for all others.
Operation Code (Op): This is a five-bit integer specifying the command function. The values are:

<table>
<thead>
<tr>
<th>Code</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>reserved</td>
</tr>
<tr>
<td>1</td>
<td>read status command/response</td>
</tr>
<tr>
<td>2</td>
<td>read variables command/response</td>
</tr>
<tr>
<td>3</td>
<td>write variables command/response</td>
</tr>
<tr>
<td>4</td>
<td>read clock variables command/response</td>
</tr>
<tr>
<td>5</td>
<td>write clock variables command/response</td>
</tr>
<tr>
<td>6</td>
<td>set trap address/port command/response</td>
</tr>
<tr>
<td>7</td>
<td>trap response</td>
</tr>
<tr>
<td>8</td>
<td>runtime configuration command/response</td>
</tr>
<tr>
<td>9</td>
<td>export configuration to file command/response</td>
</tr>
<tr>
<td>10</td>
<td>retrieve remote address stats command/response</td>
</tr>
<tr>
<td>11</td>
<td>retrieve local address stats command/response</td>
</tr>
<tr>
<td>12</td>
<td>request client-specific nonce command/response</td>
</tr>
<tr>
<td>13-30</td>
<td>reserved for future use</td>
</tr>
<tr>
<td>31</td>
<td>unset trap address/port command/response</td>
</tr>
</tbody>
</table>

Sequence: This is a 16-bit integer indicating the sequence number. Each request should use a different sequence number. Each response carries the same sequence number as its corresponding request. For asynchronous trap responses, the responder increments the sequence number by one each response, allowing trap receivers to detect missing trap responses. Note the sequence number of each fragment in a multiple-datagram response carries the same sequence number, copied from the request.

Status: This is a 16-bit code indicating the current status of the system, peer or clock, with values coded as described in following sections.

Association ID: This is a 16-bit unsigned integer identifying a valid association, or zero for the system clock.

Offset: This is a 16-bit unsigned integer indicating the offset, in octets, of the first octet in the data area. The offset must be zero in requests. Responses spanning multiple datagrams use a positive offset in all but the first datagram.

Count: This is a 16-bit unsigned integer indicating the length of the data, in octets

Data: This contains the message data for the command or response.
The maximum number of data octets is 468.

Padding: Contains zero to three octets with value zero, as needed to ensure the overall control message size is a multiple of 4 octets.

Authenticator (optional): When an NTP authentication mechanism is used, this contains the message authenticator information defined in section 7.3 of [RFC5905].

3. Status Words

Status words indicate the present status of the system, associations and clock. They are designed to be interpreted by network-monitoring programs and are in one of four 16-bit formats shown in Figure 6 and described in this section. System and peer status words are associated with responses for all commands except the read clock variables, write clock variables and set trap address/port commands. The association identifier zero specifies the system status word, while a nonzero identifier specifies a particular peer association. The status word returned in response to read clock variables and write clock variables commands indicates the state of the clock hardware and decoding software. A special error status word is used to report malformed command fields or invalid values.

3.1. System Status Word

The system status word appears in the status field of the response to a read status or read variables command with a zero association identifier. The format of the system status word is as follows:

```
0                   1
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+---+-----------+-------+-------+
|LI | ClockSrc  | Count | Code  |
+---+-----------+-------+-------+
```

Leap Indicator (LI): This is a two-bit code warning of an impending leap second to be inserted/deleted in the last minute of the current day, with bit 0 and bit 1, respectively, coded as follows: (EDITOR: this could refer to RFC 5905 section 7.3 figure 9 instead.)
<table>
<thead>
<tr>
<th>LI</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>no warning</td>
</tr>
<tr>
<td>01</td>
<td>insert second after 23:59:59 of the current day</td>
</tr>
<tr>
<td>10</td>
<td>delete second 23:59:59 of the current day</td>
</tr>
<tr>
<td>11</td>
<td>unsynchronized</td>
</tr>
</tbody>
</table>

ClockSrc: This is a six-bit integer indicating the current synchronization source, with values coded as follows:

<table>
<thead>
<tr>
<th>Code</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>unspecified or unknown</td>
</tr>
<tr>
<td>1</td>
<td>Calibrated atomic clock (e.g., PPS, HP 5061)</td>
</tr>
<tr>
<td>2</td>
<td>VLF (band 4) or LF (band 5) radio (e.g., OMEGA, WWVB)</td>
</tr>
<tr>
<td>3</td>
<td>HF (band 7) radio (e.g., CHU, MSF, WWV/H)</td>
</tr>
<tr>
<td>4</td>
<td>UHF (band 9) satellite (e.g., GOES, GPS)</td>
</tr>
<tr>
<td>5</td>
<td>local net (e.g., DCN, TSP, DTS)</td>
</tr>
<tr>
<td>6</td>
<td>UDP/NTP</td>
</tr>
<tr>
<td>7</td>
<td>UDP/TIME</td>
</tr>
<tr>
<td>8</td>
<td>eyeball-and-wristwatch</td>
</tr>
<tr>
<td>9</td>
<td>telephone modem (e.g., NIST)</td>
</tr>
<tr>
<td>10-63</td>
<td>reserved</td>
</tr>
</tbody>
</table>

System Event Counter: This is a four-bit integer indicating the number of system events occurring since the last time the System Event Code changed. Upon reaching 15, subsequent events with the same code are not counted.

System Event Code: This is a four-bit integer identifying the latest system exception event, with new values overwriting previous values, and coded as follows:
<table>
<thead>
<tr>
<th>Code</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>unspecified</td>
</tr>
<tr>
<td>1</td>
<td>frequency correction (drift) file not available</td>
</tr>
<tr>
<td>2</td>
<td>frequency correction started (frequency stepped)</td>
</tr>
<tr>
<td>3</td>
<td>spike detected and ignored, starting stepout timer</td>
</tr>
<tr>
<td>4</td>
<td>frequency training started</td>
</tr>
<tr>
<td>5</td>
<td>clock synchronized</td>
</tr>
<tr>
<td>6</td>
<td>system restart</td>
</tr>
<tr>
<td>7</td>
<td>panic stop (required step greater than panic threshold)</td>
</tr>
<tr>
<td>8</td>
<td>no system peer</td>
</tr>
<tr>
<td>9</td>
<td>leap second insertion/deletion armed for end of current month</td>
</tr>
<tr>
<td>10</td>
<td>leap second disarmed</td>
</tr>
<tr>
<td>11</td>
<td>leap second inserted or deleted</td>
</tr>
<tr>
<td>12</td>
<td>clock stepped (stepout timer expired)</td>
</tr>
<tr>
<td>13</td>
<td>kernel loop discipline status changed</td>
</tr>
<tr>
<td>14</td>
<td>leapseconds table loaded from file</td>
</tr>
<tr>
<td>15</td>
<td>leapseconds table outdated, updated file needed</td>
</tr>
</tbody>
</table>

3.2. Peer Status Word

A peer status word is returned in the status field of a response to a read status, read variables or write variables command and appears also in the list of association identifiers and status words returned by a read status command with a zero association identifier. The format of a peer status word is as follows:

0                   1
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+---------+-----+-------+-------+
| Flags   | Sel | Count | Code  |
+---------+-----+-------+-------+

Peer Status Flags: This is a set of five bits indicating the status of the peer determined by the packet procedure, with bits assigned as follows:
### Peer Selection (Sel):
This is a three-bit integer indicating the status of the peer determined by the clock-selection procedure, with values coded as follows:

<table>
<thead>
<tr>
<th>Sel</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>rejected</td>
</tr>
<tr>
<td>1</td>
<td>discarded by intersection algorithm</td>
</tr>
<tr>
<td>2</td>
<td>discarded by table overflow (not currently used)</td>
</tr>
<tr>
<td>3</td>
<td>discarded by the cluster algorithm</td>
</tr>
<tr>
<td>4</td>
<td>included by the combine algorithm</td>
</tr>
<tr>
<td>5</td>
<td>backup source (with more than sys.maxclock survivors)</td>
</tr>
<tr>
<td>6</td>
<td>system peer (synchronization source)</td>
</tr>
<tr>
<td>7</td>
<td>PPS (pulse per second) peer</td>
</tr>
</tbody>
</table>

### Peer Event Counter:
This is a four-bit integer indicating the number of peer events that occurred since the last time the peer event code changed. Upon reaching 15, subsequent events with the same code are not counted.

### Peer Event Code:
This is a four-bit integer identifying the latest peer exception event, with new values overwriting previous values, and coded as follows:

<table>
<thead>
<tr>
<th>Bit</th>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0x8000</td>
<td>configured (peer.config)</td>
</tr>
<tr>
<td>1</td>
<td>0x4000</td>
<td>authentication enabled (peer.authenable)</td>
</tr>
<tr>
<td>2</td>
<td>0x2000</td>
<td>authentication okay (peer.authentic)</td>
</tr>
<tr>
<td>3</td>
<td>0x1000</td>
<td>reachable (peer.reach != 0)</td>
</tr>
<tr>
<td>4</td>
<td>0x0800</td>
<td>broadcast association</td>
</tr>
</tbody>
</table>
### Peer Event Code Meaning

<table>
<thead>
<tr>
<th>Code</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>unspecified</td>
</tr>
<tr>
<td>1</td>
<td>association mobilized</td>
</tr>
<tr>
<td>2</td>
<td>association demobilized</td>
</tr>
<tr>
<td>3</td>
<td>peer unreachable</td>
</tr>
<tr>
<td>4</td>
<td>peer reachable</td>
</tr>
<tr>
<td>5</td>
<td>association restarted or timed out</td>
</tr>
<tr>
<td>6</td>
<td>no reply (used only with one-shot ntpd -q, known as ntpdate mode)</td>
</tr>
<tr>
<td>7</td>
<td>peer rate limit exceeded (kiss code RATE received)</td>
</tr>
<tr>
<td>8</td>
<td>access denied (kiss code DENY received), not currently implemented</td>
</tr>
<tr>
<td>9</td>
<td>leap second insertion/deletion at month’s end armed by peer vote</td>
</tr>
<tr>
<td>10</td>
<td>became system peer (sys.peer)</td>
</tr>
<tr>
<td>11</td>
<td>reference clock event (see clock status word)</td>
</tr>
<tr>
<td>12</td>
<td>authentication failed</td>
</tr>
<tr>
<td>13</td>
<td>popcorn spike suppressed by peer clock filter register</td>
</tr>
<tr>
<td>14</td>
<td>entering interleaved mode</td>
</tr>
<tr>
<td>15</td>
<td>recovered from interleave error</td>
</tr>
</tbody>
</table>

### 3.3. Clock Status Word

There are two ways a reference clock can be attached to a NTP service host, as an dedicated device managed by the operating system and as a synthetic peer managed by NTP. As in the read status command, the association identifier is used to identify which one, zero for the system clock and nonzero for a peer clock. Only one system clock is supported by the protocol, although many peer clocks can be supported. A system or peer clock status word appears in the status field of the response to a read clock variables or write clock variables command. This word can be considered an extension of the system status word or the peer status word as appropriate. The format of the clock status word is as follows:

```
 0 1 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+---------------+-------+-------+
|   Reserved    | Count | Code  |
|---------------+-------+-------+
```

Reserved: An eight-bit integer that should be ignored by requesters and zeroed by responders.

Clock Event Counter: This is a four-bit integer indicating the number of clock events that occurred since the last time the clock event

---

code changed. Upon reaching 15, subsequent events with the same code are not counted.

Clock Event Code: This is a four-bit integer indicating the current clock status, with values coded as follows:

<table>
<thead>
<tr>
<th>Clock Status</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>clock operating within nominals</td>
</tr>
<tr>
<td>1</td>
<td>reply timeout</td>
</tr>
<tr>
<td>2</td>
<td>bad reply format</td>
</tr>
<tr>
<td>3</td>
<td>hardware or software fault</td>
</tr>
<tr>
<td>4</td>
<td>propagation failure (loss of signal)</td>
</tr>
<tr>
<td>5</td>
<td>bad date format or value</td>
</tr>
<tr>
<td>6</td>
<td>bad time format or value</td>
</tr>
<tr>
<td>7-15</td>
<td>reserved</td>
</tr>
</tbody>
</table>

3.4. Error Status Word

An error status word is returned in the status field of an error response as the result of invalid message format or contents. Its presence is indicated when the E (error) bit is set along with the response (R) bit in the response. The format of the Error Status Word is:

```
0                   1
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+---------------+---------------+--------------+------------------------------------------------------------+| Error Code   |   Reserved    | Error Status |                    Meaning                                 |
+---------------+---------------+--------------+------------------------------------------------------------+|       0      | unspecified   | 0            | unspecified                                                |
|       1      | authentication failure | 1           | authentication failure                                     |
|       2      | invalid message length or format              | 2           | invalid message length or format                           |
|       3      | invalid opcode                                 | 3           | invalid opcode                                             |
|       4      | unknown association identifier                 | 4           | unknown association identifier                             |
|       5      | unknown variable name                          | 5           | unknown variable name                                      |
|       6      | invalid variable value                         | 6           | invalid variable value                                     |
|       7      | administratively prohibited                    | 7           | administratively prohibited                                |
|     8-255    | reserved                                       | 8-255        | reserved                                                   |
Reserved: Responders should use zero. Requesters should ignore the Reserved value to preserve the possibility of future use.

4. Commands

Commands consist of the header and optional data field shown in Section 3. When present, the data field contains a list of identifiers or assignments in the form
<<identifier>>[=<<value>>],<<identifier>>[=<<value>>],... where <<identifier>> is the ASCII name of a system or peer variable specified in Sections 9.1 and 11.1 of RFC 5905 and <<value>> is expressed as a decimal, hexadecimal or string constant in the syntax of the C programming language. Where no ambiguity exists, the "s." or "p." prefixes shown in Figure 5 of Section 7.1 of RFC 5905 [RFC5905] can be suppressed. Whitespace (ASCII nonprinting format effectors) can be added to improve readability for simple monitoring programs that do not reformat the data field. Internet Protocol version 4 addresses are represented as four decimal octets without leading zeros, separated by dots. Internet Protocol version 6 addresses are represented as mandated by [RFC5952], without surrounding square brackets unless a port specification is combined with the address. Timestamps, including reference, originate, receive and transmit values, as well as the logical clock, are represented in units of seconds and fractions, preferably in hexadecimal notation, while delay, offset, dispersion and distance values are represented in units of milliseconds and fractions, preferably in decimal notation. All other values are represented as-is, preferably in decimal notation.

Implementations may define variables other than those listed in Figures 6, 7, 16, 17, 18, 19, 27 and 29 of RFC 5905. Called extramural variables, these are distinguished by the inclusion of some character type other than alphanumeric or "." in the name. For those commands that return a list of assignments in the response data field, if the command data field is empty, it is expected that all available variables defined in Figures 6, 7 and 17 of RFC 5905 will be included in the response. For the read commands, if the command data field is nonempty, an implementation may choose to process this field to individually select which variables are to be returned.

Commands are interpreted as follows:

Read Status (1): The command data field is empty or contains a list of identifiers separated by commas. The command operates in two ways depending on the value of the association identifier. If this identifier is nonzero, the response includes the peer identifier and status word. Optionally, the response data field may contain other
information, such as described in the Read Variables command. If the
association identifier is zero, the response includes the system
identifier (0) and status word, while the data field contains a list
of binary-coded pairs <<association identifier>> <<status word>>, one
for each currently defined association.

Read Variables (2): The command data field is empty or contains a
list of identifiers separated by commas. If the association
identifier is nonzero, the response includes the requested peer
identifier and status word, while the data field contains a list of
peer variables and values as described above. If the association
identifier is zero, the data field contains a list of system
variables and values. If a peer has been selected as the
synchronization source, the response includes the peer identifier and
status word; otherwise, the response includes the system identifier
(0) and status word.

Write Variables (3): The command data field contains a list of
assignments as described above. The variables are updated as
indicated. The response is as described for the Read Variables
command.

Read Clock Variables (4): The command data field is empty or contains
a list of identifiers separated by commas. The association
identifier selects the system clock variables or peer clock variables
in the same way as in the Read Variables command. The response
includes the requested clock identifier and status word and the data
field contains a list of clock variables and values, including the
last timecode message received from the clock.

Write Clock Variables (5): The command data field contains a list of
assignments as described above. The clock variables are updated as
indicated. The response is as described for the Read Clock Variables
command. The reference implementation daemon requires authentication
for this command.

Set Trap Address/Port (6): The command association identifier, status
and data fields are ignored. The address and port number for
subsequent trap messages are taken from the source address and port
of the control message itself. The initial trap counter for trap
response messages is taken from the sequence field of the command.
The response association identifier, status and data fields are not
significant. Implementations should include sanity timeouts which
prevent trap transmissions if the monitoring program does not renew
this information after a lengthy interval.

Trap Response (7): This command differs from the others described
here, which are initiated by a management agent (such as ntpq) and
responded to by a NTP daemon. Trap Response is sent by a NTP daemon
to any registered trap receivers when a system, peer or clock
exception event occurs. The opcode field is 7 and the R bit is set.
The trap counter is incremented by one for each trap sent and the
sequence field set to that value. The trap message is sent using the
IP address and port fields established by the set trap address/port
command. If a system trap the association identifier field is set to
zero and the status field contains the system status word. If a peer
trap the association identifier field is set to that peer and the
status field contains the peer status word. Optional ASCII-coded
information can be included in the data field.

Configure (8): The command data is parsed and applied as if supplied
in the daemon configuration file. The reference implementation
daemon requires authentication for this command.

Save Configuration (9): Write a snapshot of the current configuration
to the file name supplied as the command data. The reference
implementation daemon requires authentication for this command.
Further, the command is refused unless a directory in which to store
the resulting files has been explicitly configured by the operator.

Read MRU (10): Retrieves records of recently seen remote addresses
and associated statistics. Command data consists of name=value pairs
controlling the selection of records, as well as a requestor-specific
nonce previously retrieved using this command or opcode 12, Request
Nonce. The response consists of name=value pairs where some names
can appear multiple times using a dot followed by a zero-based index
to distinguish them, and to associate elements of the same record
with the same index. A new nonce is provided with each successful
response.

Read local address stats (11): Retrieves the local network addresses
of the daemon with status and counters for each. Command data is not
used in the request. Similar to Read MRU, some response information
uses zero-based indexes as part of the variable name preceding the
equals sign and value, where each index relates information for a
single local address. The reference implementation daemon requires
authentication for this command.

Request Nonce (12): Retrieves a 96-bit nonce specific to the
requesting remote address, which is valid for a limited period.
Command data is not used in the request. The nonce consists of a 64-
bit NTP timestamp and 32 bits of hash derived from that timestamp,
the remote address, and salt known only to the server which varies
between daemon runs. The reference implementation honors nonces
which were issued less than 16 seconds prior. Regurgitation of the
nonce by a management agent demonstrates to the server that the agent
can receive datagrams sent to the source address of the request, making source address "spoofing" more difficult in a similar way as TCP's three-way handshake.

Unset Trap (31): Removes the requesting remote address and port from the list of trap receivers. Command data is not used in the request. If the address and port are not in the list of trap receivers, the error code is 4, bad association.

5. IANA Considerations

Editor’s Note: To be reviewed by the working group prior to completion.

6. Security Considerations

Editor’s Note: To be supplied by the working group prior to completion.

7. Acknowledgements

8. References

8.1. Normative References


8.2. Informative References


Authors’ Addresses

Dr. David L. Mills
University of Delaware
Newark, Delaware
US
Email: mills@udel.edu

Karen O’Donoghue (editor)
Internet Society
King George, Virginia
US
Email: odonoghue@isoc.org

David L. Hart
Redmond, Washington
US
Email: hart@ntp.org

Harlan M. Stenn
Network Time Foundation, Inc.
Talent, Oregon
US
Email: stenn@ntp.org